**UNIVERSIDAD DE GRANADA**

DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

PROGRAMA DE DOCTORADO EN ESTADÍSTICA MATEMÁTICA Y APLICADA

# *TÉCNICAS NO PARAMÉTRICAS Y SEMIPARAMÉTRICAS EN BASES DE DATOS PROCEDENTES DE ESTUDIOS MULTICÉNTRICOS*


UNIVERSIDAD
DE GRANADA

TESIS DOCTORAL

## *BÚSQUEDA DE PERFILES CLÍNICOS EN BASES DE DATOS DE ESTUDIOS MULTICÉNTRICOS CON DISTINTAS PATOLOGÍAS MEDIANTE DIFERENTES TÉCNICAS MULTIVARIANTES*

***Nisa Boukichou Abdelkader***

*Prof. Dr. D. Alberto Muñoz García y Prof. Dr. D. Miguel Ángel Montero Alonso*

*Granada, 2022*

**UNIVERSIDAD DE GRANADA**

DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

PROGRAMA DE DOCTORADO EN ESTADÍSTICA MATEMÁTICA Y APLICADA

# *TÉCNICAS NO PARAMÉTRICAS Y SEMIPARAMÉTRICAS EN BASES DE DATOS PROCEDENTES DE ESTUDIOS MULTICÉNTRICOS*


UNIVERSIDAD
DE GRANADA

TESIS DOCTORAL

## *BÚSQUEDA DE PERFILES CLÍNICOS EN BASES DE DATOS DE ESTUDIOS MULTICÉNTRICOS CON DISTINTAS PATOLOGÍAS MEDIANTE DIFERENTES TÉCNICAS MULTIVARIANTES*

***Nisa Boukichou Abdelkader***

*Prof. Dr. D. Alberto Muñoz García y Prof. Dr. D. Miguel Ángel Montero Alonso*

*Granada, 2022*

Editor: Universidad de Granada. Tesis Doctorales
Autor: Nisa Boukichou Abdelkader
ISBN: 978-84-1117-468-8
URI: http://hdl.handle.net/10481/76792

**UNIVERSIDAD DE GRANADA**
DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

UNIVERSIDAD
DE GRANADA

# TESIS DOCTORAL

## *BÚSQUEDA DE PERFILES CLÍNICOS EN BASES DE DATOS DE ESTUDIOS MULTICÉNTRICOS CON DISTINTAS PATOLOGÍAS MEDIANTE DIFERENTES TÉCNICAS MULTIVARIANTES*

**La presente Tesis Doctoral esta avalada hasta la fecha de su lectura por la siguiente aportación científica:**

**Boukichou-Abdelkader, N.**, Montero-Alonso, M.Á. and Muñoz-García, A. (2022). Different Routes or Methods of Application for Dimensionality Reduction in Multicenter Studies Databases. *Mathematics*, **10** (5), 696. DOI: https://doi.org/10.3390/math10050696

# AGRADECIMIENTOS

A todo el **grupo AUDIPOC España**, en especial al *Dr. D. Francisco Pozo-Rodríguez*, por facilitar sin barreras alguna, que pudiera utilizar la información de esta base de datos de la que tuve acceso durante todo el tiempo de desarrollo del proyecto y soy conocedora del amplio registro recogido en la base de datos central y sus versiones generadas, puesto que participe en el proyecto como data manager y analista de datos, pero en este momento, solo se presenta una selección reducida de la base de datos como soporte analítico para el desarrollo de esta tesis doctoral.

A mis **Directores y Tutor de Tesis**, en especial al Prof. *Dr. D. Miguel Ángel Montero-Alonso* por su interés y colaboración constante, y por creer siempre en mí y estar ahí, aunque sea en la distancia, en todos los momentos, buenos y malos.

A todo el **Personal de la Universidad de Granada**, **a los Coordinadores de este Programa de Doctorado y a la Escuela de Posgrado**, en especial al *Departamento de Estadística e Investigación Operativa*, por el compromiso incondicional con los alumnos para que puedan formarse y crecer profesionalmente, y al mismo tiempo agradecer a todos por ser siempre grandes facilitadores en la gestión con vuestro apoyo desinteresado para resolver todas mis dudas y consultas académicas orientándome de la mejor forma posible dentro de vuestras posibilidades y conocimientos normativos respecto a los diferentes procesos burocráticos que nos encontramos a lo largo de este camino doctoral.

También, a todas **aquellas personas, amigos, compañeros, jefes y conocidos** que se sumaron a este tren tan largo de la vida, mediante su ayuda y apoyo incondicional, de manera directa e indirectamente con solo su presencia para que esta investigación pudiese ver su fin algún día de estos y que ya no es tan lejano.

Asimismo, dar las gracias a todas **aquellas personas que no creyeron en mí** de ninguna de las maneras y me abandonaron a lo largo de esta travesía sin prestarme ningún tipo de ayuda, sin valorar las alternativas disponibles y existentes para intentarlo, solo deciros que no os guardo rencor alguno, porque aprendí a ser más fuerte y perseguir mi sueño propuesto, ya que como siempre digo quién la sigue la consigue al final a pesar de los obstáculos en el camino.

Y por último **a mi familia, madre, hermana y hermano**, por su apoyo incondicional, por entender que este camino era lo que quise elegir y respetarlo a su manera, y por estar siempre ahí sin apenas decir ninguna palabra, pero que con tan solo una mirada lo expresan todo, puesto que el silencio es algo positivo en la vida.

Por ello, **MUCHAS GRACIAS a TODOS por formar parte de mi vida y estar siempre ahí de cualquiera de las formas, pero siempre a mi lado, avanzando** en este fantástico camino de la vida, donde siempre nos enriquecemos mutuamente y aumentamos nuestro crecimiento personal y profesional de cualquiera de las maneras presentadas.

*A mi abuela como segunda madre,*
*In Memoriam*

# TABLA DE CONTENIDO

# INDICE DE TABLAS Y FIGURAS

## LISTADO DE TABLAS

# LISTADO DE FIGURAS

# RESUMEN

En la actualidad, el gran avance tecnológico y la transformación digital originada con el Big Data y la Inteligencia Artificial (IA), están desarrollando diversos cambios de gestión y de decisión en todos los ámbitos profesionales, concretamente en el campo de la salud y en la minería de datos, y así mismo en la sociedad en general, que de alguna manera hay que agradecer a estas transformaciones, puesto que todo este proceso implica una nueva era de novedosos métodos y algoritmos menos robustos y más eficaces, capaces de ser perfeccionados para dar diferentes vías de solución a cualquier objetivo planteado.

En paralelo, estos nuevos mecanismos están experimentando cambios en el área de la estadística computacional que en tiempos pasados eran impensables por los costes tan inmensos que eso podría suponer y los procedimientos de cálculos tan arduos que eso implicaba. Por eso, estos desarrollos tan diversos en los diferentes campos de la informática y en especial para las áreas de investigación y ciencias de datos están generando técnicas más sofisticadas y adaptables para los distintos casos que se pueden encontrar en la población de interés, en especial en el ámbito sanitario, creando mejores modelos y resultados de calidad con el fin de ayudar en la toma de decisiones, y por consiguiente, proponer mejores procedimientos de diagnósticos y de tratamientos, adaptados al individuo para intentar paliar las posibles secuelas con la finalidad de mejorar la calidad de vida en el mayor tiempo posible, cambiando los hábitos saludables mal adquiridos y fortaleciendo los nuevos que se intentan alcanzar o se desean modificar a los que ya existían en nuestras vidas.

En esta investigación nuestro objetivo ha sido el de explorar las diversas técnicas no paramétricas existentes para la búsqueda de perfiles clínicos subsanando en paralelo la problemática de la maldición de la dimensionalidad y el hándicap de los valores faltantes (*missing values*), mediante algoritmos supervisados y no supervisados con las capacidades ofrecidas para su aplicación práctica desde software estadístico R, con el fin de poder dar una vía rápida al objetivo principal de este estudio, que es la búsqueda de perfiles clínicos en base de datos multicéntricas con diferentes patologías a través de las diferentes técnicas multivariantes.

En este sentido, se aplicó el método de imputación MICE, aunque existen otros métodos mencionados para apaliar los datos faltantes, por ser una técnica que utiliza ecuaciones encadenadas en el proceso de imputación aleatoria de cada variable, y estas están condicionadas a las variables imputadas, conservando la dependencia en la estructura de correlación del algoritmo y preservando la calidad relacional del conjunto original, que es uno de los aspectos relevantes para aplicar la técnica del análisis

de componentes principales (PCA), puesto que se conoce que la estructura de correlación puede ser bastante sensible a las distintas técnicas de imputación, siendo estas necesarias de estudiar antes de aplicarlas según el tipo de variables del conjunto de datos.

Asimismo, se abordó el problema de la dimencionalidad mediante tres técnicas diferentes, como son (i) análisis de componentes principales (*PCA)*; (ii) métodos Random Forest por Gini Index & Information Value por aplicación Weight-Of-Evidence (*RF&IV)* para definir la selección de importancia de variables y disminuir eficientemente la dimensión espacial*; y (iii)* análisis paralelo con datos simulados y de remuestreo (*APS-REM)* basado en la matriz de correlaciones aleatoria, obteniendo la mejor reducción a través del análisis PCA con 12 componentes principales siendo las dos primeras las más relevantes.

Finalmente, se aplicaron varias técnicas de clasificación supervisada y no supervisada, donde el algoritmo *Cluster* es la base central para el agrupamiento, originando tres grupos óptimos de patrones clínicos afines a sus propias características, y los otros métodos clasificatorios, como son Correspondencias (*CA*), Árbol de Decisión (*DT*) y Vectores Soporte (*SVM*), sirviendo de apoyo visual para detectar posibles grupos y a la vez, como mecanismo exploratorio para confirmar resultados sobre la información existente, dando un gran valor al resultado final óptimo alcanzado.

En conclusión, se pretende mostrar que el abordaje de estas técnicas pueden servir para distintas situaciones en lo que se presente un volumen suficientemente grande de datos, donde es casi necesario una reducción del espacio dimensional a otro de menor dimensión semejante al original, supliendo los problemas de valores faltantes para un buena calidad de la información, y aplicando modelos clasificatorios para la búsqueda de patrones de perfiles clínicos con el fin de agrupar a los pacientes de forma eficiente y precisa y a su vez, poder extrapolar los resultados clínicos en estudios de investigaciones similares. Además, este planteamiento primario, será muy necesario en poco tiempo con la nueva iniciativa de la Unión Europea (UE), en el que se ha propuesto la creación del Espacio Europeo de Datos Sanitarios (EEDS) para todos los países miembros, lo que generara un volumen inmenso de datos sanitarios que requerirán de técnicas más sofisticadas para destacar la información relevante e indispensable que puedan ayudar a la toma de decisiones.

# INTRODUCCIÓN

## 1. Antecedentes y situación actual

A día de hoy, se está generando una gran cantidad de información de datos de todas las especialidades, en especial en la rama sanitaria, donde se requiere casi de forma imprescindible un abordaje rápido para poder estructurar y organizar todos estos datos creados y almacenados en grandes repositorios, que se alojan en diferentes tipos de plataformas, en nube o en local, con el fin de poder darles valor mediante la transformación y generación de datos de calidad para el desarrollo de estudios e investigaciones suficientemente fiables como para apoyarse en la toma final de decisiones clínicas, que estén persiguiendo como finalidad la mejora de terapias, tratamientos, diagnósticos y así como en paralelo, la prevención de enfermedades, en especial las crónicas, antes del inicio de la etapa de desarrollo para que se tenga un retorno positivo si se puede, y una mejor calidad de vida en estos pacientes explorados, (López-Campos et al., 2021).

En este sentido, los avances de la bioinformática apoyados en la Inteligencia Artificial (IA) y en la metodología de algoritmos de Machine Learning (ML) pueden aportar esta tecnología puntera de gran auge e impacto en la sociedad actual, sirviendo de base para el análisis, la construcción de modelos testados y fiables que ayuden a prevenir y reconocer con antelación suficiente estos diagnósticos, así como distintas mejoras que se están desarrollando en muchas de las especialidades de la medicina, con la finalidad de identificar, explicar y agrupar patrones que den soporte a las diferentes hipótesis planteadas de una investigación que este en pleno desarrollo ó para resolver un caso particular.

Por eso, este proceso dara origen a una medicina personalizada, preventiva y predictiva con algoritmos diseñados y personalizados suficientemente capaces para dar respuesta a casi todos los hallazgos que hasta ahora no han podido tener respuesta por no disponer de estas nuevas tecnologías y avances digitales que desde este momento están en marcha para cambiar y mejorar la forma de pensar y de trabajar en todos los ámbitos de la vida y en general, de la investigación clínica, (Dollfus y Petit, 1995 y Blázquez-Sánchez et al., 2020).

A lo largo de muchos años, se han venido utilizando técnicas de clasificación de objetos (o más específicamente de pacientes) en el campo de la Estadística Aplicada y/o de la medicina clínica asistencial. Estos métodos de clasificación tienen como objetivo principal determinar a cuál de las clases existentes puede pertenecer una nueva observación o individuo, cuando se tiene un conjunto de

datos definido por una muestra extraída de la población y donde cada uno de sus elementos pertenecen a una de las clases definidas.

Por tanto, los métodos no paramétricos, en concreto los modelos de Machine Learning, pueden lograr una mejor adaptación a la información disponible sin formular modelos muy rígidos, originando nuevos enfoques técnicos y analíticos que implementan distintos métodos exploratorios, aportando un gran soporte a la toma final de decisiones en todo el ámbito científico y, en particular, en estudios multicéntricos. Es decir, la utilización y desarrollo de programas informáticos que se adapten y al mismo tiempo aprenden de si mismos sin seguir ninguna pauta en concreta, mediante el uso de algoritmos y modelos estadísticos con el fin de analizar y extraer inferencias de patrones sobre los datos explorados.

## 2. Objetivo principal de esta investigación

El *objetivo de esta investigación doctoral* se centra principalmente en explorar y dar a conocer de la existencia de las diferentes técnicas multivariantes que puedan abordar los problemas que aparecen habitualmente, como son la maldición de la dimensionalidad, la búsqueda eficiente de patrones y el hándicap de los valores faltantes (*missing values*), sobre todo cuando se trabaja con un volumen grande de datos o se realizan estudios multicéntricos que registran bases de datos de alta dimensión con un gran porcentaje de datos faltantes, donde es casi necesario aplicar métodos de reducción dimensional y de imputación de missings para no perder la información relevante o el enfoque principal de la investigación desarrollada.

Paralelamente, esta tesis tiene como enfoque primordial la aplicación de distintos algoritmos para este tipo de bases de datos de estudios multicéntricos que presentan distintas patologías con la finalidad de buscar grupos de perfiles clínicos óptimos, clasificando a los diferentes pacientes según sus propias características, dando una solución factible y mejorada para el soporte de la toma de decisiones en casos similares donde se necesita realizar varias particiones del conjunto de datos poblacional para encontrar una mejor respuesta clínica basada en procesos y diagnósticos más efectivos.

## 3. Descripción de la fuente de información

En el caso que nos ocupa, los análisis de este estudio se desarrollaron con la fuente de información procedente de la base de datos original del proyecto AUDIPOC España (Pozo-Rodríguez et al., 2010), que es un estudio transversal con seguimiento a 90 días sobre la atención clínica recibida y los desenlaces experimentados por los 5.178 pacientes ingresados con exacerbación de EPOC (Enfermedad Pulmonar Obstructiva Crónica) en los 129 hospitales participantes.

En este proyecto, el criterio de inclusión del paciente en el estudio para el diagnóstico definitivo de EPOC se realizó por diferentes variables clínicas, que eran relevantes para definir el grado de la enfermedad en cuestión y poder concluir con el diagnostico final de eEPOC (exacerbación de la Enfermedad Pulmonar Obstructiva Crónica), como son:

1) "Ingresado por diagnóstico principal de eEPOC"

2) "[(Ingreso por Patología respiratoria) y (Diagnóstico previo de EPOC o espirometría documentada previa con FEV1/FVC < 70% no reversible)]"

Asimismo, las patologías asociadas a estos pacientes son diversas destacando estos nueve eventos entre otros más; y que muchas veces son causadas por el propio avance de la enfermedad principal diagnosticada clínicamente por el personal sanitario responsable.

(1) Insuficiencia Cardíaca Congestiva

(2) Comorbilidad Cardiovascular

(3) Diabetes Mellitus

(4) Enfermedad Cerebro Vascular

(5) Enfermedad Vascular Periférica

(6) Infarto de Miocardio

(7) Nefropatía

(8) Tumor sólido

(9) Edemas Maleolares

Además, de estas enfermedades asociadas a la principal, que el paciente puede tenerlas todas o alguna de ellas (*Score de patologías* "SCORE_PAT_"). Es decir, suma de las enfermedades sin contar la principal, ya que se presupone que todos estos pacientes están afectados por ella.

También se recogieron otros variables relevantes al individuo, a parte de las variables comunes (*edad*, *sexo* y *hábito tabáquico*), como son:

1) La duración de ingreso en el centro hospitalario

2) Si tuvo algún ingreso hospitalario por cualquier motivo en los 90 días posteriores

3) Si reingreso por exacerbación de EPOC a los 90 días respecto a la fecha de ingreso

4) Si los pacientes murieron a los 90 días

5) Si recibió soporte ventilatorio en cualquier momento del ingreso

6) Fallecimiento del paciente durante todo el periodo de ingreso (EXITUS)

Asimismo, para realizar un *seguimiento de estos pacientes* con profundidad tiene interés analizar los datos *espirométricos* de las pruebas asociadas a la espirometría (FVC%, FEV1%, FEV/CVF%) y también ver qué sucede con los datos *antropométricos* como pueden ser: la altura del paciente (m), el peso (kg), el índice de masa corporal (IMC), la tensión arterial diastólica (TARD), la tensión arterial sistólica (TARS) y la temperatura ($^{o}$C), así como los datos de la *exploración física* como pueden ser: la frecuencia respiratoria (FRE) y la frecuencia cardíaca (FCA), etc.; con la finalidad que estas nuevas variables incorporadas puedan ayudar a mejorar la interpretación de los resultados clínicos.

Con este planteamiento la aportación que se pretende realizar es un acercamiento más concreto de la información de la base de datos, obteniendo representaciones específicas sobre estos pacientes con el fin de agrupar las diferentes patologías anteriormente citadas, pudiendo detectar las asociaciones existentes entre ellas, originando distintos grupos de pacientes, es decir, variantes de perfiles clínicos útiles para la práctica clínica.

Por eso, para el análisis de los datos se ha realizado solo una selección de 32 variables del total de la base de datos original, por ser las más relevantes para este estudio planteado, (ver listado descrito en el *Anexo A* de esta tesis), pudiendo cumplir satisfactoriamente con el objetivo principal del mismo, sin olvidar las variables importantes que se incorporan en el estudio de clasificación, ya que estas patologías tienen bastante relación con la enfermedad principal y van apareciendo en el paciente

conforme avanza el estado de gravedad del mismo, lo que serán útil para el agrupamiento y por consiguiente, la búsqueda de patrones óptimos.

## 4. Organización de la tesis y aportación

En este sentido esta investigación se ha *estructurado en cuatro capítulos*, donde el procedimiento completo y los cálculos implementados se realizan con el software estadístico R (R Core Team, 2021) (versión 4.1.0), y todos los contrastes de hipótesis se realizan con un p-valor al 0.05.

Asimismo, el ***primer capítulo*** tiene como finalidad establecer los elementos teóricos fundamentales para abordar el gran problema de los valores missings en los registros de datos según el tipo de variables recogidas, puesto que en muchas ocasiones son un porcentaje bastante alto, que requieren de una atención analítica-exploratoria y de técnicas específicas para mejorar la información almacenada en cada uno de los registros institucionales y sanitarios. Por eso se recuerda de la existencia de diversos mecanismos, como son estos: *Amelia (Honaker et al., 2011); missForest (Stekhoven y Buehlmann, 2012 y Stekhoven, 2022); Hmisc (Harrell Jr y Dupont, 2021); mi (Su et al., 2011); regresión; redes neuronales (Venables y Ripley, 2002)*, para apaliar la mala calidad de los datos que limitan el uso de los mismos provenientes de los sistemas de información de salud de rutina, siendo los valores faltantes un componente importante de este problema y donde los propios organismos no informan de esta problemática existente para tomar medidas preventivas al respecto con el fin de mejorar el proceso de recogida diaria de la información.

En este caso, la aplicación del MICE (Van Buuren y Groothuis-Oudshoorn, 2011) es una de las alternativas acertadas para poder completar esta fuente de datos satisfactoriamente, ya que se sabe que el método MICE utiliza las ecuaciones encadenadas en el proceso de imputación aleatoria de cada variable, y estas están condicionadas a las variables imputadas, aplicando un mecanismo de cadenas dependientes en la distribución de probabilidad. Por lo tanto, la dependencia se conserva en la estructura de correlación del algoritmo de imputación, cuando se modifican los valores faltantes (*missing values*), manteniendo la calidad relacional del dataset original. Además, este aspecto es importante puesto que se conoce que la estructura de correlación puede ser bastante sensible a las distintas técnicas de imputación, y se deberían estudiar a fondo antes de aplicar cualquier método según el tipo de variables del conjunto de datos, para no perder la calidad relacional de origen, ya que es uno de los requisitos destacables para aplicar la técnica del análisis de componentes principales (PCA), que se describe en el capítulo posterior.

Por otro lado, el ***segundo capítulo*** pretende tratar el tema de la dimensionalidad mostrando distintas vías para resolverlo, mediante tres técnicas multivariantes diferentes: (i) análisis de componentes principales (*PCA*) para la reducción de la dimensión presentada; (ii) algoritmos Random Forest & Information Value (*RF&IV*) para definir la selección de importancia de variables como son los métodos RF por Gini Index & IV por aplicación weight-of-evidence (WOE) con el fin de contrastar y disminuir eficientemente la dimensión espacial de los datos-variables explorados; y (iii) análisis paralelo con datos simulados y de remuestreo (*APS-REM)* basado en la matriz de correlaciones aleatoria. Por ello, en este caso, la aplicación del PCA es la que ha obtenido la mejor reducción dimensional con 12 componentes pirncipales de las 32 iniciales para este estudio, y siendo la primera dimensión la de mayor relevancia, generando la existencia de varios grupos de pacientes con caracteristicas similares, que presentan de forma conjunta variables muy asociadas dentro de cada uno de ellos con suficiente importancia a considerar en cada perfil clínico obtenido. Así mismo, se puede confirmar que la reducción del plano ayuda significativamente a detectar a grandes rasgos, sin profundizar en la clasificación, la visualización de primeros indicios de diferentes patrones clínicos con características semejantes. También, para futuros análisis se mencionan otros procedimientos de la competencia, como son *t-SNE* (Krijthe, 2015); *Sammon mapping* (Venables y Ripley, 2002 y You, 2022); *Isomap* (Oksanen et al., 2022); *LLE* (Holger, 2015); *CCA* (González et al., 2008 y González y Déjean, 2021); *MVU* (You, 2022); *LE* (Kraemer et al., 2018 y Kraemer, 2022), que pueden ayudar a obtener mejores resultados en los métodos mencionados. Ya que se sabe que el análisis PCA tiene límites para algunos casos, puesto que este método sólo usa las combinaciones lineales de las variables originales y se puede llegar a perder mucha información.

Además, estas técnicas sólo pretenden dar soporte a la búsqueda de reducción de la selección de importancia de los factores-variables, simplificando el espacio de análisis y quedándonos sólo con la información relevante y precisa, con la finalidad de dar una mejor solución clínica que pueda ser un reflejo real de la población actual, y en paralelo, generar la clasificación de grupos de patrones con afinidades idénticas, proponiendo a la vez, esta vía de análisis para la problemática de la reducción del plano.

En este sentido, el ***tercer capítulo*** se centra primordialmente en dar respuesta al objetivo principal de esta investigación, mediante el análisis de clasificación de individuos, usando un modelo no supervisado para el acercamiento de la búsqueda de perfiles clínicos en base de datos multicéntricas con diferentes patologías. Para ello, este análisis se enfoca en tres métodos multivariantes distintos, siendo uno de ellos el más relevante para detectar la separación de los grupos por características afines, (i) es el análisis Cluster o de Conglomerados, que consiste en una técnica de clasificación por grupos no

supervisada, donde las clases no están predefinidas de antemano con el fin de identificar grupos de individuos similares sobre la segmentación de una población heterogénea, y los otros dos, que son (ii) el de *Correspondencias* (CA), que separa por grupos sin tener en cuenta afinidades, pero es un gran apoyo de visualización global en este caso clínico; y (iii) el de *Árboles de Decisión por clasificación o Decision Tree* (DT), que es otro gran soporte para detectar alguna clasificación óptima entre las distintas variables clases mostrando las diferentes hojas mediante la aplicación de pequeñas reglas de decisión de orden jerárquico que van originando la decisión final en forma de árbol.

Por eso, estos dos últimos métodos forman parte complementaria para el soporte visual con el fin de perfeccionar o detectar cualquier información a tener en cuenta en el agrupamiento de los datos de pacientes con las características de variables presentadas y exploradas.

En este sentido, el análisis Cluster o de Clasificación ha sido una buena elección de aplicación para este caso particular, ya que se ha podido obtener como resultado final óptimo a 3 grupos de perfiles clínicos dierentes entre sí, pero con caracteristicas similares y afines en el interior de ellos, culminando satisfactoriamente con el enfoque inicial de esta investigación, que pretendía buscar patrones de perfiles afines con el fin de clasificar a cada individuo de paciente en distintos grupos que presenten el mismo comportamiento, de tal manera que se pueda estudiar ampliamente sus propiedades o características con la finalidad de mejorar sus procesos clínicos y asistenciales asignados, y por consiguiente poder extrapolar sus resultados y conclusiones a la población general.

Y para finalizar este estudio, en el ***capítulo cuatro*** se desarrolla el tema de las máquinas de vectores soportes (SVMs), (Muñoz et al., 2019 y Moguerza et al., 2020), con la selección de diferentes Kernels (Martín de Diego et al., 2010; Martos et al., 2014 y Muñoz et al., 2018) para contrastar algunos resultados destacados, ya que se sabe que los SVM pueden ser otro enfoque diferente a la solución del problema planteado para bases de datos de alta dimensión, puesto que tienen como objetivo central buscar un hiperplano de separación entre las categorías de las dos clases o instancias frontera de la barrera de decisión, mediante un margen (distancia) máximo.

En esta línea, se plantea este análisis con dos enfoques diferentes, uno con datos simulados y otro con datos reales para abordar el procedimiento de clasificación con los SVM como otra buena alternativa de clasificación y reducción para diferentes casos propuestos desde esta investigación, y como soporte adicional para detectar alguna información relevante que pueda completar la búsqueda de perfiles clínicos finales.

A modo de conclusión, se puede decir que los diferentes métodos analizados han permitido acercarnos mucho más en detalle a los modelos y algoritmos de clasificación supervisados y no supervisados, con el fin de agrupar a los individuos de una manera eficiente y acertada, sin perder información importante del conjunto de datos original a través de mecanismos de reducción de dimencionalidad, y con una buena calidad de los datos mediante la aplicación de técnicas de imputación de valores perdidos, dando una solución factible y óptima de perfiles de agrupamiento, que puede ser extrapolable para un caso similar en la población de interés.

En este mismo sentido, la ***aportación que se ofrece en esta tesis doctoral*** es dar a conocer de la existencia de diferentes vías para el abordaje técnico y analítico cuando se está frente a grandes volúmenes de datos o registros de grandes dimensiones que es casi obligatorio y necesario una reducción factible sin perder información relevante y en paralelo apaliar el obstáculo de los valores faltantes para disponer de datos de calidad y precisos y por consiguiente, poder encontrar patrones afines con la finalidad de mejorar y ayudar en la toma de decisiones clínicas, mediante resultados más precisos, personalizados y preventivos para una mejor calidad de vida, en especial para los pacientes con enfermedades crónicas, que tienden a un deterioro bastante acusado con el paso del tiempo y la aparición de otras comorbilidades añadidas, pero gracias a la globalización digital que se está experimentando se podrá abordar con exactitud todos estos temas clínicos con resultados bastante significativos y de gran valor para la sociedad en general.

De hecho, este planteamiento primario, será muy necesario en poco tiempo con la nueva iniciativa de la Unión Europea (UE), en el que se ha propuesto la creación del Espacio Europeo de Datos Sanitarios (EEDS) para todos los países miembros, lo que generará un volumen inmenso de datos sanitarios que requerirán de técnicas más sofisticadas, (como son las de reducción de dimensionalidad, de imputación múltiple de valores faltantes o las de análisis de clasificación mixta de algoritmos de aprendizaje supervisado y no supervisado), para destacar la información esencial e indispensable que puedan ayudar a la toma de decisiones (Comisión Europea, 2022).

# CAPÍTULO I

# Imputación de valores faltantes

## I. 1. Introducción

En la actualidad, la era tecnológica y la transformación digital originada con el *Big Data* y la *Inteligencia Artificial* (IA) están cambiando la forma de pensar y actuar en todos los ámbitos de trabajo, así como en el sanitario, originando nuevos enfoques técnicos y analíticos que implementan distintos métodos exploratorios, que están aportando un gran soporte a la toma final de decisiones en todo el ámbito científico y, en particular, en estudios multicéntricos.

Este avance de innovación tecnológico, paralelamente, genera nuevas problemáticas al trabajar con grandes repositorios de datos, como es el inconveniente de los datos *missings*, (Santana, 2015; Van Buuren y Groothuis-Oudshoorn, 2011 y Villagarcía y Muñoz, 1997), el problema de la maldición de la dimensionalidad (Choubey et al., 2020), o como en este estudio, destaca la necesidad de explorar formas más efectivas, enfocado en mejorar nuevas aportaciones con la búsqueda de perfiles clínicos.

En la mayoría de los campos de investigación, la *falta de datos* es un problema común e introduce un elemento de ambigüedad en el análisis de datos, que puede surgir por diferentes motivos, bien por el mal manejo de las muestras, un error de medición, un valor anómalo eliminado o simplemente por la falta de análisis. En particular, se sabe que en los estudios en el ámbito de ciencias de la salud, a menudo, están plagados de valores perdidos (Blazek et al., 2021) que pueden reducir en gran medida el tamaño de la muestra si solo se consideran casos completos para el análisis, y por consiguiente, si estos análisis ignoran los datos faltantes tienen el potencial de introducir sesgos en las estimaciones de los parámetros.

Por este motivo, la *imputación múltiple por ecuaciones encadenadas* (*MICE*) (Van Buuren y Groothuis-Oudshoorn, 2011) ha surgido como una estrategia líder para imputar datos epidemiológicos faltantes debido a su facilidad de implementación y capacidad para mantener estimaciones de efectos no sesgadas e inferencias válidas. Dentro del algoritmo *MICE*, la imputación se puede realizar utilizando una variedad de métodos paramétricos o no paramétricos. En este aspecto, la literatura indica que los métodos de imputación no paramétricos basados en árboles superan a los métodos paramétricos en

términos de sesgo y cobertura cuando hay interacciones u otros efectos no lineales entre las variables. Sin embargo, estos estudios no proporcionan una comparación justa, ya que no siguen la recomendación establecida de que cualquier efecto en el modelo de análisis final (incluidas las interacciones) debe incluirse en el modelo de imputación paramétrica. De hecho, se ha demostrado mediante simulación que la incorporación adecuada de interacciones en el modelo de imputación paramétrica conduce a un rendimiento mucho mejor, (Shah et al., 2014 y Hong y Lynn, 2020).

Por ello, en este capítulo se aborda la problemática de los valores perdidos, utilizando dos técnicas diferentes del MICE, el *método sample* que realiza una muestra aleatoria simple desde los valores observados y los devuelve como imputaciones, suponiendo que los datos son MAR (*Missing at Random,* que faltan por el azar), y el *método Predictive Mean Matching* (*PMM*) que se encarga de imputar por coincidencias de medias predictivas, es deicr el método PMM del paquete MICE es un método de imputación semiparamétrico, donde los valores imputados coinciden con alguno de los valores observados en la misma variable, preservando las relaciones no lineales incluso cuando parte de la estructura del modelo de imputación es incorrecta.

En este caso, se descarto el método *mi* de R (Su et al., 2011) porque puede originar una imputación de valores missings poco adecuada o incorrecta, a pesar de que también emplea la imputación múltiple por variables, generando diferentes modelos mediante la coincidencia de medias predictivas para acercarse a los valores faltantes. Es decir, para cada observación de la variable con un valor faltante, se encuentra la observación con la media predictiva más cercana para esa variable (entre aquellas con valores observados en esa variable), usando este valor observado de esta "coincidencia" como valor imputado.

Obviamente, uno de los motivos por el cual no se ha optado por este algoritmo *mi* es debido a que, este método puede ser problemático cuando las tasas de faltantes son altas o cuando los valores faltantes se encuentran fuera del rango de los datos observados, puesto que las predicciones se obtienen con la función bayesiana de familia gaussiana.

Además, el proceso con *mi* puede tardar bastante tiempo en converger con grandes conjuntos de datos con una alta tasa de faltantes en muchas variables. Puesto que, este enfoque de imputación múltiple requiere de cuatro pasos para su aplicación, como son (i) configuración, (ii) imputación, (iii) análisis y (iv) validación, donde cada uno de ellos se subdividen en otros pasos diferentes dentro del principal.

También, es cierto que, aunque esta técnica dispone de una selección razonablemente amplia de modelos, aún es posible que ninguno proporcione un ajuste completamente adecuado para los datos, lo que podría conducir a una imputación incorrecta.

Por lo tanto, para lograr un ajuste apropiado puede ser complicado cuando existen restricciones en los datos, pero en este sentido, el propio algoritmo de imputación ofrece una solución ante este problema, dando la opción del método de coincidencia de medias predictivas, que ya se ha mencionado anteriormente, similar al aplicado con el MICE, donde precisamente es uno de los procesos implementados en este caso particular para resolver el tema de los valores faltantes.

## I. 2. Métodos

Para tratar la problemática de los missings data y en paralelo, el tema de la dimensionalidad que se tratará en el capítulo II de este trabajo, y motivados por este aspecto de agrupamiento tan característico, en la actualidad, las guías de práctica clínica inciden con gran énfasis en mejorar los modelos de atención sanitaria existentes mediante la clasificación de los pacientes atendidos, a través de la realización de un diagnóstico adecuado y una correcta gestión asistencial-clínica, ya que, la evolución de las comorbilidades sobre este tipo de pacientes representan un reto asistencial enorme que dependen de otros factores, como pueden ser, la adhesión a los tratamientos asignados, las intervenciones en hábitos de vida saludables o la capacidad de desarrollar ciertas habilidades para reconocer los signos y síntomas de la exacerbación con el fin de prevenir-tratarlos mediante un plan de autogestión propio.

De esta problemática existente, esta investigación se centra en una selección de variables epidemiológicas, clínicas y de desenlaces (32 variables) con el fin de dar sentido al objetivo principal planteado de mejorar la reducción dimensional para un adecuado agruapamiento de los pacientes, y en paralelo indagar en la búsqueda de perfiles clínicos con varias características asociadas al desarrollo de una enfermedad principal crónica, permitiendo dar una nueva vía de acceso para situaciones similares en estudios multicéntricos.

Debido a esta necesidad, se aborda este problema planteado con una solución factible de imputación multivariante para resolver la temática de los valores perdidos mediante el método *MICE* (Karthe, 2016) con diferentes técnicas para contrastar resultados, como *Sample y Predictive Mean Matching* (Miri et

al., 2020), que ofrecen un funcionamiento correcto con estos datos clínicos, reduciendo el sesgo en el proceso de selección de características.

En este sentido, los *missing values* se consideran el primer obstáculo en el modelado cuando existe una pérdida de información entre el 5-20% o más, promoviendo la realización de determinadas técnicas de imputación para dicha problemática, ya que se conoce que la elección del método de imputación influye en gran medida en la capacidad predictiva del modelo, (Marston et al., 2010 y Shah et al., 2014). En la mayoría de los métodos de análisis estadístico, la eliminación por lista es el método predeterminado que se utiliza para imputar los missings. Aunque, no es tan bueno, ya que conduce a la pérdida de información y algunas ocasiones se hace casi necesario que permanezcan esos datos para dar significado a otras variables-factores poco relevantes con interés de análisis.

Por esta razón, algunos paquetes funcionan mejor con variables continuas y otros con categóricas, por lo que se puede tomar una mejor decisión al elegir el paquete que mejor se adapte a los datos en estudio. En nuestro caso, se ha elegido utilizar MICE (*imputación multivariante mediante ecuaciones encadenadas*) por ser uno de los paquetes más utilizados en R (R Core Team, 2021) y por la creación de múltiples imputaciones, que *ayudan a reducir el sesgo y aumentar la eficiencia,* en comparación con una sola imputación (como la media) que se encarga de la incertidumbre en los valores perdidos.

Además, una de las características relevante es que MICE asume que los datos faltantes son *Missing at Random* (*MAR*), lo que significa que la probabilidad de que falte un valor depende solo del dato observado y se puede predecir usándolos.

Se recuerdan los diferentes tipos de definiciones para valores perdidos (Ortiz y González, 2015), que pueden encontrarse en cualquier registro de datos incompleto, que son los siguientes descritos:

- *MCAR* (*Missing Completely at Random*): no existe relación entre los datos faltantes ni son una submuestra del conjunto de datos por lo que se debe simplemente a la aleatoriedad. Es decir, el valor perdido falta por completo al azar y el proceso no depende de ninguna de las observaciones por ser un mecanismo aleatorio.

- *MAR* (*Missing at Random*): se da cuando la propensión o tendencia a que un dato (observación completa de una variable) falte no está relacionada con los datos faltantes sino con alguno de los datos observados de la variable. Es decir, dado un conjunto de datos $\{X_1, \ldots, X_m\}$ con $n$ variables $X_1, \ldots, X_n$, donde $X_{i1\text{-}m}$ es un valor del mismo, este puede ser explicado por el resto de valores $X_i$, puesto que

su falta es aleatoria y no sigue ningún patrón. Por lo tanto, el valor faltante depende solo del dato observado de la variable, caso descrito para este estudio.

- *MNAR (Missing Not at Random)*: se presenta cuando no se aplican las definiciones anteriores de MCAR y MAR. Los datos faltantes dependen de los propios datos no observados de las variables y no se puede predecir el modelo de otras observaciones sin tener sesgo.

Para este caso particular, se asume que los datos perdidos son datos MAR, porque el valor faltante está relacionado con algunos de los datos observados de estas variables medidas, que no se completó debido a la falta de seguimiento clínico. En este sentido, se visualizan los datos observados y faltantes para ver si presentan algún tipo de comportamiento específico, y el valor faltante muestra un patrón general con datos faltantes dispersos aleatoriamente a lo largo de la matriz de datos, pudiendo estar condicionado por las observaciones de otras variables del conjunto de datos diferentes al principal, por lo que se pueden estimar a partir de estas observaciones, y por lo tanto, se asume que los datos missings son MAR.

También, una de las ventajas del MICE es que realiza la imputación de los datos variable por variable (especificando un modelo de imputación por variable) y asimismo, es capaz de manejar diferentes tipos de variables y puede gestionar la imputación de variables definidas en un subconjunto de datos.

Para solventar esta problemática, existen otros métodos según el tipo de variables que se este tratando de completar, como son *Amelia* (Honaker et al., 2011), *missForest* (Stekhoven y Buehlmann, 2012 y Stekhoven, 2022), *Hmisc* (Harrell Jr y Dupont, 2021) *y mi* (Su et al., 2011)*,* todos ellos para la imputación de datos faltantes, (Ispirova et al., 2020).

- El método *Amelia* (Honaker et al., 2011; Zahid y Heumann, 2019 y Zhang, 2016) aplica el algoritmo EMB basado en bootstrap (*Expectation-Maximization with Bootstrapping*) que lo hace más rápido y robusto. Y aunque, realiza imputaciones múltiples reduciendo el sesgo y aumentando la eficiencia, utiliza un enfoque de modelado conjunto en las estimaciones basado en una distribución normal multivariante (*MVN*) por lo que las variables deben distribuirse o acercarse a la normalidad. Sin embargo, *MICE* imputa datos variable por variable, siendo capaz de manejar diferentes tipos de variables, y pudiendo gestionar la imputación de variables en un subconjunto de datos, aspecto procedimental que no puede realizar *Amelia* (*MVN*).

- Por otro lado, *missForest* (Stekhoven y Buehlmann, 2012; Alsaber et al., 2021; Hong y Lynn, 2020; Lenz et al., 2020 y Stekhoven, 2022) es una implementación del algoritmo *Random Forest* (*RF*), por lo que este método iterativo no paramétrico es aplicable a varios tipos de variables, y consiste en crear un modelo *RF* para cada variable con el fin de usarlo para predecir los *missings* en la variable, ayudandosé de los valores observados. Este método suele funcionar bien en variables categóricas, por lo que no es necesario eliminarlas del conjunto, ya que se ocupa del valor perdido perteneciente a cualquier tipo de variable, considerando el *error en la estimación del modelo* (*OOB*), y originando diferentes umbrales OOB (*out-of-bag*). Con esta técnica se proporciona un alto nivel de control sobre el proceso de imputación, pudiendo ajustar el número de variables muestreadas aleatoriamente en cada división. Además, este método puede superar a *Hmisc* si las variables observadas y suministradas contienen suficiente información.

- Asimismo, el método *Hmisc* (Delisle et al., 2018 y Harrell Jr y Dupont, 2021) es un paquete de usos múltiples, útil para análisis de datos, gráficos de alto nivel, imputación de valores perdidos, creación avanzada de tablas, ajuste y diagnóstico de modelos. Para la imputación de valores perdidos reconoce automáticamente los tipos de variables y utiliza una muestra de arranque y una coincidencia de medias predictivas para imputar los *missings*. Dentro de su amplia gama de funciones tenemos dos potentes: impute() y *aregImpute()*, la primera simplemente imputa el valor perdido utilizando el método estadístico definido por el usuario (media, máxima, mediana siendo el valor predeterminado), y la segunda permite la imputación de medias mediante regresión aditiva, bootstrapping y coincidencia de medias predictivas, y destacando que este método asume linealidad en las variables que predice y utiliza la puntuación óptima de *Fisher* para predecir las variables categóricas sin necesidad de separarlas del conjunto como *MICE*.

- Y finalmente, el método *mi* (Su et al., 2011 y Luo et al., 2017) es un paquete para *Multiple imputation with diagnostics*. Este procedimiento consiste en crear varios modelos de imputación para aproximar los valores perdidos, utilizando el método predictivo de coincidencia de medias (PMM) como uno de ellos, ya que por defecto usa el método bootstrap (arranque), completando los datos faltantes con valores muestreados aleatoriamente de los observados. Paralelamete, puede detectar automáticamente irregularidades en los datos, como la alta colinealidad entre variables y agrega ruido al proceso de imputación para resolver el problema de las restricciones aditivas. Destacar que este método, puede ser complejo en los casos donde las tasas de missings son altas o se encuntran fuera del rango de los datos observados por utilizar en las predicciones la función bayesiana con familia gaussiana

"*bayesglm*()". Por eso, no es muy recomendable si se tiene este problema, puesto que puede llevar a imputaciones incorrectas, que se pueden solucionar usando el algoritmo de coincidencia de medias predictivas o estudiar otros métodos de imputación más eficientes y precisos que den un ajuste más acertado para los datos explorados.

Aunque, es cierto que este algoritmo emplea un proceso de cuatro pasos bastante exhaustivo, como son: (i) *Configuración* que se dividen en (a) Visualización de patrones de datos faltantes. (b) Identificación de problemas estructurales en los datos y pre-procesamiento. (c) Especificación de los modelos condicionales; (ii) *Imputación* que se divide en (a) Imputación iterativa basada en el modelo condicional. (b) Comprobación del ajuste de los modelos condicionales y comprobación para ver si los valores imputados son razonables. (c) Comprobación de la convergencia del procedimiento; (iii) *Análisis* que se divide en (a) Obtención de datos completos. (b) Agrupación del análisis de caso completo en conjuntos de datos de imputación múltiple; (iv) *Validación* que se divide en (a) Análisis de sensibilidad. (b) Validación cruzada. (c) Comprobación de compatibilidad. Pero, como bien se indica tiene ciertas limitaciones en determinados casos (restricciones de los datos, porcentaje alto de missings, existencia de heterocedasticidad en un modelo, es decir cuando la varianza de los errores no es igual en todas las observaciones realizadas, etc;), pudiendo abordarlos con el método *PMM*, que es semejante al implementado con el MICE.

Asimismo, cabe destacar en esta sección, que también existen otras técnicas de imputación de datos bastante conocidas y usuales en el área analítica computacional, como pueden ser los *métodos de regresión* y la *red neuronal* (Beysolow, 2017 y Theano, 2015), donde ambos procedimientos ayudan a completar esos valores faltantes en determinadas variables registradas de la base de datos, que por cualquier motivo se hayan detectado en el transcurso de la investigación, ocasionando la existencia de ciertas instancias incompletas del dataset con datos missings, y que son necesarias completar para obtener resultados de calidad que puedan ser extrapolados a la población general.

Más concretamente, el método de *regresión* (Santana, 2015), realiza una estimación de estos valores sobre los datos existentes, es decir, los valores faltantes para cada individuo se estiman mediante predicción a partir de la regresión de las variables conocidas para ese participante especifico, con la finalidad de completar la información ausente en determinados participantes de la base de datos analizada, debido a que, el personal clínico encargado de la entrevista no pudo tomar todas las medidas exploratorias cuando se estaba desarrollando la recogida de datos del estudio.

Por otro lado, el método de *redes neuronales* (Vaquerizo, 2014; López y Fernández, 2008 y Beck, 2013), podría ser una posible vía de mejora para una buena adaptación de estos datos, ya que este método se puede realizar desde el paquete *"nnet"* (Venables y Ripley, 2002) del propio software estadístico R. Esta técnica consiste en que la red está formada por un conjunto de neuronas (unidades) unidas unas a otras, donde cada una de las conexiones de las neuronas tiene un peso asociado y se utiliza mediante un algoritmo para encontrar los pesos de las conexiones entre las neuronas, organizándose las neuronas de la red en capas. Del mismo modo, la red organiza sus neuronas en capas, siendo la *primera capa la entrada a la red* a través de las observaciones de entrenamiento y la *capa final las predicciones* de la red neuronal para cualquier caso que se encuentre en la neurona de entrada. Asimismo, entre la capa inicial y la final existen más de una capa oculta, permitiendo que las salidas de la red se comporten según los casos presentados al modelo, mediante el *algoritmo de "back-propagation"* que trata de obtener los pesos de las conexiones, optimizando un cierto criterio de error para asegurar las salidas de la red, puesto que la gran mayoria de las funciones de activación son no lineales.

No obstante, aunque es un método bastante útil, tiene el mismo inconveniente que la gran mayoría de los procedimiento de imputación, y es que las redes neuronales son conocidas por ser sensibles a diferentes escalas de las variables utilizadas en un problema de predicción. En este sentido, se debe realizar la transformación de los datos antes de introducirlos a la red, evitando un posible impacto negativo en el desempeño de la misma.

En este caso, se normalizan los datos con la finalidad de que todas las variables tengan media cero y desviación estándar igual a uno, aplicando sencillamente la siguiente transformación a cada columna del conjunto de datos:

$$y_i = \frac{x_i - \bar{x}}{\sigma_x}$$

donde $\bar{x}$ es el valor medio de la variable original $X$ y $\sigma_x$ es la desviación estándar.

En este trabajo, se ha realizado una prueba de este método para experimentar como se adapta este algoritmo en las variables a imputar con esta técnica mediante el software estadístico R, aunque se conoce que no es el más acertado cuando la información tiene un alto porcentaje de missings.

Este proceso tiene como paso primero, iniciar con la división de la información en dos conjuntos "set" de datos para obtener mayor efectividad en la imputación. En primer lugar, hay que dividir el fichero

general en dos subarchivos llamados de entrenamiento *"Train"*, que contiene la información de las variables a imputar para el *set* de entrenamiento y de prueba *"Test"*, que dispone de los mismos datos perdidos pendientes de ser completados para el *set* de prueba del modelo pero con una muestra (por ejemplo de *n=2500*) más pequeña del total de participantes (*N=5178*).

A continuación, para la implementación de la técnica hay que realizar algunos ajustes en los parámetros del modelo y también es necesario hacer la imputación de las observaciones ausentes solo con las variables cuantitativas relacionadas con la variable imputada, ya que la *función nnet ()* asigna los pesos iniciales de los enlaces entre los nodos con valores aleatorios entre el intervalo [-0.5 , 0.5], así que, si se desea tener una respuesta entre 0 y 1, solo se tiene que dividir por 50 ya que por defecto los pesos salen entre -0.5 y 0.5.

Para asegurar que se obtiene los mismos resultados presentados, se añade la función *set.seed ()* que inicializa el generador de números aleatorios en un número de semilla, en este caso por ejemplo se fija en un valor de *500* y del mismo modo, el propio software para facilitar la parte computacional utiliza la *función scale ()* que se encarga de realizar la transformación de los datos en el proceso de la red neuronal. Asimismo, el parámetro *size* permite especificar cuantos nodos tendrá la capa oculta, el *linout=T* indica que la salida será lineal y no binomial, la sentencia *trace=F* evita que el programa presente todas las iteraciones necesarias para llegar a la convergencia del modelo y, si se precisa se puede añadir la opción *decay* (como por ejemplo se puede fijar el *valor de 0.00001* e ir ajustando dicho parámetro) para evitar el problema de los pesos más altos.

Además, en la implementación del método de red neuronal, se pudo observar que los datos de entrenamiento "train" y los de prueba "test" de la mariz de origen fueron muy similares a las predicciones obtenidas en cada uno de los dataset (*train-test*) con tendencia ascendente y lineal (*Figura 1*), por lo que se presupone que estos datos predichos son bastantes acertados para la imputación por tener un *error cuadrático medio* muy bajo (*inferior a 0.0001*) en casi todas las ejecuciones, pero si se observan detenidamente las salidas recogidas de los vectores soportes, la clasificación no es tan idónea como se esperaba tal y como reflejan las salidas de las tablas de clasificación y el valor del grado de acuerdo o de concordancia, que indca que es bastante bajo (débil) entre determinadas variables.

Dicho esto, aunque esta primera técnica se realizo como una de las primeras vías de abordaje sobre el dataset de datos faltantes para suplir uno de los objetivos relevantes de esta exploración analítica, se ha podido comprobar que el problema de la imputación de los valores missings existentes en este caso particular, se debe abordar con otro algoritmo más acertado, ya que los datos completados no son tan buenos si se observa el gráfico representado (*Figura 1*).

Probablemente, el fallo detectado en los datos puede ser de que estén sobreestimados por diferir en comparación con los promedios originales, por lo que se descarta este método por la limitación, el tiempo computacional y la poca flexibilidad que tiene con los datos missings, optando por otro que mejor refleje y se adapte a la información existente en este registro.

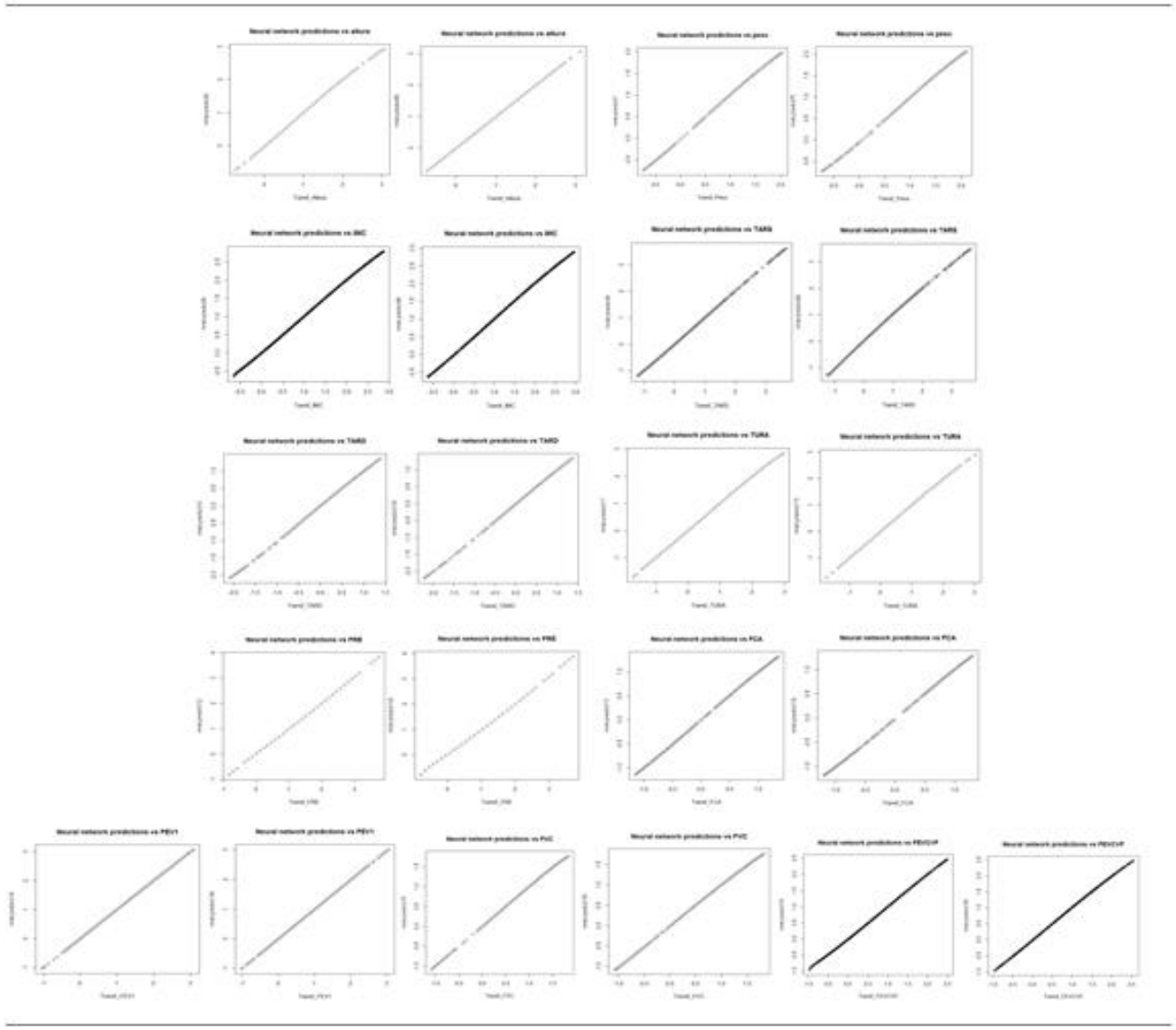


**Figura 1. (a)** NNET. Situación real dataset con > 5-20% de datos missings después de la imputación. Dataset Modificado

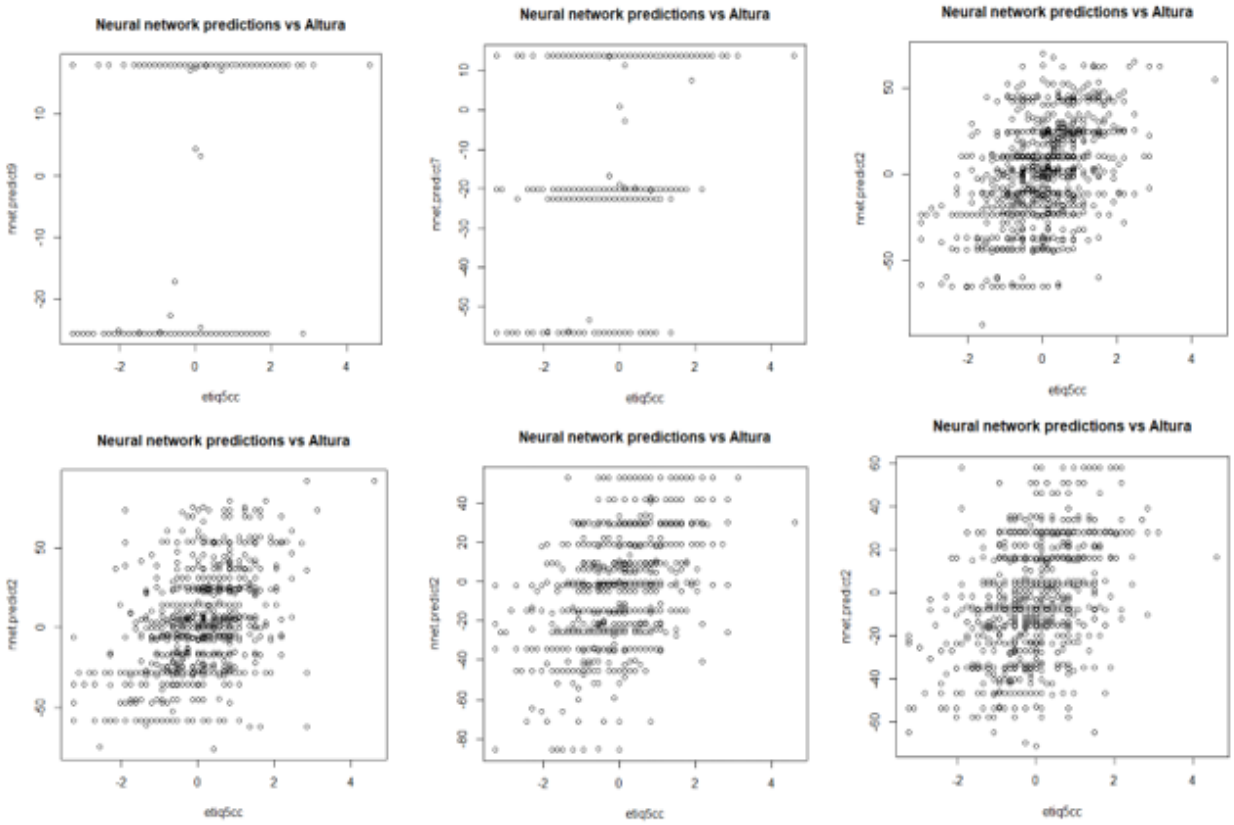



**Figura 1. (b)** NNET. Situación real dataset con > 5-20% de datos missings después de la imputación. Dataset Modificado

Por todo ello, a pesar de las ventajas de los métodos decritos anteriormente en este estudio, se puede pensar que la elección *MICE* (Van Buuren y Groothuis-Oudshoorn, 2011) para la imputación no es la acertada, pero todo lo contrario, es una técnica muy versátil y adaptable para todo tipo de variables clínicas basado en los datos de cada una de ellas, construyendo los modelos apartir de los conjuntos de datos por separado y luego, combinando sus resultados, lo que ayuda a reducir la incertidumbre en los valores perdidos y aumentar la eficiencia, en comparación con una imputación única que solo se ocupa de la incertidumbre en los datos faltantes.

El procedimiento MICE denominado como *Imputación múltiple con ecuaciones encadenadas (en inglés "Multiple Imputation with Chained Equations"*) consiste básicamente en actualizar cada una de las variables con datos faltantes mediante series completas de distribuciones condicionadas, a través de tres pasos principales de imputación múltiple, que se pueden resumir en los siguientes puntos de forma esquemática: (i) imputación, (ii) análisis y (iii) agrupación.

- ***Estado inicial:*** es donde esta el primer dataset con las variables numéricas, que se corresponde con el conjunto de datos original, sin la aplicación del algoritmo.

- ***Fase primera o de imputación:*** donde el dataset esta formado por los conjuntos de datos que se han imputado. Este paso principal se suele llamar fase *mice*(), donde en esa fase se le indica alguno de los método de imputación considerados por el algoritmo, mediante el argumento "method" que en este caso se aplico a través de dos vías seleccionadas para contrastar resultados.

    - La primera con el ***método simple*** (Van Buuren y Groothuis-Oudshoorn, 2011), donde este mecanismo consiste en que para cada una de las variables con datos faltantes, se reemplazan sus valores perdidos por muestras aleatorias de los datos observados de su propia variable. Por eso, este procedimiento de imputación es una de las técnicas más habituales y utilizadas en computación, ya que se puede aplicar para cualquier tipo de escala (numérica o categórica).

    - Y la segunda opción es la del ***método por coincidencias de medias predictivas*** (PMM) (Van Buuren y Groothuis-Oudshoorn, 2011) o como se denomina en inglés "predictive mean matching", que sólo se aplica para variables numéricas. En este procedimiento los valores imputados coinciden con alguno de los valores observados en la misma variable, manteniendo la estrucura relacional del conjunto original al aplicar dentro del proceso distirbuciones condicionadas.

      Más concretamente, la técnica PMM consiste en calcular para cada variable valores predichos mediante un modelo de regresión para las observaciones faltantes y las observadas (no faltantes o completas), con el fin de completar los datos faltantes seleccionando aleatoriamente una de las observaciones completas cuyos valores predichos son los más cercanos al valor predicho de la observación faltante. En definitiva, este método tiene como cometido construir la métrica que haga coincidir que las observaciones con datos faltantes sean similares a las observaciones con valores completos u observados para usarlos dentro del proceso de imputación.

- ***Fase segunda o de análisis:*** donde el dataset presenta las estimaciones de las incógnitas de interés. A este paso intermedio se le suele denominar también fase *with*().

- ***Fase última, tercera o de agrupación:*** se concluye con el dataset que ya contiene la agrupación final de los resultados de las estimaciones obtenidas. Y finalmente, esta fase final con las imputaciones realizadas se le denomina fase *pool*().

Por otro lado, manejar valores perdidos es un paso crucial en el preprocesamiento de datos con *Machine Learning*, donde la mayoría de los algoritmos disponibles para analizar conjuntos de datos en el proceso de selección de características y en el proceso de clasificación o estimación, analizan conjuntos de datos completos. En consecuencia, en muchos casos, la estrategia para lidiar con los valores perdidos, (Mera-Gaona et al., 2021) es usar solo instancias con datos completos o reemplazar los valores perdidos con una media, moda, mediana o un valor constante. Y por lo general, descartar las muestras faltantes o reemplazar los valores faltantes mediante técnicas fundamentales provoca sesgos en los análisis posteriores de los conjuntos de datos. Por todo esto, en este trabajo se resolverá el problema aplicando la imputación multivariante mediante *MICE*.

En este sentido, el planteamiento se desarrolla en varios pasos de ejecución, primero se intenta suplir la falta de valores perdidos en el *capítulo I* mediante el método *MICE.* Después en el *capítulo II*, se continua con la aplicación de las diferentes técnicas de análisis para reducción dimensional del conjunto de datos-variables, a través de distintos mecanismos, y por último, se finaliza en el *capítulo III* describiendo la clasificación de perfiles clínicos originados, que mejor definen a cada patrón clinico generado según sus características internas en el grupo formado. Y como extensión a este trabajo, en el *capítulo IV*, se podrá ver una pincelada del análisis con vectores soportes (SVM) y métodos Kernel en un experimento con casos simulados y otro con caso real con los datos de este dataset.

Para la realización de esta tesis doctoral, la fuente de información de análisis procede de la base de datos original del proyecto AUDIPOC España (Pozo-Rodríguez et al., 2010). Los resultados que se desprenden del análisis descriptivos-exploratorios (Table 1) realizado a los 5.178 pacientes ingresados por *exacerbación de la Enfermedad Pulmonar Obstructiva Crónica* (eEPOC), donde un 87% son hombres y un 13% mujeres con una edad media de 73 años, el 83% son fumadores y solo el 68% tienen espirometria realizada al ingreso o al alta. Todos ellos presentan diversas características y patologías clínicas que pueden ser bastantes dispares dependiendo de la gravedad o del avance de la enfermedad primordial, donde el 35% ingresa con una estancia media de 10 días con necesidades de soporte ventilatorio en un 11%, y el 28% vuelve a reingresar por exacerbación y casi todos (27%) lo hacen a los 90 días, con el 5% de exitus positivo.

## I. 3. Resultados

Para realizar el análisis se ha utilizado el *software estadístico R* (R Core Team, 2021) (versión 4.1.0), siendo la solución factible para suplir algunos aspectos de imputación de *missings values* (Faquih et al., 2020) en determinadas variables del dataset, mediante diversos métodos presentados por el paquete MICE (Ferguson et al., 2018 y Slade y Naylor, 2020) (*Sample y Predictive Mean Matching*) (Van Buuren y Groothuis-Oudshoorn, 2011) con el objetivo de completar y mejorar los resultados finales.

Con este planteamiento, en la *Figura 2* se representa el estado de valores perdidos en la base de datos donde se ha realizado un proceso de imputación en determinadas variables de este dataset, mediante diversos métodos presentados por el paquete MICE (Luo et al., 2018), con el fin de completar la información de este estudio y mejorar los resultados finales.

```
Variables sorted by number of missings:
 Variable      Count
      ALP 0.6237930
     PPKG 0.6210892
      FRE 0.5264581
     FVCP 0.4103901
    FEV1P 0.3810351
      FCA 0.1568173
     TURA 0.1465817
     TARS 0.1145230
     TARD 0.1145230
```

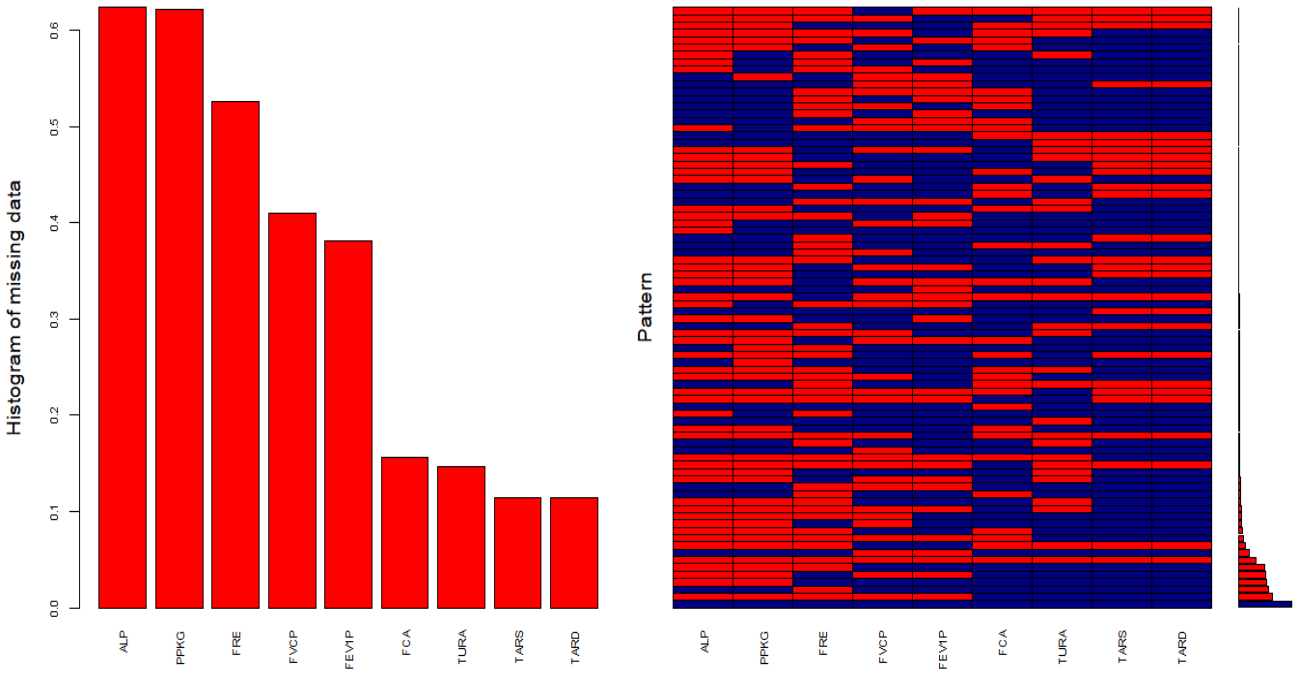



**Figura 2.** MICE. Situación real dataset con > 5-20% de datos missings antes de la imputación. Dataset Original

A la vista de las salidas obtenidas con las técnicas de Sample y Predictive Mean Matching, se ha confirmado que se han conseguido los mismos resultados originales con ambos procedimientos sin afectar a la información existente, quedando resuelto este dilema satisfactoriamente, y pudiendo aplicar nuevos análisis exploratorios. Es decir, ambos procedimientos de imputación sobre el dataset final completo muestra los mismos resultados exploratorios, por lo que se supone que tienen el mismo impacto en los resultados finales, ya que no hay mucha variación o diferencia entre ambos métodos de imputación para este caso particular.

Finalmente, la imputación obtuvo un conjunto de datos completo y algo más preciso para aplicar técnicas multivariantes sobre estos datos reales, siendo prueba de ello, la visualización gráfica mostrada (*Figura 3*) donde en cada una de las variables el porcentaje es cero en el *pattern of missing data*, y por consiguiente la coincidencia en el diagrama de dispersión, se aprecia una información libre de valores perdidos e idéntica al original, pudiéndose extrapolar los resultados obtenidos a la población general.

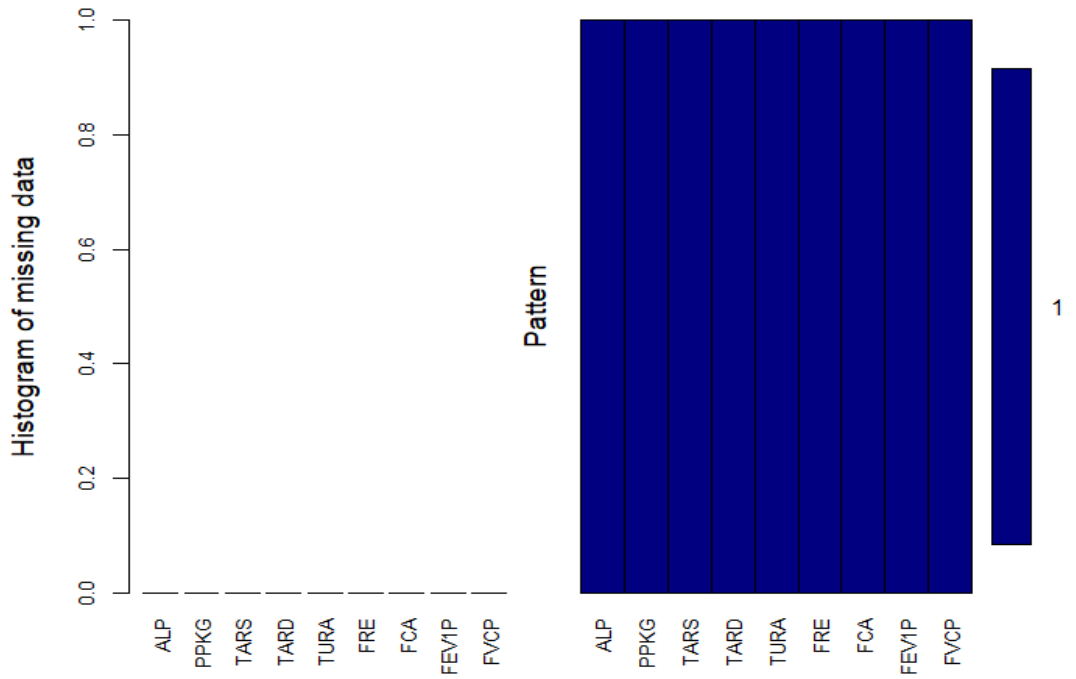



**Figura 3. (a)** MICE. Situación real dataset con > 5-20% de datos missings después de la imputación. Dataset Modificado

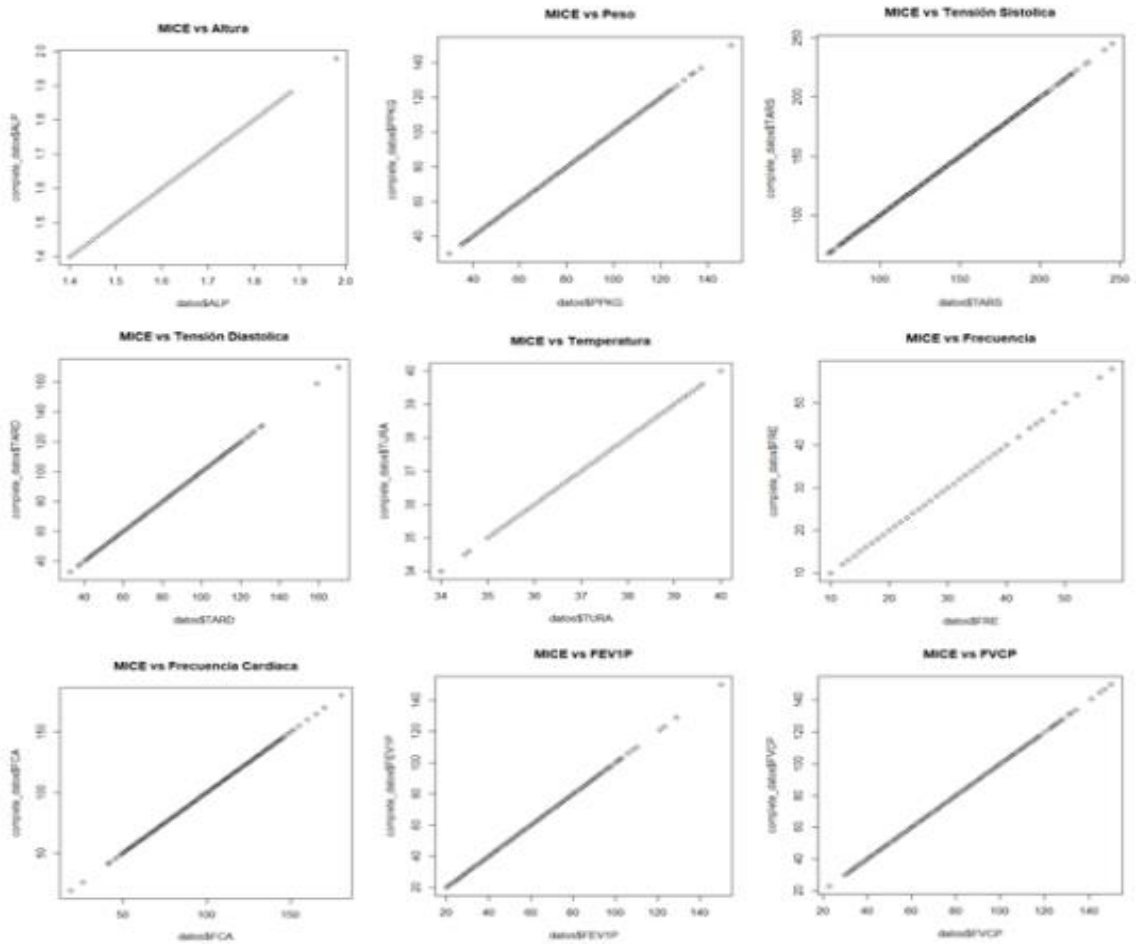



**Figura 3. (b)** MICE. Situación real dataset con > 5-20% de datos missings después de la imputación. Dataset Modificado

En base a los resultados obtenidos por los diferentes procedimientos de imputación (Figura 2), se aseguran los resultados asignados pre-post aplicación con éxito del 100% de los datos completados (Figura 3) en las diferentes variables epidemiológicas-clínicas (Tabla 1), con resultados promedios ligeramente alterados como la Altura (ALP) 1.64 metros; Peso (PPKG) 74.85 kg; Índice de Masa Corporal (IMC) 27.88 kg/m$^2$;

Tensión Arterial Sistólica (TARS) 136.4 mmHg, superior al rango de normalidad 90-120 mmHg indicando posible riesgo de padecer un ECV (Evento Celebro Vascular); Tensión Arterial Diastólica (TARD) 75.04 mmHg, dentro del parámetro de normalidad 60-80 mmHg; Temperatura (TURA) 36.78 $^0$C; Frecuencia Respiratoria (FRE) 24.26 resp./min, alterada por estar fuera del rango de normalidad 12-18 respiraciones por minuto; Frecuencia Cardíaca (FCA) 94.44 lat/min, dentro del rango normal 60-100 latidos por minuto;

En cuanto a la FEV1 espirometría en % del teórico (FEV1P) 45.02%, muestra alteración severa, puesto que el Volumen Espiratorio Forzado en el primer segundo % en condiciones normales esta alrededor del 80%; FVC espirometría en % del teórico (FVCP) 64.88%, refleja sintomatología grave, ya que la Capacidad Vital Forzada % se considera normal cuando es mayor del 70% y varía con la edad, de aquí su gravedad; Cociente relacional FEV1/FVC espirometría previa o alta (FEVCVF) 72.67%, indica presencia de alteración indefinida en cuanto obstrucción cuando el rango de referencia se encuentra entre el 70-85%.

**Tabla 1.** DESCRIPTIVO. Resumen de resultados epidemiológicos-clínicos

| Variables | Mean | SD | Variables | n (%) – (No/Yes) | |
|---|---|---|---|---|---|
| EDAD | 73.39 | 10.08 | SEXO (Male/Female) | 4526 (87.4) | 652 (12.6) |
| DURING | 9.96 | 7.82 | HT (Smoking) | 906 (17.5) | 4272 (82.5) |
| ALP | 1.64 | 0.08 | ESPIROMETRIA_PA | 1644 (31.8) | 3534 (68.2) |
| PPKG | 74.85 | 15.55 | INGRESOS | 3343 (64.6) | 1835 (35.4) |
| IMC | 27.88 | 6.03 | SV | 4596 (88.8) | 582 (11.2) |
| TARS | 136.4 | 23.85 | EXACER_90DIAS | 3793 (73.2) | 1385 (26.8) |
| TARD | 75.04 | 13.84 | REING_EXAC | 3751 (72.4) | 1427 (27.6) |
| TURA | 36.78 | 0.82 | MUERTOS_90DIAS | 4873 (94.1) | 305 (5.90) |
| FRE | 24.26 | 6.62 | EXITUS | 4919 (95.0) | 259 (5.00) |
| FCA | 94.44 | 18.50 | ICHARICC | 4058 (78.4) | 1120 (21.6) |
| FEV1P | 45.02 | 16.82 | CCVSDM | 2961 (57.2) | 2217 (42.8) |
| FVCP | 64.88 | 19.21 | ICHAR_DM | 3844 (74.2) | 1334 (25.8) |
| FEVCVF | 72.67 | 28.48 | EV | 3588 (69.3) | 1590 (30.7) |
| | | | ICHARECV | 4620 (89.2) | 558 (10.8) |
| | | | ICHAREVP | 4422 (85.4) | 756 (14.6) |
| | | | ICHARIM | 4505 (87.0) | 673 (13.0) |
| | | | ICHARNEF | 4691 (90.6) | 487 (9.40) |
| | | | ICHAR_TS | 4506 (87.0) | 672 (13.0) |
| | | | EP | 3828 (73.9) | 1350 (26.1) |

Asimismo, el porcentaje detectado en cada una de las patologías asociadas al desarrollo de la enfermedad principal son las siguientes: *Insuficiencia cardíaca congestiva* (ICHARICC) 22%; *Comorbilidad Cardiovascular* (CCVSDM) 43%; *Diabetes Mellitus* (ICHAR_DM) 26%; *Enfermedad Vascular*

(EV) 31% con *cerebro vascular* (ICHARECV) 11% + *vascular periférica* (ICHAREVP) 15%; *Infarto de miocardio* (ICHARIM) 13%; *Nefropatía* (ICHARNEF) 9%; *Tumor sólido* (ICHAR_TS) 13% y *Edemas maleolares* (EP) 26% y además, la presencia única de *solo EPOC* fue del 32%, ya que la gran mayoría de los pacientes (68%) presentan *otras patologías adheridas*, que dependiendo de su avance y estado de gravedad de la propia enfermedad principal, pueden coexistir una o varias de ellas, además de la predominante.

## I. 4. Conclusiones

A la vista de los resultados descriptivos, se ha podido completar la fuente de datos satisfactoriamente con la aplicación *MICE*, originando datos más exactos y limpios para continuar con los siguientes análisis multivariantes en los diferentes capitulos indicados en este trabajo, con el fin de llevar acabo el objetivo planteado sobre la búsqueda de perfiles clíninos según características o patologías de agrupación.

En este sentido, destacar que dada que la estructura de correlación puede ser bastante sensible a las distintas técnicas de imputación, se debe estudiar este aspecto antes de proceder a la imputación, según el tipo de variables del conjunto de datos para no perder la calidad relacional del original, por lo que es que es necesario para aplicar la técnica del PCA, que se describe en el capítulo posterior.

Para este caso, se sabe que el método MICE utiliza las ecuaciones encadenadas en este proceso de imputación aleatoria de cada variable, y estas están condicionadas a las variables imputadas, aplicando un mecanismo de cadenas dependientes en la distribución de probabilidad. Por lo tanto, se asume que esta dependencia se conserva en la estructura de correlación del algoritmo de imputación utilizado, cuando se modifican los valores missings, manteniendo la calidad relacional del dataset original.

Asimismo, este avance rápido requiere aprender de técnicas sofisticadas para que la complejidad no sea tan elevada en los estudios multicéntricos, puesto que se disponen de multitud de parámetros-variables de análisis que hacen que este proceso se complique exageramente, incluyendo la problemática de los missings, que en muchas ocasiones son un porcentaje bastante alto, que requieren de una atención analítica-exploratoria y de técnicas especificas para mejorar la información almacenada en cada uno de los registros institucionales y sanitarios.

También, se recuerda de la existencia de diversos mecanismos para apaliar la mala calidad de los datos que está limitando el uso de los mismos provenientes de los sistemas de información de salud de

rutina, siendo los valores faltantes (Chia et al., 2020) un componente importante de este problema y donde los organismos de salud, por diversas razones, no informan al sistema central de esta problemática existente para tomar medidas preventivas al respecto con el fin de mejorar el proceso de recogida diaria de la información (Feng et al., 2021).

En este sentido, con el desarrollo de este estudio se quiere recordar de la existencia de diferentes vías para subsanar el problema de la falta de datos en los diferentes registros clínicos con el fin de obtener una mayor calidad y mejor información en las distintas fuentes de datos sanitarias.

En definitiva, la calidad de los datos y los diversos métodos de implementación para la mejora de la informacion recogida, son requisitos fundamentales para poder obtener un registro clinico de alta calidad y por consiguiente, obtener unos buenos resultados con diferentes salidas que puedan ser extrapolables a la población general con la máxima garantía del análisis realizado en cualquier tipo de fuente de datos clínicos.

# CAPÍTULO II

# Reducción de la dimensionalidad mediante diferentes métodos

## II. 1. Introducción

Durante los últimos años el avance tecnológico, la transformación digital originada con el *Big Data y la* Inteligencia Artificial (IA) y por otro lado, la pandemia generada con el COVID19 están afectando de forma atroz a todo el entorno económico-social, y en particular, al ámbito sanitario-investigación, llevando a las instituciones sanitarias-científicas a plantearse diversas vías y estrategias para establecer sinergias entre ambas, con el fin de poder analizar la gran cantidad de información existente entre los distintos registros clínicos, explotando los infinitos repositorios originados con el fin de dar soporte y apoyar en la toma de decisiones para la mejora y calidad de los pacientes atendidos (López-Campos et al., 2021).

En este sentido, el campo de la *Estadística Computacional* se ha adaptado rápidamente a las necesidades requeridas y ha experimentado un crecimiento exponencial, aportando un gran avance con la aparición de nuevas herramientas de análisis estadístico, algoritmos específicos y adaptados a las nuevas necesidades que se están generando en todas las áreas de conocimiento y, en especial, en el área de biomedicina (Dollfus y Petit, 1995 y Blázquez-Sánchez et al., 2020).

Este avance tecnológico hace aflorar nuevas problemáticas que con los inmensos repositorios de datos deben ser mejorados, como es el caso de la dimensionalidad o el hándicap de los datos missings tratados en el capítulo I, con la finalidad de poder mejorar los nuevos análisis ajustándose a la línea principal de investigación.

En este sentido, toma especial importancia las aplicaciones de datos en el contexto de *Functional Data Analysis* (FDA), permitiendo dar una nueva vía de acceso para situaciones similares en estudios multicéntricos (Cheney, 2001 y Ferraty y Vieu, 2006). Las técnicas FDA son las más destacadas por su gran utilización y uso generalizado, como son el *Functional Principal Components Analysis* (FPCA), (Choubey et al., 2020), *Time Series* and *Functional Linear Regression Models*.

En este capítulo se aborda la problemática de la dimensionalidad, utilizando varias técnicas multivariantes de reducción dimensional con el fin de contrastar resultados, como *Principal Components Analysis* (PCA) (Gil, 2018) y otros métodos como el *Random Forest* (Deng y Wang, 2021 y Wang et al., 2021b) *por Gini Index & Information Value por aplicación WOE* (RF&IV) (Bhalla, 2015 y Larsen, 2015), y también, el *parallel analysis con datos simulados y remuestreo* (APS-REM) que pretenden buscar la mejor selección de importancia de factores-variables (óptima) para simplificar el espacio de análisis con el fin de poder generar nuevos grupos-perfiles de clasificación con características similares (Boukichou-Abdelkader et al., 2022).

Por tanto, con las técnicas actuales de análisis de datos, toda base de datos clínica con una gran dimensión de información multicéntrica o cualquier repositorio en *Cloud*, requiere de métodos y técnicas estandarizadas con perfiles clínicos cualificados para una adecuada *gobernanza del dato*, que asegure un buen tratamiento y procesamiento de la información, evitando incoherencias y datos faltantes no disponibles, mediante la implementación de *técnicas de calidad* e *inferencia estadística* que resalten el gran valor de la información, mejorando los resultados finales y dando conclusiones reales a la investigación en estudio para los casos de gran interés poblacional.

## II. 2. Métodos

Siguiendo el hilo de esta tesis doctoral, para tratar el tema de la dimensionalidad se aplican distintas técnicas multivariantes: (i) el análisis de componentes principales (PCA) para la reducción de la dimensión presentada en este conjuto de datos (Sánchez, 2019); (ii) la técnica Random Forest e Information Value (RF&IV) para definir la selección de importancia de variables mediante los métodos Random Forest por Gini Index (Hanko et al., 2021 y Yang et al., 2020) & Information Value por aplicación weight-of-evidence (WOE) con el fin de contrastar ambos y si es posible disminuir eficientemente algo más la dimensión de los datos-variables explorados (Prabhakaran, 2016); y (iii) el análisis paralelo de simulación y remuestreo (APS-REM) para realizar un análisis de comparación con datos simulados y de remuestreo basado en la matriz de correlaciones aleatoria. Estas técnicas solo pretenden dar soporte a la búsqueda de reducción de la selección de importancia de los factores-variables, simplificando el espacio de análisis y quedándonos solo con la información relevante y precisa, con la finalidad de dar una mejor solución clínica que pueda ser un reflejo real de la población actual, y en paralelo, generar la clasificación de grupos con afinidades idénticas y proponer esta vía de análisis para la problemática de la reducción de la dimensionalidad.

El objetivo, tras aplicar estas fases técnicas, se espera mejorar la capacidad de rendimiento de la base de datos clínica con alta dimensión, extrayendo de la mejor manera posible los resultados esperados y enfocando de forma más específica, el problema de la reducción dimensional existente y suplir la problemática de los *missing values*, con la finalidad de intentar clasificar adecuadamente a los pacientes por sus características similares.

El planteamiento se desarrolla en varios pasos de ejecución, primero que era de suplir la falta de valores perdidos se ha descrito en el capítulo anterior. Ahora en el *capítulo II*, se continua con la aplicación de las diferentes técnicas de análisis para reducción dimensional del conjunto de datos-variables, a través de los distintos mecanismos de comprobación (*PCA, RF&IV, APS-REM*), y una vez seleccionado el mejor método de reducción que viene dado por PCA, se finaliza el análisis describiendo la clasificación preliminar de los grupos-perfiles originados, destacando características principales con el fin de resaltar el resultado clínico obtenido mediante la mejor reducción dimensional.

## II. 3. Resultados

En la misma línea del capítulo anterior, también para realizar el análisis se ha utilizado el *software estadístico R* (R Core Team, 2021) (versión 4.1.0), siendo la base principal para aplicar las distintas técnicas de *análisis multivariante* mediante PCA (Siuly y Li, 2015), RF&IV y APS-REM con el fin de reducir y optimizar la dimensión de los datos estudiados.

Con este planteamiento de implementación de distintas técnicas multivariantes (El Boujnouni et al., 2021) y del análisis de datos funcional, se llegó a la finalidad perseguida, buscar la mejor reducción posible dimensionalmente sin perder información relevante con el fin paralelo, de una correcta detección y visualización de la generación de diferentes grupos con características similares.

A continuación, se detallan los tres procedimientos implementados y los resultados alcanzados en cada uno de ellos para valorar la mejor técnica aplicada y las posibilidades que aportan a este análisis.

### *II. 3. 1. Análisis de Componentes Principales (Principal Components Analysis)*

La elección de aplicar el *Análisis de Componentes Principales (ACP - PCA)* (Fernández-Crehuet et al., 2019 y Pinheiro et al., 2021), es debido a que la inercia de las primeras dimensiones muestra si existen

fuertes relaciones entre las variables y sugiere el número de dimensiones que deben estudiarse, por lo que se acerca rápidamente al objetivo perseguido.

No obstante, el método PCA está diseñado para trabajar con variables numéricas, por lo que se debe realizar una conversión de las variables categóricas binarias o dicotómicas a numéricas. Esta transformación de variables categóricas nominales u ordinales al convertirlas en variables numéricas, se denomina Análisis Multivariante con Escalamiento Óptimo (MVAOS) (Rossiter, 2021), técnica que cuantifica factores, reemplazando valores categóricos por números reales, de modo que PCA se puede aplicar (Meulman et al., 2004; Linting, 2007; Jolliffe y Cadima, 2016 y Saucedo, 2019), transformaciones que se calculan junto con el PCA para maximizar la varianza explicada por cada componente, (Meulman et al., 2002; Meulman et al., 2004; Linting, 2007; Manisera et al., 2010; Molina y Espinosa, 2010; Kuroda et al., 2013; Jolliffe y Cadima, 2016; De Leeuw et al., 2017; Saucedo, 2019 y Rossiter, 2021).

Esta técnica se puede aplicar con el paquete *Gifi* de R (Mair y De Leeuw, 2019) desde la función *"princals"*, o bien con el software SPSS (SPSS Inc., 2008) usando el método CATPCA, que deriva pesos de los datos de entrada produciendo relaciones lineales óptimas en los datos de salida con un escalado óptimo (Meulman et al., 2002; Molina y Espinosa, 2010 y Manisera et al., 2010). Y del mismo sentido, el software SAS (SAS Institute Inc., 2018) proporciona el procedimiento PRINQUAL basado en PRINCIPALS (Kuroda et al., 2013). Por lo tanto, el Análisis de Componentes Principales (PCA) se puede aplicar en variables categóricas con diferentes métodos y el resultado de la reducción dimensional es significativo.

No obstante, existe el procedimiento análogo para variables categóricas, que es el Análisis de Correspondencias Múltiples (MCA) que se utiliza en la extensión de estos análisis para obtener mejores resultados, y se encuentra implementado en el paquete *FactoMineR* (Lê et al., 2008) mediante diferentes funciones de aplicación como pueden ser *CA()* o *MCA()*, donde esta última proporciona valores propios mucho más pequeños, en comparación con los eigenvalues obtenidos de un PCA o CA, que son aplicados en este caso.

A la vista de los resultados aplicados finalmente con la función *PCA()* del paquete Factoshiny (Vaissie et al., 2021), aunque también se utilizaron previamente las funciones *dudi.pca()* del paquete *ade4* (Dray et al., 2007; Bougeard y Dray, 2018 y Thioulouse et al., 2018) y la función *prcomp()* directa de R (R Core Team, 2021), se ha obtenido que las dos primeras dimensiones expresan el 19,24% (Tabla 2) de la inercia total del conjunto de datos; eso significa que la variabilidad total de la nube de los individuos (o variables) se explica por el plano 1:2. Obviamente, este es un porcentaje muy pequeño, pero significativo para este dataset, y el primer plano representa una pequeña parte de la variabilidad de los

datos, aunque este valor es mayor que el valor de referencia (7,19% que equivale al percentil 95 de la distribución de porcentajes de inercia obtenida simulando 501 tablas de datos de tamaño equivalente sobre la base de una distribución normal), por lo que la variabilidad explicada por este plano es significativa. Si es preciso para obtener mejores resultados a partir de estas observaciones en algunas variables, se recomienda interpretar también las dimensiones mayores o iguales a la tercera para completar la información.

**Tabla 2.** Reducción Dimensional con PCA – Eigenvalues

| Eigenvalues | Dim.1 | … | … | … | … | … | … | … | Dim.32 |
|---|---|---|---|---|---|---|---|---|---|
| | **Dim.1** | **Dim.2** | **Dim.3** | **Dim.4** | **Dim.5** | **Dim.6** | **Dim.7** | **Dim.8** | **Dim.9** |
| Variance | 3.246 | 2.911 | 1.981 | 1.744 | 1.601 | 1.472 | 1.387 | 1.281 | 1.235 |
| % of var. | 10.143 | 9.098 | 6.191 | 5.451 | 5.004 | 4.600 | 4.336 | 4.002 | 3.860 |
| Cumulative % of var. | 10.143 | **19.241** | 25.432 | 30.883 | 35.887 | 40.487 | 44.823 | 48.825 | 52.685 |
| | **Dim.10** | **Dim.11** | **Dim.12** | **Dim.13** | **Dim.14** | **Dim.15** | **Dim.16** | **Dim.17** | **Dim.18** |
| Variance | 1.144 | 1.097 | 1.041 | 0.982 | 0.964 | 0.950 | 0.920 | 0.910 | 0.874 |
| % of var. | 3.574 | 3.427 | 3.254 | 3.069 | 3.012 | 2.970 | 2.876 | 2.843 | 2.731 |
| Cumulative % of var. | 56.259 | 59.687 | **62.941** | 66.010 | 69.022 | 71.992 | 74.868 | 77.711 | 80.442 |
| | **Dim.19** | **Dim.20** | **Dim.21** | **Dim.22** | **Dim.23** | **Dim.24** | **Dim.25** | **Dim.26** | **Dim.27** |
| Variance | 0.846 | 0.830 | 0.806 | 0.793 | 0.734 | 0.720 | 0.597 | 0.407 | 0.218 |
| % of var. | 2.645 | 2.593 | 2.519 | 2.478 | 2.294 | 2.250 | 1.864 | 1.272 | 0.682 |
| Cumulative % of var. | 83.087 | 85.680 | 88.199 | 90.677 | 92.971 | 95.221 | 97.085 | 98.357 | 99.039 |
| | **Dim.28** | **Dim.29** | **Dim.30** | **Dim.31** | **Dim.32** | | | | |
| Variance | 0.183 | 0.058 | 0.043 | 0.020 | 0.004 | | | | |
| % of var. | 0.571 | 0.183 | 0.133 | 0.061 | 0.013 | | | | |
| Cumulative % of var. | 99.610 | 99.792 | 99.925 | 99.987 | 100.000 | | | | |

Por tanto, el análisis indica que una estimación del número correcto de ejes a interpretar sugiere restringir el análisis a la descripción de los primeros 12 ejes, puesto que son los que llevan la gran mayoria de la información real y estos ejes presentan una cantidad de inercia del 62,94% mayor que las obtenidas por el percentil 95 de distribuciones aleatorias (40,78%), por lo que la descripción relevante se situará en estos ejes, siendo los dos primeros los que más contribuyen a la inercia total (Figura 4).

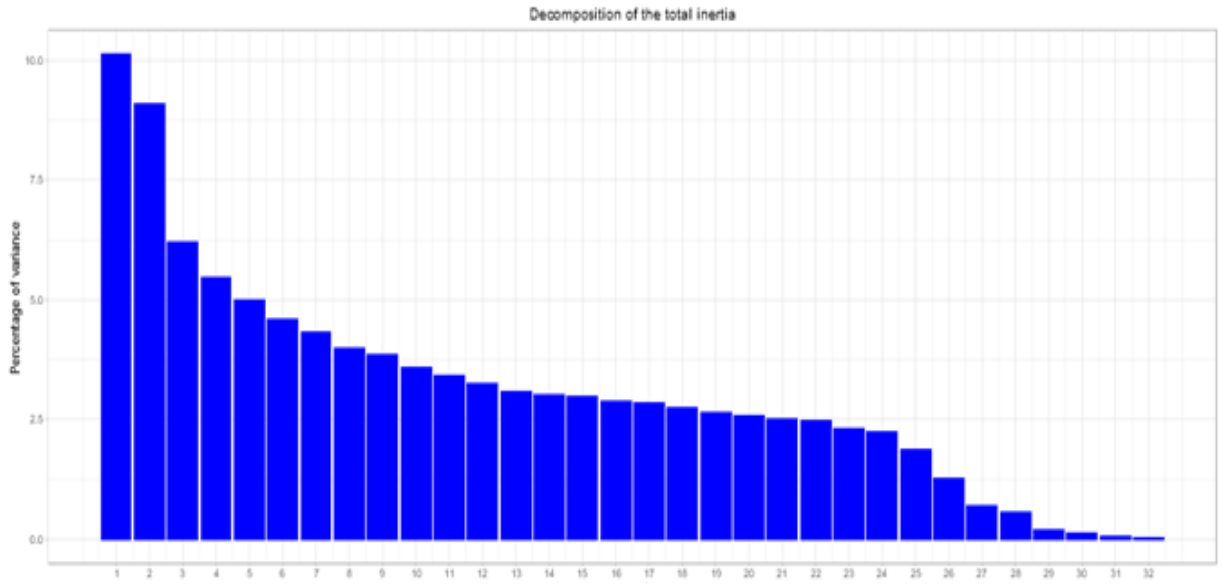



**Figura 4.** Descomposición de la inercia total en porcentaje de la varianza explicada

Por otro lado, también para confirmar la salida anterior se puede visualizar este corte de ejes, mediante el gráfico de sedimentación, como otra ayuda rápida sobre las componentes finales correctas, donde se muestra la cantidad óptima de componentes a tomar en los datos, siendo los valores por encima de la línea del punto 1.0 los más aceptables (Figura 5), por lo que se verifica que siguen siendo las 12 componentes anteriores (autovalores mayores que uno) los que disponen de la gran mayoria de la información válida. En paralelo, se puede observar que el número de factores representativos para estos datos crudos pueden estar alrededor de 3 factores, proporcionando una idea de la información suficiente para explicar la agrupación de estos datos.

Asimismo, también se muestra una de las hipótesis fundamentales que requiere el analisis PCA, tras el proceso de estandarización de los datos originales (crudos) por la mediciones de diferentes escalas, como es la matriz de correlaciones, donde estas deben ser altas para aseguridad su aplicabilidad. En el siguiente plot (Figura 6) se muestra el color correlación que existe entre ellas, donde el color *azul*, indice una correlación positiva y el *rojo* una correlación negativa.

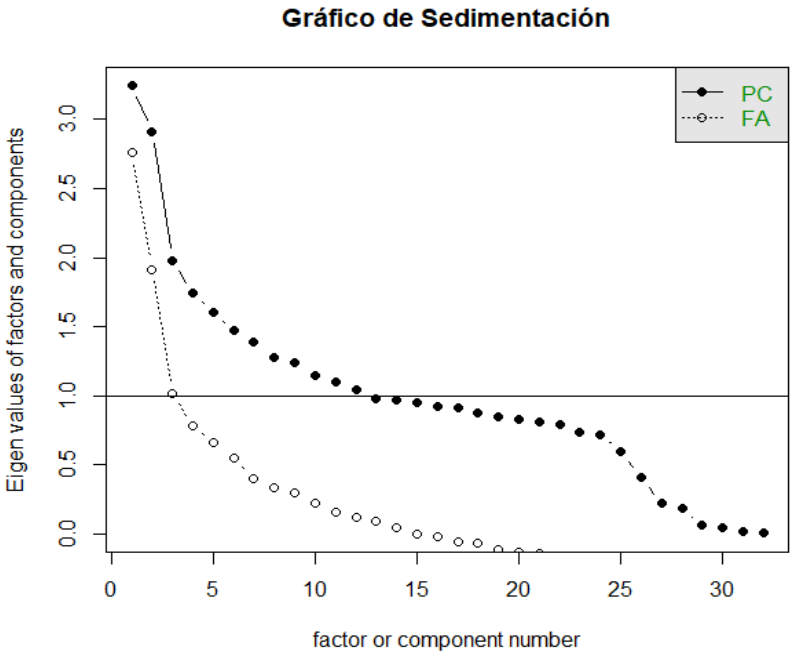



**Figura 5.** Gráfico de Sedimentación mediante PCA

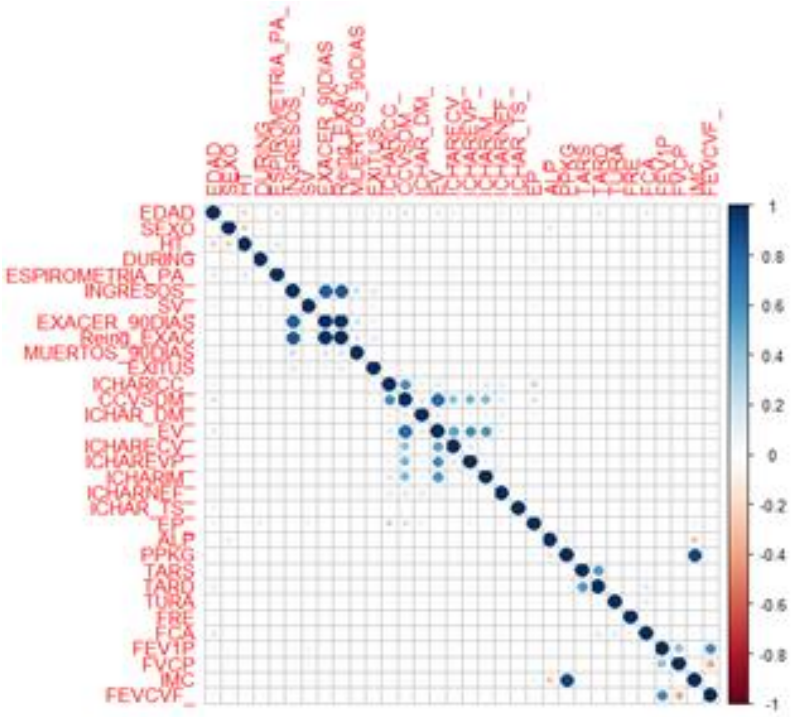



**Figura 6.** Matriz de Correlaciones mediante PCA

### *II. 3. 2. Análisis Paralelo (Parallel Analysis)*

Análogamente, se realizó una segunda comprobación con el Parallel Analysis (Figura 7) de datos simulados y remuestreo (Ayers et al., 2021) (APS-REM) para la información existente mediante la función *fa.parallel()* del paquete "*psych*" (Revelle, 2021), que presenta diferentes opciones para visualización gráfica de componentes principales através de la opción "pc" o la de análisis factorial del eje principal ("fa"), con la finalidad de confirmar el número de componentes finales de forma óptima.

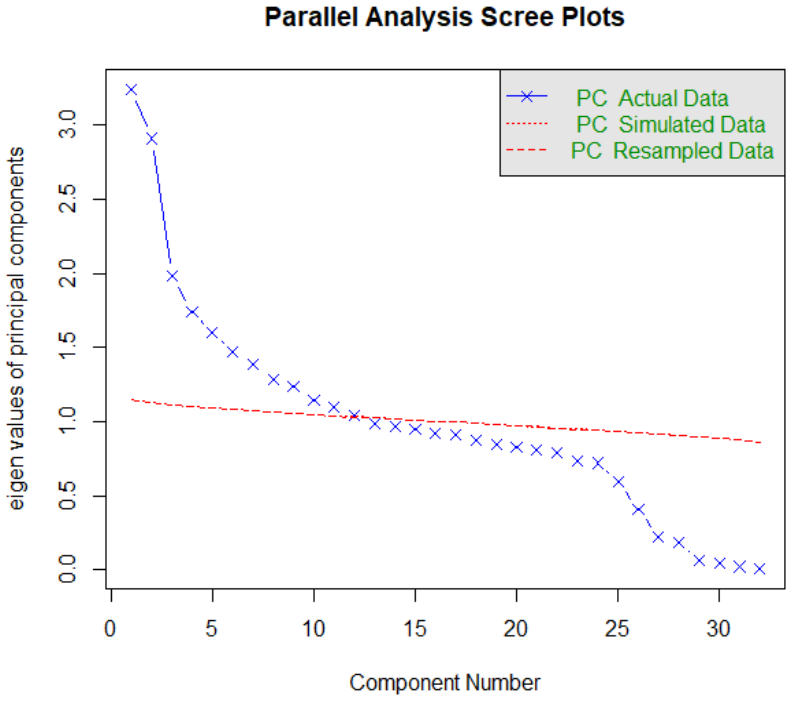



**Figura 7.** Análisis Paralelo con datos simulados y remuestreo

Este análisis es una técnica alternativa que compara la pantalla de los datos observados con la de una matriz de datos (o una matriz de correlación) aleatorios del mismo tamaño que la original. La función traza los valores propios para una solución de componentes principales y factor principal, y hace lo mismo para matrices aleatorias del mismo tamaño que la matriz de datos original. Asimismo, si se tienen datos brutos, las matrices aleatorias tienen dos funcionalidades: (i) primero son una matriz de datos normales univariados y (ii) segundo, son muestras aleatorias (aleatorizadas en filas) de los datos originales, (Revelle, 2008).

En definitiva, a la vista de estos resultados obtenidos, este análisis también sugiere que son necesarios 12 componentes a estudiar, por lo que viene mostrando y confirmando el mismo número de componentes que el análisis PCA.

### *II. 3. 3. Análisis con Random Forest & Information Value*

Siguiendo la línea anterior de optimizar el número de componentes cuando la lista de variables es muy grande, se realizó una tercera comprobación (*Tabla 3*) con los métodos *Random Forest* (Bouzid et al., 2021 y Li et al., 2021) *(GINI Index*) y el de *Information Value* (WOE, *Weight Of Evidence*) (Parsai y Kumar, 2021 y Wurst et al., 2020) desde los paquetes "*randomForest* y *smbinning*" (Liaw y Wiener, 2002 y Jopia, 2019), con la finalidad de contrastar si era posible disminuir eficientemente el espacio dimensional por la importancia de variables sobre los datos-variables explorados.

Para este caso, la técnica *Random Forest* (Bouzid et al., 2021; Deng y Wang, 2021; Hanko et al., 2021; Li et al., 2021; Prabhakaran, 2016; Wang et al., 2021b y Yang et al., 2020), también conocida como *Decision Tree Ensembles*, son útiles para la selección de características además de ser clasificadores efectivos. Este algoritmo ayuda a reducir la selección de variables de importancia simplificando el espacio de análisis, y solo guarda y considera la información relevante y precisa, optimizando el número de componentes cuando la lista de variables es muy grande. Este proceso se realiza con el índice de Gini (*GINI Index)*, ó también con la técnica *WOE* (Weight Of Evidence), calculando la importancia de las variables, a partir de los datos-variables explorados, para reducir el espacio dimensional.

El método utiliza una selección de todas las características de las variables en el modelo de predicción, dando un valor de importancia (score de peso que indica los atributos más predictivos) para cada variable en el modelo de generación aleatoria. Asimismo, para la reducción de la dimensionalidad, el proceso construye un gran conjunto de árboles contra un atributo de destino, utilizando las estadísticas de cada atributo con una puntuación "score" calculada (en relación con los otros atributos), con el fin de encontrar el subconjunto de características más informativo. Si este atributo seleccionado es el mejor, entonces indica que la variable es una característica informativa para retener en el análisis.

**Tabla 3.** Aplicación de los métodos RF&IV (Gini Index y WOE)

| | VARIABLE | IMP_RF | IMP_IV | RANKING_RF | RANKING_IV | RANKING_TOT |
|---|---|---|---|---|---|---|
| 1 | PPKG | 224.86344 | 0.0244 | 3 | 5 | 8 |
| 2 | FEV1P | 206.09781 | 0.0487 | 6 | 3 | 9 |
| 3 | EDAD | 171.82978 | 0.2186 | 10 | 2 | 12 |
| 4 | Reing_EXAC | 242.19560 | 0.0000 | 1 | 12 | 13 |
| 5 | FVCP | 192.07075 | 0.0310 | 9 | 4 | 13 |
| 6 | ALP | 158.79108 | 0.2265 | 13 | 1 | 14 |
| 7 | INGRESOS_ | 196.81619 | 0.0000 | 7 | 9 | 16 |
| 8 | CCVSDM_ | 209.72344 | 0.0000 | 5 | 16 | 21 |
| 9 | EXACER_90DIAS | 164.83231 | 0.0000 | 12 | 11 | 23 |
| 10 | DURING | 140.31719 | 0.0000 | 16 | 7 | 23 |
| 11 | E V_ | 194.26879 | 0.0000 | 8 | 18 | 26 |
| 12 | HT_ | 36.84869 | 0.0000 | 25 | 6 | 31 |
| 13 | ESPIROMETRIA_PA_ | 39.30613 | 0.0000 | 24 | 8 | 32 |
| 14 | FEVCVF_ | 233.34052 | 0.0000 | 2 | 31 | 33 |
| 15 | IMC | 212.88434 | 0.0000 | 4 | 30 | 34 |
| 16 | ICHARICC_ | 87.92548 | 0.0000 | 20 | 15 | 35 |
| 17 | TARS | 166.60532 | 0.0000 | 11 | 25 | 36 |
| 18 | ICHAREVP_ | 90.87474 | 0.0000 | 19 | 20 | 39 |
| 19 | SV_ | 23.62432 | 0.0000 | 30 | 10 | 40 |
| 20 | TARD | 157.85915 | 0.0000 | 15 | 26 | 41 |
| 21 | ICHARECV_ | 78.67005 | 0.0000 | 22 | 19 | 41 |
| 22 | ICHARIM_ | 85.35488 | 0.0000 | 21 | 21 | 42 |
| 23 | FCA | 158.19131 | 0.0000 | 14 | 29 | 43 |
| 24 | ICHAR_DM_ | 36.56857 | 0.0000 | 26 | 17 | 43 |
| 25 | TURA | 129.89077 | 0.0000 | 17 | 27 | 44 |
| 26 | MUERTOS_90DIAS | 22.99425 | 0.0000 | 31 | 13 | 44 |
| 27 | FRE | 128.07834 | 0.0000 | 18 | 28 | 46 |
| 28 | EXITUS | 22.95276 | 0.0000 | 32 | 14 | 46 |
| 29 | EP_ | 39.31523 | 0.0000 | 23 | 24 | 47 |
| 30 | ICHARNEF_ | 25.97349 | 0.0000 | 28 | 22 | 50 |
| 31 | ICHAR_TS_ | 23.82815 | 0.0000 | 29 | 23 | 52 |

A la vista de las salidas basadas en la selección de importancia de las variables finales, se llegó a la misma conclusión que el análisis PCA, donde los resultados obtenidos muestran que se podría contemplar una reducción con 11 ó 12 características-variables mediante RF referencia GINI Index. Este método para la selección de importancia de variables finales es el más utilizado y menos restrictivo, ya que es un proceso aleatorio, donde cada vez ejecutado puede mostrar diferentes variables de salida.

En paralelo, otra reducción más exigente con solo 5 variables se realiza mediante el método IV aplicación WOE, que es bastante restrictivo para la selección final, por ser una medida que determina el poder predictivo de una característica-variable. Por eso, este método es poco recomendado cuando no se desea perder información relevante contenida en algunas variables principales del dataset, y donde se sabe a priori que pueden dar sentido a la búsqueda final de perfiles clínicos, tal y como es en este caso estudiado.

En este sentido, la mejor solución viene proporcionada por el análisis PCA con 12 componentes principales. Por ello, se visualiza en detalle los *resultados descritos por el plano 1:2* (*Figura 8*) sobre los dos ejes más relevantes, que pueden aportar datos significativos a la búsqueda de perfiles-grupos con el fin de ampliar a otros más específicos de clasificación idóneos para el agrupamiento de patrones.

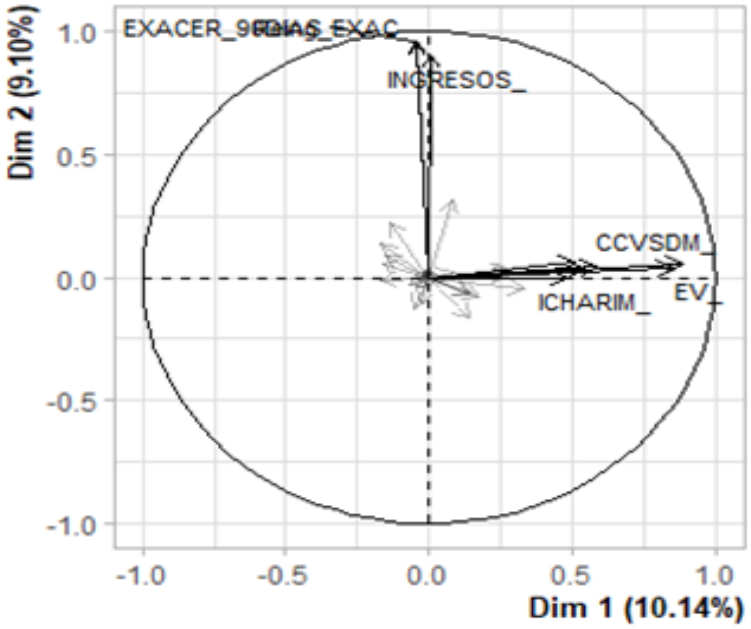



**Figura 8. (a)** Descripción del plano 1: 2 (Variables)

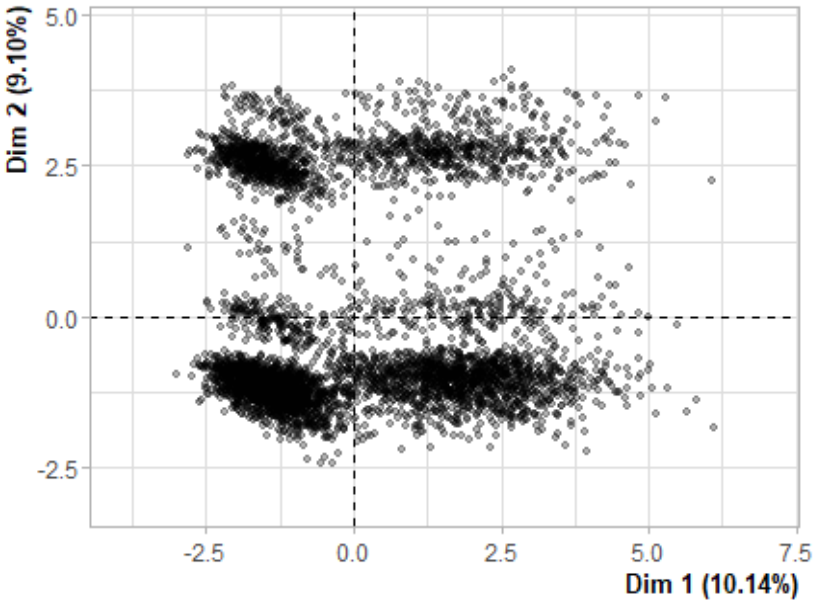



**Figura 8. (b)** Descripción del plano 1: 2 (Pacientes)

Ante estos resultados, la *dimensión 1* muestra individuos caracterizados por una *coordenada fuertemente positiva* en el eje (*a la derecha del gráfico*) frente a una *negativa* (*a la izquierda del gráfico*), donde:

- ✓ *Group 1* con coordenada positiva en el eje, muestra valores altos para estas variables ordenas desde las más fuertes a menos, es decir, variables que presentan una correlación alta con datos elevados y en contraposición, las otras muestran datos menos elevados con una correlación baja: Comorbilidad Cardiovascular *(CCVSDM_),* Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_), E*nfermedad Vascular *(EV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de Miocardio *(ICHARIM_),* Muertos a 90 días *(MUERTOS_90DIAS) y* Enfermedad Cerebro Vascular *(ICHARECV_)*; y valores bajos para estas otras variables ordenadas por las más débiles: SEXO*,* FVC espirometría en % del teórico *(FVCP),* Tensión Arterial Diastólica *(TARD),* Temperatura *(TURA),* Frecuencia Cardíaca *(FCA) y* Hábito Tabáquico *(HT_)*.

- ✓ *Group 2* también positiva, tiene valores altos en estas variables: Enfermedad Vascular *(EV_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de

Miocardio *(ICHARIM_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Cerebro Vascular *(ICHARECV_),* EDAD*,* Diabetes Mellitus *(ICHAR_DM_),* Nefropatía *(ICHARNEF_) y* cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF)*; y valores bajos en estas otras: Exacerbación a 90 días (*EXACER_90DIAS*), Reingresos por exacerbación (*Reing_EXAC*), Ingresos hospitalarios por cualquier motivo (*INGRESOS_*), Espirometría realizada previa al ingreso o al alta (ESPIROMETRIA_PA_), EXITUS, Hábito Tabáquico (*HT_*), Frecuencia Cardíaca (*FCA*), Tensión Arterial Diastólica (*TARD*), Muertos a 90 días *(MUERTOS_90DIAS*) y Tensión Arterial Sistólica (*TARS*).

- ✓ *Group 3* con coordenada negativa en el eje, muestra valores altos para estas variables ordenadas por las más fuertes: Tensión Arterial *D*iastólica *(TARD),* Hábito Tabáquico *(HT_),* FVC espirometría en % del teórico *(FVCP),* SEXO*,* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Frecuencia Cardíaca *(FCA),* EXITUS, Tensión Arterial Sistólica *(TARS)* y Temperatura *(TURA)*; y valores bajos para estas otras ordenadas desde las más débiles: Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular Periférica *(ICHAREVP_),* EDAD, Edemas maleolares *(EP_)* y Diabetes Mellitus *(ICHAR_DM_)*.

- ✓ *Group 4* también con coordenada negativa, observa valores altos para estas variables: Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_),* Frecuencia Cardíaca *(FCA),* Soporte Ventilatorio *(SV_),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Muertos a 90 días *(MUERTOS_90DIAS),* EXITUS, Tumor Sólido *(ICHAR_TS_)* y Duración del Ingreso hospitalario *(DURING)*; y valores bajos en estas otras: Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de Miocardio *(ICHARIM_),* Enfermedad Cerebro Vascular *(ICHARECV_),* FEV1 espirometría en % del teórico *(FEV1P),* Nefropatía *(ICHARNEF_),* Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF)* y Diabetes Mellitus *(ICHAR_DM_)*.

Por otro lado, la dimensión 2 enfrenta individuos con una coordenada fuertemente positiva en el eje (en la parte superior del gráfico) a una negativa (en la parte inferior del gráfico).

Y a la vista de los resultados, se observa que existen dos variables (Exacerbación a 90 días *(EXACER_90DIAS)* y Reingresos por exacerbación *(Reing_EXAC)*) que están altamente correlacionadas, y podrían resumir este eje (*correlación 0, 0*).

- ✓ *Group 1* con una coordenada positiva en el eje, comparte valores altos para estas variables: Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_),* Frecuencia Cardíaca *(FCA),* Soporte Ventilatorio *(SV_),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Muertos a 90 días *(MUERTOS_90DIAS),* EXITUS, Tumor Sólido *(ICHAR_TS_)* y Duración del Ingreso hospitalario *(DURING)*; y bajos para estas otras: Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de Miocardio *(ICHARIM_),* Enfermedad Cerebro Vascular *(ICHARECV_),* FEV1 espirometría en % del teórico *(FEV1P),* Nefropatía *(ICHARNEF_),* Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF)* y Diabetes Mellitus *(ICHAR_DM_)*.

- ✓ *Group 2* también positiva, muestra valores altos en las variables: Comorbilidad Cardiovascular (*CCVSDM_*), Exacerbación a 90 días (*EXACER_90DIAS*), Reingresos por exacerbación (*Reing_EXAC*), Ingresos hospitalarios por cualquier motivo (*INGRESOS_*), Enfermedad Vascular (*EV_*), Insuficiencia Cardíaca Congestiva (*ICHARICC_*), Enfermedad Vascular Periférica (*ICHAREVP_*), Infarto de Miocardio (*ICHARIM_*), Muertos a 90 días (*MUERTOS_90DIAS*) y Enfermedad Cerebro Vascular (*ICHARECV_*); y bajos en estas otras: SEXO, FVC espirometría en % del teórico *(FVCP),* Tensión Arterial Diastólica *(TARD),* Temperatura *(TURA),* Frecuencia Cardíaca *(FCA)* y Hábito Tabáquico *(HT_)*.

- ✓ *Group 3* con una coordenada negativa, comparte valores altos para las variables: Tensión Arterial Diastólica *(TARD),* Hábito Tabáquico *(HT_),* FVC espirometría en % del teórico *(FVCP),* SEXO, Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Frecuencia Cardíaca *(FCA),* EXITUS, Tensión Arterial Sistólica *(TARS)* y Temperatura *(TURA)*; y bajos para estas otras: Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular Periférica *(ICHAREVP_),* EDAD, Edemas maleolares *(EP_)* y Diabetes Mellitus *(ICHAR_DM_)*.

- ✓ Group 4 también negativa, muestra valores altos para estas variables: Enfermedad Vascular *(EV_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de Miocardio *(ICHARIM_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Cerebro Vascular *(ICHARECV_),* EDAD, Diabetes Mellitus *(ICHAR_DM_),* Nefropatía *(ICHARNEF_)* y Cociente relacional FEV1/FVC espirometría previa o alta (*FEVCVF)*; y bajos para estas otras: Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por

cualquier motivo *(INGRESOS_),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* EXITUS, Hábito Tabáquico *(HT_),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Diastólica *(TARD),* Muertos a 90 días *(MUERTOS_90DIAS)* y Tensión Arterial Sistólica *(TARS)*.

No obstante, aunque se ha visto que las dos primeras dimensiones predominantes explican suficientemente este caso particular, también se pueden visiualizar las otras dimensiones del plano que muestra el análisis PCA, como puede ser el *plano 3:4* (*Figura 9*).

Según los resultados obtenidos, este tercer eje es capaz de aportar el 11,64% a la variabilidad total (30,88%), llegando a explicar algo más de información sobre los datos para determinadas variables analizadas que sean necesarias describirlas o simplemente disponer del apoyo visual del resto de dimensiones existentes (*planos 5:6 hasta el 11:12*) poco significativos y de difícil interpretación sobre los perfiles clíninos perseguidos por su complejidad dimensional.

Por esta misma razón de complejidad, aunque se muestra una pincelada de la dimensión 3:4, en este caso particular no se han tomado en consideración, ya que no aportan casi ninguna mejora en las conclusiones finales de los datos analizados que puedan añadir mejoras a nuestro objetivo, pero en otros casos exploratorios donde se estén analizando diferentes intancias muy similares entre sí con multitud de muestras genéticas, pueden ser muy útil y una herramienta de descarte de estás si se desea mejorar el resultado final y por consiguiente, la interpretación de los resultados sobre las metas clínicas que se desean alcanzar en la investigación.

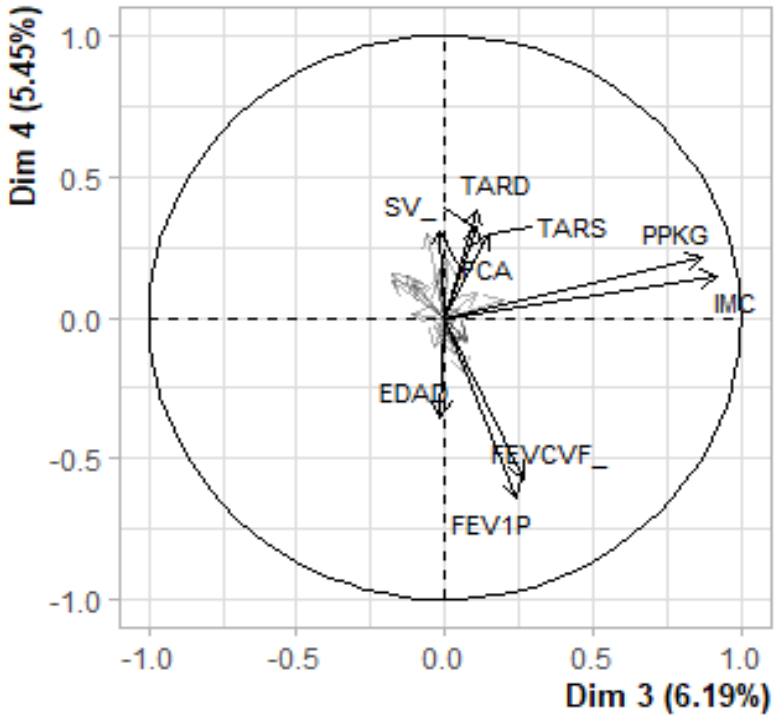



**Figura 9. (a)** Descripción del plano 3: 4 (Variables)

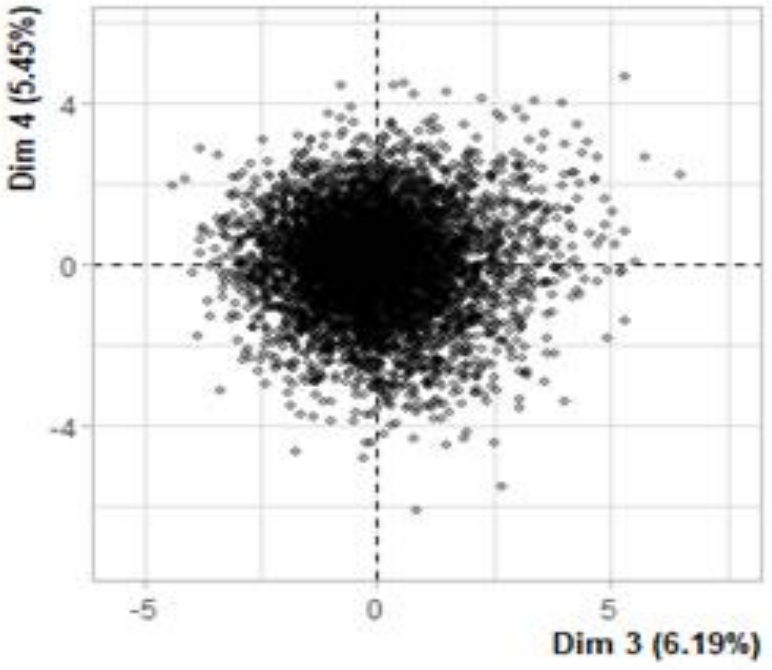



**Figura 9. (b)** Descripción del plano 3: 4 (Pacientes)

No obstante, los resultados de la *dimensión 3* muestra individuos caracterizados por una *coordenada fuertemente positiva* en el eje (*a la derecha del gráfico*) frente a una *negativa* (*a la izquierda del gráfico*), donde:

- ✓ *Group 1* con coordenada positiva en el eje, muestra valores altos para estas variables ordenas desde las más fuertes a menos: Peso *(PPKG),* Índice de Masa Corporal *(IMC),* Tensión Arterial Diastólica *(TARD),* Tensión Arterial Sistólica *(TARS),* Soporte Ventilatorio *(SV_),* Edemas maleolares *(EP_),* Frecuencia Cardíaca *(FCA),* Hábito Tabáquico *(HT_),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_)* y Diabetes Mellitus *(ICHAR_DM_)*; y valores bajos para estas otras variables ordenadas por las más débiles: FEV1 espirometría en % del teórico *(FEV1P*), EDAD, Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FVC espirometría en % del teórico *(FVCP),* SEXO, Enfermedad Cerebro Vascular *(ICHARECV_),* EXITUS, Nefropatía *(ICHARNEF_)* y ALTURA *(ALP)*.

- ✓ *Group 2* con coordenada negativa en el eje, tiene valores altos en estas variables ordenadas por las más fuertes: Enfermedad Vascular *(EV_),* Altura *(ALP),* Enfermedad Vascular Periférica *(ICHAREVP_),* Infarto de Miocardio *(ICHARIM_),* Hábito Tabáquico *(HT_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Cerebro Vascular *(ICHARECV_),* Frecuencia Respiratoria *(FRE),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_)* y Frecuencia Cardíaca *(FCA)*; y valores bajos en estas otras ordenadas desde las más débiles: Índice de Masa Corporal (*IMC*), Peso (*PPKG*), FEV1 espirometría en % del teórico (*FEV1P*), Cociente relacional FEV1/FVC espirometría previa o alta (*FEVCVF*), Edemas maleolares (*EP_*), SEXO, Ingresos hospitalarios por cualquier *motivo (INGRESOS_*), Tensión Arterial Sistólica (*TARS*), Reingresos por exacerbación (*Reing_EXAC*) y Exacerbación a 90 días (*EXACER_90DIAS*).

Por otro lado, la *dimensión 4* enfrenta individuos con una coordenada fuertemente positiva en el eje (en la parte superior del gráfico) a una negativa (en la parte inferior del gráfico).

- ✓ *Group 1* con una coordenada positiva en el eje, comparte valores altos para estas variables: Enfermedad Vascular (*EV_*), Altura (*ALP*), Enfermedad Vascular Periférica (*ICHAREVP_*), Infarto de Miocardio (*ICHARIM_*), Hábito Tabáquico (*HT_*), Comorbilidad Cardiovascular (*CCVSDM_*), Enfermedad Cerebro Vascular (*ICHARECV_*), Frecuencia Respiratoria (*FRE*), Espirometría realizada previa al ingreso o al alta (*ESPIROMETRIA_PA_*) y Frecuencia Cardíaca (*FCA*); y bajos para estas otras: Índice de Masa Corporal *(IMC),* Peso *(PPKG),* FEV1 espirometría en % del teórico *(FEV1P),*

Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* Edemas maleolares *(EP_),* SEXO, Ingresos hospitalarios por cualquier motivo *(INGRESOS_),* Tensión Arterial Sistólica *(TARS),* Reingresos por exacerbación *(Reing_EXAC)* y Exacerbación a 90 días *(EXACER_90DIAS)*.

- ✓ *Group 2* también positiva, muestra valores altos en las variables: Peso (*PPKG*), Índice de Masa Corporal (*IMC*), Tensión Arterial Diastólica (*TARD*), Tensión Arterial Sistólica (*TARS*), Soporte Ventilatorio (*SV_*), Edemas maleolares (*EP_*), Frecuencia Cardíaca (*FCA*), Hábito Tabáquico (*HT_*), Espirometría realizada previa al ingreso o al alta (*ESPIROMETRIA_PA_*) y Diabetes Mellitus (*ICHAR_DM_*); y bajos en estas otras: FEV1 espirometría en % del teórico *(FEV1P),* EDAD, Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FVC espirometría en % del teórico *(FVCP),* SEXO, Enfermedad Cerebro Vascular *(ICHARECV_),* EXITUS, Nefropatía *(ICHARNEF_)* y Altura *(ALP)*.

- ✓ *Group 3* con una coordenada negativa, comparte valores altos para las variables: FEV1 espirometría en % del teórico (*FEV1P*), Cociente relacional FEV1/FVC espirometría previa o alta (*FEVCVF*), EDAD, SEXO, Nefropatía (*ICHARNEF_*), FVC espirometría en % del teórico (*FVCP*), Ingresos hospitalarios por cualquier motivo (*INGRESOS_*), Reingresos por exacerbación (*Reing_EXAC*), Exacerbación a 90 días (*EXACER_90DIAS*) y EXITUS; y bajos para estas otras: Tensión Arterial Diastólica (*TARD*), Hábito Tabáquico (*HT_*), Frecuencia Cardíaca (*FCA*), Tensión Arterial Sistólica (*TARS*), Espirometría realizada previa al ingreso o al alta (*ESPIROMETRIA_PA_*), Soporte Ventilatorio (*SV_*), Frecuencia Respiratoria (*FRE*), Peso (*PPKG*), Enfermedad Vascular (*EV_*) y Comorbilidad Cardiovascular (*CCVSDM_*).

En definitiva, a la vista de los resultados obtenidos en este estudio, aunque cada método tiene sus peculiaridades de implementación, se puede decir que el que mejor se ajusta es PCA (Tăuţan et al., 2020), observándose 4 grupos de perfiles con la misma afinidad y características. Ante este resultado, se necesita analizar el contexto de la información y el objetivo que se desea alcanzar para seleccionar una vía correcta, ya que, en muchas ocasiones, se dispone de información relevante que no se desea rechazar por completar el parámetro explorado y su posible relación directa con el objetivo de la investigación planteada.

## II. 4. Conclusiones

Según los resultados desarrollados, se puede decir que se ha obtenido la mejor reducción dimensional con 12 componentes principales de las 32 iniciales que se partían para este estudio, siendo la primera dimensión la de mayor relevancia, y generando la existencia de varios grupos de pacientes con caracteristicas similares, que presentan de forma conjunta variables muy asociadas dentro de cada uno de ellos con suficiente importancia a considerar en cada perfil clínico obtenido.

Por lo que, esta hipótesis de asociación asegura la correcta aplicación del método PCA, ya que uno de los requisitos es que la matriz de correlación entre variables debe ser bastante alta, de lo contrario su aplicación no tendría sentido. En este aspecto, antes de proceder a la imputación, dado que la estructura de correlación puede ser bastante sensible a las distintas técnicas de imputación, se debe estudiar según el tipo de variables del conjunto de datos para no perder la calidad relacional del original, que es necesario para la aplicación del PCA.

Para este caso, como se ha mencionado en el capítulo anterior, se sabe que el método MICE utiliza las ecuaciones encadenadas en la imputación aleatoria de cada variable, y estas están condicionadas a las variables imputadas, aplicando un mecanismo de cadenas dependientes en la distribución de probabilidad. Por lo tanto, se asume que esta dependencia se conserva en la estructura de correlación del algoritmo, manteniendo la calidad relacional del dataset original.

También en este estudio, se confirma que la reducción del plano ayuda significativamente a detectar a grandes rasgos, sin profundizar en la clasificación, la visualización de primeros indicios de diferentes patrones clínicos con características semejantes e inciden en que el desarrollo de otras patologías (enfermedades crónicas) a lo largo del tiempo influyen en los parámetros clínicos de forma negativa (con el aumento irregular de las constantes vitales y el ingreso hospitalario frecuente de este tipo de episodios), ocasionados por la propia severidad de la enfermedad, por la edad avanzada que presentan estos pacientes unidos a malos hábitos saludables y por la gravedad de las exacerbaciones del propio paciente EPOC, que se encuentra en un cuadro clínico bastante severo ocasionado por el deterioro de su enfermedad en el transcurso del tiempo y dentro de un periodo demasiado prolongado, por lo que conocer esta información apriori puede mejorar la calidad asistencial-social del mismo.

Asimismo, para futuros análisis se quiere mencionar otros procedimientos de la competencia que pueden ayudar a obtener mejores resultados en los métodos mencionados. Ya que, se sabe que el

análisis PCA tiene límites para algunos casos, puesto que este método solo usa las combinaciones lineales de las variables originales y puede llegar a perder mucha información.

En este aspecto, existen otros métodos no lineales desarrollados para reducir la dimensionalidad de los datos a un espacio más pequeño, que podrían estudiarse para este estudio, y paralelamente, ser una solución factible para ayudar a superar esta limitación anteriormente mencionada.

Algunos de estos algoritmos son los siguientes que se describen (*t-SNE* (Krijthe, 2015); S*ammon mapping* (Venables y Ripley, 2002 y You, 2022); *Isomap* (Oksanen et al., 2022); *LLE* (Holger, 2015); *CCA* (González et al., 2008 y González y Déjean, 2021); *MVU* (You, 2022); *LE* (Kraemer et al., 2018 y Kraemer, 2022), pero hay muchos más en la literatura, (Amat, 2017a, 2017b y Valencia-Aguirre et al., 2010).

- *t-SNE* (*t-distributed Stochastic Neighbor Embedded ó Incrustación de Vecinos Estocásticos distribuidos en t*), (Krijthe, 2015), es una extensión del algoritmo *SNE* (*Stochastic Neighbor Embedding*). Este algoritmo calcula la probabilidad condicional de cada par de observaciones para minimizar la suma de las diferencias entre las probabilidades de la dimensión superior e inferior. Este proceso es muy lento cuando se trabaja con muchas observaciones y la aplicación de este algoritmo con la misma información puede dar resultados diferentes. En este sentido, existe una aproximación más rápida de este método llamado *Barnes-Hut* para muchas observaciones.
  Asimismo, la reducción de dimensionalidad t-SNE se basa en un espacio d-dimensional dentro de un espacio D-dimensional, donde d < D, y para otros casos no se puede aplicar. Este método no genera una serie de ecuaciones de fácil interpretación como PCA, y tiene la desventaja de que no es incremental, es decir, no se puede aplicar directamente con nuevas observaciones agregadas, lo que requiere ejecutar todo el algoritmo, incluidos todos los datos antiguos y nuevos.

- *Sammon mapping* (*Mapeo de Sammon*), (Venables y Ripley, 2002 y You, 2022), es un algoritmo que mapea un espacio de alta dimensión a uno de menor dimensión, preservando la estructura original de distancias entre puntos en la proyección de menor dimensión.

- *Isomap* (*Isomapa*), (Oksanen et al., 2022), es un método de reducción de dimensionalidad no lineal basado en la teoría espectral, que preserva las distancias geodésicas en la dimensión inferior a través del mapeo isométrico, combinando varios algoritmos diferentes para reducir las dimensiones, mientras se preservan las estructuras locales.

- *LLE* (Locally Linear Embedding ó *Incrustación Lineal Local*), (Holger, 2015), es un algoritmo que busca un espacio de menor dimensión de los datos, preservando las distancias dentro de los vecindarios locales. Este puede ser un método similar a los análisis de componentes principales locales.

- *CCA* (*Canonical Correlation Analysis ó Análisis de Correlación Canónica*), (González et al., 2008 y González y Déjean, 2021), es un método para medir las asociaciones entre dos conjuntos multivariados de variables.

- *MVU* (*Maximum Variance Unfolding ó Despliegue de Varianza Máxima*), (You, 2022), es una técnica para la reducción de la dimensionalidad no lineal de los datos vectoriales de entrada de una dimensión más alta a una más baja. Este se puede considerar como una Generalización del Análisis de Componentes Principales.

- *LE* (*Laplacian Eigenmaps ó Mapas Propios Laplacianos*), (Kraemer et al., 2018 y Kraemer, 2022), es un algoritmo para la reducción de dimensionalidad no lineal que tiene propiedades de conservación de la localidad y utiliza los vectores propios en un espacio de baja dimensión, de modo que los datos se aproximan al operador de Laplace Beltrami.

En este sentido, con este desarrollo analítico-técnico se pretende mostrar la existencia de estos métodos computacionales para suplir la reducción de la dimensionalidad (Pasha y Latha, 2020 y Salvador et al., 2020) en bases de datos multicéntricas, donde es casi necesaria esta reducción de dimensionalidad a un espacio de dimensión menor y semejante al original, puesto que se dispone de un gran volumen de datos y de infinidad de variables clínicas, y realizar una selección de las más importantes o relevantes según el objetivo perseguido ayuda a enfocar con más precisión los resultados buscados, con la garantía de poder extrapolarlos a una población general.

# CAPÍTULO III

# Clasificación grupal para la identificación y búsqueda de patrones afines mediante diferentes técnicas multivariantes

## III. 1. Introducción

En la actualidad, existen otros métodos de análisis de clasificación (Pedregosa et al., 2011a) bastante utilizados y con gran impacto en el área de la Estadística Computacional y de la investigación, como pueden ser el Análisis *Cluster o de Clasificación* (Mirzal, 2020 y Fratello et al., 2022), que sirve para detectar la separación de los grupos por características similares, el de *Correspondencias* (CA) (Tobón et al., 2020), que separa por grupos sin tener en cuenta afinidades pero es un gran apoyo de visualización global en este caso clínico y el de *Árboles de Decisión por clasificación o Decision Tree* (DT) (Rokach y Maimon, 2007 y Rajaguru y S R, 2019), que es un gran soporte para detectar alguna clasificación óptima entre las distintas variables clases motrando las diferentes hojas como las salidas del árbol en varias capas o niveles que son los posibles nodos de resultados mediante la aplicación de pequeñas reglas de decisión de orden jerárquico que van originando la decisión final en forma de árbol.

Por esta misma razón, el avance tecnológico existente junto con la transformación digital puede contribuir a mejorar estas técnicas disponibles, ya que reconoce que el proceso de aprendizaje en clasificación se centra principalmente en construir una función capaz de distinguir entre las distintas clases o grupos de salida representadas por un caso especifico de uso, donde la función de separación se conoce como función *discriminante*, utilizando los valores de las variables de entrada definidas por el propio caso de uso que se quiere estudiar creando una frontera de separación.

Asimismo, este procedimiento realiza en el conjunto de datos una división de corte que los separa en dos o más partes, constituyendo diferentes áreas fácilmente identificables por cada una de las clases definidas.

Por esta misma cuestión, en el campo de clasificación existen varios tipos de funciones discriminantes que sirven de soporte para determinar las diferentes áreas del espacio al que pertenece cada una de las

clases formadas en un mismo caso, encontrándose funciones sencillas como puede ser una recta que son las denominadas funciones lineales, u otras más complejas como pueden ser las del tipo polinomial.

No obstante, y sin duda, si se realiza una revisión del tema actualmente se conoce que una de las técnicas de aprendizaje más utilizadas son los modelos supervisados de regresión (Pedregosa et al., 2011b), en particular la Regresión Logística (RL) por su eficacia, validez y desempeño ampliamente demostrados en la literatura y en paralelo, por ser el modelo más flexible y moldeable, ya que genera una separación más suave entre las observaciones de la variable clase de salida, pero recientemente, muy en auge se tiene un método alternativo que se mencionará en el capítulo IV de esta tesis doctoral como son las Máquinas de Vectores Soportes o SVM *(Support Vector Machine),* (Legorreta, 2015), cuya base central son los procesos algorítmicos buscando un hiperplano de separación entre las instancias de las dos clases mediante un margen máximo, es decir, una distancia máxima entre las instancias frontera de la barrera de decisión, dando así otro enfoque diferente a la solución del problema planteado para bases de datos de alta dimensión, como puede ser este caso presentado.

En definitiva, todas estas técnicas de aprendizaje autómatico (Machine Learning) junto con el soporte de la minería de datos, de alguna manera pretenden evaluar o medir la calidad de los modelos estimados, mediante diferentes tipos de métodos (como puede ser el algoritmo de las K-medias, el vecino más cercano KNN o K-Nearest-Neighbor) y distintas métricas, como puede ser la tasa de clasificación o de aciertos a travez de la matriz de confusión, donde se proporciona una mayor información sobre las clases que tienen dificultad de ser clasificadas o aquellas que son fáciles para mezclarse.

Asimismo, existen otras métricas aparte de la matriz de confusión, como puede ser el coeficiente kappa que mide el acuerdo entre dos variables, el área bajo la curva ROC que indica el valor óptimo o de descarte en el modelo estimado según se varia el umbral discriminatorio representando el grado de sensibilidad y especificidad de la prueba realizada para dicho modelo de clasificación binario, el Error Cuadrático Medio (ECM), el coeficiente de determinación ($R^2$) y otras muchas más, que están disponibles según el nivel de medición y calidad que se quiera precisar en los modelos construidos y ajustados (Granville, 2019 y Wickham, 2021).

Esta claro, que la información de todas estas mediciones ayudan a soportar los resultados finales con el fin de identificar y buscar la mejor opción de agrupamiento que pueda clasificar a los diferentes pacientes que componen el conjunto de datos explorado, tratando de encontrar cualquier patrón de comportamiento que los separe en diferentes secciones de perfiles clínicos afines y acorde a sus

características propias, generando de esta forma un soporte adicional, que permite dar esa ayuda necesaria en los análisis desarrollados y a su vez, completar la toma de decisiones, que se propone por personal cualificado en cualquier ámbito de estudio, como puede ser la salud en investigación, la bioinformática u otro campo biotecnológico de las ciencias en general, donde su papel fundamental es la de colaborar y mejorar la toma de decisiones en la investigación.

## III. 2. Métodos

Tras solventar los temas anteriores referentes al problema de la reducción de la dimensionalidad (Choubey et al., 2020 y Pinheiro et al., 2021), y el de la imputación de valores missing (Karthe, 2016 y Miri et al., 2020), que se han desarrollado en los capítulos I y II, se continua con la visión central de este capítulo para dar respuesta al gran hándicap propuesto como es la de identificar y buscar perfiles clínicos mediante diferentes métodos de clasificación con el objetivo de agrupar adecuadamente los distintos tipos de pacientes según sus propias características y afinidades, encontrando el modelo de patrón más ajustado para datos clínicos, en concreto, para este caso expuesto.

Para ello, este abordaje de clasificación grupal que cada vez toma más validez en el terreno analítico-estadístico, así como en la interpretación de resultados, llevando a ser una parte casi fundamental en los procesos de modelado, se ha desarrollado con diferentes metodologías y métricas, buscando la mejor separación, por no decir la solución óptima, que represente y evalue la calidad de estos datos dando como origen el mejor resultado de agrupamiento final con unas conclusiones acorde al objetivo y extrapolables a la población de interés, y por consiguiente, para la práctica cotidiana del ámbito biosanitario, sirviendo como soporte a la toma de decisiones finales con la intención de mejorar la atención y calidad de los pacientes diagnosticados.

El planteamiento se desarrolla en varios pasos de ejecución, primero se realizara una orientación de como están representados grupalmente estos datos tras conocer que los dos primeros ejes son los que màs contribuyen a la variabilidad explicada de este registro. Ahora en el *capítulo III*, se continua con la aplicación de las diferentes técnicas de análisis (supervisadas y no supervisadas) para la clasificación del conjunto de datos-variables, a través de los distintos mecanismos de comprobación (*Cluster, Correspondencia (CA), Árbol de Decisión (DT)*), y una vez seleccionado y encontrado el mejor método de clasificación que viene dado por el análisis Cluster con la creación de tres grupos finales óptimos, se finaliza la exploración analítica describiendo el agrupamiento o patrón de clasificación conseguido para los distintos grupos-perfiles formados, destacando sus características relevantes con la finalidad de

resaltar el resultado obtenido mediante la identificación óptima de separación, que viene representada por los perfiles clínicos finales generados para este caso particular.

No obstante, cabe destacar que el desarrollo de esta investigación, aunque se analizan otras técnicas alternativas para evaluar resultados, se enfoca principalmente en el análisis clúster como el camino central a lo que se pretende conseguir en esta exploración, ya que el análisis clúster tiene como búsqueda primordial clasificar a los individuos formando grupos o conglomerados de tal manera que los pacientes dentro de cada conglomerado presenten homogeneidad en cuanto a los valores adoptados por cada tipo de variable.

Por ello, tiene sentido que en este tipo de técnicas el concepto de distancia espacial tiene bastante relevancia, ya que define el grado de similaridad o disimilaridad, (en términos de relación de distancias) en cada una de las agrupaciones, es decir, que una observación será lo más parecida a otra en la medida que la distancia que las separa sea lo menor posible (o sea, que tenga distancia mínima), o por el contrario, que la similaridad sea máxima. Para este caso especifico, aunque existen distintos métodos de distancias, (como pueden ser la Euclídea para variables numéricas como es este caso, la de Levenshtein para tipo texto y el coeficiente de Jaccard para variables enteras), se opta por la distancia euclídea con el fin de poder evaluar los patrones de agrupación que son la clave principal de este planteamiento, y en paralelo, realizar un enfoque simple en el análisis de las diferentes características de variables para aportar ese valor añadido a la exploración analítica de los grupos formados, dando un nivel de calidad y validez a los perfiles finales generados.

También, hay que tener en cuenta que existen diferentes métodos para la aplicación del análisis cluster basados todos ellos en el concepto de jerárquia (Amat, 2017c), más concretamente se dividen en dos grandes bloques denominados jerárquicos y no jerárquicos, que a su vez los primeros pueden ser de dos tipos: (i) asociativos o aglomerativos y (ii) disociativos o divisivos; y los otros que son de repartición (Sancho, 2020).

Además, existen varias aproximaciones a los métodos de clustering, los más conocidos son los (i) algoritmos *jerárquicos*, que crean una descomposición jerárquica del conjunto de datos usando algún tipo de criterio, y cuyo objetivo principal es construir un dendrograma en forma de árbol; (ii) también están los *algoritmos de particionamiento*, que construyen distintas particiones evaluándolas de acuerdo al criterio selecionado, donde un ejemplo de ello es el procedimiento K-means donde se requiere definir el parámetro inicial de agrupación, que en este caso particular se utilizo un número de cluster entre el intervalo cinco y tres como fase inicial de testeo hasta analizar los otros métodos de soporte,

definiendo el óptimo final que termino siendo K igual a tres, como el resultante de los grupos-clusters con el máximo óptimo; y finalmente (iii) existen otros muchos, que estan basados en funciones de conectividad y densidad, los basados en rejillas que utilizan una estructura de granularidad de múltiples niveles o los que están basados en modelos donde se presuponne modelos diferentes de clusters buscando entre ellos el mejor modelo ajustable que alcance el óptimo.

En este sentido, y a la vista esta que con la gran información disponible a nuestro alcance, y en paralelo, la variedad de técnicas de aprendizaje automàtico que se pueden utilizar en el algoritmo de agrupamiento de los datos clínicos almacenados para clasificación, es imposible que no se pueda llegar a la identificación de modelos de patrones afines dentro de un conjunto de datos si a priori se establecen y se definen correctamente las bases del modelo deseado, alcanzando diferentes soluciones òptimas de las cuales siempre se puede encontrar la máxima óptima que refleje a todos los individuos de datos explorados.

## III. 3. Resultados

Siguiendo la misma línea de los capítulos anteriores, también para realizar el análisis se ha utilizado el *software estadístico R* (R Core Team, 2021) (versión 4.1.0), siendo el programa informático principal para aplicar las distintas técnicas de *análisis de clasificación* mediante Cluster, CA (López, 2018) y DT con el fin de buscar y optimizar la búsqueda de patrones que mejor agrupen a estos datos exploradoss.

Con la siguiente base de desarrollo y aplicación de distintas técnicas de aprendizaje (Liu et al., 2008 y Lowie et al., 2021) a través del análisis supervisado y no supervisado, se llego a la finalidad perseguida, buscar e identificar la mejor agrupación posible de clasificación ajustada a los datos y sin perder información relevante, de manera acorde y afin a sus características internas y visuales, generando diversos grupos-perfiles muy similares internamente dentro del grupo y diferentes entre si respecto a otros grupos, originando tipos de patrones basados en sus propias propiedades clínicas.

A continuación, se detallan los tres procedimientos analíticos implementados y los resultados alcanzados en cada uno de ellos para soportar y valorar la búsqueda y la identificación de perfiles clínicos adecuados, originando un agrupamiento de clasificación ajustado a estos datos-pacientes y en paralelo, aportando un valor diferencial y adicional a este análisis de segmentación para la toma de decisiones.

### *III. 3. 1. Análisis Cluster ó de Clasificación no supervisada por grupos*

Debido a la creciente aparición de nuevos métodos, en los últimos años este tipo de análisis de agrupamiento no supervisado ha sido objeto de una amplia investigación, que recientemente se ha activado con nuevos avances significativos en diferentes áreas de investigación, como la sanitaria, especialmente en la investigación clínica, en la ingeniería particularmente en la bioinformática y en las ciencias en general, donde en todas ellas se ha considerado una buena alternativa de aplicación para la búsqueda de soluciones de agrupamiento en la que se disponga de distintas observaciones de muestras o instancias de individuos.

En este sentido, la opción de aplicar la técnica del *Análisis Cluster o de Clasificacion no supervisada por grupos* (Vega-Pons y Ruiz-Shulcloper, 2011; Ortiz-Gonçalves et al., 2018 y Niño-Ramírez et al., 2021), o como bien se conoce *Análisis de Conglomerados*, se realiza desde el paquete *cluster* de R (Maechler et al., 2021) y con el de *factoextra* (Kassambara y Mundt, 2020), este método viene determinado por el objetivo principal del propio proceso algorítmico, que tiene implementado identificar grupos de individuos similares y, por consiguiente, el enfoque de ayudar a descubrir la distribución de patrones y correlaciones relevantes en grandes conjuntos de datos, evaluando la calidad de los grupos resultantes.

Más concretamente, se ha seleccionado el analisis cluster por ser una técnica de clasificación por grupos no supervisada, donde las clases no están predefinidas de antemano, teniendo como función principal la segmentación de una población heterogénea (es decir, diferente entre sí) en un número de subgrupos o particiones homogéneos (o sea iguales entre sí dentro del mismo) denominados clusters.

Asimismo, en este estudio se ha mencionado varias veces que el análisis cluster ha demostrado ser una opción adecuada para generar un conjunto de clusterings a partir de un mismo dataset, pudiendo este combinarlos en un único cluster final con la finalidad de mejorar la calidad de los clusters de datos-pacientes individuales.

También, según estudios desarrollados se sabe que esta nueva aportación de métodos de agrupación ofrece resultados prometedores y un gran número de aplicaciones que mejoran a las técnicas existentes, siendo estas muy útiles y de fácil escalabilidad para la investigación y en especial para la toma de decisiones en proyecciones futuras dentro de los diversos campos tecnológicos y de investigación con la finalidad de identificar patrones afines que puedan mejorar la calidad de vida de los individuos, y en paralelo, buscar nuevas estrategias más favorables y menos dañinas de diagnóstico

y tratamiento para que tengan menor impacto sobre los pacientes tratados con el fin de ayudarles a tener una vida más saludable y satisfactoria.

Asimismo, por la literatura descrita, se conoce que el concepto de agrupación espacial viene determinado por el enfoque de identificar subespacios para la formación de clusters, clasificando a los datos (pacientes) en diferentes segmentos o grupos de clusters. Por ello, se sabe que los algoritmos convencionales de agrupación espacial tienden a explorar los grupos densos en todos los subespacios posibles, lo que esto conlleva a sufrir el hándicap de la maldición de la dimensionalidad, que aparece con el aumento en el número de dimensiones, que por consiguiente se presenta en el número de subespacios a explorar, aumentando exponencialmente el número de los clusters de subespacios.

Por esta razón, las funcionalidades de la minería de datos, como es el análisis cluster, se vuelven más complejas a medida que aumenta el número de dimensiones, cuando lo único que se pretende es extraer perfiles o patrones relevantes de grandes volúmenes de datos, mediante procedimientos algorítmicos que sean eficientes y que ayuden a analizar los grandes registros de datos para una adecuada extracción de patrones de individuos (pacientes).

En esta misma línea, este problema se resolvió en el capítulo II, mediante diferentes enfoques técnicos para la reducción de la dimensionalidad y la selección de características importantes, ya que se conoce que dificulta el análisis de resultados para la agrupación de los clusters, debido a que puede presentar una alta probabilidad de información redundante de agrupación en los clusters creados en los diferentes subespacios.

Asimismo, como se ha mencionado anteriormente, se debe tener en cuenta que la calidad del clustering resultante depende tanto de la medida de similaridad utilizada, que normalmente es una función de distancia (que por defecto es la euclídea para atributos numéricos, al igual que este caso), como de su implementación, puesto que las funciones de distancia son muy sensibles al tipo de variables usadas (continuas, categóricas nominales u ordinales) y al rango o escala con las que están medidas, por lo que es necesario aplicar un proceso de normalización para estandarizar todas las variables a la misma escala con la finalidad de que todas presenten media cero y desviación típica uno (N(0,1)), asegurando que dentro del proceso de formación de clustering todas las características de variables tengan la misma importancia o peso, evitando el sobreajuste u overfitting por alguna de ellas en la generación de los grupos finales.

Además, existen diferentes métodos para aplicar el análisis cluster basados en el concepto de jerarquía (Zheng et al., 2009), que se dividen en dos grandes bloques denominados (1) jerárquicos y (2) no jerárquicos (Boutros y Okey, 2005).

(1) Respecto a los ***métodos jerárquicos*** se tiene que la salida es una jerarquía entre clusters y dependiendo del nivel de corte se obtiene un clustering distinto, sin la necesidad de requerir el parámetro del número de clusters. La relevancia de este método es que su principal objetivo es construir un dendrograma con estructura de árbol, donde se puede visualizar fácilmente la solución final. Además, estos pueden ser a su vez de dos tipos: (i) asociativos o aglomerativos y (ii) disociativos o divisivos.

(2) Respecto a los ***métodos no jerárquicos o de repartición*** se tiene que inicialmente se parte de una solucion previa, clasificando a los individuos en un número de k grupos prefijados a priori, donde posteriormente son analizados en las diferentes particiones de estos k grupos con el fin de obtener la mejor partición de todas.

Asimismo, se tiene que las *técnicas jerárquicas* (1) también se pueden dividir en dos tipos de niveles, como pueden ser (i) los aglomerativos o (ii) los divisivos, que se describen a continuación (Halkidi et al., 2001 y Jaya et al., 2020):

(i) Los **jerárquicos aglomerativos** (también denominados *hacia arriba o bottom-up*), se basan en medir la distancia entre clusters, donde la situación inicial de partida es un cluster por cada instancia de individuo, siendo cada individuo un grupo independiente, y en cada paso se fusionan los dos clusters más cercanos (minimizando distancias entre ellos), hasta poder alcanzar que todos los individuos formen un solo grupo.

Existen varios métodos de aplicación para llevar a cabo el proceso de división o unión de estos grupos basados en medidas de distancias, como pueden ser:

(a) *Simple* (también *Simple or Single Linkage o Vecino más próximo*): minimizar la distancia mínima entre elementos o individuos de cada grupo.

(b) *Completo* (también *Complete Linkage o Vecino más alejado*): minimizar la distancia máxima entre elementos de cada grupo.

(c) *Average* (también *Promedio, Average Linkage o UPGMA*): distancia promedio entre grupos. También se le conoce como *UPGMA* que se refiere al método de grupos de pares no

ponderados con media aritmética (o en inglés *Unweighted Pair Group Method with Arithmetic mean*), y que consiste en un agrupamiento jerárquico aglomerativo (ascendente) simple. Además, existe la versión ponderada denominada WPGMA (*Weighted Pair Group Method with Arithmetic mean*).

(d) *Ward* (conocido como *Varianza mínima*): consiste en fusionar el par de clusters que genera un agrupamiento con mínima varianza (o sea obtener el menor incremento en el valor total de la suma de los cuadrados de las diferencias, dentro de cada cluster y de cada individuo al centroide de cada cluster). Es decir, el método se calcula como la media de la distancia cuadrática de cada elemento o individuo al centroide.

(e) *Centroide* (también *Centroid Linkage o WPGMC)*: distancia entre los centroides, donde el centroide se define como la distancia entre un representante de cada grupo, usando el valor medio de todas ellas para crear el centroide principal.
También se le conoce como W*PGMC* que se refiere al método de grupos de pares *ponderados* con el centroide (o en inglés *Weighted Pair Group Method with Centroid*), y que consiste en un agrupamiento jerárquico aglomerativo (ascendente) simple. Además, se tiene que la versión *no ponderada* (UPGMC) es similar al método de la mediana donde el tamaño de los clusters no se considera.

(f) *Mediana:* distancia mediana entre grupos. Este método también tiene como basa formar un nuevo centroide donde se utiliza la media de los centroides de los grupos que se unen, pero sin ponderar, ya que el peso de los clusters es irrelevante. Este es análogo al método UPGMC.

(ii) Los ***jerárquicos divisivos*** (también llamados *de arriba hacia abajo o top-down*), aunque son los menos utilizados que los aglomerativos, consisten en partir con un único cluster o grupo, que engloba a todos los invidivuos, y en cada paso sucesivo se selecciona este cluster y se divide en dos subgrupos disjuntos, hasta que cada cluster o grupo contiene un único individuo, y este sea independiente, entonces es el momento en que se detiene el proceso.
También, esta estrategia a su vez dispone de muchos métodos, dependiendo de la manera de medir las similitudes entre los grupos, como pueden ser por ejemplo el linkage simple, el linkage completo, el promedio entre grupos, el centroide, la mediana y el método de análisis de asociación.

Además, con este método se pueden distinguir dos variantes según la forma de realizar la partición:

(a) *Unidimensional* (también conocida como Monotéticos o *Monothetic*): donde sólo se considera una variable (atributo) para realizar la partición, y normalmente esta clase de métodos se utiliza cuando las variables son de tipo binario.

(b) *Multidimensional* (también llamada Politéticos o *Polythetic*): donde todas las variables se consideran para crear la partición, usando una distancia entre clusters para medirlas. Es decir, que las divisiones se basan centralmente en las observaciones recogidas por todas las variables (atributos) de la base de datos registrada.

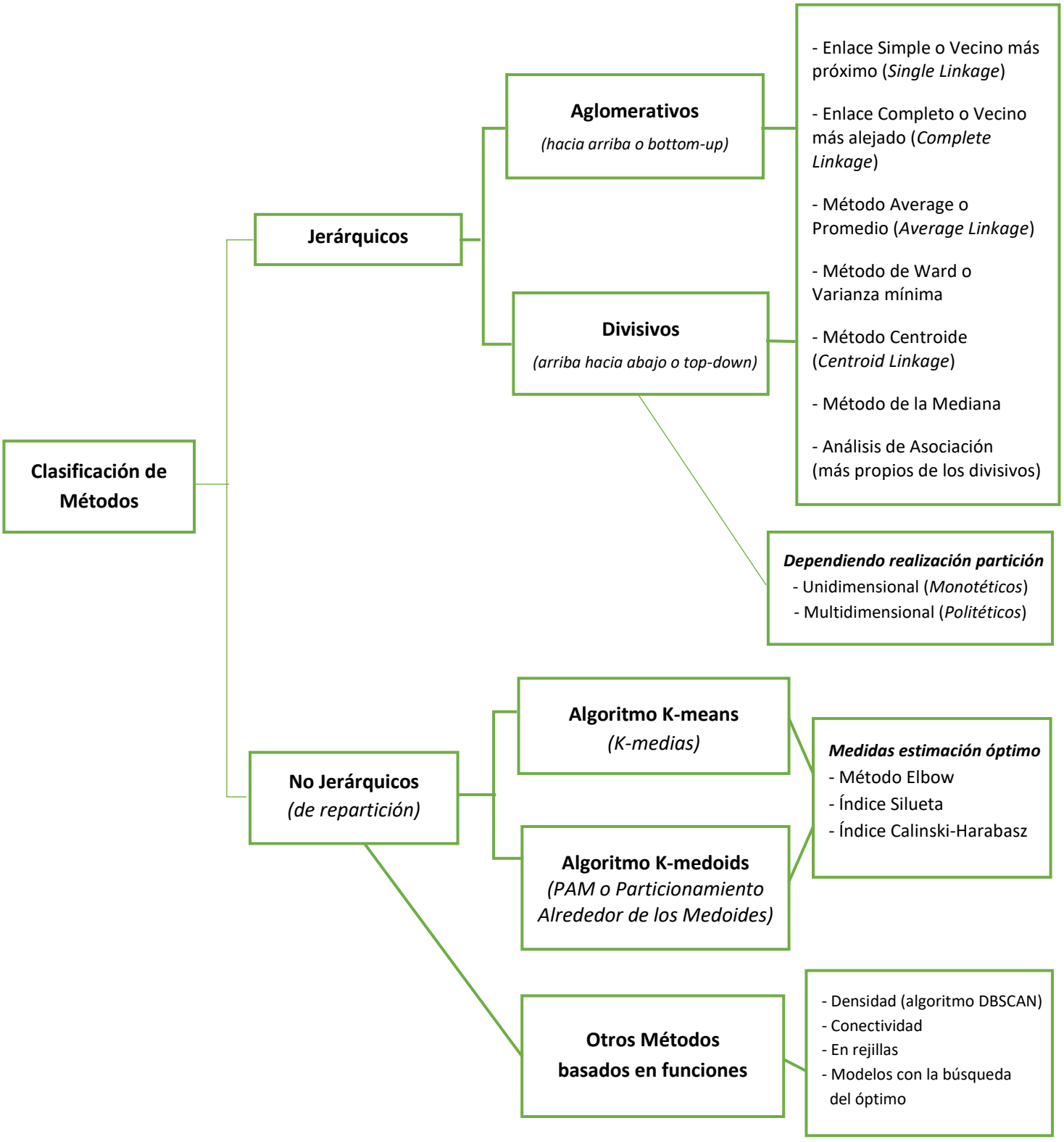
Más concretamente, en la *Tabla 4* se puede ver de forma esquemática todos los métodos mencionados para la clasificación de métodos jerárquicos y no jerárquicos mediante el análisis Cluster.

Como se ha mencionado anteriormente, existen varias aproximaciones al cluster, mediante la aplicación de distintos algoritmos, como pueden ser: (i) los jerárquicos, (ii) los de particionamiento y (iii) otros más, que se basan en funciones de densidad como puede ser el algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) que marca como outliers aquellos puntos que no superan un umbral de densidad establecido (Sancho, 2020); u otros basado en conectividad, en rejillas o en modelos con la búsqueda del óptimo.

En este mismo sentido, se puede destacar la existencia de otros ***métodos basados en particionamiento***, cuyo enfoque principal es construir una partición de la base de datos *"D"* formada por un número *"n"* de individuos en un conjunto de *"K"* clusters. Es decir, dicho de otra manera, si se tiene un valor para "K*"*, se intentar encontrar la partición de "D" en los "K" clusters, optimizando el criterio de particionamiento seleccionado.

A continuación, se describen dos métodos heurísticos bastante utilizados y conocidos por su gran extensión e impacto en el entorno ciencitfico, como son: el algoritmo de las *k-means* y el de las *k-medoids o PAM*, (Amat, 2017c).

**Tabla 4.** Clasificación de métodos jerárquicos y no jerárquicos – Análisis Cluster



(a) ***K-means*** (*K-medias*): es un algoritmo de agrupamiento por particiones, iterativo y relativamente eficiente en el que las instancias de individuos se van moviendo entre clusters hasta que se alcanza el conjunto de clusters deseado, finalizando en un óptimo local, que depende de la elección inicial de los centros (centroides) de clusters, que estos (centroides) normalmente son uno diferente a los elementos del conjunto y donde su valor es el promedio de los elementos del cluster para cada variable.

Es decir, el método consiste en dividir los individuos en K conjuntos disjuntos minimizando la suma de distancias dentro de los individuos de cada grupo, donde cada cluster representa el centro (centroide) del cluster, por eso es necesario indicar el parámetro de entrada con el número de clusters deseado.

Además, se conoce que una de las desventajas del algoritmo es que hay que fijar anticipadamente el número de clusters (*K*), pero esto se puede solucionar con una medida de rendimiento.

Existen varias medidas que permiten estimar el mejor número de grupos (es decir, el número óptimo de clusters), como pueden ser: el *método Elbow*, el *Coeficiente Silhouette o Índice Silueta* que es uno de los más conocidos, y el Índice de *Calinski-Harabasz*.

- *Elbow method* (*método Elbow*): esta basado en la estrategia de encontrar el valor óptimo en un hiperparámetro, probando un rango de valores de este hiperparámetro y representando gráficamente sus resultados con el fin de identificar el punto de corte de la curva donde no exista ninguna mejora o deja de ser sustancial (principio de verosimilitud). En el caso de particionamiento, este método calcula la varianza total intra-cluster en función del número de clusters y toma como óptimo aquel valor en el que al añadir otros clusters no presenta ninguna mejoría.

- *Coeficiente Silhouette* (*Índice Silueta*): que mide la calidad global del agrupamiento, es decir mide cómo de similares son los elementos o individuos de un mismo cluster (cohesión) comparado con otros clusters (separación), donde toma valores en [-1,1], siendo mejor cuanto más cercano a 1 está el valor ya que significa que los clusters resultantes son buenos y tienen mejor separación, por el contrario, si se acerca a cero significa que el individuo está en el borde de dos clusters. Este método es similar al de *Elbow*, pero con la diferencia de que en vez de minimizar el total inter-cluster de la suma de cuadrados, se maximiza la media de los coeficientes Silhouette, donde este coeficiente indica como de buena es la asignación realizada de una observación comparando su similitud con el resto de observaciones de su

cluster frente a las de los otros clusters. Por eso, los valores altos son indicativo de que la observación asignada al cluster es correcta.

- *Calinski-Harabasz* (Índice de *Calinski-Harabasz o Indice CH*), también conocido como *criterio de relación de varianza*. Este método es una razón entre la dispersión intra-clusters y la dispersión inter-clusters. Es decir, es una medida que indica como de similar es un elemento o individuo a su propio grupo (cohesión) en comparación con otros grupos (separación). Por tanto, cuanto mayor es el valor, pues mejor es el agrupamiento. Desafortunadamente, no existe un valor de corte aceptable, aunque se puede elegir esa solución que dé un pico o al menos un cambio abrupto en la gráfica lineal de lo índices CH representados.

(b) ***K-medoids*** (*PAM o Particionamiento Alrededor de los Medoides*): esta relacionado con el método de las K-means puesto que, ambos agrupan las observaciones o individuos en K clusters, siendo el valor K prefijado de antemano. Este método difiere en que cada cluster se representa por su centroide (medoide) que se corresponde con el punto más central del cluster, en vez de las medias (que toma el valor promedio de todas las observaciones de los individuos del cluster), es decir, cada cluster figura por uno de los elementos o individuos incluidos en el cluster.
No obstante, es un método más robusto que K-means, y se ve menos afectado por outliers o ruido, considerándose como la analogía entre media y mediana.
Además, el algoritmo más conocido para aplicar esta técnica del K-medoids se denomina PAM, que minimiza la suma de las diferencias de cada observación o individuo respecto a su medoid. Asimismo, hay que destacar que este método se utiliza cuando se sospecha o se tiene conocimiento de la presencia de outliers (valores atípicos).

Como se ha mencionado previamente, existen diversos métodos para el proceso de separación y fusión de los grupos, donde en este caso particular, se han aplicado distintos métodos: (1) *Linkage Completo*, minimizando distancia máxima, (2) *Linkage Simple*, minimizando distancia mínima, (3) *Average*, calculando la distancia promedio y (4) *Ward*, obteniendo la minima varianza, con el fin de seleccionar el mejor entre ellos.

Para su aplicación, se utilizan datos tipificados en los diferentes métodos para normalizarlos y como medida de distancia la euclídea. Tras la implementación de cada técnica jerárquica, se calculan los coeficientes de correlación cofenéticos entre la matriz de distancias y la matriz cofenética en cada uno de ellos, obteniéndose los correspondientes resultados gráficos a través del dendrograma (*Figura 10*).

Y sus coeficientes de correlación en las salidas del editor, donde los valores mostrados para cada método son de 0.53 para el completo; de 0.76 para el simple; de 0.74 para el promedio y de 0.29 para el Ward, indicando que el mejor método es el del cluster 2 con un coeficiente cofenético de 0.76 correspondiente al linkage simple, aunque se puede observar por la información que existen pocas diferencias entre el cluster 3 (average) con un coeficiente cofenético de 0.74 y el cluster 2.

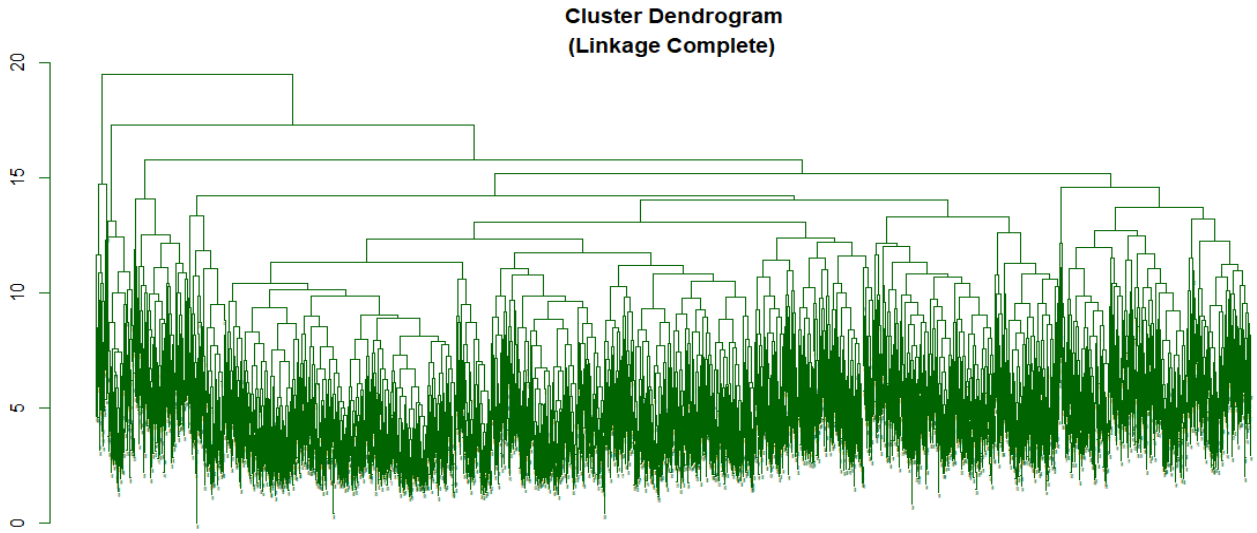



**Figura 10. (a)** Dendrograma de diferentes métodos mediante el Linkage Completo

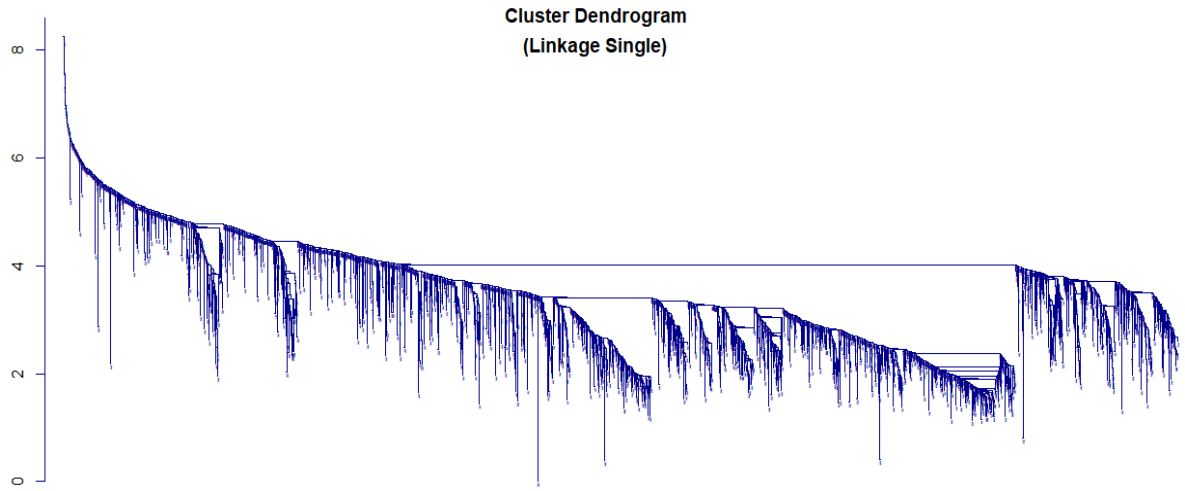



**Figura 10. (b)** Dendrograma de diferentes métodos mediante el Linkage Simple

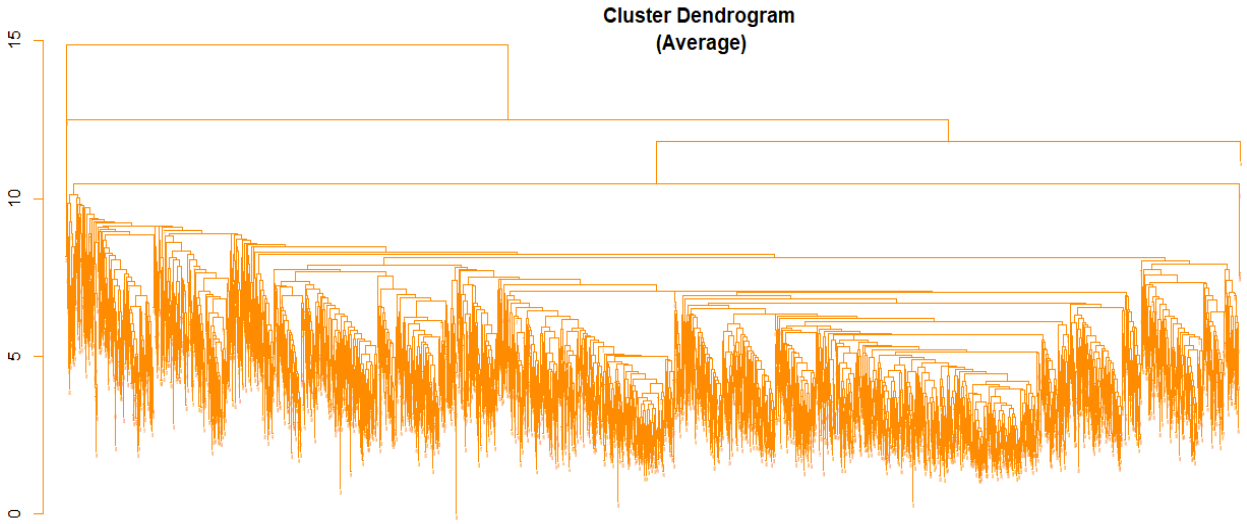



**Figura 10. (c)** Dendrograma de diferentes métodos mediante el Average

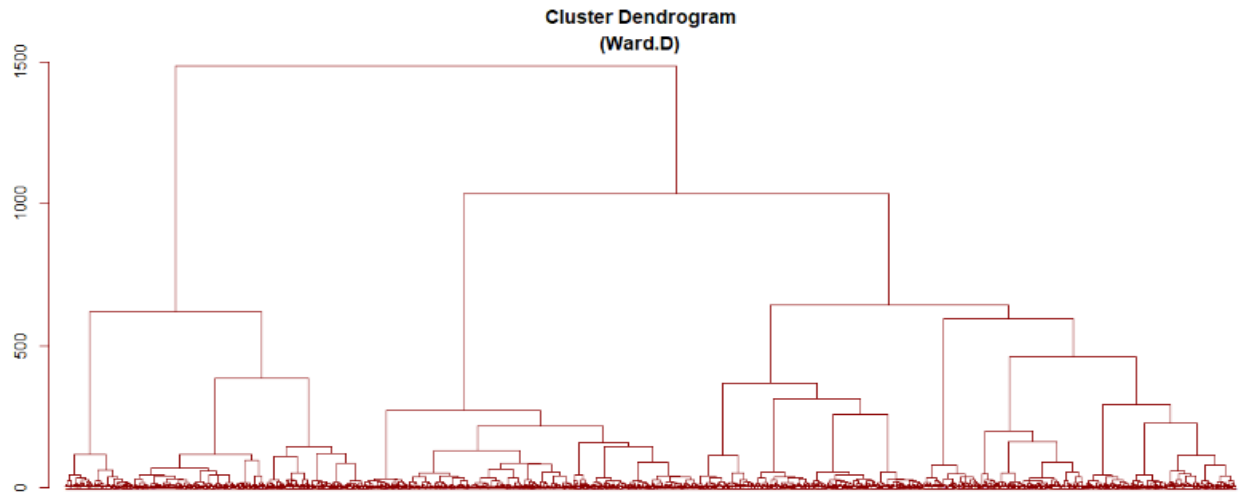



**Figura 10. (d)** Dendrograma de diferentes métodos mediante el Ward

En este mismo sentido, como se pudo visualizar previamente en los resultados obtenidos con el análisis PCA que esta detallado en el capitulo II, se encontró que se observan cuatro grupos de perfiles por afinidad y características para la información de este dataset.

Por ello, se ha explorado ampliamente clasificándolos en diferentes grupos de 5, 4 y 3 obteniendose los siguientes resultados visuales que presentan los diferentes dendrogramas o árboles jerárquicos del clúster (*Figuras 11 y 12*), así como sus mapas factoriales representados a la derecha del dendrograma con la intención de mejorar la percepción visual de los grupos formados por los individuos coloreados de acuerdo con sus perfiles clusters.

A la vista de los dendrogramas presentados (*Figura 11*), se puede apreciar que la clasificación con 4 grupos de clusters, muestra que estos individuos pueden ser agrupados algo más por estar más cerca entre ellos y seguramente son bastnate parecidos internamente, especialmente los clusters 1 y 3 donde se puede generar un nuevo grupo de cluster mixto con la finalidad de reducir al máximo (óptimo) el número de conglomerados finales.

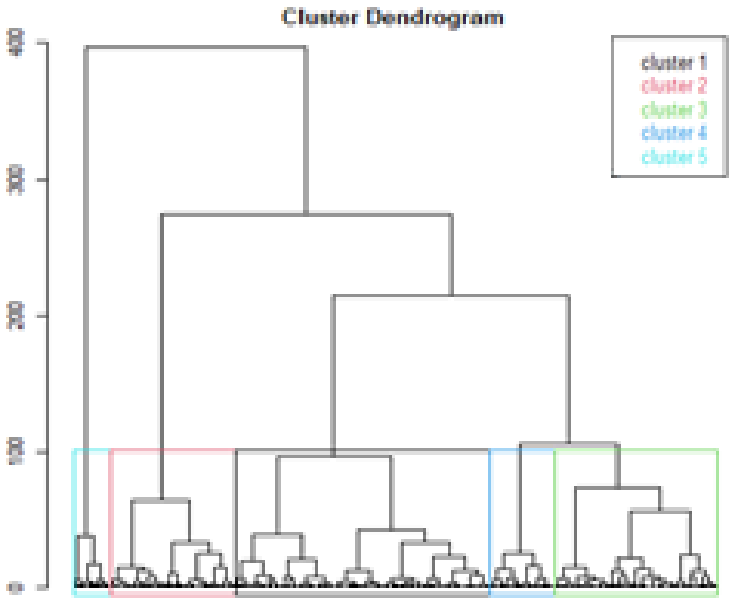



**Figura 11. (a)** Dendrograma o Árbol jerárquico para 5 grupos de clusters

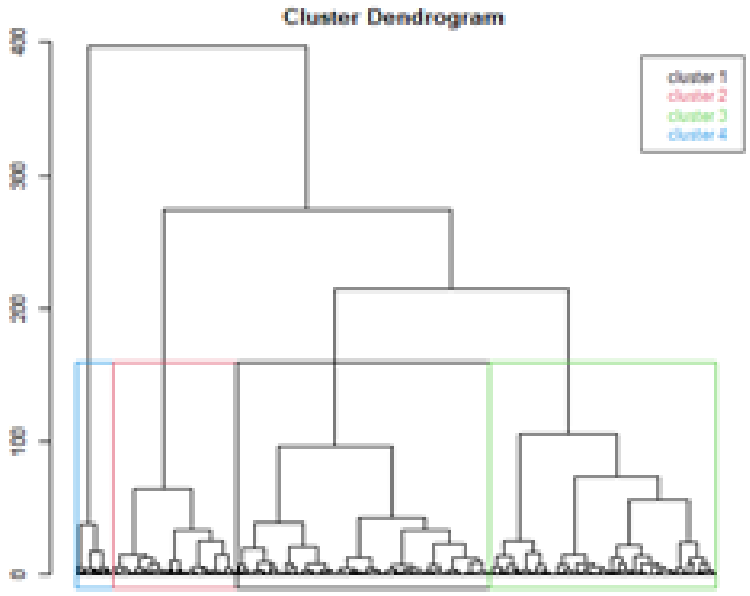



**Figura 11. (b)** Dendrograma o Árbol jerárquico para 4 grupos de clusters

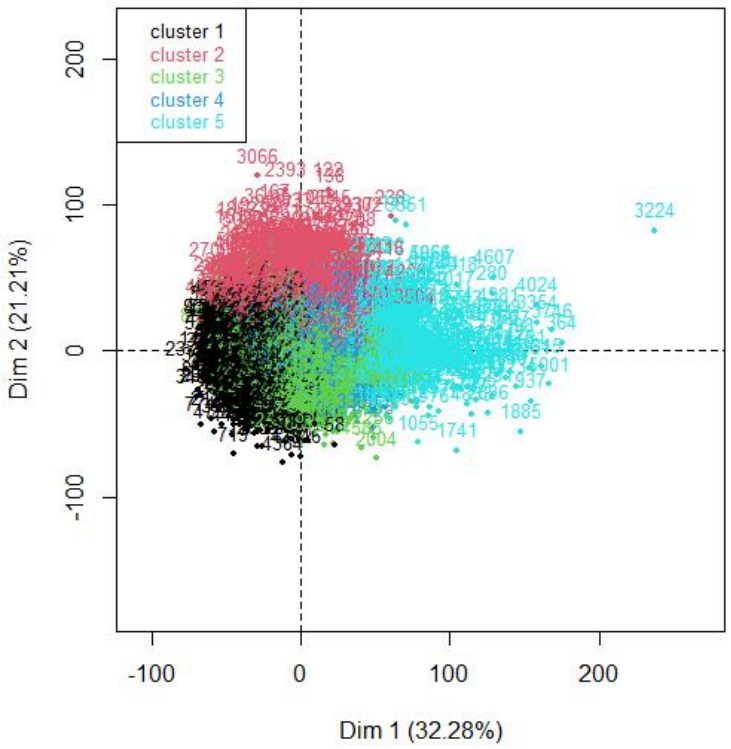



**Figura 11. (c)** Mapa factorial para 5 grupos de clusters

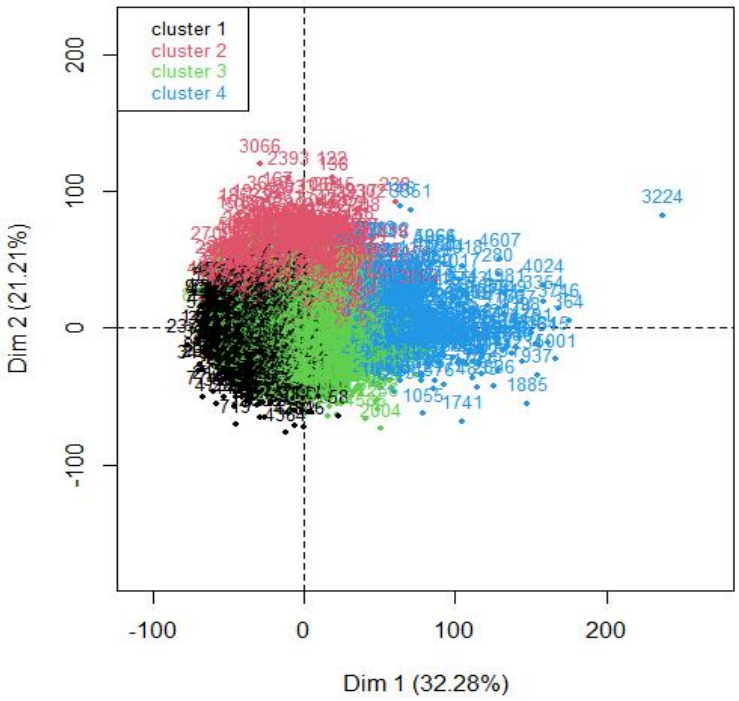



**Figura 11. (d)** Mapa factorial para 4 grupos de clusters

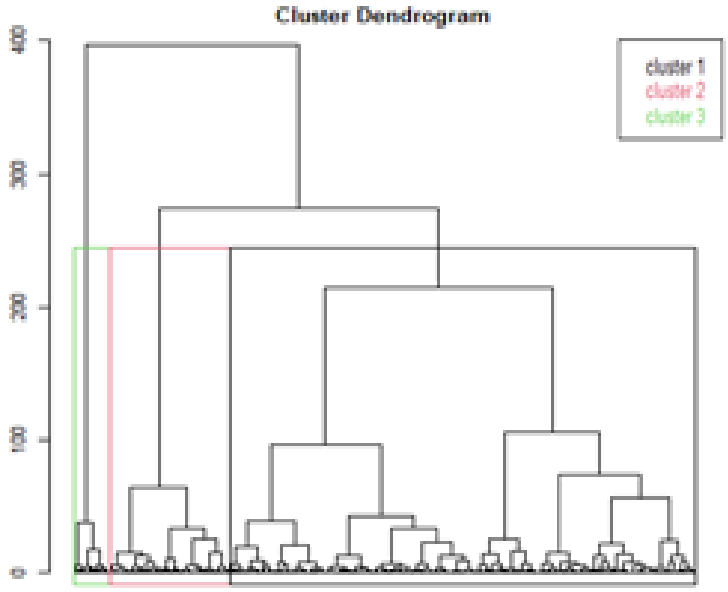



**Figura 12. (a)** Dendrograma o Árbol jerárquico para 3 grupos de clusters

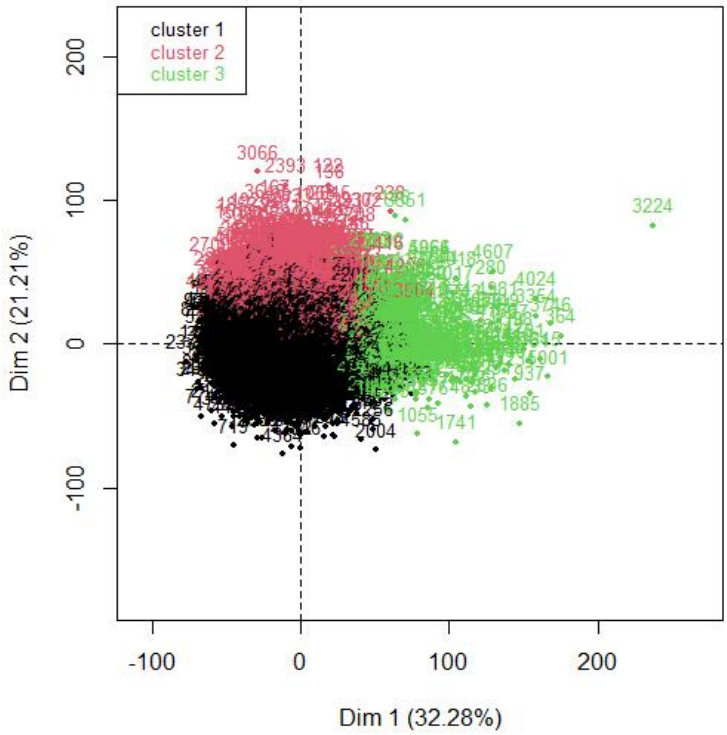



**Figura 12. (b)** Mapa factorial para 3 grupos de clusters

Precisamente, en la *Figura 12* se puede observar el árbol jerárquico formado por los 3 nuevos clusters, que son más homogéneos entre sí dentro del mismo grupo-perfil y más heterogeneos entre ellos, pudiendo interpretar mejor los perfiles finales generados.

Por otro lado, como no se dispone de información adicional en la que basarse para saber exactamente el número óptimo K, (excepto lo extraído del análisis PCA donde estos podrían ser de 4 grupos y el reflejado por el soporte visual del análisis AC con 3 posibles grupos), aun asi, se aplico el algoritmo de K-means para un rango de valores de K (cambiando dicho valor K desde 20, 15, 10 y bajando hasta 5 para métodos diferentes y luego para solo los dos métodos mejores de cluster según coeficiente cofenético, el 2 es Simple y el 3 es Average), tal y como se puede visualizar un ejemplo en la *Figura 13* para intentar identificar aquel valor a partir del cual la reducción en la suma total de varianza intra-cluster deja de ser sustancial.

Precisamente, este punto de corte significativo mediante este mecanismo se le conoce como estrategia del codo o método Elbow, que con la utilización de la función llamada “fviz_nbclust” del paquete *factoextra* de R (Kassambara y Mundt, 2020) se puede automatizar este proceso y se genera una

representación de los resultados que ayudan a detectar el punto de inflexión del cambio para seleccionar el número de cluster más efectivo.

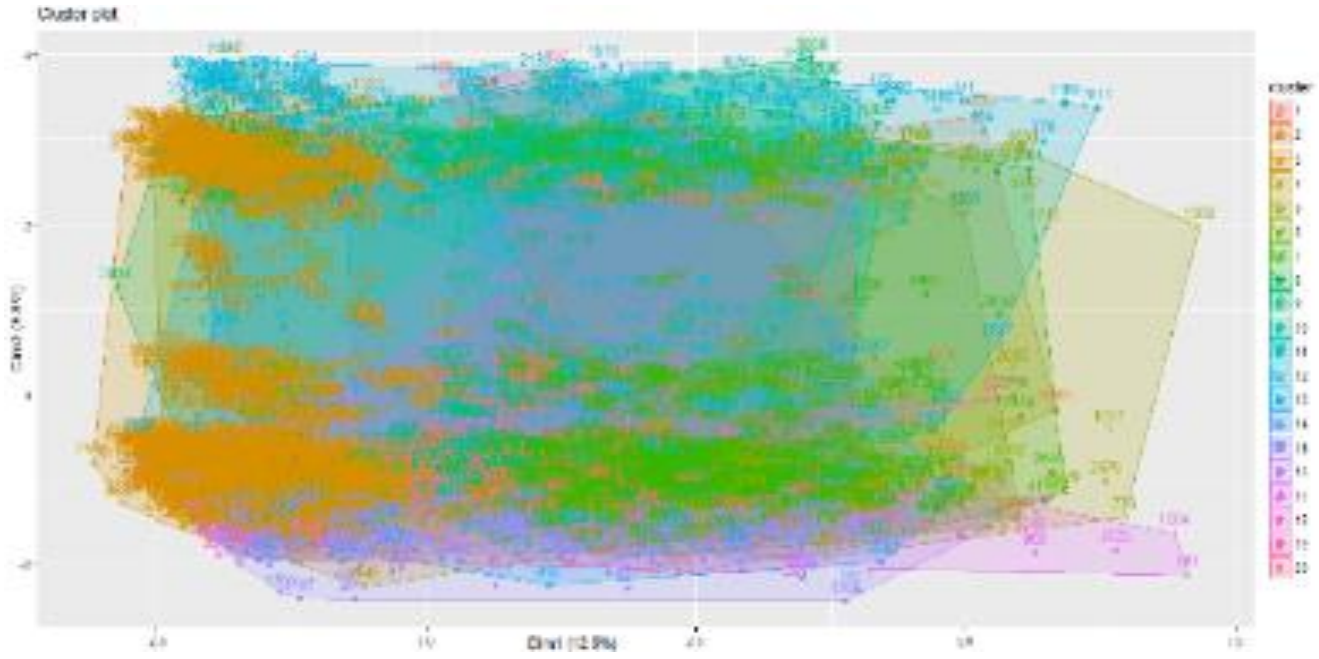


**Figura 13. (a)** Procedimiento K-means variando valor K para método seleccionado (K=20). Completo

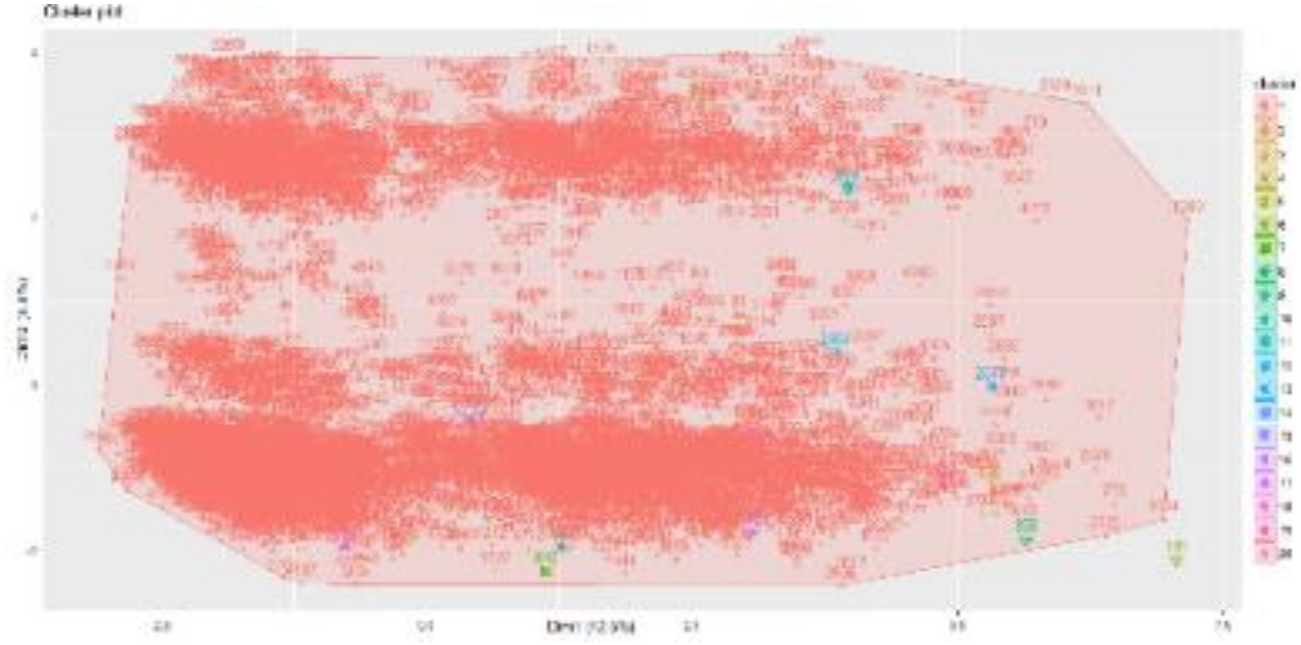


**Figura 13. (b)** Procedimiento K-means variando valor K para métodos seleccionados (K=20). Simple

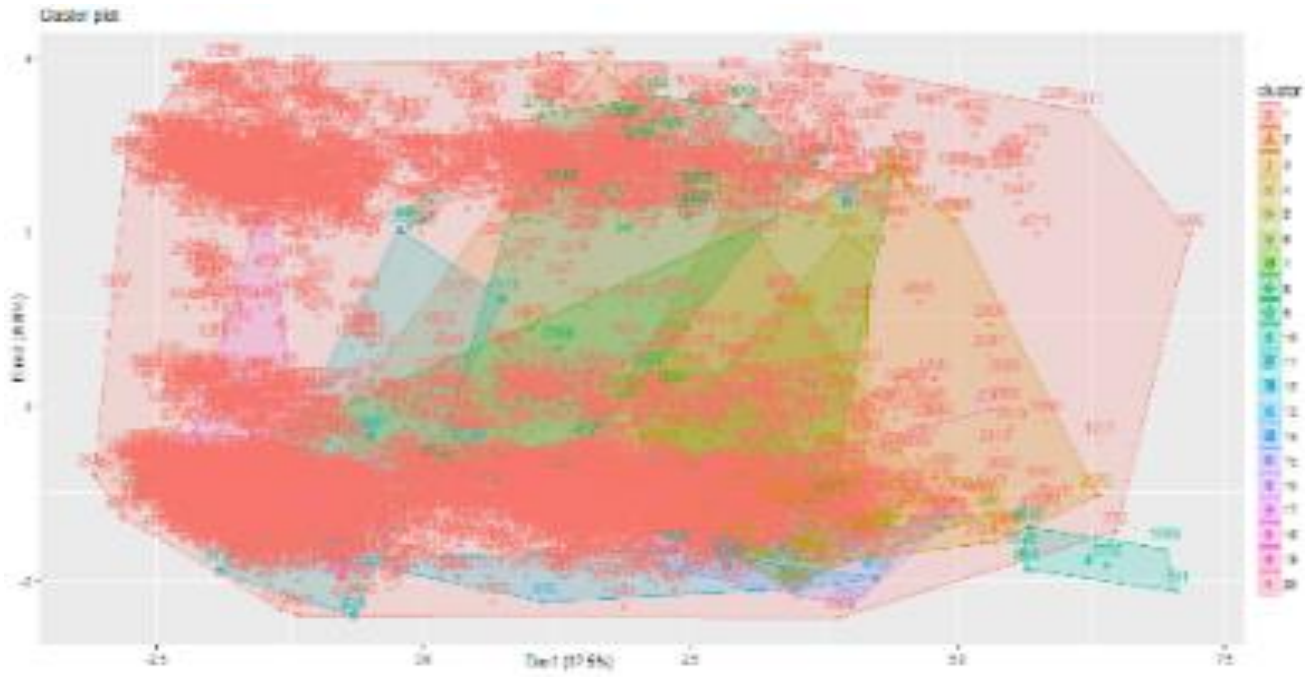


**Figura 13. (c)** Procedimiento K-means variando valor K para métodos seleccionados (K=20). Average

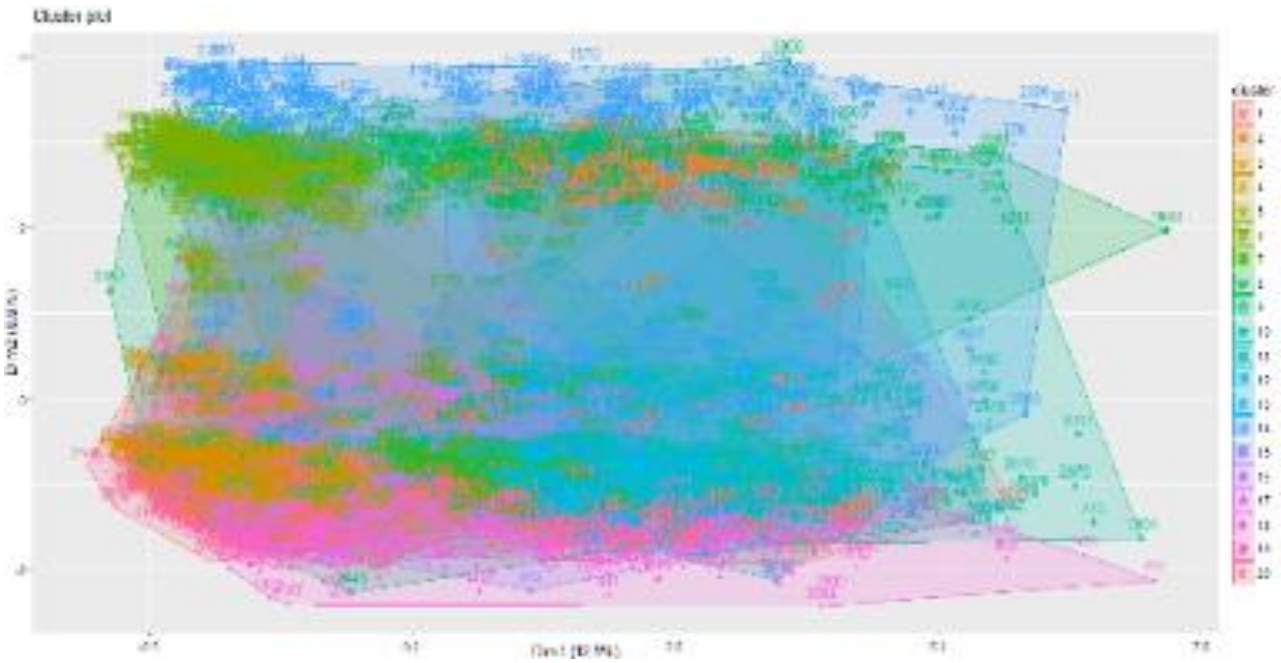


**Figura 13. (d)** Procedimiento K-means variando valor K para métodos seleccionados (K=20). Ward

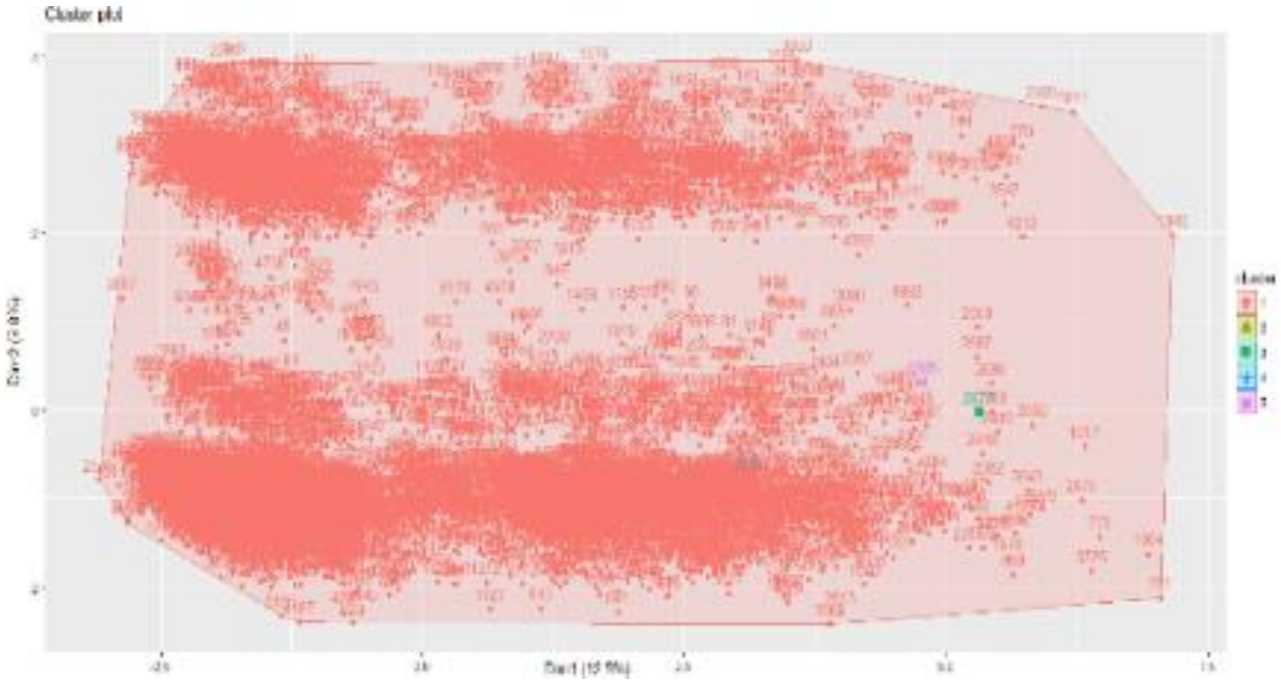


**Figura 13. (e)** Procedimiento K-means variando valor K para método final (K=5). Simple

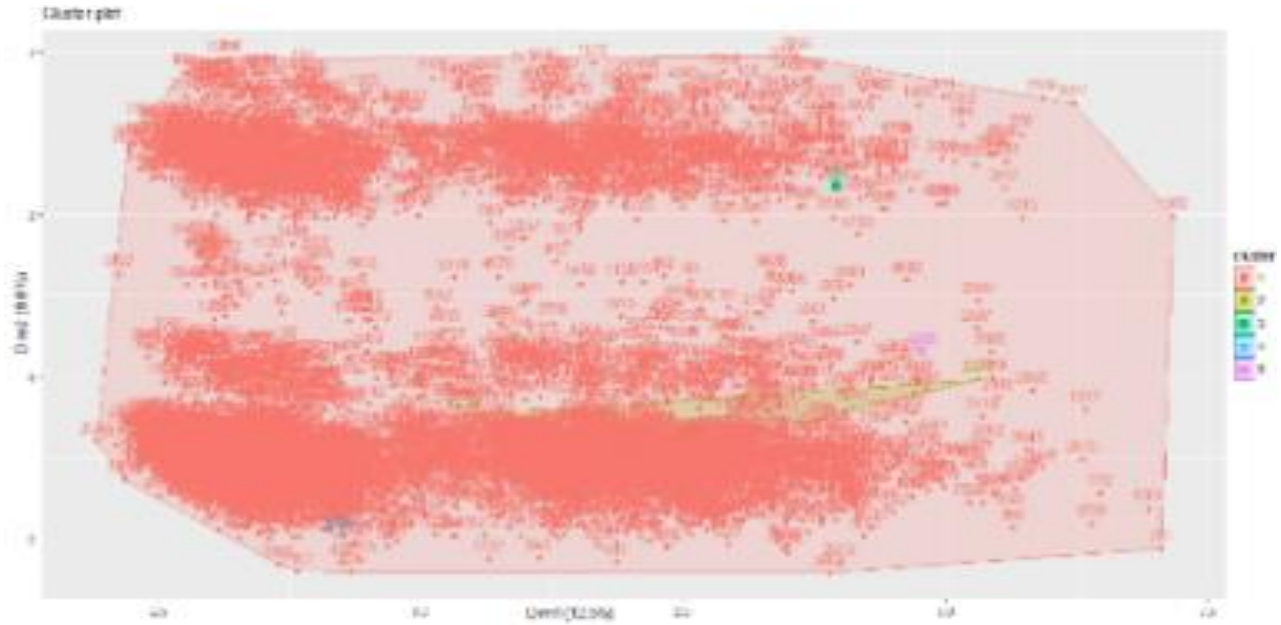


**Figura 13. (f)** Procedimiento K-means variando valor K para método final (K=5). Average

Por lo tanto, según los resultados visuales del K-means y los coeficientes de correlación anteriores, se ha detectado que el mejor método viene dado por el linkage simple (definido por cluster 2) de distancia mínima con un 76.5% y compuesto por la creación de 4 grupos por lo que se puede intuir que es posible generar una clasificación con un número óptimo que sea menor de 4 grupos.

En este mismo sentido, el análisis final parecía que revelaba que el número óptimo puede estar entre el rango de 5 y 3 conglomerados de grupos, por lo que se podría optimizar el número de clusters a 3 grupos-perfiles. Por este motivo, se ha aplicado el método *Elbow* para averiguar este punto sustancial de cambio, que minimiza el total intra-cluster de la suma de cuadrados, y por los resultados obtenidos en la curva se puede apreciar que a partir de 5, 4 y 3 clusters la mejora es mínima como se muestra visulamente en la *Figura 14*.

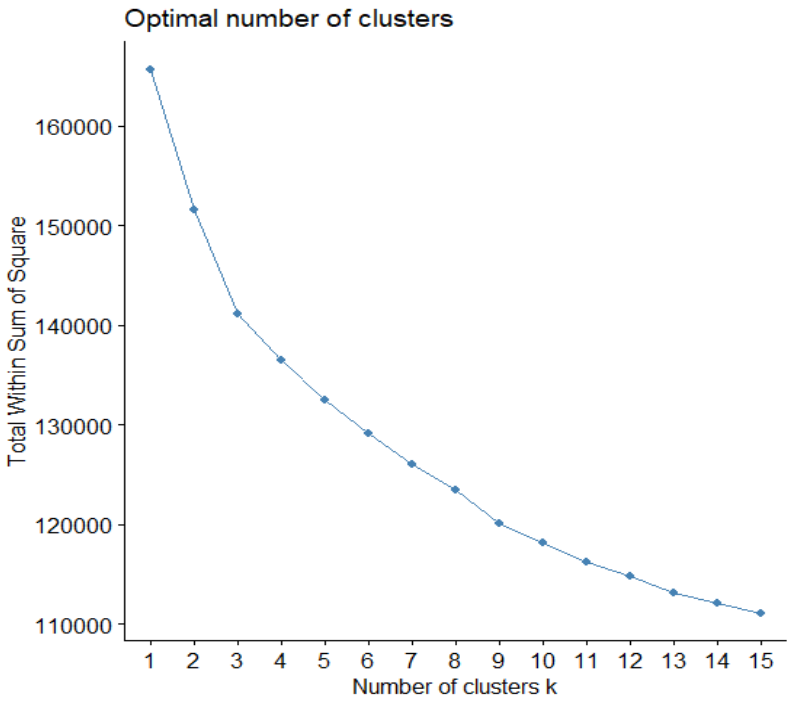



**Figura 14.** Método Elbow – Clusters posibles de 5 a 3 grupos según total intra-cluster de suma cuadrados

Asimismo, a la vista de estos datos relevantes y signficativos, se aplica un segundo método llamado *Silhouette*, para verificar esto último indicado, que es una técnica muy similar al de *Elbow*, con la diferencia de que se maximiza la media de los coeficientes o índices silhouette de todas las observaciones de individuos (*Figura 15*). Dicho esto, los resultados obtenidos de la silueta de la curva

señala como número definitivo 3 clusters óptimos como mejor separación de agrupamiento de los datos, reafirmando un poco más lo que ya se visualizaba en las salidas anteriores.

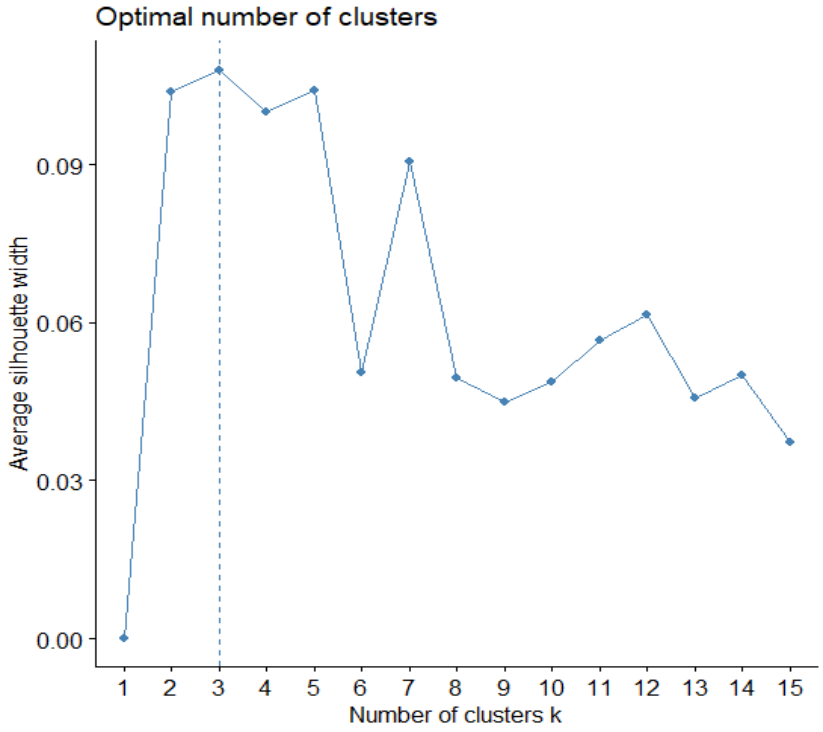



**Figura 15.** Método Silhouette – Cluster óptimo de 3 grupos según máximo valor medio de los índices

Visto este último resultado confirmatorio de tres grupos de perfiles optimo, si se recuerda que anteriormente cuando se estuvo explorando con el análisis PCA se detecto que el Análisis Factorial (AF o FA) del gráfico de sedimentación mediante PCA (*Figura 5*) del capítulo II aparte de las 12 componentes principales obtenidas, además se indicaba 3 factores relevantes por encima de la línea de corte. Dicho esto, también el AF se define como una técnica estadística de reducción de datos usada para explicar las correlaciones entre las variables observadas en términos de un número menor de variables no observadas llamadas factores. Las variables observadas se modelan como combinaciones lineales de factores más las expresiones de error.

Finalmente, tras explorar también el análisis CA (Correspondencia) que más adelante se detalla, se ha determinado visualizar los 3 clusters de salida por los resultados obtenidos anteriormente y sus correlaciones altas y bajas presentadas mediante los diferentes clusters óptimos generados frente a las variables, que describen las características más relevantes de los individuos (*Tabla 5*).

**Tabla 5.** Descripción correlacional de los Clusters finales óptimo vs Variables

| Variables | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| CCVSDM_ | 0.734 | **-2.12** | **2.22** |
| DURING | **2.34** | **-2.45** | -0.203 |
| EDAD | -1.48 | -1.41 | **5.15** |
| ESPIROMETRIA_PA_ | **5.04** | **2.51** | **-13.6** |
| EV_ | 0.227 | -1.74 | **2.52** |
| FCA | **-12.8** | **15.5** | **-2.54** |
| FEV1P | **-11.9** | -0.992 | **23.9** |
| FEVCVF_ | **-24.3** | -1.03 | **46.9** |
| FRE | -1.28 | **2.18** | -1.31 |
| FVCP | **9.46** | -0.63 | **-16.5** |
| HT_ | **2.96** | -0.112 | **-5.32** |
| ICHARECV_ | -0.334 | **-2.35** | **4.6** |
| ICHARICC_ | -0.114 | -1.38 | **2.54** |
| IMC | 1.84 | **-3** | 1.66 |
| MUERTOS_90DIAS | 1.69 | **-2.37** | 0.873 |
| PPKG | **2.51** | **-3.77** | 1.72 |
| SV_ | -0.936 | **2.2** | **-1.98** |
| TARD | **-26.7** | **30.8** | **-2.62** |
| TARS | **-40.3** | **44.4** | -0.136 |

Por tanto, se describe la clasificación formada por los 3 grupos perfiles óptimos.

- ✓ **Cluster 1** formado por individuos que comparten valores altos en las variables (ordenas desde las más fuertes): FVC espirometría en % del teórico *(FVCP),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Hábito Tabáquico *(HT_),* Peso *(PPKG)* y Duración del Ingreso hospitalario *(DURING)*; y valores bajos en estas otras (por orden de más débil): Tensión Arterial Sistólica (*TARS*), Tensión Arterial Diastólica (*TARD*), Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* Frecuencia Cardíaca *(FCA)* y FEV1 espirometría en % del teórico *(FEV1P)*.

- ✓ **Cluster 2** formado por individuos que comparten valores altos en las variables: Tensión Arterial Sistólica *(TARS),* Tensión Arterial Diastólica *(TARD),* Frecuencia Cardíaca *(FCA),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Soporte Ventilatorio *(SV_)* y Frecuencia Respiratoria *(FRE)*; y valores bajos en estas otras: Peso *(PPKG),* Índice de Masa Corporal *(IMC),* Duración del Ingreso hospitalario *(DURING),* Muertos a 90 días *(MUERTOS_90DIAS),* Enfermedad Cerebro Vascular *(ICHARECV_)* y Comorbilidad Cardiovascular *(CCVSDM_).*

- ✓ **Cluster 3** formado por individuos que comparten valores altos en las variables: Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FEV1 espirometría en % del teórico *(FEV1P),* EDAD, Enfermedad Cerebro Vascular *(ICHARECV_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Enfermedad Vascular *(EV_)* y Comorbilidad Cardiovascular *(CCVSDM_)*; y valores bajos en estas otras: FVC espirometría en % del teórico *(FVCP),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Hábito Tabáquico *(HT_),* Tensión Arterial Diastólica *(TARD),* Frecuencia Cardíaca *(FCA)* y Soporte Ventilatorio *(SV_).*

### *III. 3. 2. Análisis de Correspondencias*

En este caso, el Análisis de Correspondencias (*Correspondence Analysis*, CA) del paquete central de R "*ca*" (Nenadic y Greenacre, 2007) y se completo también con los paquetes *ade4, FactoMineR y factoextra* (Dray et al., 2007; Lê et al., 2008; Bougeard y Dray, 2018; Thioulouse et al., 2018 y Kassambara y Mundt, 2020) se ha uttilizado como parte adicional del análisis principal (Cluster) para completar el resultado de clasificación por agrupamiento, ya que es una técnica multivariante bastante descriptiva visualmente, que tiene como enfoque central representar la relación de las variables, mediante una medida de asociación, en un espacio de baja dimensión con la menor pérdida de información posible, (De La Fuente, 2011 y Rangel et al., 2020).

Es decir, es un método de reducción de dimensiones, equivalente a los análisis de PCA, Factorial (AF) o Discriminante (AD) para variables categóricas, nominales u ordinales, ayudando a visualizar la nube de puntos multidimensional en dos dimensiones, o sea, dicho de otra manera más sencilla, este método es básicamente un procedimiento de transformación de una nube de puntos definida en un espacio de alta dimensión a otro menor, concretamente al de dos dimensiones para ser interpretado fácilmente, pudiéndose visualizar la posición de los puntos u observaciones según las características de las variables exploradas, respetando al máximo sus posiciones en la nube de puntos original.

Para ello, existen dos tipos de aplicación según el número de variables relacionadas que se quiera representar en el espacio dimensionl exploratoriamente:

- *Análisis de Correspondencia Simple* (ACS): permite identificar las dimensiones básicas dependiendo del número de categorías de cada variable, para la combinación de modalidades o niveles de dos variables cualitativas o categóricas. Es decir, analiza y detecta desde un punto de vista gráfico, las relaciones de dependencia e independencia de un conjunto de variables categóricas a partir de los datos de una tabla de contingencia o de doble entrada, asociando para cada una de las modalidades o factores de la tabla un punto en el espacio dimensional de forma que las relaciones de distancia (minima o máxima) entre los puntos reflejan las interacciones de asociación entre ellas.

  Además, este análisis utiliza una distancia euclídea ponderada como medida de asociación denominada chi-cuadrado, donde esta medida (chi-cuadrado) tiene por condición ponderar a cada perfil con un peso, lo que significa que cada fila o columna es representa por un peso proporcional a su importancia en el conjunto, entendiéndose el término importancia por su frecuencia para evitar dar mayor peso a las categorías de menor relevancia dentro del conjunto. Y en esta misma línea, el método de normalización usado es el simétrico, con la finalidad de que la inercia se reparta de igual manera entre filas y columnas, pudiéndose detectar las diferencias entre las categorías de las variables.

- *Análisis de Correspondencia Múltiple* (ACM): es una extensión de la técnica ACS para un conjunto grande de variables categóricas, es decir, cuando se desea explorar un número superior a dos atributos o variables. Por eso, a este método se le conoce como ACM, permite analizar en conjunto las tablas de contigencia multidimensionales como es el caso de este estudio presentado, donde se aplica el algoritmo CA a la matriz de individuos-variables, también conocida como matriz indicadora, mostrando a los individuos en las filas y a las categorías de las variables en las columnas.

  Dicho de otra manera, este procedimiento se basa en realizar un análisis de correspondencia sobre la tabla de contingencia múltiple o la denominada matriz de Burt, que es una matriz simétrica, que contiene todas las tabulaciones cruzadas para cada par de variables categóricas. Esta matriz o tabla es el análogo a la matriz de covarianzas para el caso de variables continuas.

  Asimismo, el proceso consiste en construir superposiciones de pequeños bloques, donde en las zonas transversales (diagonales) se encuentran las matrices diagonales que muestran las frecuencias marginales de cada una de las variables exploradas, y en la zona superior o por encima

de la diagonal, se encuentran las frecuencias cruzadas correspondientes a todas las combinaciones pares de las variables analizadas.

Además, esta técnica realiza representaciones gráficas similares a las del ACS, en las que cada categoría de las variables o la de los individuos (o pacientes), se representan por puntos.

Como se ha mencionado anteriormente y similar a las conclusiones obtenidas con el PCA, el análisis CA también indica que la inercia de las primeras dimensiones muestra si existen fuertes relaciones entre las variables y sugiere el número de dimensiones que deben estudiarse, por lo que es un soporte adicional a nuestra exploración.

**Tabla 6.** Descomposición de la inercia total con CA – Eigenvalues

| Eigenvalues | Dim.1 | … | … | … | … | … | … | … | Dim.31 |
|---|---|---|---|---|---|---|---|---|---|
| | **Dim.1** | **Dim.2** | **Dim.3** | **Dim.4** | **Dim.5** | **Dim.6** | **Dim.7** | **Dim.8** | **Dim.9** |
| Variance | 0.018 | 0.010 | 0.008 | 0.006 | 0.005 | 0.004 | 0.003 | 0.003 | 0.002 |
| % of var. | **24.132** | 14.053 | 11.038 | 8.358 | 7.037 | 4.832 | 3.713 | 3.485 | 3.145 |
| Cumulative % of var. | 24.132 | **38.185** | 49.223 | 57.581 | 64.618 | 69.449 | 73.162 | **76.647** | 79.792 |
| | **Dim.10** | **Dim.11** | **Dim.12** | **Dim.13** | **Dim.14** | **Dim.15** | **Dim.16** | **Dim.17** | **Dim.18** |
| Variance | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| % of var. | 2.415 | 1.896 | 1.723 | 1.669 | 1.608 | 1.520 | 1.491 | 1.370 | 1.307 |
| Cumulative % of var. | 82.206 | 84.103 | 85.826 | 87.495 | 89.104 | 90.624 | 92.115 | 93.485 | 94.792 |
| | **Dim.19** | **Dim.20** | **Dim.21** | **Dim.22** | **Dim.23** | **Dim.24** | **Dim.25** | **Dim.26** | **Dim.27** |
| Variance | 0.001 | 0.001 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| % of var. | 1.204 | 1.040 | 0.813 | 0.621 | 0.331 | 0.290 | 0.273 | 0.239 | 0.211 |
| Cumulative % of var. | 95.996 | 97.036 | 97.849 | 98.470 | 98.801 | 99.091 | 99.364 | 99.603 | 99.814 |
| | **Dim.28** | **Dim.29** | **Dim.30** | **Dim.31** | | | | | |
| Variance | 0.000 | 0.000 | 0.000 | 0.000 | | | | | |
| % of var. | 0.084 | 0.073 | 0.027 | 0.002 | | | | | |
| Cumulative % of var. | 99.898 | 99.971 | 99.998 | 100.000 | | | | | |

A la vista de los resultados, las dos primeras dimensiones expresan el 38,19% (*Tabla 6*) de la inercia total del conjunto de datos; eso significa que l variabilidad total de la nube de las filas (o columnas) se

explica por el plano 1:2. Obviamente, este es un porcentaje intermedio, pero significativo para estos datos, y el primer plano representa una parte de la variabilidad de los datos, que es mucho mayor que el valor de referencia (7,45% que equivale al percentil 95 de la distribución de porcentajes de inercia obtenida simulando 101 tablas de datos de tamaño equivalente sobre la base de una distribución uniforme), por lo que la variabilidad explicada por este plano es muy significativa.

En la misma línea del análisis PCA, si fuese necesario para obtener mejores resultados podría ser interesante a partir de estas observaciones, considerer las siguientes dimensiones que también expresan un alto porcentaje de la inercia total, interpretando las dimensiones mayores o iguales a la tercera para completar la información.

Por lo tanto, la aplicación del CA indica que una estimación del número correcto de ejes a interpretar sugiere restringir el análisis a la descripción de los primeros 8 ejes, puesto que son los que llevan la gran mayoria de la información real y estos ejes presentan una cantidad de inercia del 76,65% mayor que las obtenidas por el percentil 95 de distribuciones aleatorias (28,61%), por lo que la descripción relevante se situará en estos ejes, siendo los dos primeros, los que más contribuyen a la inercia total (*Figura 16*), donde el eje primero lo hace con 24,13% algo más que el segundo que contribuye con el 14,05%.

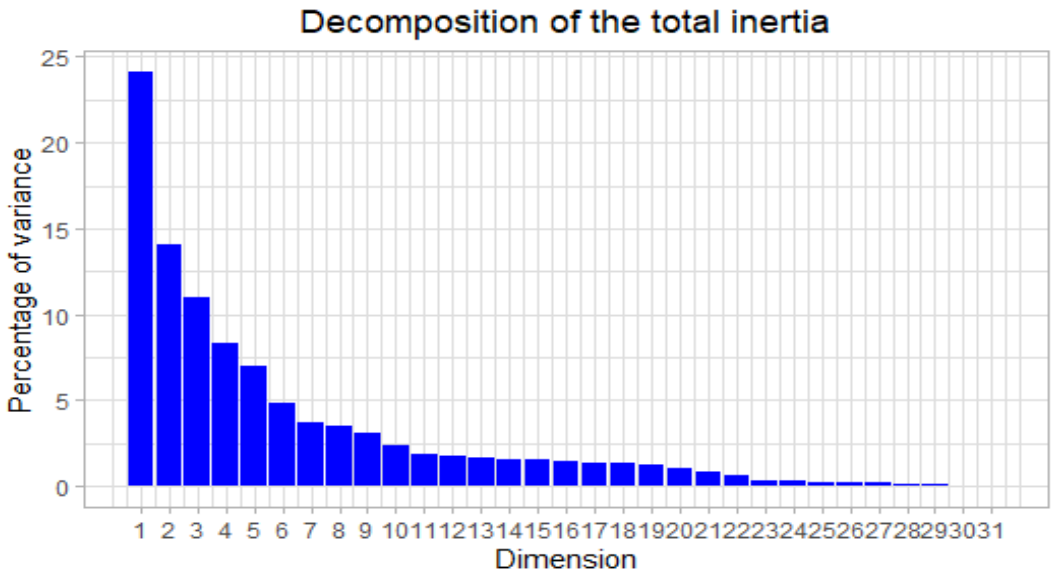



**Figura 16.** Descomposición de la inercia total en porcentaje de la varianza explicada mediante CA

Asimismo, también para confirmar la salida anterior se puede visualizar este corte de ejes, mediante el gráfico de sedimentación, como otra ayuda visual sobre los ejes finales correctos, donde se muestra la cantidad óptima de componentes a tomar en los datos, siendo los valores por encima del punto 1.0 los más aceptables (*Figura 17*), por lo que se verifica que siguen siendo las 8 componentes anteriores (autovalores mayores que uno) los que disponen de la gran mayoria de la información válida. En paralelo, se puede observar que la línea discontinua de color rojo nos indica cuál sería la contribución (en términos de porcentaje de variabilidad explicada en cada dimensión) si fueran homogéneas.

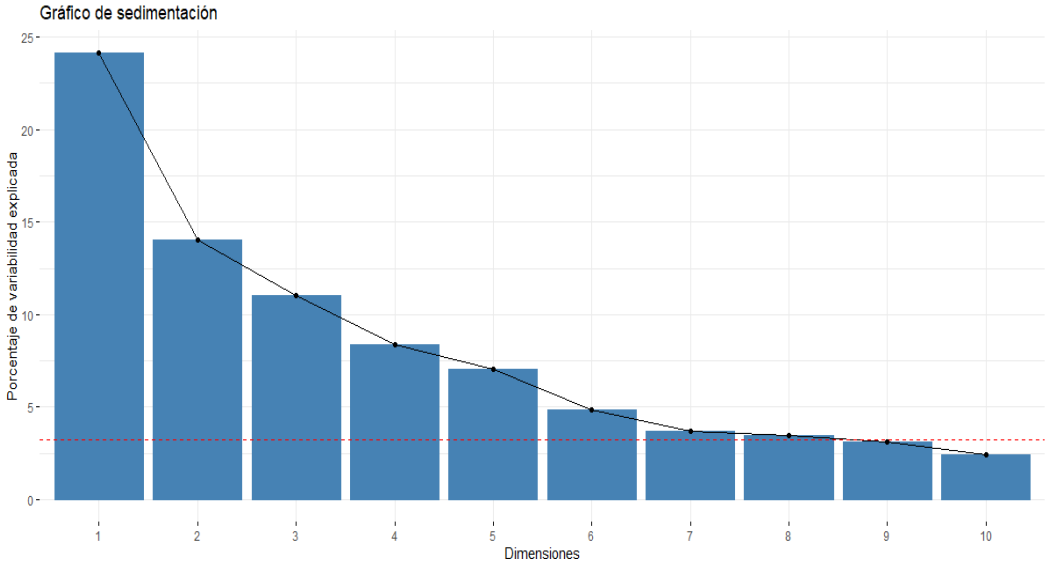



**Figura 17.** Gráfico de Sedimentación mediante CA

Además, se puede observar visualmente la descripción de la contribución dimensional en el plano 1:2 para las diferentes variables (columnas) por orden de importancia en el eje explorado (*Figura 18*).

A la vista de la representación, se aprecia que existen tres variables relevantes con alto porcentaje de contribución al eje, y de igual manera, existen otras dos variables más (justo en el corte de la línea discontinua) con un porcentaje menor de contribución al plano, pero que también son bastante importantes y a tener en cuenta para el análisis de este eje (1-2), y así como para el resto de variables con menor peso pero, con la misma relevancia como para completar la descripción del plano.

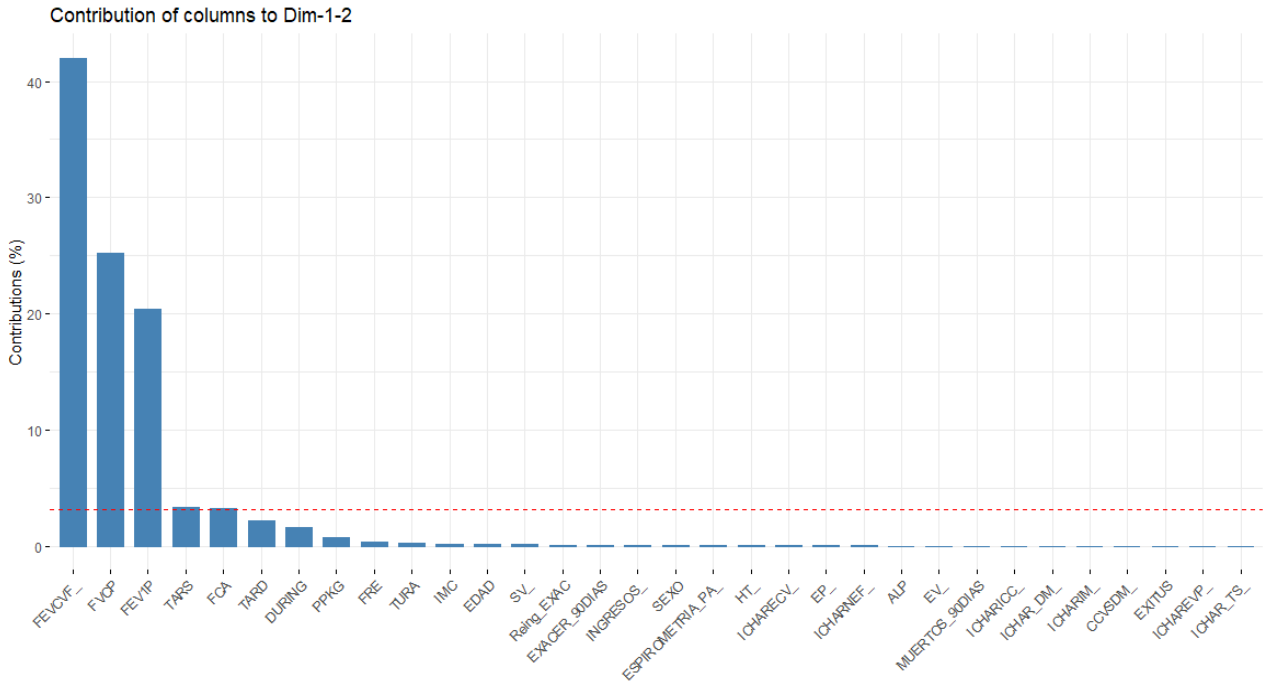



**Figura 18.** Descripción de la contribución de cada variable (columnas) en el plano 1:2 con CA

A continuación, también se puede visualizar la matriz de correlaciones entre variables (*fuerte-débil)* asegurando que estas sean altas como uno de los requisitos fundamentales, mediante la representación dimensional (*Figura 19*) de la contribución de cada atributo-variable en cada eje individual, especialmente el plano 1:2 interpretado y destacando la dimensión 3 y 5 por su importancia, aunque con una alta complejidad para su descripción.

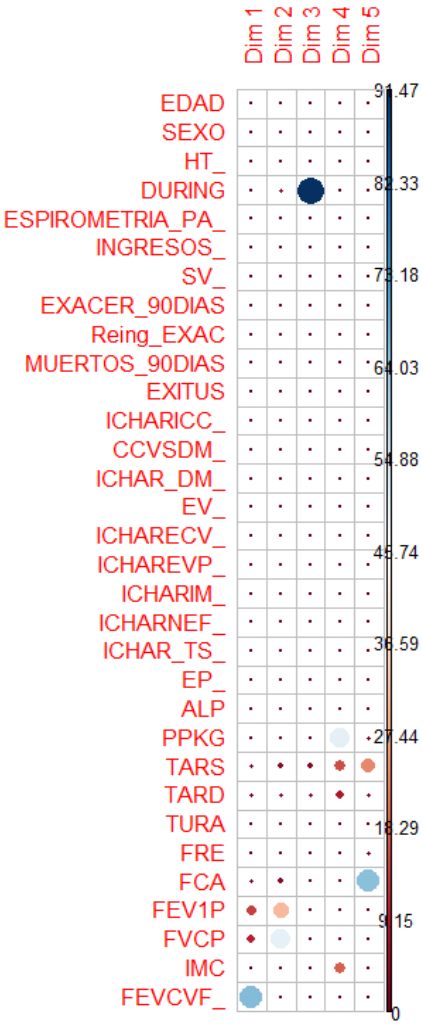



**Figura 19.** Representación de la contribución por variables en cada dimensión por separado con CA

En esta línea y con los resultados obtenidos por los diferentes métodos aplicados, se ha llegado a la conclusión de que el número óptimo de grupos se encuentra en tres perfiles, lo que viene asegurando las salidas del análisis cluster anteriormente realizado.

Dicho esto, se visualiza en detalle los resultados del plano 1:2 y las representaciones visuales por separado (*Figura 20*) para pacientes (*rows*) y variables (*cols*) detectando en cada una de ellas los tres grupos generados por características similares dentro del mismo perfil (pacientes o variables) mediante el grado de la escala de color implementada.

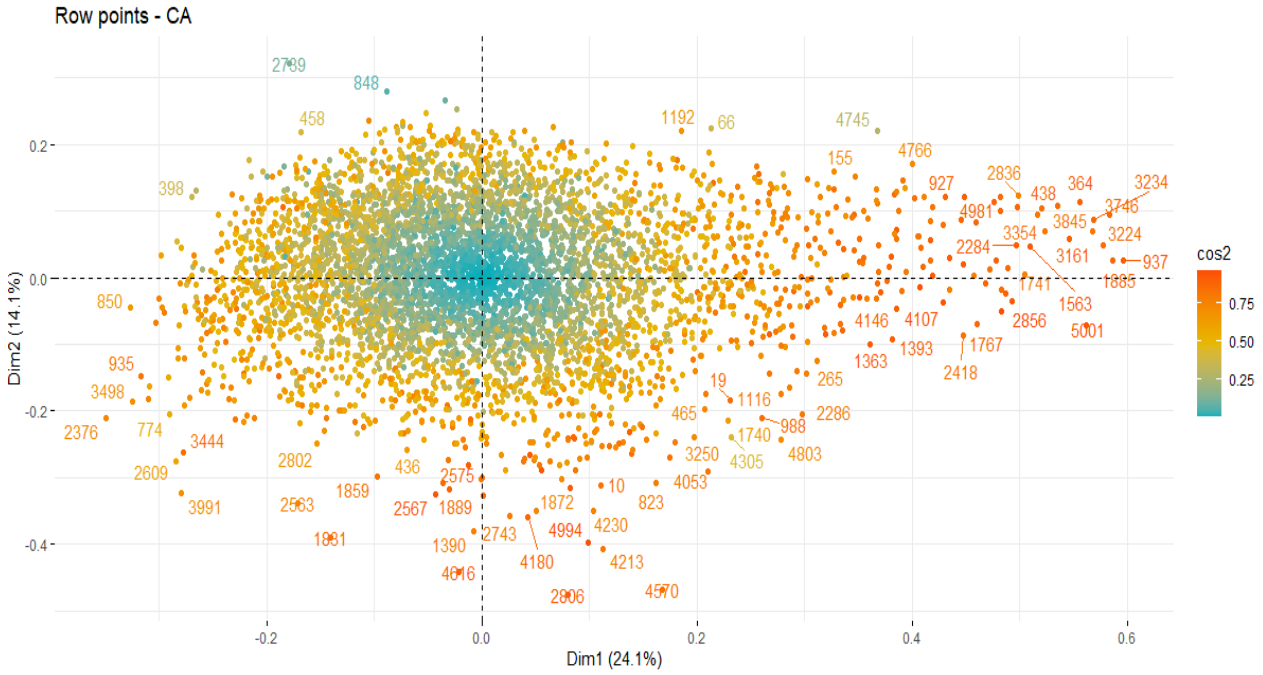



**Figura 20. (a)** Descripción del plano 1: 2 (Pacientes) por grupos separados mediante CA

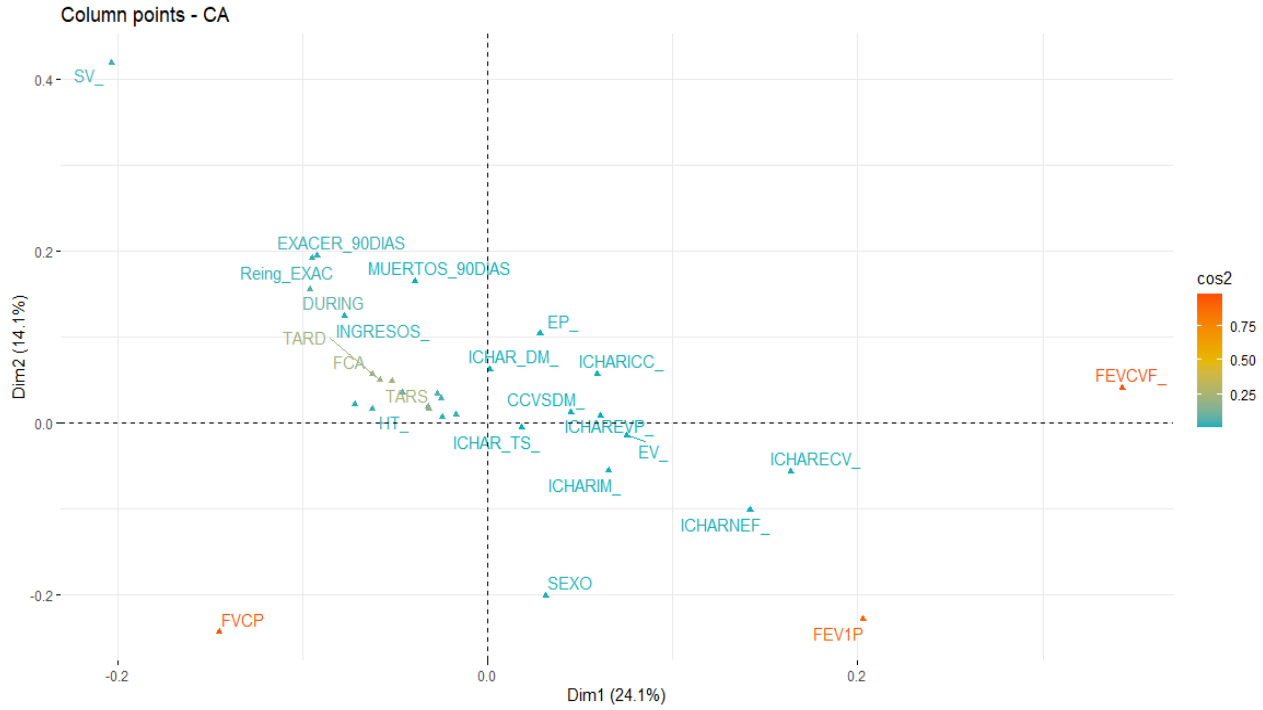



**Figura 20. (b)** Descripción del plano 1: 2 (Variables) por grupos separados mediante CA

Asimismo, también se puede visualizar en conjunto (*Figura 21*) a pacientes y variables para una mayor exploración descriptiva, ya que este procedimiento es un soporte adicional al análisis principal (Cluster).

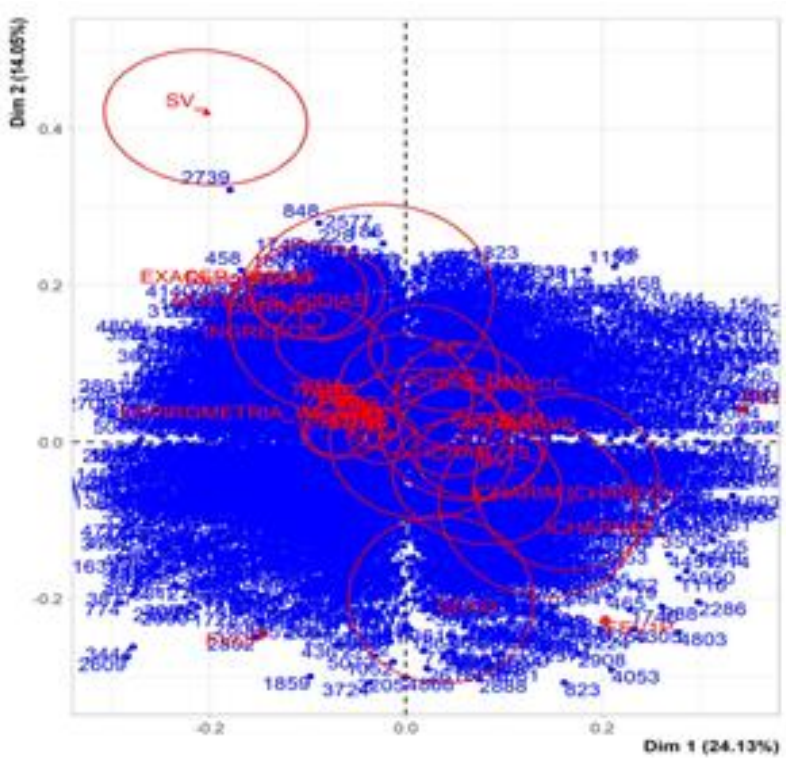



**Figura 21.** Descripción del plano 1: 2 (Pacientes vs Variables) en conjunto mediante CA

En este sentido, los resultados descritos por la *dimensión 1* muestra factores de individuos caracterizados por una *coordenada fuertemente positiva* en el eje (*a la derecha del gráfico*) frente a una negativa (*a la izquierda del gráfico*), donde se tiene que:

- ✓ *Grupo 1* con coordenada positiva en el eje, comparte alta frecuencia para estas variables por orden de importancia entre las más fuerte a menos: Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FEV1 espirometría en % del teórico *(FEV1P),* EDAD, Enfermedad Cerebro Vascular *(ICHARECV_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Nefropatía *(ICHARNEF_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Edemas maleolares *(EP_)* y Enfermedad Vascular Periférica *(ICHAREVP_)*; y con baja frecuencia para estas otras variables ordenadas por las más débiles: FVC espirometría en % del teórico *(FVCP),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Tensión Arterial Diastólica *(TARD),* Frecuencia

Cardíaca *(FCA),* Tensión Arterial Sistólica *(TARS),* Hábito Tabáquico *(HT_),* Frecuencia Respiratoria *(FRE),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_)* y Duración del Ingreso hospitalario *(DURING)*.

- ✓ *Grupo 2* con coordenada negativa en el eje, muestra alta frecuencia para estos factores-variables: FVC espirometría en % del teórico *(FVCP),* FEV1 espirometría en % del teórico *(FEV1P),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Hábito Tabáquico *(HT_)* y SEXO; y con baja para estas otras: Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* Tensión Arterial Sistólica *(TARS),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Diastólica *(TARD),* Duración del Ingreso hospitalario *(DURING),* Peso *(PPKG),* Soporte Ventilatorio *(SV_),* Índice de Masa Corporal *(IMC),* Edemas maleolares *(EP_)* y Altura *(ALP)*.

- ✓ *Grupo 3* también con coordenada negativa en el eje, indica alta frecuencia para estos factores de variables: Tensión Arterial Diastólica *(TARD),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Sistólica *(TARS),* Duración del Ingreso hospitalario *(DURING),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Soporte Ventilatorio *(SV_),* Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Frecuencia Respiratoria *(FRE)* e Ingresos hospitalarios por cualquier motivo *(INGRESOS_)*; y con baja para estas otras: FEV1 espirometría en % del teórico *(FEV1P),* Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FVC espirometría en % del teórico *(FVCP),* EDAD, Enfermedad Cerebro Vascular *(ICHARECV_),* Enfermedad Vascular *(EV_),* Comorbilidad Cardiovascular *(CCVSDM_),* SEXO, Nefropatía *(ICHARNEF_)* e Enfermedad Vascular Periférica *(ICHAREVP_).*

A la vista de los resultados, se observa que existen ciertos factores de individuos, que están altamente correlacionados, y podrían resumir este eje (*correlación bastante cerca de 0,97*).

Por otro lado, aunque con la primera dimensión se tendría suficiente información, por ser la que más contribuye a la inercia total, se describe la *dimensión 2* que enfrenta factores de individuos con una coordenada fuertemente positiva en el eje (*en la parte superior del gráfico*) a una negativa (*en la parte inferior del gráfico*).

- ✓ *Grupo 1* con una coordenada positiva en el eje, comparte alta frecuencia para estas variables: Tensión Arterial Diastólica *(TARD),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Sistólica *(TARS),* Duración del Ingreso hospitalario *(DURING),* Espirometría realizada previa al ingreso o al alta

*(ESPIROMETRIA_PA_),* Soporte Ventilatorio *(SV_),* Exacerbación a 90 días *(EXACER_90DIAS),* Reingresos por exacerbación *(Reing_EXAC),* Frecuencia Respiratoria *(FRE)* e Ingresos hospitalarios por cualquier motivo *(INGRESOS_)*; y con baja para estas otras: FEV1 espirometría en % del teórico *(FEV1P),* Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FVC espirometría en % del teórico *(FVCP),* EDAD, Enfermedad Cerebro Vascular *(ICHARECV_),* Enfermedad Vascular *(EV_),* Comorbilidad Cardiovascular *(CCVSDM_),* SEXO, Nefropatía *(ICHARNEF_)* y Enfermedad Vascular Periférica *(ICHAREVP_)*.

- ✓ *Grupo 2* también positiva, muestra alta frecuencia para estos factores-variables: Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* FEV1 espirometría en % del teórico *(FEV1P),* EDAD, Enfermedad Cerebro Vascular *(ICHARECV_),* Comorbilidad Cardiovascular *(CCVSDM_),* Enfermedad Vascular *(EV_),* Nefropatía *(ICHARNEF_),* Insuficiencia Cardíaca Congestiva *(ICHARICC_),* Edemas maleolares *(EP_) y* Enfermedad Vascular Periférica (*ICHAREVP)*; y con baja para estas otras: FVC espirometría en % del teórico *(FVCP),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Tensión Arterial Diastólica *(TARD),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Sistólica *(TARS),* Hábito Tabáquico *(HT_),* Rrecuencia Respiratoria *(FRE),* Reingresos por exacerbación *(Reing_EXAC),* Ingresos hospitalarios por cualquier motivo *(INGRESOS_)* y Duración del Ingreso hospitalario *(DURING).*

- ✓ *Grupo 3* con una coordenada negativa, indica alta frecuencia para estos factores de variables: FVC espirometría en % del teórico *(FVCP),* FEV1 espirometría en % del teórico *(FEV1P),* Espirometría realizada previa al ingreso o al alta *(ESPIROMETRIA_PA_),* Hábito Tabáquico *(HT_)* y SEXO; y con baja para estas otras: Cociente relacional FEV1/FVC espirometría previa o alta *(FEVCVF),* Tensión Arterial Sistólica *(TARS),* Frecuencia Cardíaca *(FCA),* Tensión Arterial Diastólica *(TARD),* Duración del Ingreso hospitalario *(DURING),* Peso *(PPKG),* Soporte Ventilatorio *(SV_),* Índice de Masa Corporal *(IMC),* Edemas maleolares *(EP_)* y Altura *(ALP)*.

### *III. 3. 3. Análisis de clasificación por Árboles de Decisión*

En línea con el planteamiento de búsqueda de perfiles óptimos para este caso, también se ha tomado la opción de explorar el Análisis de clasificación por Árboles de Decisión (*Decision Tree*, DT), (Navarro-Mateu et al., 2010; Bosco, 2018 y Wang et al., 2021a), como otra vía adicional de soporte al análisis principal (Cluster). Precisamente, esta opción vino motivada por detectar cualquier información

relevante o existencia de otra clasificación oculta entre los clusters óptimos que pueda mejorar el resultado definitivo de agrupamiento generado por los 3 grupos de perfiles finales.

Este método por Árboles de Decisión (DT) del paquete *rpart* de R (Therneau y Atkinson, 2019) es una técnica de *aprendizaje supervisado*, tal y como lo es el análisis de Vectores Soportes (SVM), que se mencionara brevemente en el capítulo IV, donde estos algoritmos de supervisión tienen como base la clasificación (o la regresión) de los datos previamente etiquetados (dependiendo del tipo de variable a explorar si es categórica o continua), con el fin de encontrar relaciones entre ellos através de una función, que pueda asociar a las variables de entradas con las de salida, generando un árbol de decisión que permite clasificar o predecir observaciones de individuos en el futuro. Por eso, si la variable objetivo o clase toma valores categóricos o discretos se denomina árbol de clasificación, y si por el contrario, toma valores continuos se llama árbol de regresión. Asimismo, todas las variables de entrada pueden ser continuas (numéricos) o categóricas (discretas, nominal, ordinal).

Más concretamente, estos modelos de clasificación tratan de predecir el valor de una variable predefinida (discreta o categórica) en un conjunto de posibles valores mediante la clasificación de la información en función de otras variables, como es el caso presentado. Sin embargo, en regresión ocurre todo lo contrario, desean predecir el valor de una variable no predefinida (continua) que puede tomar cualquier posible valor en función de otras variables, las cuales son independientes entre sí, (Martínez y Esparducer, 2013 y Im et al., 2021).

Por esta razón, a diferencia del análisis Cluster, del CA y del PCA, que son *procedimientos no supervisados*, donde se tratan de describir y encontrar estructuras de datos, que definan grupos según su similaridad (distiancia) o que simplificándolas a otra menor se mantenga sus características esenciales para aumentar la calidad de los datos através de la mejora del rendimiento de estos algoritmos implementados.

En este sentido, los árboles de decisión (Franchuk et al., 2020; Karacan et al., 2020 y Gheondea-Eladi, 2019), son uno de los enfoques algorítmicos más en auge y poderosos actualmente en la gran mayoría de los campos de investigación sobre aprendizaje automático (Machine Learning), en especial en los ámbitos sanitarios y de minería o ciencia de datos, para la búsqueda de comportamientos diferentes que puedan definir grupos de patrones eficientes y más precisos con la finalidad de que puedan ser aplicables sobre la población general.

Este algoritmo de clasificación utiliza reglas de decisión sencillas o restricciones dicotómicas, que usualmente son de dos grupos o clases, como parte de su análisis clasificatorio, manteniendo un orden jerárquico en su aplicación, y construye una estructura en forma de árbol, para representarlo visualmente como resultado final, (donde los colores indican las categorías de las clases de salida en cada nodo, y muestran que a mayor grado de color mejor es la separación de las clases), ayudando de esta manera a la toma de decisiones mediante la descripción de los datos sin tomar la resolución final sobre ellos.

Es decir, esta técnica usa el modelo predictivo (Orellana, 2018; Martínez, 2020 y IBM Corporation, 2021), para representar el mapeado de las observaciones de individuos del conjunto de datos hasta dar con la decisión final, donde las hojas son las categorías o etiquetas de clase, los nodos son las reglas (o pruebas simples) a realizar y las ramas (o descendientes de cada nodo) son las uniones o enlaces de características que llevan a esas categorías de clase (o posibles resultados de las pruebas del nodo).

Dicho de otra manera, estos métodos consisten en explorar todas las instancias del conjunto de datos con la finalidad de detectar el valor o categoría que mejor proporcione una correcta clasificación o predicción separando los datos en subgrupos, donde todas las divisiones son binarias creando dos particiones de grupos. Este procedimiento de entrenamiento se aplica de forma repetitiva (recurrente) con la identificación inicial de la raíz del árbol y se continua hasta llegar a cada una de las hojas, dividiendo los subgrupos en unidades cada vez más pequeñas hasta completar el árbol (que es cuando se alcanza el umbral de pureza del nodo que esta determinado por el valor minimo o máximo de la medida de división) o hasta cumplir con algún criterio de parada previamente definido, (que sucede cuando se determina al inicio un límite de profundidad máximo prefijado, donde esta profundidad viene medida como el número de nodos desde la raíz hasta la hoja, que en caso de no especificar nada, los nodos se expanden hasta que todas las hojas sean puras o contengan el mínimo de categorías o instancias necesarias para dividir un nodo interno, que por defecto es dos).

En este sentido, se conoce que a mayor profundidad del árbol, esto implicara mayor complejidad en las reglas de decisión, generando un modelo muy ajustado y por consiguiente, este ajuste puede originar un sobreaprendizaje del algoritmo, por lo que hay que tener cuidado con el criterio de parada que se fija.

Asimismo, cabe destacar que este proceso solo se queda con los atributos o categorías que realmente importan en la toma decisiones, descartando aquellos atributos que no contribuyan a la precisión del

árbol, por lo que esta información es de gran valor y puede servir para reducir los datos a las variables más relevantes o de mayor importancia antes de aplicar cualquier otra técnica diferente.

Además, esta regla de división, que mide la calidad de una partición, se puede realizar através de dos tipos de medidas: (i) una por defecto que se denomina "Índice Gini" que mide la impureza y (ii) otra que se llama "entropía" que mide la ganancia de información, con la finalidad de conseguir que la conbinación de los valores anteriores se minimicen o se maximicen (entropía o gini, respectivamente), provocando que todas las categorías o instancias de una clase se puedan unir a la parte positiva de la condición, mientras que el resto permanece en la parte contraria (negativa). Así pues, se tiene que este proceso solo considerara un nodo puro, si el 100% de los casos del nodo corresponden a una categoría específica de la variable objetivo (clase).

Por esta misma razón de precisión y seguridad, los modelos de árboles de decisión están muy extendidos recientemente, y su utilización es bastante alta en diversos campos de investigación por ser una técnica sencilla, autoexplicativa y de fácil escalabilidad, ya que son capaces de manejar una gran variedad de datos-variables de entradas diferentes con un costo mínimo de computación, y además, pueden procesar conjuntos de datos con valores faltantes originando un alto poder predictivo con poco esfuerzo computacional y asimismo, pueden manejar datos ruidosos utilizando mecanismos internos (como los denominados "poda") que reducen la profundidad del árbol de tal manera que resulte la mejor generalización. Y en paralelo, pueden servir de soporte para diferentes funciones según el objetivo planteado, bien para el análisis de clasificación, regresión, agrupamiento o clusters y finalmente, también se pueden utilizar como técnica adicional para el procedimiento de selección de variables (o categorías) mediante el cribado final de atributos (o variables).

Asimismo, cabe destacar que este método DT tiene algunas limitaciones, concretamente dos inconvenientes: (i) cuando se aplica un factor alto de profundidad, puede presentar ciertos problemas de sobreaprendizaje complicando el proceso algorítmico sobre las categorías de clase, y (ii) no detecta las correlaciones entre las variables, puesto que cada nodo de decisión se obtiene de forma independiente, sin tener en cuenta al resto de nodos o pruebas.

No obstante, como se ha comentado anteriormente, para este caso particular, la técnica de árboles de decisión (DT) se ha utilizado como soporte exploratorio adicional de clasificación para la búsqueda de perfiles-pacientes, detectando cualquier cambio en los datos que pueda ayudar en la definición final de los grupos perfiles formados. Dicho esto, se han explorado varios casos (clases), aunque solo dos de ellos han sido los más relevantes por el interés clínico que puedan aportar con relación a los ingresos

hospitalarios y al hábito tabáquico, ya que se intuye que puedan ofrecer algún resultado extra o simplemente ser un sopoorte adicional de confirmación para los datos existentes.

Para ello, se ha iniciado este procedimiento fijando una semilla de "seed=500" y posteriormente, se ha ido variando (seed=1000, 1500) para poder detectar cambios significativos. Asimismo, se ha realizado una división del conjunto de datos ("dataset") en dos partes, donde el 70% son datos de entrenamiento "train" y el 30% son datos de prueba "test" para su posterior validación sobre el resultado obtenido.

A la vista de los resultados del primer árbol de decisión sobre los datos de *hábito tabáquico* (*Figura 22*), se puede observar que el 82% son fumadores y la clasificación se inicia con la variable de género (Sexo etiquetada como (1) Mujer y (0) Hombre) en el primer tramo del flujo de decisión, finalizando el proceso con la realización de la prueba espirométrica y el corte de edad avanzado.

Este árbol concluye mostrando varios resultados para tomar en consideración, como son: (i) que el 12% de mujeres fumadoras mayores de 73 años (6%) tienen una predisposición a gravedad en un 6%; y (ii) que el 88% de hombres fumadores sin espirometria realizada (27%) tienen una predisposicion a gravedad en un 60%.

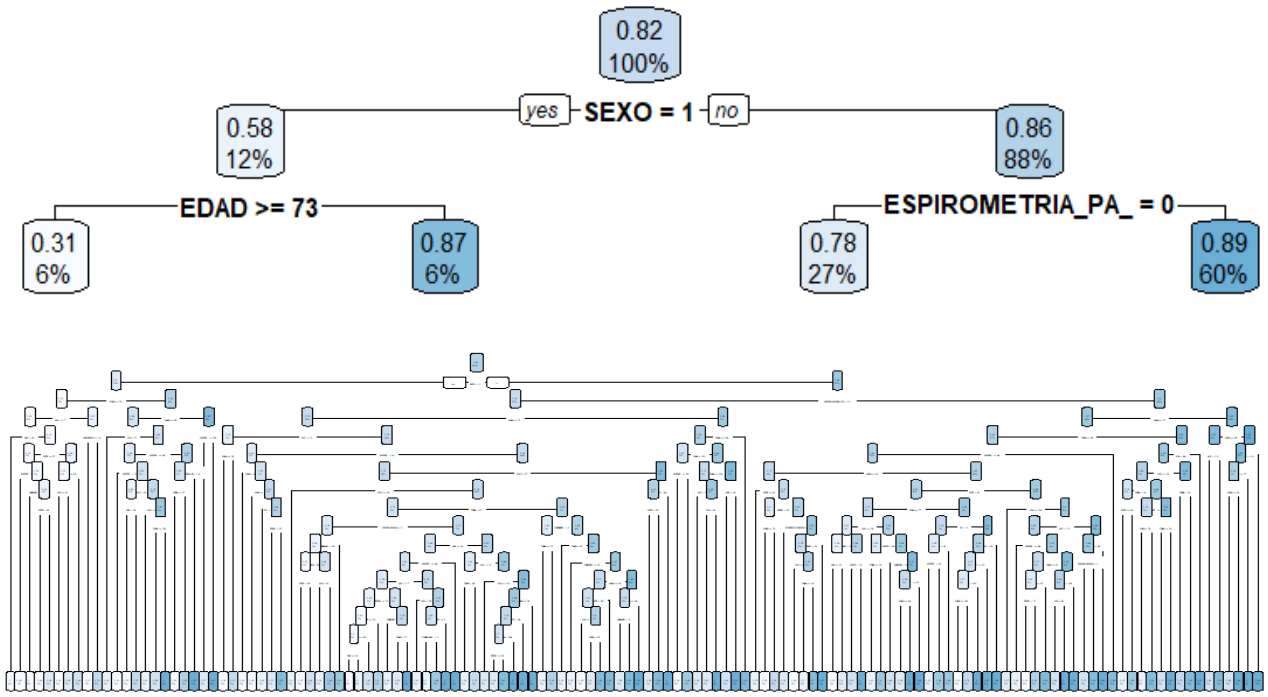



**Figura 22.** Visualización del árbol de decisión (DT) por Hábito Tabáquico

Asimismo, se puede visualizar la salida completa del árbol (*Tabla 7*) cuando la representación visual del mismo es bastante borrosa o poco clara, que lleva a provocar que no se puedan apreciar los nodos finales del mismo. Desafortunadamente, este problema de mapeado es muy común cuando el árbol tiene una mayor profundidad, lo que genera que sus reglas de decisión se compliquen llevando a mostrar una estructura del árbol muy poco nítida.

**Tabla 7.** Salida del árbol de decisión por Hábito Tabáquico – Forma abreviada y extendida

Forma abreviada (*arbol_4f*)

```
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 527.65350 0.8231724
#  2) SEXO>=0.5 442 107.72850 0.5791855
#   4) EDAD>=72.5 226  47.93363 0.3053097 *
#   5) EDAD< 72.5 216  25.10648 0.8657407 *
#  3) SEXO< 0.5 3183 389.95920 0.8570531
#    6) ESPIROMETRIA_PA_< 0.5 991 171.71540 0.7769929 *
#    7) ESPIROMETRIA_PA_>=0.5 2192 209.02010 0.8932482 *
```

Forma extendida (arbol_4ff)

```
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
    1) root 3625 520.5103000 0.82620690
      2) SEXO>=0.5 474 113.8397000 0.59915610
        4) EDAD>=77.5 168  28.2857100 0.21428570
          8) ALP< 1.705 147  22.0408200 0.18367350
           16) FRE< 18.5 31   0.0000000 0.00000000 *
           17) FRE>=18.5 116  20.7155200 0.23275860
             34) FVCP>=56.5 74   9.3648650 0.14864860
               68) FEVCVF_< 70.57 43   1.9069770 0.04651163 *
               69) FEVCVF_>=70.57 31   6.3870970 0.29032260
                138) DURING< 5.5 8   0.0000000 0.00000000 *
                139) DURING>=5.5 23   5.4782610 0.39130430
                  278) FVCP< 67.5 11   1.6363640 0.18181820 *
                  279) FVCP>=67.5 12   2.9166670 0.58333330 *
             35) FVCP< 56.5 42   9.9047620 0.38095240
               70) PPKG>=65.5 28   5.2500000 0.25000000
                140) FCA>=81 19   1.7894740 0.10526320 *
                141) FCA< 81 9   2.2222220 0.55555560 *
               71) PPKG< 65.5 14   3.2142860 0.64285710 *
          9) ALP>=1.705 21   5.1428570 0.42857140
           18) INGRESOS_< 0.5 12   1.6666670 0.16666670 *
```

```
   19) INGRESOS_>=0.5 9   1.5555560 0.77777780 *
  5) EDAD< 77.5 306  47.0065400 0.81045750
  10) EDAD>=64.5 145  32.7586200 0.65517240
   20) FCA< 71.5 16   3.7500000 0.37500000 *
   21) FCA>=71.5 129  27.5969000 0.68992250
    42) FEVCVF_>=68.695 73  17.6712300 0.58904110
     84) FEVCVF_< 71.62 8   0.8750000 0.12500000 *
     85) FEVCVF_>=71.62 65  14.8615400 0.64615380
     170) IMC< 28.23 28   6.9642860 0.46428570
      340) FRE< 29 19   4.1052630 0.31578950 *
      341) FRE>=29 9   1.5555560 0.77777780 *
     171) IMC>=28.23 37   6.2702700 0.78378380
      342) IMC>=37.255 7   1.7142860 0.42857140 *
      343) IMC< 37.255 30   3.4666670 0.86666670
       686) TARS>=138 13   2.7692310 0.69230770 *
       687) TARS< 138 17   0.0000000 1.00000000 *
    43) FEVCVF_< 68.695 56   8.2142860 0.82142860
     86) ESPIROMETRIA_PA_< 0.5 26   5.8846150 0.65384620
     172) TURA>=37.1 9   2.0000000 0.33333330 *
     173) TURA< 37.1 17   2.4705880 0.82352940 *
     87) ESPIROMETRIA_PA_>=0.5 30   0.9666667 0.96666670 *
  11) EDAD< 64.5 161   7.6024840 0.95031060
   22) FEVCVF_>=133.6 7   1.7142860 0.57142860 *
   23) FEVCVF_< 133.6 154   4.8376620 0.96753250 *
3) SEXO< 0.5 3151 378.5592000 0.86036180
  6) ESPIROMETRIA_PA_< 0.5 983 170.7630000 0.77619530
  12) EDAD>=70.5 733 140.7503000 0.74079130
   24) ALP< 1.505 29   7.2413790 0.51724140
    48) EDAD>=79 14   2.8571430 0.28571430 *
    49) EDAD< 79 15   2.9333330 0.73333330 *
   25) ALP>=1.505 704 132.0000000 0.75000000
    50) FEVCVF_< 35.81 61  14.5573800 0.60655740
     100) TARD< 59.5 7   0.8571429 0.14285710 *
     101) TARD>=59.5 54  12.0000000 0.66666670
      202) EDAD>=85.5 11   2.5454550 0.36363640 *
      203) EDAD< 85.5 43   8.1860470 0.74418600
       406) EP_>=0.5 11   2.7272730 0.45454550 *
       407) EP_< 0.5 32   4.2187500 0.84375000
        814) PPKG< 61.5 7   1.7142860 0.57142860 *
        815) PPKG>=61.5 25   1.8400000 0.92000000 *
    51) FEVCVF_>=35.81 643 116.0684000 0.76360810
     102) IMC< 36.54 588 110.2500000 0.75000000
      204) IMC>=35.7 10   2.4000000 0.40000000 *
      205) IMC< 35.7 578 106.6038000 0.75605540
       410) TURA< 36.75 323  65.7956700 0.71517030
        820) EXACER_90DIAS>=0.5 64  15.6093800 0.57812500
         1640) DURING< 13.5 49  12.2449000 0.48979590
          3280) EDAD< 79.5 27   6.0000000 0.33333330
           6560) FVCP< 65.5 16   1.7500000 0.12500000 *
           6561) FVCP>=65.5 11   2.5454550 0.63636360 *
          3281) EDAD>=79.5 22   4.7727270 0.68181820
           6562) FCA< 80.5 8   1.8750000 0.37500000 *
```

```
      6563) FCA>=80.5 14   1.7142860 0.85714290 *
    1641) DURING>=13.5 15   1.7333330 0.86666670 *
   821) EXACER_90DIAS< 0.5 259  48.6872600 0.74903470
    1642) FRE>=23.5 142  30.7394400 0.68309860
     3284) FEVCVF_>=44.28 130  29.4230800 0.65384620
      6568) ALP< 1.675 71  17.4647900 0.56338030
       13136) PPKG>=71.5 39   9.6923080 0.46153850
        26272) FVCP>=52.5 29   6.5517240 0.34482760
         52544) TARS>=124.5 20   3.2000000 0.20000000
          105088) FRE>=24.5 12   0.0000000 0.00000000 *
          105089) FRE< 24.5 8   2.0000000 0.50000000 *
         52545) TARS< 124.5 9   2.0000000 0.66666670 *
        26273) FVCP< 52.5 10   1.6000000 0.80000000 *
       13137) PPKG< 71.5 32   6.8750000 0.68750000
        26274) DURING< 4.5 8   1.8750000 0.37500000 *
        26275) DURING>=4.5 24   3.9583330 0.79166670
         52550) TARD< 71 13   3.0769230 0.61538460 *
         52551) TARD>=71 11   0.0000000 1.00000000 *
      6569) ALP>=1.675 59  10.6779700 0.76271190
       13138) IMC< 19.61 7   1.7142860 0.42857140 *
       13139) IMC>=19.61 52   8.0769230 0.80769230
        26278) DURING>=7.5 32   6.8750000 0.68750000
         52556) ICHAR_DM_>=0.5 16   4.0000000 0.50000000 *
         52557) ICHAR_DM_< 0.5 16   1.7500000 0.87500000 *
        26279) DURING< 7.5 20   0.0000000 1.00000000 *
     3285) FEVCVF_< 44.28 12   0.0000000 1.00000000 *
    1643) FRE< 23.5 117  16.5812000 0.82905980
     3286) ALP< 1.595 23   5.4782610 0.60869570
      6572) TARD< 68.5 8   1.8750000 0.37500000 *
      6573) TARD>=68.5 15   2.9333330 0.73333330 *
     3287) ALP>=1.595 94   9.7127660 0.88297870
      6574) FCA>=112.5 7   1.7142860 0.57142860 *
      6575) FCA< 112.5 87   7.2643680 0.90804600
       13150) EDAD>=78.5 56   6.8571430 0.85714290
        26300) FEV1P< 53.5 37   6.2702700 0.78378380
         52600) TARS< 139.5 27   5.6296300 0.70370370
          105200) PPKG>=65.5 16   3.9375000 0.56250000 *
          105201) PPKG< 65.5 11   0.9090909 0.90909090 *
         52601) TARS>=139.5 10   0.0000000 1.00000000 *
        26301) FEV1P>=53.5 19   0.0000000 1.00000000 *
       13151) EDAD< 78.5 31   0.0000000 1.00000000 *
  411) TURA>=36.75 255  39.5843100 0.80784310
   822) ICHARNEF_>=0.5 34   8.0294120 0.61764710
    1644) TURA>=37.25 16   3.7500000 0.37500000 *
    1645) TURA< 37.25 18   2.5000000 0.83333330 *
   823) ICHARNEF_< 0.5 221  30.1357500 0.83710410
    1646) TARD< 47 7   1.7142860 0.42857140 *
    1647) TARD>=47 214  27.2149500 0.85046730
     3294) DURING>=34.5 11   2.7272730 0.54545450 *
     3295) DURING< 34.5 203  23.4088700 0.86699510
      6590) EDAD>=82.5 61  10.7868900 0.77049180
       13180) FVCP< 66 36   8.0000000 0.66666670
```

```
          26360) FCA>=98.5 11   2.5454550 0.36363640 *
          26361) FCA< 98.5 25   4.0000000 0.80000000
            52722) EDAD< 85.5 10   2.4000000 0.60000000 *
            52723) EDAD>=85.5 15   0.9333333 0.93333330 *
         13181) FVCP>=66 25   1.8400000 0.92000000 *
        6591) EDAD< 82.5 142  11.8098600 0.90845070
         13182) TARD>=79.5 48   7.9166670 0.79166670
          26364) DURING< 7.5 19   4.6315790 0.57894740 *
          26365) DURING>=7.5 29   1.8620690 0.93103450 *
         13183) TARD< 79.5 94   2.9042550 0.96808510 *
    103) IMC>=36.54 55   4.5454550 0.90909090
      206) FEV1P< 43.5 27   4.0740740 0.81481480
        412) FVCP< 54.5 9   2.2222220 0.55555560 *
        413) FVCP>=54.5 18   0.9444444 0.94444440 *
      207) FEV1P>=43.5 28   0.0000000 1.00000000 *
  13) EDAD< 70.5 250  26.4000000 0.88000000
    26) TARS>=129.5 142  20.5985900 0.82394370
      52) TARS< 132.5 27   6.0000000 0.66666670
        104) FEVCVF_>=86.44 11   2.5454550 0.36363640 *
        105) FEVCVF_< 86.44 16   1.7500000 0.87500000 *
      53) TARS>=132.5 115  13.7739100 0.86086960
        106) TARS>=152.5 37   7.2972970 0.72972970
          212) ALP< 1.625 16   3.9375000 0.56250000 *
          213) ALP>=1.625 21   2.5714290 0.85714290
            426) FCA< 91 8   1.8750000 0.62500000 *
            427) FCA>=91 13   0.0000000 1.00000000 *
        107) TARS< 152.5 78   5.5384620 0.92307690
          214) FRE< 19 11   2.5454550 0.63636360 *
          215) FRE>=19 67   1.9402990 0.97014930 *
    27) TARS< 129.5 108   4.7685190 0.95370370 *
7) ESPIROMETRIA_PA_>=0.5 2168 197.6753000 0.89852400
  14) EDAD>=70.5 1420 160.8979000 0.86971830
    28) TARD>=68.5 972 126.1595000 0.84670780
      56) FEVCVF_>=49.255 846 118.1655000 0.83215130
        112) EDAD>=82.5 159  29.4339600 0.75471700
          224) FVCP>=90.5 19   4.7368420 0.52631580 *
          225) FVCP< 90.5 140  23.5714300 0.78571430
            450) PPKG< 62 25   6.0000000 0.60000000
              900) TARD>=77.5 11   2.5454550 0.36363640 *
              901) TARD< 77.5 14   2.3571430 0.78571430 *
            451) PPKG>=62 115  16.5217400 0.82608700
              902) MUERTOS_90DIAS>=0.5 12   2.9166670 0.58333330 *
              903) MUERTOS_90DIAS< 0.5 103  12.8155300 0.85436890
               1806) PPKG< 80.5 67  11.0746300 0.79104480
                 3612) IMC>=31.05 7   1.4285710 0.28571430 *
                 3613) IMC< 31.05 60   7.6500000 0.85000000
                   7226) TURA>=37.6 8   2.0000000 0.50000000 *
                   7227) TURA< 37.6 52   4.5192310 0.90384620 *
               1807) PPKG>=80.5 36   0.9722222 0.97222220 *
        113) EDAD< 82.5 687  87.5575000 0.85007280
          226) FVCP< 62.5 341  52.6099700 0.80938420
            452) EDAD>=76.5 162  30.1234600 0.75308640
```

```
  904) IMC>=30.46 48  11.2500000 0.62500000
   1808) PPKG< 89 23   5.4782610 0.39130430
     3616) FRE>=20.5 13   2.3076920 0.23076920 *
     3617) FRE< 20.5 10   2.4000000 0.60000000 *
   1809) PPKG>=89 25   3.3600000 0.84000000
     3618) FEV1P< 35 12   2.6666670 0.66666670 *
     3619) FEV1P>=35 13   0.0000000 1.00000000 *
  905) IMC< 30.46 114  17.7543900 0.80701750
   1810) FEV1P< 28.5 24   5.6250000 0.62500000
     3620) TURA>=36.15 14   3.4285710 0.42857140 *
     3621) TURA< 36.15 10   0.9000000 0.90000000 *
   1811) FEV1P>=28.5 90  11.1222200 0.85555560
     3622) DURING< 5.5 19   4.1052630 0.68421050 *
     3623) DURING>=5.5 71   6.3098590 0.90140850
       7246) FEVCVF_>=87.145 21   3.8095240 0.76190480
        14492) EDAD>=79.5 12   2.9166670 0.58333330 *
        14493) EDAD< 79.5 9   0.0000000 1.00000000 *
       7247) FEVCVF_< 87.145 50   1.9200000 0.96000000 *
 453) EDAD< 76.5 179  21.5083800 0.86033520
  906) EV_>=0.5 58  10.0862100 0.77586210
   1812) TURA< 36.45 22   5.3181820 0.59090910
     3624) FEVCVF_< 83.195 14   3.4285710 0.42857140 *
     3625) FEVCVF_>=83.195 8   0.8750000 0.87500000 *
   1813) TURA>=36.45 36   3.5555560 0.88888890 *
  907) EV_< 0.5 121  10.8099200 0.90082640
   1814) IMC< 28.225 70   9.2714290 0.84285710
     3628) IMC>=25.12 35   7.1428570 0.71428570
       7256) TARS< 153 23   5.6521740 0.56521740
        14512) TARS>=141.5 9   1.5555560 0.22222220 *
        14513) TARS< 141.5 14   2.3571430 0.78571430 *
       7257) TARS>=153 12   0.0000000 1.00000000 *
     3629) IMC< 25.12 35   0.9714286 0.97142860 *
   1815) IMC>=28.225 51   0.9803922 0.98039220 *
227) FVCP>=62.5 346  33.8265900 0.89017340
 454) FCA>=137.5 7   1.7142860 0.57142860 *
 455) FCA< 137.5 339  31.3864300 0.89675520
  910) ALP< 1.635 153  20.2352900 0.84313730
   1820) FEVCVF_>=99.22 9   2.2222220 0.44444440 *
   1821) FEVCVF_< 99.22 144  16.4930600 0.86805560
     3642) TARD>=80.5 56   9.9821430 0.76785710
       7284) FEV1P>=51.5 30   6.9666670 0.63333330
        14568) PPKG< 70.5 12   2.6666670 0.33333330 *
        14569) PPKG>=70.5 18   2.5000000 0.83333330 *
       7285) FEV1P< 51.5 26   1.8461540 0.92307690 *
     3643) TARD< 80.5 88   5.5909090 0.93181820
       7286) TARS< 123.5 27   4.0740740 0.81481480
        14572) FCA< 91 16   3.4375000 0.68750000 *
        14573) FCA>=91 11   0.0000000 1.00000000 *
       7287) TARS>=123.5 61   0.9836066 0.98360660 *
  911) ALP>=1.635 186  10.3494600 0.94086020
   1822) FCA< 94.5 94   8.9361700 0.89361700
     3644) EXACER_90DIAS>=0.5 18   3.6111110 0.72222220 *
```

```
          3645) EXACER_90DIAS< 0.5 76   4.6710530 0.93421050 *
         1823) FCA>=94.5 92   0.9891304 0.98913040 *
   57) FEVCVF_< 49.255 126   6.6111110 0.94444440 *
 29) TARD< 68.5 448  33.1071400 0.91964290
   58) FCA< 98.5 296  28.5405400 0.89189190
    116) IMC>=27.965 126  16.8254000 0.84126980
     232) TURA< 36.15 33   6.5454550 0.72727270
      464) FVCP>=67.5 15   3.7333330 0.53333330 *
      465) FVCP< 67.5 18   1.7777780 0.88888890 *
     233) TURA>=36.15 93   9.6989250 0.88172040
      466) FEVCVF_>=89.56 23   4.4347830 0.73913040
       932) EDAD< 78.5 10   2.5000000 0.50000000 *
       933) EDAD>=78.5 13   0.9230769 0.92307690 *
      467) FEVCVF_< 89.56 70   4.6428570 0.92857140
       934) TURA>=37.7 11   2.1818180 0.72727270 *
       935) TURA< 37.7 59   1.9322030 0.96610170 *
    117) IMC< 27.965 170  11.1529400 0.92941180
     234) TURA< 35.95 16   3.0000000 0.75000000 *
     235) TURA>=35.95 154   7.5844160 0.94805190 *
   59) FCA>=98.5 152   3.8947370 0.97368420 *
15) EDAD< 70.5 748  33.3623000 0.95320860
  30) EDAD>=65.5 279  20.2652300 0.92114700
    60) ALP< 1.545 13   2.7692310 0.69230770 *
    61) ALP>=1.545 266  16.7819500 0.93233080 *
  31) EDAD< 65.5 469  12.6396600 0.97228140
    62) ALP< 1.595 80   5.5500000 0.92500000
     124) TURA>=36.75 38   5.0526320 0.84210530
      248) EDAD>=61.5 14   3.2142860 0.64285710 *
      249) EDAD< 61.5 24   0.9583333 0.95833330 *
     125) TURA< 36.75 42   0.0000000 1.00000000 *
    63) ALP>=1.595 389   6.8740360 0.98200510 *
```

Del mismo modo que en el caso anterior, los resultados del segundo árbol de decisión sobre los datos de *ingresos hospitalarios* (*Figura 23*) muestran que el 35% tuvieron algún ingreso hospitalario y la clasificación se inicia con la variable de reingresos por exacerbación (etiquetada como (1) Si/Yes y (0) No) en el primer tramo del flujo de decisión y finaliza con la comprobación del evento de mortalidad a los tres meses.

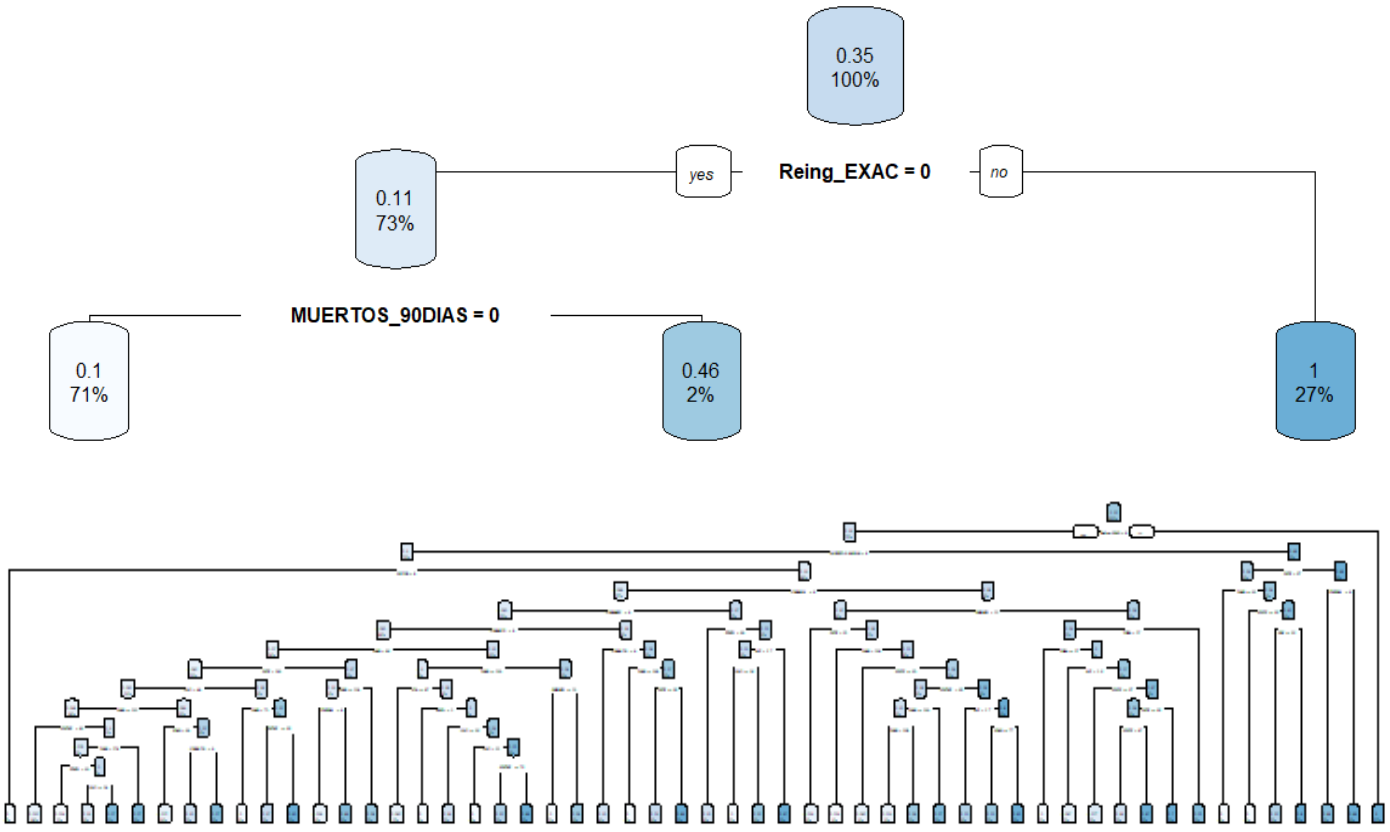



**Figura 23.** Visualización del árbol de decisión (DT) por Ingresos Hospitalarios

También, este otro árbol indica algunos resultados concluyentes para tomar en consideración, como son: (i) que el 73% de los pacientes que no reingresan por exacerbación, solo un 2% tienen predisposición a fallecer; y (ii) que el 27% de los pacientes que si reingresan por exacerbación, tienen una predisposicion muy alta a fallecer en los 90 días siguientes del reingreso por exacerbación por la propia gravedad de la enfermedad y por reiterados reingresos hospitalarios sin mejorar en las terapias asignadas.

Asimismo, se puede visualizar la salida completa del árbol (*Tabla 8*) cuando esta representación visual no es bastante limpia y no se alcance a ver los nodos finales del mismo, que normalmente esto ocurre cuando el árbol tiene una alta profundidad que provocan que sus reglas de decisión se compliquen demasiado no pudiendo mostrar un gráfico del árbol que sea algo más claro y transparente.

**Tabla 8.** Salida del árbol de decisión por Ingresos Hospitalarios – Forma abreviada y extendida

Forma abreviada **(***arbol_2f*)

```
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 826.2549 0.3514483
#  2) Reing_EXAC< 0.5 2653 267.6223 0.1138334
#    4) MUERTOS_90DIAS< 0.5 2566 235.2486 0.1021044 *
#    5) MUERTOS_90DIAS>=0.5 87  21.6092 0.4597701 *
#  3) Reing_EXAC>=0.5 972   0.0000 1.0000000 *
```

Forma extendida (*arbol_2ff*)

```
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
   1) root 3625 826.2549000 0.35144830
     2) Reing_EXAC< 0.5 2653 267.6223000 0.11383340
       4) MUERTOS_90DIAS< 0.5 2566 235.2486000 0.10210440
         8) EXITUS< 0.5 188   0.0000000 0.00000000 *
         9) EXITUS>=0.5 2378 233.1337000 0.11017660
          18) ICHARIM_< 0.5 2071 183.1019000 0.09802028
            36) ICHARNEF_< 0.5 1903 158.9070000 0.09196006
              72) ICHARICC_< 0.5 1592 119.3844000 0.08165829
               144) TARD< 85.5 1236  80.8762100 0.07038835
                 288) FVCP< 101.5 1177  71.9626200 0.06542056
                   576) IMC< 39.655 1133  65.6752000 0.06178288
                    1152) TARS>=120.5 694  31.4308400 0.04755043
                      2304) FEVCVF_< 91.92 568  19.2957700 0.03521127 *
                      2305) FEVCVF_>=91.92 126  11.6587300 0.10317460
                        4610) TARS< 173.5 119   8.3193280 0.07563025
                          9220) PPKG< 82.5 84   1.9523810 0.02380952 *
                          9221) PPKG>=82.5 35   5.6000000 0.20000000
                            18442) IMC>=30.47 28   2.6785710 0.10714290 *
                            18443) IMC< 30.47 7   1.7142860 0.57142860 *
                        4611) TARS>=173.5 7   1.7142860 0.57142860 *
                    1153) TARS< 120.5 439  33.8815500 0.08428246
                      2306) EDAD< 87.5 415  28.6843400 0.07469880 *
                      2307) EDAD>=87.5 24   4.5000000 0.25000000
                        4614) ICHAR_TS_< 0.5 17   1.7647060 0.11764710 *
```

```
          4615) ICHAR_TS_>=0.5 7   1.7142860 0.57142860 *
        577) IMC>=39.655 44   5.8863640 0.15909090
        1154) TARD< 72.5 24   0.0000000 0.00000000 *
        1155) TARD>=72.5 20   4.5500000 0.35000000
          2310) FEVCVF_>=55.435 12   1.6666670 0.16666670 *
          2311) FEVCVF_< 55.435 8   1.8750000 0.62500000 *
      289) FVCP>=101.5 59   8.3050850 0.16949150
        578) TARS>=118.5 45   4.4444440 0.11111110
        1156) CCVSDM_< 0.5 36   0.9722222 0.02777778 *
        1157) CCVSDM_>=0.5 9   2.2222220 0.44444440 *
        579) TARS< 118.5 14   3.2142860 0.35714290 *
    145) TARD>=85.5 356  37.8061800 0.12078650
      290) TARS>=127.5 324  30.4321000 0.10493830
        580) FCA>=96.5 184  11.2173900 0.06521739 *
        581) FCA< 96.5 140  18.5428600 0.15714290
        1162) SEXO>=0.5 28   0.0000000 0.00000000 *
        1163) SEXO< 0.5 112  17.6785700 0.19642860
          2326) IMC>=25.74 80   9.4875000 0.13750000 *
          2327) IMC< 25.74 32   7.2187500 0.34375000
            4654) IMC< 20.62 11   0.0000000 0.00000000 *
            4655) IMC>=20.62 21   5.2380950 0.52380950
              9310) FEVCVF_>=74.7 8   1.5000000 0.25000000 *
              9311) FEVCVF_< 74.7 13   2.7692310 0.69230770 *
      291) TARS< 127.5 32   6.4687500 0.28125000
        582) DURING>=11.5 8   0.0000000 0.00000000 *
        583) DURING< 11.5 24   5.6250000 0.37500000 *
    73) ICHARICC_>=0.5 311  38.4887500 0.14469450
    146) ICHAR_TS_< 0.5 273  29.7655700 0.12454210 *
    147) ICHAR_TS_>=0.5 38   7.8157890 0.28947370
      294) TARS>=160 8   0.0000000 0.00000000 *
      295) TARS< 160 30   6.9666670 0.36666670
        590) FVCP>=55.5 19   3.1578950 0.21052630 *
        591) FVCP< 55.5 11   2.5454550 0.63636360 *
  37) ICHARNEF_>=0.5 168  23.3333300 0.16666670
    74) PPKG< 87.5 128  13.2421900 0.11718750 *
    75) PPKG>=87.5 40   8.7750000 0.32500000
    150) ALP< 1.71 31   5.4193550 0.22580650
      300) IMC>=36.08 11   0.0000000 0.00000000 *
      301) IMC< 36.08 20   4.5500000 0.35000000 *
    151) ALP>=1.71 9   2.0000000 0.66666670 *
 19) ICHARIM_>=0.5 307  47.6612400 0.19218240
  38) DURING< 10.5 210  26.4238100 0.14761900
    76) FVCP< 50.5 41   0.9756098 0.02439024 *
    77) FVCP>=50.5 169  24.6745600 0.17751480
    154) TARS< 119.5 42   0.9761905 0.02380952 *
    155) TARS>=119.5 127  22.3779500 0.22834650
      310) FEV1P>=62.5 31   1.8709680 0.06451613 *
      311) FEV1P< 62.5 96  19.4062500 0.28125000
        622) FEVCVF_< 64.735 50   7.3800000 0.18000000
        1244) TARS>=127.5 43   4.4186050 0.11627910
          2488) TARS< 159 32   0.9687500 0.03125000 *
          2489) TARS>=159 11   2.5454550 0.36363640 *
```

```
          1245) TARS< 127.5 7   1.7142860 0.57142860 *
          623) FEVCVF_>=64.735 46  10.9565200 0.39130430
          1246) ALP< 1.665 26   4.6153850 0.23076920 *
          1247) ALP>=1.665 20   4.8000000 0.60000000
            2494) EDAD>=76.5 9   2.0000000 0.33333330 *
            2495) EDAD< 76.5 11   1.6363640 0.81818180 *
      39) DURING>=10.5 97  19.9175300 0.28865980
        78) TURA< 37.25 80  14.4875000 0.23750000
        156) TURA>=36.85 16   0.0000000 0.00000000 *
        157) TURA< 36.85 64  13.3593800 0.29687500
          314) ALP< 1.595 15   0.9333333 0.06666667 *
          315) ALP>=1.595 49  11.3877600 0.36734690
            630) FEV1P>=56.5 13   0.9230769 0.07692308 *
            631) FEV1P< 56.5 36   8.9722220 0.47222220
            1262) FVCP>=54.5 26   6.1538460 0.38461540
              2524) FEV1P< 46.5 14   1.7142860 0.14285710 *
              2525) FEV1P>=46.5 12   2.6666670 0.66666670 *
            1263) FVCP< 54.5 10   2.1000000 0.70000000 *
        79) TURA>=37.25 17   4.2352940 0.52941180 *
    5) MUERTOS_90DIAS>=0.5 87  21.6092000 0.45977010
      10) FVCP< 66.5 47   9.4042550 0.27659570
        20) TARD>=81 9   0.0000000 0.00000000 *
        21) TARD< 81 38   8.5526320 0.34210530
          42) FEV1P>=55 7   0.0000000 0.00000000 *
          43) FEV1P< 55 31   7.5483870 0.41935480
            86) FRE>=24.5 16   3.0000000 0.25000000 *
            87) FRE< 24.5 15   3.6000000 0.60000000 *
      11) FVCP>=66.5 40   8.7750000 0.67500000
        22) CCVSDM_< 0.5 18   4.4444440 0.44444440 *
        23) CCVSDM_>=0.5 22   2.5909090 0.86363640 *
  3) Reing_EXAC>=0.5 972   0.0000000 1.00000000 *
```

## III. 4. Conclusiones

A la vista de los resultados presentados por los diferentes métodos de clasisifcación y de agrupamiento, se puede decir que el análisis Cluster o por Conglomerado ha sido una buena elección de aplicación, por ser uno de los que mejor representa a los datos cuando se desea realizar particiones de grupos homogéneos entre sí y heterogéneos entre ellos. Para este caso particular, esta técnica ha dado como resultado final óptimo a 3 grupos de perfiles clínicos dierentes entre sí, pero con caracteristicas similares y afines en el interior de ellos, culminando satisfactoriamente con el enfoque inicial de esta investigación, que pretendía buscar patrones de perfiles afines con el fin de clasificar a cada individuo de paciente en distintos grupos que presenten el mismo comportamiento, de tal manera que se pueda estudiar ampliamente sus propiedades o características con la finalidad de mejorar sus procesos clinicos y asistenciales asignados, y por consiguiente poder extrapolar sus resultados y concluisones a la población general.

Para ello, se analizaron diversas técnicas como soporte visual para explorar y describir los datos registrados y en paralelo, se implementaron diferentes métricas con el fin de poder comparar resultados que arrojen luz al objetivo principal, y a la vez, seleccionar la mejor estrategia de uso que proporcione una información de calidad através de distintas técnicas de reducción dimensional sin perder datos relevantes, y asimismo, poder detectar un algoritmo eficiente para la creación de diferentes grupos con el fin de generar de forma correcta y ajsutada un resutlado final con los perfiles clínicos óptimos.

Por otro lado, cabe destacar que los análisis de Correspondencia (CA) y de Decisión (DT) que se han utilizado como parte adicional a la exploración, han proporcionado una complementación visual bastante adecuada como para poder perfilar y ajustar la información originada y en paralelo, se ha podido verificar que el resultado final óptimo de agrupamiento es el acertado para resolver la hipótesis planteada en este estudio.

Dicho esto y por los resultados desarrollados para este caso particular, se puede decir que el análisis por conglomerados o cluster implementado, ha obtenido el número óptimo de clusters con la existencia de varios grupos-perfiles de pacientes con caracteristicas similares, que presentan de forma conjunta variables muy asociadas dentro de cada uno de ellos con suficiente relevancia a considerar en cada perfil clínico obtenido, concretamente se han formado estos tres grupos-perfiles con los siguientes detalles.

- ✓ ***Perfil 1*** agrupados por pacientes fumadores con bastante relevancia en datos espirométricos y probablemente problemas de dieta, de valores tensionales y del ritmo cardíaco, y asimismo estos pueden provocar estancias hospitalarias de larga duración (es decir, estas variables del análisis: *FVCP, ESPIROMETRIA_PA_, HT_, PPKG, DURING, TARS, TARD, FEVCVF_, FCA, FEV1P*), agravando y empeorando el estado del paciente que esta debilitado por otras patologías además de la principal.

- ✓ ***Perfil 2*** agrupados por pacientes hipertensos con espirometria realizada, datos espirométricos variantes, posiblemente requirió la necesidad de soporte ventilatorio por la presencia de varias comorbilidades, entre ellas la del evento celebro vascular, y que esto añadido a los problemas alimentarios por peso y registrar estancias hospitalarias largas pueden originar un desenlace bastante negativo tras un periodo de tiempo corto (es decir, estas variables del análisis: *TARS, TARD, FCA, ESPIROMETRIA_PA_, SV_, FRE, PPKG, IMC, DURING, MUERTOS_90DIAS, ICHARECV_, CCVSDM_*), por lo que se recomienda encarecidamente adoptar hábitos de salud que puedan mejorar su calidad de vida hasta la etapa final de vida, por posible desgaste general originado por distintas patologías además de la central.

- ✓ ***Perfil 3*** agrupados por pacientes bastante graves por presentar varias cormobilidades y patologías (destacando la del evento cerebro vascular y la de insuficiencia cardíaca congestiva), con los valores espirométricos y antropométricos variables debido a los años avanzados del paciente y seguramente es hipertenso y fumador con necesidades de soporte ventilatorio, por presencia de comorbilidades cardiovasculares, (es decir, estas variables del análisis: *FEVCVF_, FEV1P, EDAD, ICHARECV_, ICHARICC_, EV_, CCVSDM_, FVCP, ESPIROMETRIA_PA_, HT_, TARD, FCA, SV_*), lo que indica que su estado esta empeorando rápidamente debido a la patología principal y se debería revisar terapias y otras alternativas para mantener una buena calidad de vida.

No obstante, estos resultados no son concluyentes clínicamente, puesto que se ha detectado que todos los algoritmos tienen algunas limitaciones, y posiblemente se debe completar con otras técnicas específicas si se desea reproducir estos modelos en otros diferentes con el mismo enfoque.

En definitiva, como bien se ha comentado en los resultados, estos grupos-clusters resumidamente presentan caracteristicas muy similares clinicamente e indican que el desarrollo de otras patologías (en término de enfermedades crónicas) a lo largo del tiempo con el aumento irregular de las constantes vitales y el ingreso hospitalario frecuente de este tipo de episodios, vienen ocasionado por la propia

severidad de la enfermedad, por la edad avanzada que presentan estos pacientes unidos a malos hábitos saludables y por la gravedad de las exacerbaciones del propio paciente EPOC, que se encuentra en un cuadro clínico bastante severo ocasionado por el deterioro de su enfermedad en el transcurso del tiempo y dentro de un periodo demasiado prolongado.

En este sentido, con este desarrollo analítico-técnico se pretende mostrar la existencia de estos métodos computacionales de aprendizaje supervisado y no supervisado para apalizar la búsqueda de patrones en conjuntos de datos de alta dimencionalidad que provienen de estudios multicéntricas o de repositorios de Big Data, donde casi siempre es necesario aplicar un procedimiento de clasificación por agrupamient, que pueda aportar un valor añadido a todas esas variables clínicas que son recogidas por los profesionales y que previamente a cualquier algoritmo, se necesita realizar un mecanismo de filtrado bastnate importante para seleccionar aquellas variables o datos más relevantes con la finalidad de dar validez a la toma de decisiones, que muchas veces se precisa de esta información adicional para acertar correctamente en diagnósticos propuestos y en paralelo, poder personalizar procedimientos y tratamientos que no requieran de muchos recursos clínicos para la sociedad actual y poder tener de más tiempo para mejorar los hábitos saludables, que realmente es lo que proporciona una buena calidad de vida para nuestra salud.

# CAPÍTULO IV

# Caso experimental con datos simulados y reales mediante SVM y métodos Kernel

## IV. 1. Introducción

En los últimos años el campo de la Estadística Computacional ha experimentado un gran avance con la aparición de nuevas herramientas estadísticas de estudio, en concreto, algoritmos específicos y modernos adaptados a la necesidad requerida por estos tiempos en el área de la investigación como son la recogida de datos y el análisis de clasificación para las diferentes ramas de la ciencia, en especial, para las ciencias sociales y la biomedicina.

Por este motivo, el planteamiento de esta tesis doctoral también se ha propuesto abordar el problema de la búsqueda de perfiles clínicos en bases de datos procedentes de estudios multicéntricos sobre el contexto de análisis de datos funcionales (ADF), (Muñoz y González, 2010 y González y Muñoz, 2013), permitiendo dar una nueva vía de acceso para situaciones similares en estudios multicéntricos (Ramsay y Silverman, 1997).

Para ello, se pretende introducir una metodología de regularización o penalización de la rugosidad, puesto que se conoce que tiene la ventaja tanto de aproximarse mediante funciones básicas como de técnicas de suavizamiento local, con el fin de obtener las proyecciones del conjunto de datos sobre diferentes implantaciones de los RKHS (Reproducing Kernel Hilbert Spaces), es decir, sobre el Espacio de Hilbert del Núcleo Reproductor, conocidos por RKHS, (Aronszajn, 1950; Berlinet y Thomas-Agnan, 2004 y Cucker y Zhou, 2007), permitiendo a su vez, obtener distintas representaciones sobre los datos clínicos propuestos.

Además, se pretende usar otra técnica multivariante distinta como son los vectores soportes (Hamel, 2009; Muñoz et al., 2019 y Moguerza et al., 2020), más conocidos como SVM (Moguerza y Muñoz, 2006a, 2006b y Moguerza et al., 2007), como medio de análisis de clasificación entre los individuos de la bases de datos de tal forma que se puedan agrupar los distintos participantes según la patología clínica asociada, generando diferentes perfiles de separación.

Como ya se mencionó, estas patologías clínicas pueden ser bastantes dispares dependiendo del la gravedad o del avance de la enfermedad primordial, en este caso la EPOC (*Enfermedad Pulmonar Obstructiva Crónica*), que es la forma más común de nombrarla en el ámbito sanitario (Pozo-Rodríguez et al., 2010).

Con esta idea, se supone que el procedimiento señalado puede mejorar la capacidad de rendimiento de las bases de datos clínicas con alta dimensión, extrayendo de la mejor manera posible los resultados esperados y enfocando de forma más específica, el problema de la búsqueda de perfiles en pacientes con varias características clínicas asociadas por el desarrollo de una patología central.

Por otro lado, en la aplicación práctica, una de las cuestiones más importantes para obtener una buena representación analítica para la clasificación de los datos es la elección de un núcleo apropiado (Martín de Diego et al., 2010; Martos et al., 2014 y Muñoz et al., 2018). En este caso como alternativa de solución, para evitar este tipo de problemas se propone un enfoque de fusión de información donde se consideran varios núcleos para el conjunto de datos estudiados y posteriormente, se combinan todos ellos obteniendo el mejor resultado según los datos facilitados por el algoritmo implementado.

Si es cierto, que esta técnica SVM con *datos simulados* puede apreciar mucho mejor esta representación vectorial al tratarse de datos aleatorios y ajustables al modelo estudiado sin tener en cuenta otros factores externos, que puedan influir sobre la información analizada, y es en este punto donde precisamente recae la dificultad de trabajar con *datos reales*.

Puesto que, si se necesita extrapolar al caso real para la reducción de información con el fin de representar los datos lo mejor posible, mediante un modelo ajustable y adecuado se requiere tener en cuenta otros parámetros que vienen impuestos al aplicar este procedimiento, y que hace que se presenten varios inconvenientes en la práctica, que en ocasiones aumentan el grado de complejidad de los datos (como pueden ser los missings, penalizaciones, costes dispuestos asumir, etc); con la finalidad de poder obtener una buena representación y visualización de dichos datos-variables reales.

Por eso, se han mostrado ambas alternativas, tanto con datos reales como con simulados, con el fin de contrastar resultados finales y verificar los inconvenientes que se pueden encontrar a la hora de implementar el algoritmo sobre datos reales, ya que en muchas ocasiones, se necesita saber que parámetros se requieren ajustar hasta conseguir un modelo ajustable y de buena calidad que pueda mostrar información final fiable y precisa como para extrapolar al ámbito poblacional.

## IV. 2. Métodos

Desde el punto de vista computacional se ha continuado en la investigación de intentar mejorar los resultados presentados, aunque ya se ha obtenido el agrumamiento óptimo buscado para los diferentes perfiles clínicos, mostrando características afines y distintas entre ellos.

Aun así, se ha querido explorar este tema con gran auge y de reciente impacto en el ámbito computacional y de investigación clínica, que puede ayudar a clasificar a los individuos de otra manera diferente, ya que en sus procesos algorítmicos busca un hiperplano de separación entre las instancias de las dos clases mediante un margen máximo, es decir, distancia máxima entre las instancias frontera de la barrera de decisión, dando otro enfoque alternativo a la solución del problema planteado para bases de datos de alta dimensión, que se ha analizado.

Por este motivo, se ha testado el análisis de datos funcional, en especial los SVMs (Quantsignals, 2012; Neto, 2013 y Legorreta, 2015), como otro soporte visual donde en el proceso se indican los máximos puntos destacados mejorando interpretaciones grupales, mediante la selección del kernel apropiado, (que pueden ser de tipo *lineal, radial, polinomial y sigmoidal*), y ajustando por la constante de regularización o de penalización (“*λ”*) para corregir el sobreajuste de los datos que se están analizando. Puesto que, se conoce que es una buena vía de análisis de clasificación entre los individuos de la base de datos y puede ayudar a agrupar los distintos participantes según la patología clínica asociada, cumpliendo y asegurando el objetivo central de este estudio.

Cabe destacar, que es una técnica poco recomendable cuando el dataset presenta un porcentaje alto de missings mostrando datos poco fiables en la salida si no se completa previamente la información. Por este mismo motivo, se ha descartado la aplicación del método de redes neuronales por la complejidad planteada en estos datos que a priori necesitaba un proceso de imputación, que se realizó con el método MICE desarrollado en el capítulo I de este trabajo, y una vez solventado el problema de los valores faltantes, se aplico una reducción dimensional para la exploración de la información y la obtención de los grupos óptimos de datos-pacientes, con la finalidad de conseguir resultados apropiados y fiables para la base de datos clínica presentada.

En este sentido, en el Análisis de Datos Funcional (ADF) como bien se conoce por la literatura (Saxe, 2002 y Ferraty y Vieu, 2006), la información de los datos analizados de cualquier tipo de estudio son tratados como un conjunto de curvas y no como un conjunto de vectores para cualquiera de las ramas de las ciencias objeto de estudio.

En concreto, el análisis de datos funcionales (ADF) es aquella parte de la estadística que trabaja con muestras de funciones aleatorias, donde las medidas de tendencia central, de dispersión y de relación entre variables aplicadas a muestras de variables aleatorias se pueden definir de manera análoga para muestras de datos funcionales.

Ahora, en este caso, simplemente hace falta considerar que se esta trabajando en un espacio vectorial distinto, más concretamente, el espacio $L^2$, donde las funciones poseen cuadrado integrable.

No obstante, las técnicas de ADF más utilizadas son el Análisis en Componentes Principales Funcional (ACPF), las Series Temporales y los Modelos de Regresión Lineal Funcional. Estos últimos modelos presentan distintas variantes dependiendo de si la variable respuesta o la variable explicativa son funcionales o por el contrario, si solo una de ellas lo es (Ramsay y Silverman, 1997).

En la práctica las funciones muestrales son evaluadas en un conjunto finito de puntos que pueden ser desigualmente espaciados y diferentes para los individuos observados.

Por ello, en primer lugar la metodología seguida es reconstruir la verdadera forma funcional de las curvas a partir de sus observaciones discretas, o sea, transformar nuestros puntos evaluados en funciones para poder representar la información objeto de estudio.

Desde el punto de vista de la práctica, uno de los métodos más usados, para aproximar las funciones muestrales consiste en representarlas en términos de bases de funciones y aproximar sus coeficientes mediante interpolación, en el caso de datos observados sin error, o mediante mínimos cuadrados, en el caso de datos con ruido.

Esta metodología proporciona buenas aproximaciones cuando las funciones básicas tienen esencialmente las mismas características que el proceso que genera los datos. En otro caso este método de aproximación no tiene control sobre el grado de suavización de la curva y puede llevar a aproximaciones poco precisas y con resultados bastante pobres.

Con este planteamiento, la aportación que se pretende realizar es un acercamiento más concreto de la información de la base de datos, obteniendo representaciones específicas sobre estos pacientes con el fin de agrupar las diferentes patologías, pudiendo detectar las asociaciones existentes entre ellas, y originando distintos grupos de pacientes, es decir, variantes de perfiles clínicos útiles para la práctica clínica.

Como ya se ha comentado anteriormente, se debe reconstruir la forma funcional de las curvas a partir de sus observaciones discretas con el objetivo de descubrir las estructuras intrínsecas que con una simple representación no sería posible evidenciar. De esta forma, se plantea formalmente el problema de la siguiente manera:

Sea un conjunto de datos funcionales $\{X_1, \ldots, X_n\}$ se tiene que la observación de *n* variables funcionales $X_1, \ldots, X_n$ son idénticamente distribuidas.

Además, se dice que una variable aleatoria X es una variable funcional si toma valores en un espacio funcional E que esta definido como un espacio normado o seminormado completo, (Ferraty y Vieu, 2006).

En general, la representación de un dato funcional en una base ortonormal proporcionará ventajas tanto desde el punto de vista teórico como práctico sirviendo de puente entre el dato funcional y su verdadera forma funcional.

Una base es un conjunto de funciones conocidas e independientes $\{\phi_k\}_{k\in\mathrm{N}}$ tales que cualquier función puede ser aproximada, tan bien como se quiera, mediante una combinación lineal de *K* de ellas con *K* suficientemente grande. De esta forma, la observación funcional puede aproximarse como $x(t) \approx \sum_{k=1}^{K} c_k \phi_k(t)$.

Luego, sea $x_1(t),.x_2(t),..,x_n(t)$ el conjunto de funciones que constituyen la información muestral relacionada con una variable funcional. Se puede considerar que las observaciones pertenecen a un proceso estocástico de la forma: $X = \{X(t) : t \in T\}$ y que además es de segundo orden y las funciones pertenecen al espacio de Hilbert $L^2(T)$ con cuadrado integrable, definiendo el producto interior de la siguiente manera:

$$\langle f, g \rangle = \int_T f(t)g(t)dt, \qquad \forall f, g \in L^2(T)$$

Muchas veces, dada la complejidad de observar las funciones en tiempo continuo, para la práctica se utilizan observaciones de las funciones en un conjunto finito de tiempo diferente para cada individuo.

De este modo la información muestral estaría representada por vectores de la forma: $x_i = (x_{i0},..., x_{im})'$ y se asume que las trayectorias pertenecen a un espacio finito-dimensional generado por la base anteriormente citada $\{\phi_k\}_{k\in\mathrm{N}}$ quedando la observación funcional de la siguiente manera:

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t) \qquad i = 1,..., n$$

Con este planteamiento se pretende abordar los datos propuestos del estudio multicéntrico analizando los datos observados como funciones, donde cada individuo es una función con las diferentes patologías anteriormente nombradas y que se citarán en cada supuesto aplicado.

No obstante, se sabe que la representación puede ser o no bastante buena dependiendo de la calidad de los datos recogidos y de la existencia o no de ruido en la representación de la función.

Desde el punto de vista computacional, en algunos casos será necesario ajustar el modelo mediante una *constante de regularización* ($\lambda$) para que permita corregir el sobreajuste de los datos que se están analizando (Evgeniou et al., 2000).

En la actualidad, este procedimiento esta teniendo un gran impacto en la literatura estadística ya que, el método se utiliza en un conjunto amplio de técnicas de minería de datos, como regresión lineal, regresión logística, SVM, etc.;

La *técnica de regularización* consiste en reducir la importancia de los parámetros $\theta_j$ que aparecen en la función de coste y este efecto se consigue mediante la inclusión de los parámetros $\theta_j$ en la función de coste.

De esta forma, en el caso de regresión lineal la función de coste se ve modificada por la adición de un sumatorio de todos los parámetros $\theta_j$ con un factor llamado parámetro de regularización, $\lambda$.

Por eso, el valor de este parámetro $\lambda$ tiene que ser elegido cuidadosamente, ya que tanto valores elevados como reducidos dan lugar a distorsiones en la función de coste y estas distorsiones pueden ocasionar que se produzca un mal ajuste con los datos manejados.

En definitiva, la expresión del problema a minimizar sería de la siguiente manera:

$$\min_{f \in H} H[f] = \frac{1}{l}\sum_{i=1}^{l}(y_i - f(x_i))^2 + \lambda \| f \|_K^2$$

donde:

$\| f \|_K^2$ es la norma de un RKHS en $H$ definido por la función definida positiva $K$.

$l$ es el número de puntos de datos, es decir, los pares $(x_i, y_i)$ de $l$.

$\lambda$ es el parámetro de regularización.

Por tanto, con esta corrección introducida por el parámetro de regularización se puede conseguir el mejor modelo ajustado para los datos, en concreto para los datos de este estudio que se están analizando en cada uno de los casos presentados para la exploración de las clases con el hábito tabáquico, el estado de exitus, la realización de espirometría y varias patologías relevantes.

## IV. 3. Resultados

Con esta misma idea mencionada, se ilustra el problema planteado utilizando el *software R* (R Core Team, 2021) como medio de análisis estadístico, más específicamente los paquetes *e1071* (Meyer et al., 2021), *kernlab* (Karatzoglou et al., 2004) y *fda* (Ramsay et al., 2020), con el fin de realizar un acercamiento más concreto a los datos estudiados.

Y por consiguiente, buscar representaciones más específicas sobre estos pacientes que ayuden al agrupamiento de las diferentes patologías, detectando la existencia de asociaciones entre ellas, pudiendo formar distintas variantes de perfiles clínicos útiles para la práctica.

Dicho esto, para el análisis de los datos, las variables que se incorporan en el estudio de clasificación son las citadas anteriormente en el apartado de introducción, ya que tienen bastante relación con la

enfermedad principal y van apareciendo en el paciente conforme avanza el estado de gravedad del mismo, concretamente, las patologías asociadas al paciente que se desean explorar en este estudio son: (i) *Insuficiencia Cardíaca Congestiva*; (ii) *Comorbilidad Cardiovascular*; (iii) *Diabetes Mellitus*; (iv) *Enfermedad Cerebro Vascular*; (v) *Enfermedad Vascular Periférica*; (vi) *Infarto de Miocardio*; (vii) *Nefropatía*; (viii) *Tumor sólido* y (ix) *Edemas Maleolares*.

En este sentido, a continuación se detallan los procedimientos utilizados en cada tipo de datos (simulados y reales), destacando la gran ventaja que se dispone cuando se trabaja con los datos simulados (ejemplo con una muestra de entrenamiento n=100), y que no se puede apreciar cuando se exploran datos reales (en este caso una muestra final de entrenamiento n=5178 para mantener la representatividad del caso clínico), puesto que en la práctica suelen aparecer múltiples inconvenientes relacionados con los temas de ajustes y adaptabilidad a los distintos modelos predefinidos.

Pero es cierto, que ambas vías analíticas aportan un valor diferencial y adicional a este análisis de clasificación aplicado con el método SVM para el soporte de la toma de decisiones sin perder información relevante, y pudiendo encontrar una buena representación visual de los datos para la generación de diferentes grupos de patologías que presenten características similares.

### *IV. 3. 1. Análisis con datos simulados*

En primer lugar, desde el punto de vista computacional, el procedimiento de datos simulados se inicia primero con la creación de las dos variables "*x*" e "*y*" mediante distribuciones aleatorias normales "*rnorm ()*" cuyos parámetros son el tamaño muestral (*n*) definido como la cantidad de números que desea generar y los argumentos estándares que son la media (*mean*) y la desviación típica (*sd*). Después, se sigue con la construcción de la variable clasificatoria "*clases*" que se realiza como una réplica de ellas para definir las categorías. En este caso, se usa un *kernel* del tipo lineal y se utilizan distintos *costes* (1, 10 y 0.1).

Por otro lado, para el caso del hiperplano, se define primero la función del problema a minimizar como "*kfunction ()*" que anteriormente se menciono y está se utiliza como *kernel*, y por último, se aplica la propia función "*ksvm"* para obtener los vectores soportes en el hiperplano.

A continuación, se muestra la representación gráfica obtenida tras aplicar la sintaxis del método *SVM* y perfeccionado con el paquete *kernlab* (Karatzoglou et al., 2004) para los datos simulados con el objetivo de poder comparar ambas salidas (reales y simulados).

A simple vista, se puede apreciar que los resultados son muy diferentes, teniendo grandes dificultades cuando se trabaja con datos reales. No obstante, en los datos simulados no se pueden apreciar estos problemas que a lo mejor pueden ayudar a mejorar el modelo seleccionado y por tanto, las conclusiones podrían ser algo más rápidas enfocando el problema para encontrar el ajuste ideal de los datos analizados.

Por los resultados de este caso presentado, se puede observar que con los datos simulados cuando el *coste* es 1 ò 10 se obtiene el mismo número de vectores soportes (es decir, los SVM son 3) como se indica en la *Figura 24*, pero cuando se disminuye el *coste* del modelo a 0.1 se puede ver que se obtiene el doble de vectores soportes (o sea, ahora los SVM son 6) que con un coste alto, *Figura 25*, por lo que cuando el coste es más bajo se puede tener más vectores soportes que representen al modelo estudiado.

En este aspecto, el aumentar o disminuir el coste del modelo dependerá en muchos casos de lo que se desea analizar y la influencia que presenta la variable ante los datos representados.

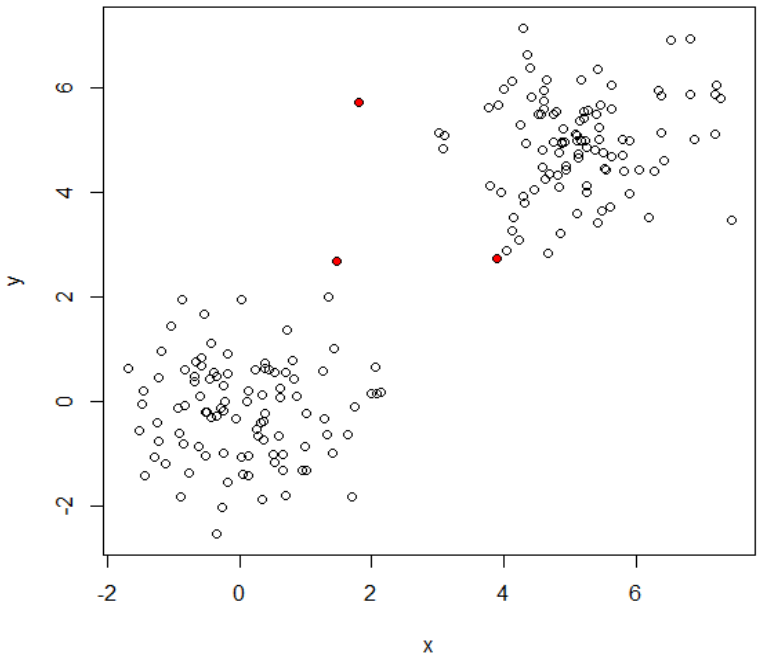



**Figura 24.** Simulado con coste 1 ó 10 – SVM

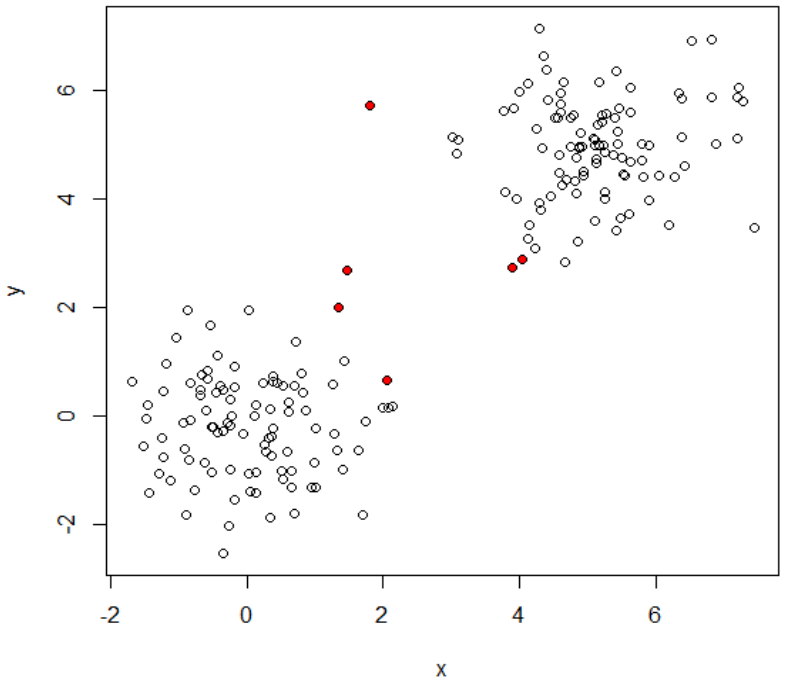



**Figura 25.** Simulado con coste 0.1 – SVM

A continuación, se muestra en la *Figura 26* como se predispone la información de los datos en el hiperplano en el modelo de *coste estándar* (es decir, C=1) que anteriormente se ha presentado en la *Figura 24*, donde los nuevos vectores soportes son tres con identificación etiquetada (es decir, son los individuos 1, 17 y 16).

Dicho esto, en esta representación gráfica se puede reflejar el hiperplano del modelo con la información de los vectores soportes obtenidos mediante los datos simulados con un coste estándar (C=1).

Por lo que, a la vista de los resultados de esta última implementación con este tipo de gráfico, se puede apreciar una mejora visual en la representación de los vectores soportes, que han sido seleccionados dando un aspecto más sencillo, y mejorando la interpretación de la toma de decisiones a la hora de elegir la opción más correcta sobre los datos analizados.

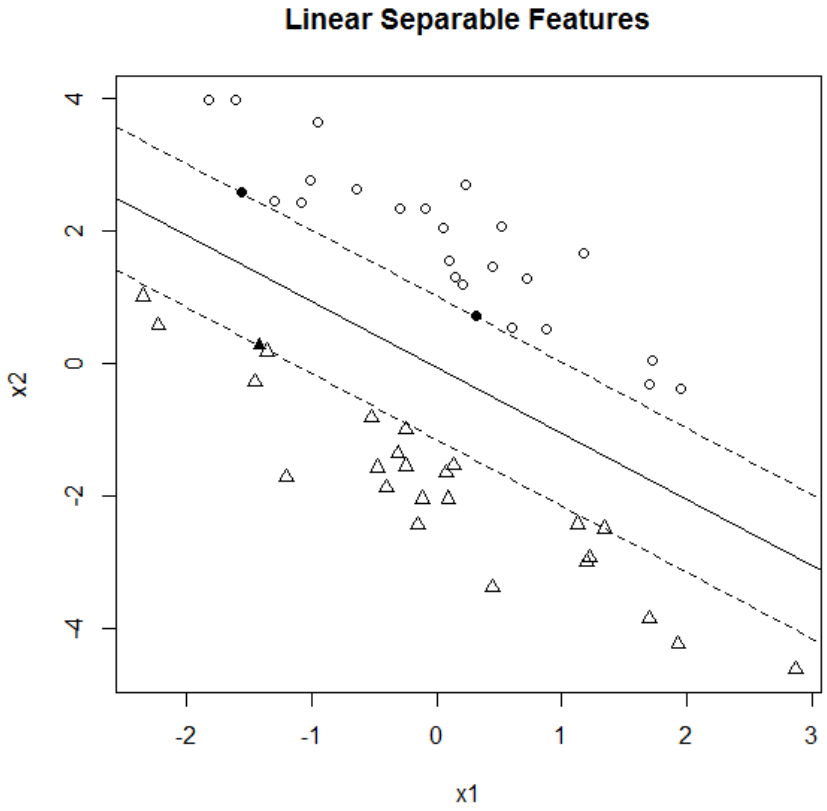



**Figura 26.** Simulado en el hiperplano con coste 1 – SVM

Obviamente, en este caso simulado con una muestra de datos pequeña (n=100) es fácil detectar los cambios cuando se esta explorando el modelo diseñado con los parámetros definidos por el propio usuario, y por consiguiente, los vectores soportes son de fácil apreciación a la vista como muestra la *Figura 26*, puesto que es un algoritmos diseñado a priori para mostrar que esta técnica funciona correctamente, siendo bastante útil si se desea destacar información relevante para clasificar a diferentes individuos de una clase definida, resumiendo los datos en el menor número posible de vectores soporte sin perder información esencial de lo que se necesita analizar y sacando el mayor potencia a los datos para dar el mejor resultado final.

Además, destacar que el paquete *kernlab* (Karatzoglou et al., 2004) tiene disponible la opción de un optimizador alternativo que puede utilizarse para la solución de problemas de regresión, clasificación y de clase única, si se precisa ampliar en esta vía exploratoria (Moguerza et al., 2020).

### *IV. 3. 2. Análisis con datos reales*

Por otro lado, para el análisis de los datos reales también se ha utilizado un *kernel* de tipo *lineal* como iniciación del análisis exploratorio, dejando el valor por defecto de la *constante de coste* (*C=1),* ya que se ha observado que se puede conseguir de esta forma un buen número de vectores soportes (o sea, un valor menor de estos), que representan casi perfectamente a las variables de estudio, auqnue también se ha variado la constante para ver los diferentes resultados de contraste con el fin de seleccionar el mejor valor de coste.

Puesto que, como se ha mencionado anteriormente, este parámetro de coste puede tomar diferentes valores, ampliándolo hasta *C*=10 y abarcando en la representación gráfica el máximo número de vectores soportes, aunque también se puede reducir hasta un coste de 0.1, disminuyendo el número de vectores soportes para el análisis.

En algunos casos, se podrá observar que es necesario incluir una *constante de regularización* o penalización (λ) para mejorar el ajuste del modelo cuya finalidad mediante esta técnica es reducir el sobreajuste de los datos que se están analizando.

Dicho esto, la decisión de este parámetro *C*, que es la constante del término de regularización en la formulación de Lagrange, dependerá de cual sea la influencia de la variable o el problema real que se desea mostrar en los resultados finales. En este caso particular, se puede apreciar que bajando el *coste* a *0.1*, aunque se dispone de un menor número de vectores soportes se tiene una mejor clasificación respecto a la relación de las variables.

También, se ha empleado el método de *validación cruzada* (*cross=2*) para estimar la probabilidad de clasificar erróneamente una observación dividiendo la muestra en dos partes. Y asimismo, con la finalidad de usar datos *no estandarizados,* se ha incluido la sentencia *scale=False* ya que por defecto, la técnica muestra los estandarizados.

A continuación, con este planteamiento y tras implementar la sintaxis sobre los datos clínicos de pacientes reales, se refleja en cada apartado los resultados obtenido de este estudio particular, donde ya se conocía que los pacientes presentaban una media de 75 años, (aunque la edad de participación oscila entre 31 y 99 años, de aquí la gravedad avanzada), con un alto porcentaje predominante de hombres, y donde la mediana de duración por ingreso hospitalario es de 8 días por paciente, siendo el tiempo mínimo de estancia de 1 día y de 130 días como periodo más largo de ingreso clínico.

En este mismo sentido, el procedimiento de análisis para cada caso es presentar como se comporta la edad "*x1*" con la duración de los ingresos hospitalarios en días "*x2*" para reflejar la relación existente entre ambas variables respecto al avance de la enfermedad principal, según la variable "*clase*" utilizada y definida como los diferentes casos descritos con las distintas patologías.

Hay que destacar, que para estos resultados alcanzados el número total de entrenamiento es alto (n=5178) con el fin de mantener la representatividad de los casos del dataset original, aunque inicialmente se dividió en 70% entrenamiento "*Train*" (n=3625) y 30% prueba "*Test*" (n=1553), pero se observó que existía una reducción en el porcentaje de casos analizados perdiendo la esencia del estudio clínico a explorar. Por ello, se mantuvo la muestra completa para el "training", donde esto posiblemente pueda afectar a la capacidad de clasificación o tasa de aciertos, no siendo está demasiado buena en determinados casos. Dicho esto, si se desea extender este estudio con esta alternativa de análisis clasificatorio con SVM, seguramente será necesario optar por otros kernels más complejos, por la optimización de los parámetros mediante *validación cruzada* usando la función genérica "*tune ()*" del paquete *e1071* de R (Meyer et al., 2021), o seguir ajustando con la penalización para mejorar el modelo.

*1. Fumadores y no fumadores*

A continuación, se realiza un estudio entre *fumadores y no fumadores* con la variable hábito tabáquico "*HT_*" para averiguar si existe algún tipo de influencia respecto a las variables: *edad y duración de ingreso hospitalario*.

Para ello, se desarrolla la sintaxis y se realizan las verificaciones pertinentes en cada ítem para la mejor elección de los parámetros, como son la del *Kernel* y la constante de coste "*C"*, con el fin de obtener el menor número de SVMs, presentando resultados óptimos. En paralelo, los resultados de este análisis requieren de la introducción de una *constante de regularización* para corregir el ajuste del modelo, donde el valor de la constante inicialmente será 1 (*coef0=1*) y luego se realizan otras variaciones de esta constante para su ajuste.

Además, para este caso particular los resultados indican que no es ajustable mediante un núcleo lineal, puesto que la clasificación de las clases no es muy buena, como bien muestra la tabla de clasificación que indica solo una de las clases y no las dos. Por este motivo, se intenta modificar la constante de regularización mediante varios valores diferentes (0.1, 0.3, 0.5, 1, 10, 50 y 100) y la función coste (0.1, 1, 10 y 100), obteniendo nuevos resultados, que probablemente confirman que para este caso el kernel no es lineal y se debe modificar a otro tipo diferente de núcleo (radial, polinomial y sigmoidal).

Es decir, al obtener los mismos resultados en este caso particular tras los ajustes de los parámetros, se confirma que los datos no son lineales por lo que se debe modificar el kernel aplicado, puesto que la tabla de clasificación muestra valores de clase incompletos y un porcentaje de acierto bajísimo, lo que hace pensar que los resultados no se ajustan al tipo de kernel lineal.

Por ello, se modifica y se comparan otros tipos de kernels (*radial, polinomial* y *sigmoidal*) con el fin de obtener el que mejor se ajusta al caso expuesto, ya que anteriormente se ha confirmado que el kernel lineal no es ajustable al caso. En este sentido, se utilizan los mismos costes definidos (*0.1, 1.0* y *10*), con la finalidad de buscar el mejor modelo que represente este ítem clínico mostrado.

A la vista de los resultados, se tiene que con respecto a los diferentes kernels, se tiene que si se utiliza el kernel "radial" el mejor es con coste=10, dando una buena clasificación y un porcentaje alto de aciertos; el kernel "polinomial" es mejor con coste=0.1 con una clasificación bastante aceptable que presenta un porcentaje alto aunque también se podría seleccionar el de coste=10 con un porcentaje medio y con un grado de concordancia (o sea, coeficiente Kappa) negativo, por lo que la relación es inversa.

Asimismo, si se selecciona el kernel "sigmoid" se tiene que ninguno de los costes es el mejor y se introduce una constante de regularización con valor 1 (*coef0=1*) para corregir el ajuste del modelo sigmoidal, pero los resultados muestran que tampoco se ha conseguido mejorarlo al añadirla.

En resumen, a continuación en la *Tabla 9* se disponen los resultados obtenidos para una mejor lectura del caso clínico de *fumadores y no fumadores*.

**Tabla 9.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 1

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 1827 (*) | 1833 (*) | 1828 (*) | 1827 (*) | 1833 (*) | 1828 (*) |
| Radial | 2331 (*) | 2396 (83,37%) | 2327 (84,72%) | - | - | - |
| Polynomial | 1464 (78,43%) | 1458 (*) | 1473 (40,46%) | - | - | - |
| Sigmoid | 1812 (*) | 1812 (*) | 1812 (*) | 1812 (*) | 1812 (*) | 1812 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En definitiva, a la vista de los resultados se puede comprobar que el modelo de kernel **radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por hábito tabáquico), consiguiendo un número de vectores soportes (**2327**) adecuados para el estudio con una tasa de clasificación del **84,72%** y con un coste del 10.

No obstante, se aprecia que aparentemente el modelo de kernel **polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**1464**) menor, pero manteniendo una representación algo más baja en ambos niveles (**78,43%**) según el tipo de coste seleccionado (C=0.1).

Al igual que en el caso anterior, ambos modelos de kernels pueden ser empleados para futuros análisis, pero *el mejor es el modelo de tipo radial con coste 10 dando 2327 SVMs,* como bien se puede visualizar en la *Figura 27.*

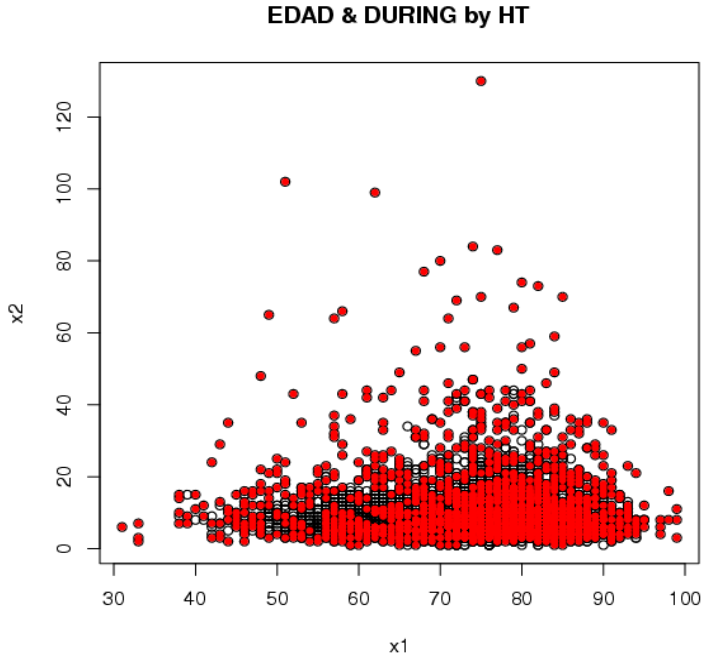



**Figura 27.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by HT – Caso 1

Con la salida de los resultados obtenidos, se puede visualizar que *la variable hábitos tabáquico tiene bastante relación con la edad y por consiguiente con el tiempo de ingreso en el centro hospitalario*, es decir, que el ser fumador es una de las características primordiales en el avance de la enfermedad junto a otras patologías como ya se anunciaba desde el inicio del estudio.

Además, en este mismo caso (*fumadores y no fumadores)*, también se ha verificado que muchas veces disminuyendo el *coste a 0.1* no siempre se mejora la representación de los datos del modelo, puesto que en este mismo ítem, cuando se aumenta el coste se percibe una mejora en el ajuste del modelo disminuyendo el número de vectores soportes.

En esta misma línea, y a la vista de los resultados de la tabla expuesta anteriormente (*Tabla 9*), se intenta mejorar la salida de los análisis obetenidos, ***cambiando los parámetros internos*** para obtener un mejor ajuste del modelo para este caso (*fumadores y no fumadores*), ya que los resultados no son lo bastante buenos como para dar como definitivo al modelo final de hábito tabáquico (fumador vs no fumador).

Dicho esto, se aplica el mismo procedimento con la selección del **Kernel=*Radial*** e introduciendo el parámetro sigma con diferentes valores (0.05; 0.1; 1; 10) para cada coste C (0.1; 1; 10) y el parámetro lambda que es "1/C", siendo sus valores distintos (10; 1; 0.1) para cada sigma y coste, con el fin de poder mejorar aun más el modelo si es posible, obteniendo los siguientes resultados de la *Tabla 10*, que indican que no se ha podido mejorar el modelo radial al añadir los diferentes parámetros sigma y lambda por cada coste C.

**Tabla 10.** SVMs con tasa de clasificación mediante la selección del Kernel Radial mejorando ajuste

| KERNEL | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| Radial | C=0.1 | C=1 | C=10 | Lambda=0.1 | Lambda=1 | Lambda=10 |
| sigma=0.05 | 2331 (*) | 2396 (83,37%) | 2327 (84,72%) | 2327 (84,72%) | 2396 (83,37%) | 2331 (*) |
| sigma=0.1 | 2331 (*) | 2396 (83,37%) | 2327 (84,72%) | 2327 (84,72%) | 2396 (83,37%) | 2331 (*) |
| sigma=1 | 2331 (*) | 2396 (83,37%) | 2327 (84,72%) | 2327 (84,72%) | 2396 (83,37%) | 2331 (*) |
| sigma=10 | 2331 (*) | 2396 (83,37%) | 2327 (84,72%) | 2327 (84,72%) | 2396 (83,37%) | 2331 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

Asimismo, se implementa el mismo proceso con la selección del **Kernel=*Polynomial***, y se añade el parámetro grado con distintos valores (degree=0; 1; 2; 3) para cada coste C (0.1; 1; 10), con la finalidad de mejorar el modelo si es posible, obteniendo los siguientes resultados de la *Tabla 11*, que a la vista de los datos parece que no se ha podido mejorar el modelo polynomial al disminuir el grado del polinomio por lo que el mejor modelo sería el de grado 3 y coste 0.1 como se puede apreciar en la tabla.

**Tabla 11.** SVMs con tasa de clasificación mediante la selección del Kernel Polynomial mejorando ajuste

| KERNEL | SVM (Tasa clasificación) | | |
|---|---|---|---|
| Polynomial | C=0.1 | C=1 | C=10 |
| Degree=0 | 1812 (*) | 1812 (*) | 1812 (*) |
| Degree=1 | 1826 (*) | 1828 (*) | 1828 (*) |
| Degree=2 | 1570 (*) | 1449 (*) | 1403 (*) |
| Degree=3 | 1464 (78,43%) | 1458 (*) | 1473 (40,46%) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros.

Y finalmente, también se realizo el mismo mecanismo con la selección del **Kernel=*Sigmoid***, introduciendo el parámetro gamma con diferentes valores (0.5; 0.1; 0.001; 1) para cada coste C (0.1; 1; 10), con el fin de mejorar el modelo si es posible ya que con la corrección de ajuste (es decir, la constante de regularización *coef0*) no se pudo mejorar en casi nada al modelo.

Dicho esto, a la vista de los resultados de la *Tabla 12* parece que tampoco se ha podido mejorar el modelo sigmoidal al disminuir o aumentar el parámetro gamma. Y asimismo ni mejoro con el ajuste de corrección al modelo por lo que se puede descartar esta función para estos datos porque no son ajustables al tipo de kernel sigmoidal.

**Tabla 12.** SVMs con tasa de clasificación mediante la selección del Kernel Sigmoid mejorando ajuste

| KERNEL | SVM (Tasa clasificación) | | |
|---|---|---|---|
| Sigmoid | C=0.1 | C=1 | C=10 |
| Gamma=0.001 | 1812 (*) | 1812 (*) | 1812 (*) |
| Gamma=0.1 | 1812 (*) | 1812 (*) | 1812 (*) |
| Gamma=0.5 | 1812 (*) | 1812 (*) | 1812 (*) |
| Gamma=1 | 1812 (*) | 1812 (*) | 1812 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros.

*2. Exitus*

Siguiendo la misma dinámica del caso anterior, para este segundo caso explorado se presenta la relación de la edad del paciente y el tiempo de ingreso por la variable de evento "*EXITUS"*, observando si se confirma la existencia de asociación entre ambas variables.

Esta variable "EXITUS" indica si el paciente continúa vivo o fallecido en el transcurso de todo el periodo de ingreso que estuvo en el hospital por alguna complicación de la enfermedad que padecía o de las que se le añadieron en el avance de la patología principal, agravando el estado del paciente hasta su fase final.

Para ello, se desarrolla la misma sintaxis y se realizan las verificaciones pertinentes en cada punto para la mejor elección de los parámetros, (*Kernel*, coste "C" y constante de regularización "*coef0"*) , con el fin de obtener el menor número de SVMs, presentando resultados óptimos.

También, para este caso los resultados muestran que los datos no son lineales, por lo que se debe modificar el kernel aplicado, puesto que la tabla de clasificación señala un porcentaje de aciertos muy escaso, lo que hace pensar que los resultados no se ajustan al tipo de kernel lineal.

Por ello, se aplican los diferentes tipos de kernels (*radial, polinomial* y *sigmoidal*) con el fin de obtener el que mejor se ajusta a este caso, ya que como se ha visto en la *Tabla 13* el kernel lineal no se adapta correctamente.

A la vista de los resultados obtenidos, se tiene que con respecto a los diferentes kernels, el kernel "radial" es el mejor con coste=10, dando una buena clasificación y un porcentaje alto de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable que presenta un porcentaje alto aunque también se podría seleccionar el de coste=0.1 con un porcentaje bueno, muy similar al de coste 10.

Asimismo, si se selecciona el kernel "sigmoid" se tiene que ninguno de los costes es el mejor y se introduce la constante de regularización o de penalización con valor 1 (*coef0=1*) para corregir el ajuste del modelo sigmoidal, pero los resultados muestran que tampoco se ha conseguido mejorarlo.

En resumen, a continuación en la *Tabla 13* se disponen los resultados obtenidos para una mejor lectura del caso clínico *EXITUS* (vivo o muerto).

**Tabla 13.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 2

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 528 (*) | 526 (*) | 529 (*) | 528 (*) | 526 (*) | 529 (*) |
| Radial | 1051 (*) | 1290 (95,29%) | 1275 (96,14%) | - | - | - |
| Polynomial | 493 (94,63%) | 489 (*) | 470 (94,73%) | - | - | - |
| Sigmoid | 518 (*) | 518 (*) | 518 (*) | 518 (*) | 518 (*) | 518 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En definitiva, a la vista de los resultados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del participante y duración de ingreso hospitalario por EXITUS), consiguiendo un número de vectores soportes (**1275**) satisfactorio para el análisis con una tasa de clasificación del **96,14%** y con un coste del 10.

Sin embargo, también se aprecia que el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**470**) bastante menor, pero manteniendo una representación algo más baja en ambos niveles (**94,73%**) según el tipo de coste seleccionado (C=10).

Al igual que en el caso anterior, ambos modelos de kernels pueden ser empleados para futuros análisis con la finalidad de mejorar la interpretación clínica de los resultados, pero *el mejor es el modelo de tipo radial con coste 10 dando 1275 SVMs,* como bien se puede visualizar en la *Figura 28.*

A la vista de los resultados, se puede ver que la relación de las variables (*duración de ingreso y edad por exitus*) es bastante buena, ya que la *tasa de clasificación es del 96,14%* entre ambos niveles y el *número de vectores* es *1275* con un coste de *C=10*.

Además, es normal que los gráficos no sean tan claros visualmente porque el número de pacientes en estudio (n=5178) es demasiado grande. No obstante, sería bueno pensar que si se desea analizar mejor esta relación o cualquiera de las que se presenten en este estudio, simplemente habría que seleccionar los *datos elegidos por los SVMs* y tomarlos como una s*ubmuestra* de análisis donde los resultados que se obtengan pueden ser *extrapolados a la muestra general*, ya que son los vectores soportes más relevantes y los que mejor representan a la muestra de estudio.

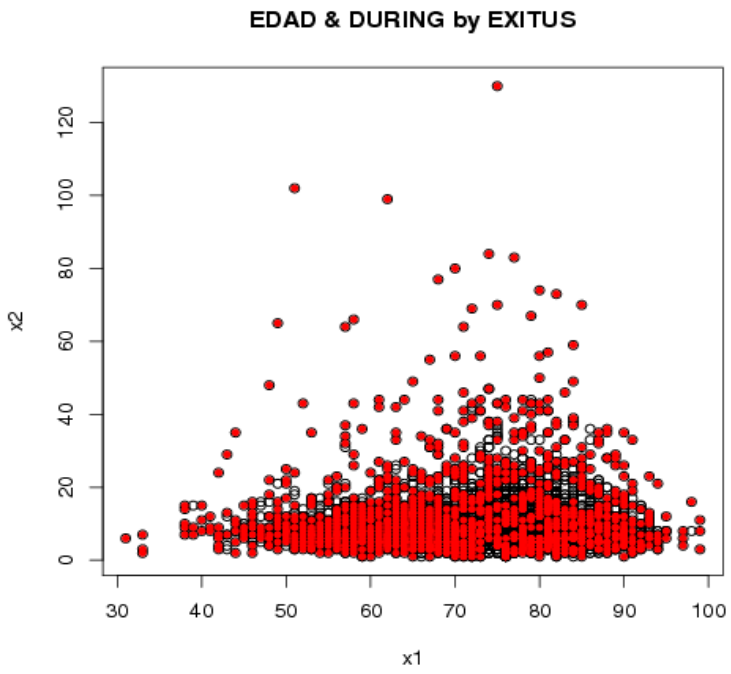



**Figura 28.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by EXITUS – Caso 2

*3. Prueba de espirometría realizada y no realizada*

Para este otro caso, se quiere detectar si al paciente se le realizó la prueba de espirometría "*ESPIROMETRIA_PA_"* mediante la relación existente entre la variable de duración hospitalaria y la edad, donde se desea comprobar si esta prueba se le realizo o no al paciente en algún momento del tiempo para medir y valorar el grado de avance de la enfermedad o para descartar el diagnóstico de presencia de la misma en el individuo objeto de estudio.

Para ello, se implementa la sintaxis nuevamente con las verificaciones específicas para la mejor elección del *Kernel*, de la constante *C* definidos con los distintos valores (*0.1, 1.0* y *10*), y del parámetro de regularización para corregir el ajuste del modelo, siendo el valor de la constante 1 (*coef0=1*), todo ello con el fin de obtener el menor número de SVMs, presentando los siguientes resultados en la *Tabla 14*.

Asimismo, para este caso los resultados muestran que los datos no son lineales, por lo que se debe modificar el kernel aplicado, puesto que la tabla de clasificación señala un porcentaje de aciertos muy baja, y tampoco se ha mejorado con la introducción de la constante de regularización, lo que hace pensar que los resultados no se ajustan al tipo de kernel lineal.

En este sentido, se aplican los diferentes kernels (*radial, polinomial* y *sigmoidal*) para obtener el mejor ajusta en este caso, ya que como se ha visto en la tabla el kernel lineal no se adapta correctamente.

Dicho esto, a la vista de los resultados obtenidos, se tiene que el kernel "radial" es el mejor con coste=10, dando una buena clasificación y un porcentaje alto de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable que presenta un porcentaje alto, aunque también se podría seleccionar el de coste=1 con un porcentaje bueno, muy similar al otro (C=10) y se descarta el coste=0.1 por tener un porcentaje muy bajo, aunque presente un número menor de SVMs.

Asimismo, con el kernel "sigmoid" se tiene que ninguno de los costes es el mejor y se introduce la constante de regularización con valor 1 (*coef0=1*) para corregir el ajuste del modelo sigmoidal, pero los resultados muestran que tampoco se ha conseguido mejorarlo.

En resumen, a continuación en la *Tabla 14* se disponen los resultados obtenidos para una mejor lectura del caso clínico *ESPIROMETRIA*.

**Tabla 14.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 3

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 3323 (*) | 3320 (*) | 3326 (*) | 3323 (*) | 3320 (*) | 3326 (*) |
| Radial | 3645 (*) | 3598 (71,80%) | 3470 (73,66%) | - | - | - |
| Polynomial | 2177 (37,29%) | 2195 (68,10%) | 2134 (68,11%) | - | - | - |
| Sigmoid | 3288 (*) | 3288 (*) | 3288 (*) | 3288 (*) | 3288 (*) | 3288 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

De este mismo modo, por los resultados obtenidos se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por espirometría realizada), consiguiendo un número de vectores soportes (**3470**) adecuados para el estudio con una tasa de clasificación del **73,66%** y con un coste del 10.

Analogamente, se aprecia que aparentemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**2134**) menor, pero manteniendo una representación algo más baja en ambos niveles (**68,11%**) según el tipo de coste seleccionado (C=10).

Al igual que en el caso anterior, ambos modelos de kernels pueden ser empleados para futuros análisis, pero *el mejor es el modelo de tipo radial con coste 10 dando 3470 SVMs,* como bien se puede visualizar en la *Figura 29.*

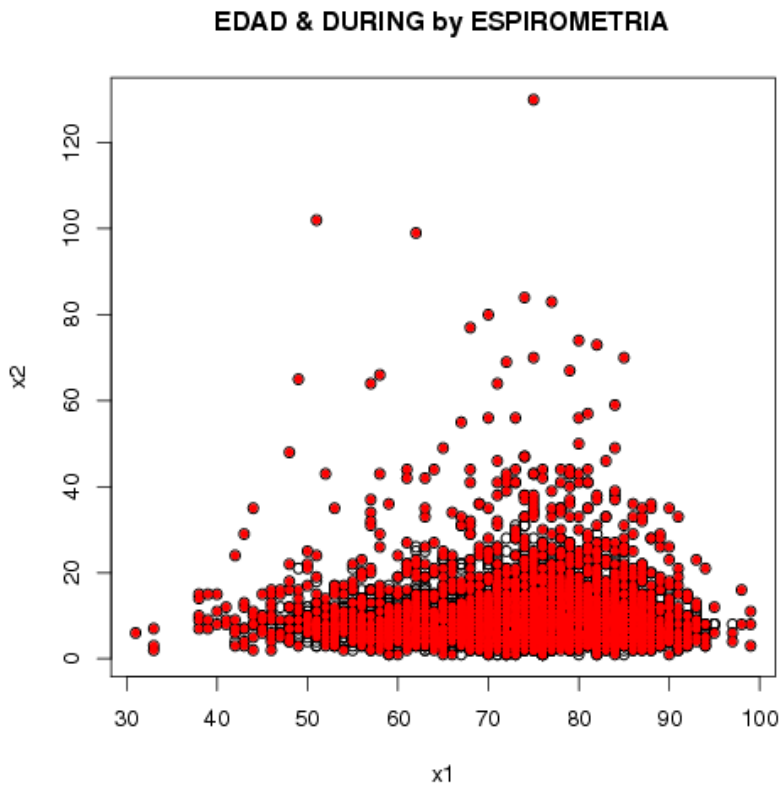



**Figura 29.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ESPIROMETRIA – Caso 3

A la vista de los resultados, se puede ver que la relación de las variables (*duración de ingreso y edad por espirometría realizada*) es bastante buena, ya que la *tasa de clasificación es del 73,66%* entre ambos niveles y el *número de vectores* es *3470* con un coste de *C=10*.

Además, como se ha mencionado anteriormente se puede tomar la selección con los *datos elegidos por los SVMs* para mejorar este análisis o cualquier otro de los presentados, tomándolos como una s*ubmuestra* de análisis donde los resultados que se obtengan pueden ser *extrapolados a la población general*, puesto que son los vectores soportes más relevantes y que mejor representan a la muestra de estudio.

### *4. Patologías detectadas en el participante*

A continuación, se exploran algunas de las patologías detectadas en el paciente a lo largo de su seguimiento clínico, siendo los siguientes eventos los que están más asociadas a la enfermedad principal, intentado explorar cada caso por separado con el fin de obtener alguna conclusión relevante, ayudando a completar los perfiles óptimos alcanzados y por consiguiente, valorar esta vía clasificatoria como otra alternativa más para la toma de decisiones.

En este sentido, se pretende observar como afectan cada una de estas patologías respecto al ingreso hospitalario y a la edad del individuo, valorando si la estancia y la edad avanzada han influido en la determinación o no de la presencia de la enfermedad, o de lo contrario, estas patologías vienen directamente influenciadas por la gravedad y el avance de la enfermedad principal.

Para ello, se aplica el procedimiento mencionado anteriormente, verificando y ajustando los parámetros específicos con el fin de obtener la mejor elección del Kernel, de la constante C y del argumento de regularización para corregir el ajsute del modelo y por consiguiente, obtener el menor número de SVMs, presentando los resultados finales en cada una de las tablas indicadas de forma resumida.

#### *4.1. Insuficiencia cardíaca congestiva*

Por ello, tras implementar los cambios oportunos, se tiene que para este caso los resultados muestran que los datos no se ajustan al tipo de kernel lineal, por lo que se comprueban otros kernels diferentes (*radial, polinomial* y *sigmoidal*) con el fin de mostrar el mejor ajuste.

Dicho esto, a la vista de los resultados obtenidos, se tiene que el kernel "radial" es el mejor con coste=10, dando una buena clasificación y un porcentaje alto de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable que presenta un porcentaje alto, aunque también se podría seleccionar el de coste=1 con un porcentaje bueno, muy similar al del coste C=10, y se descarta el coste=0.1 por tener un porcentaje muy bajo, aunque presenta un número menor de SVMs.

Respecto, al kernel "sigmoid" se tiene que ninguno de los costes es el mejor, ni con la modificación de la constante de regularización (*coef0=1*) para corregir el ajuste del modelo, se ha podido mejorar los resultados.

En resumen, en la *Tabla 15* se disponen los resultados obtenidos para una mejor lectura del caso clínico *INSUFICIENCIA CARDIACA CONGESTIVA.*

**Tabla 15.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.1

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 2276 (*) | 2317 (*) | 2380 (*) | 2276 (*) | 2317 (*) | 2380 (*) |
| Radial | 2775 (*) | 2846 (79,22%) | 2746 (80,82%) | - | - | - |
| Polynomial | 1766 (69,10%) | 1740 (73,31%) | 1774 (74,93%) | - | - | - |
| Sigmoid | 2240 (*) | 2240 (*) | 2240 (*) | 2240 (*) | 2240 (*) | 2240 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En este sentido, por los resultados obtenidos se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Insuficiencia Cardíaca Congestiva), consiguiendo un número de vectores soportes (**2746**) adecuados para el estudio con una tasa de clasificación del **80,82%** y con un coste del 10.

Del mismo modo, se aprecia que supuestamente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**1774**) menor, pero manteniendo una representación algo más baja en ambos niveles (**74,93%**) según el tipo de coste seleccionado (C=10).

Por tanto, similar al caso anterior, ambos modelos de kernels pueden ser empleados para futuros análisis, pero el *mejor es el modelo de tipo radial con coste 10 dando 2746 SVMs,* como bien se puede visualizar en la *Figura 30.*

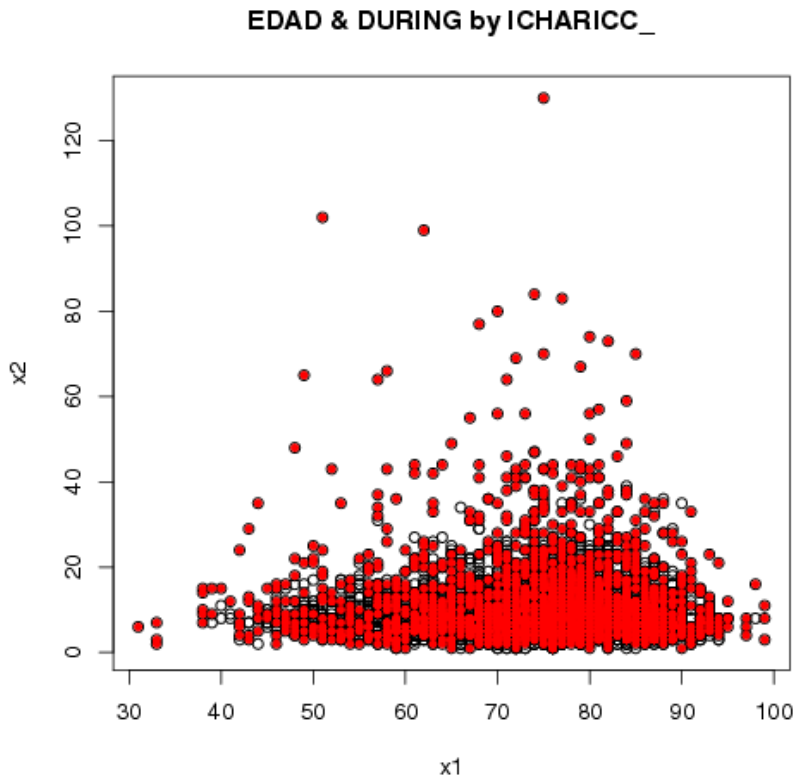



**Figura 30.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ICHARICC – Caso 4.1

*4.2. Comorbilidad cardiovascular*

Asimismo, para este otro caso los resultados muestran que los datos no se ajustan al tipo de kernel lineal y se aplican los diferentes kernels (*radial, polinomial* y *sigmoidal*) para mostrar el mejor ajuste.

Dicho esto, los resultados obtenidos muestran que el kernel "radial" es el mejor con coste=10, dando una buena clasificación y un porcentaje alto de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable que presenta un porcentaje medio de aciertos; y el kernel "sigmoid" se tiene que ninguno de los costes es el mejor, ni mejora los resultados con la modificación de la constante de regularización.

A continuación, en la *Tabla 16* se disponen los resultados obtenidos para una mejor lectura del caso clínico *COMORBILIDAD CARDIOVASCULAR.*

**Tabla 16.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.2

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 4478 (*) | 4505 (*) | 4483 (*) | 4478 (*) | 4505 (*) | 4483 (*) |
| Radial | 4636 (58,34%) | 4478 (65,28%) | 4271 (67,81%) | - | - | - |
| Polynomial | 2445 (55,35%) | 2546 (57,42%) | 2499 (58,21%) | - | - | - |
| Sigmoid | 4434 (*) | 4434 (*) | 4434 (*) | 4434 (*) | 4434 (*) | 4434 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En resumen, por los resultados mostrados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Comorbilidad Cardiovascular), consiguiendo un número de vectores soportes (**4271**) adecuados para el estudio con una tasa de clasificación del **67,81%** y con un coste del 10.

Además, se aprecia que aparentemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**2499**) menor, pero manteniendo una representación algo más baja en ambos niveles (**58,21%**) según el tipo de coste seleccionado (C=10).

De la misma manera que los otros casos mencionados, ambos modelos de kernels pueden ser empleados para futuros análisis, pero *el mejor es el modelo de tipo radial con coste 10 dando 4271 SVMs,* como bien se puede visualizar en la *Figura 31.*

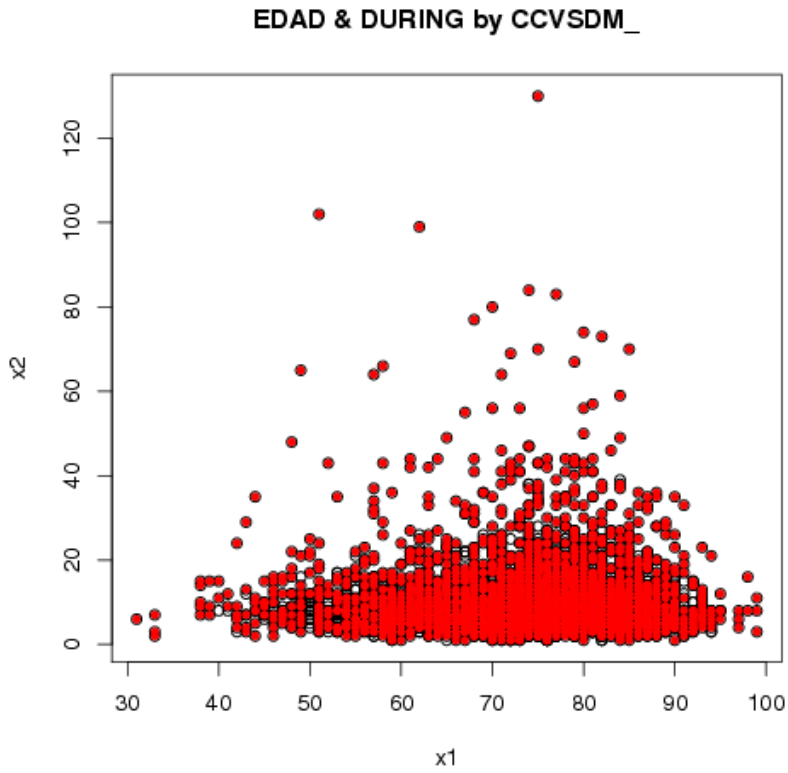



**Figura 31.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by CCVSDM – Caso 4.2

### *4.3. Diabetes mellitus*

También, para este caso los resultados muestran que los datos no se ajustan al tipo de kernel lineal, por lo que modificamos a otros kernels (*radial, polinomial* y *sigmoidal*) para mostrar el mejor ajuste.

A la vista de los resultados obtenidos, se tiene que el kernel "radial" es el mejor con coste=10, dando buena clasificación y alto porcentaje de aciertos; el kernel "polinomial" es mejor con coste=0.1 con una clasificación bastante aceptable que presenta un porcentaje alto de aciertos, aunque se podría seleccionar el de coste=1 con un porcentaje bueno, muy similar al del coste C=0.1; y el kernel "sigmoid" se tiene que ninguno de los costes es el mejor, ni mejora resultados con la constante de regularización.

En resumen, a continuación en la *Tabla 17* se disponen los resultados obtenidos para una mejor lectura del caso clínico *DIABETES MELLITUS.*

**Tabla 17.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.3

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 2693 (*) | 2694 (*) | 2696 (*) | 2693 (*) | 2694 (*) | 2696 (*) |
| Radial | 3138 (*) | 3165 (75,38%) | 3074 (77,40%) | - | - | - |
| Polynomial | 1967 (74,28%) | 1966 (72,05%) | 2030 (56,45%) | - | - | - |
| Sigmoid | 2668 (*) | 2668 (*) | 2668 (*) | 2668 (*) | 2668 (*) | 2668 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

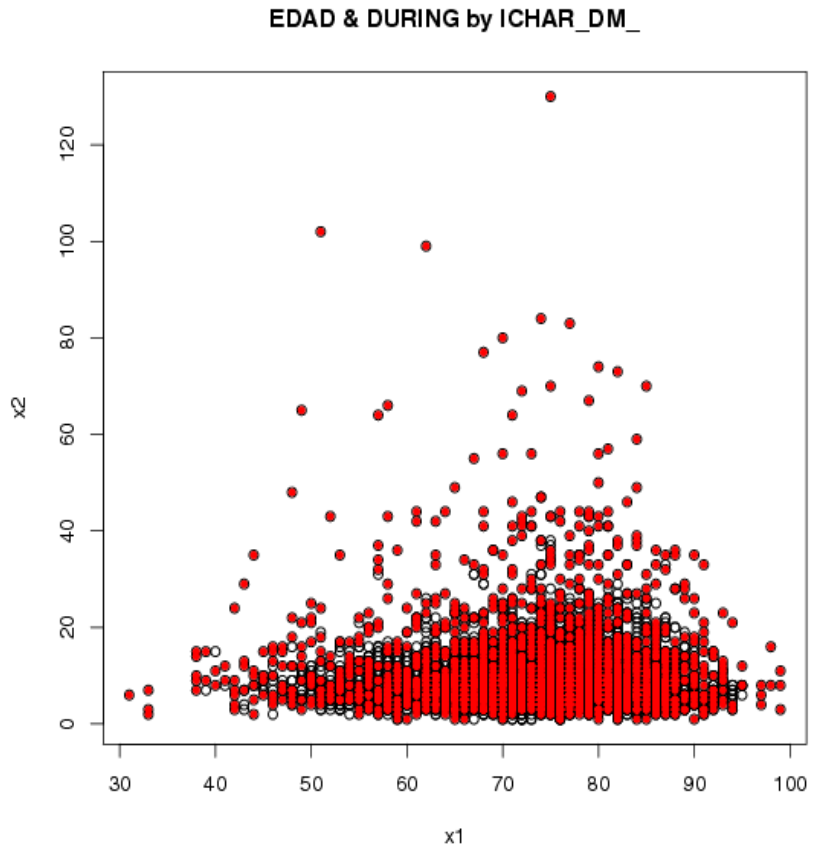



**Figura 32.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ICHAR_DM – Caso 4.3

En definitiva, según los resultados presentados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Enfermedad de Diabetes Mellitus), consiguiendo un número de vectores soportes (**3074**) adecuados para el estudio con una tasa de clasificación del **77,40%** y con un coste del 10.

Por otro lado, se aprecia que posiblemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**1967**) menor, pero manteniendo una representación algo más baja en ambos niveles (**74,28%**) según el tipo de coste seleccionado (C=0.1).

Dicho esto, ambos modelos de kernels pueden ser empleados para futuros análisis, pero *el mejor es el modelo de tipo radial con coste 10 dando 3074 SVMs,* como bien se puede visualizar en la *Figura 32.*

### *4.4. Enfermedad vascular con cerebro vascular y vascular periférica*

Siguiendo la mimsa línea anterior, para este caso los resultados muestran que no existe un ajuste de tipo lineal y se aplican los kernels *radial, polinomial* y *sigmoidal* para mejorar el ajuste.

Por tanto, los resultados obtenidos muestran que el kernel "radial" es el mejor con coste=10 con buena clasificación de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable que presenta un porcentaje medio de aciertos, aunque se podría seleccionar el de coste=1 con un porcentaje bajo, pero presenta un número menor de SVMs; y con el kernel "sigmoid" se ratifica que ninguno de los costes es el mejor, ni mejora resultados con la constante de regularización.

En resumen, a continuación en la *Tabla 18* se disponen los resultados obtenidos para una mejor lectura del caso clínico *ENFERMEDAD VASCULAR.*

**Tabla 18.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.4

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 3210 (*) | 3217 (*) | 3210 (*) | 3210 (*) | 3217 (*) | 3210 (*) |
| Radial | 3564 (*) | 3592 (71,53%) | 3458 (73,95%) | - | - | - |
| Polynomial | 2220 (33,93%) | 2216 (42,72%) | 2245 (68,54%) | - | - | - |
| Sigmoid | 3180 (*) | 3180 (*) | 3180 (*) | 3180 (*) | 3180 (*) | 3180 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En conclusión, por los resultados observados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Enfermedad Vascular abarcando la Cerebro Vascular y Vascular Periférica), consiguiendo un número de vectores soportes (**3458**) adecuados para el estudio con una tasa de clasificación del **73,95%** y con un coste del 10.

Sin embargo, se aprecia que visiblemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**2245**) menor, pero manteniendo una representación algo más baja en ambos niveles (**68,54%**) según el tipo de coste seleccionado (C=10).

Por tanto, ambos modelos de kernels pueden ser empleados para futuros análisis, pero *el mejor es el modelo de tipo radial con coste 10 dando 3458 SVMs,* como bien se puede visualizar en la *Figura 33.*

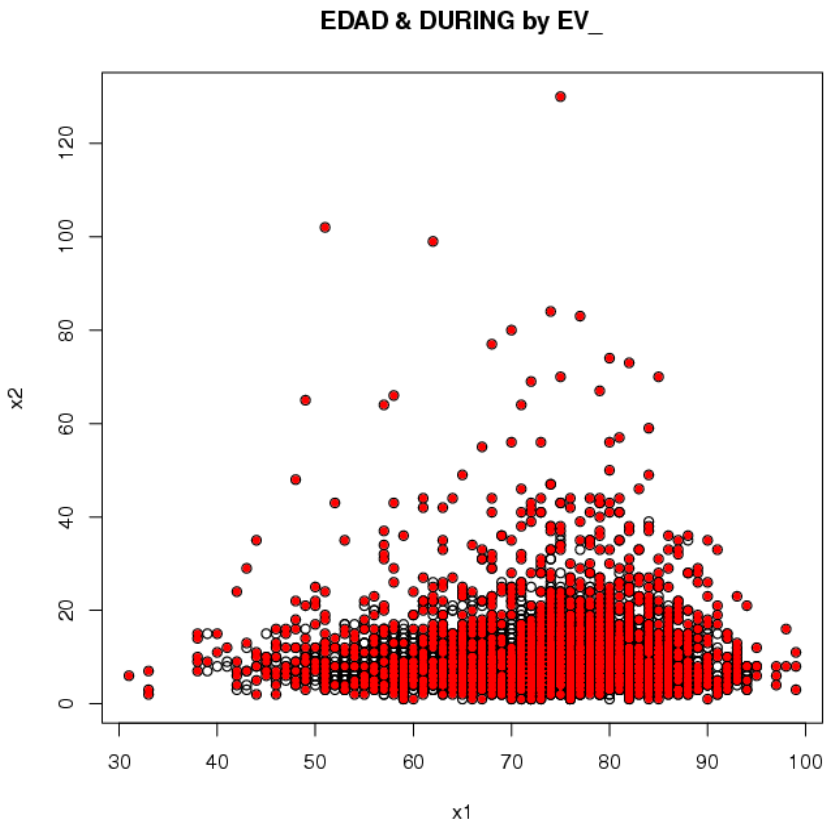



**Figura 33.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by EV – Caso 4.4

*4.5. Infarto de miocardio*

Con el mismo procedimiento de los casos anteriores, en este caso los resultados también muestran que los datos no se ajustan al tipo lineal y se aplican los diferentes kernels para mejorarlo.

A la vista de los resultados obtenidos, se tiene que el kernel "radial" es el mejor con coste=10 con buena clasificación y un alto porcentaje de aciertos; el kernel "polinomial" es mejor con coste=1 con una clasificación bastante aceptable que presenta un porcentaje alto de aciertos, aunque se podría seleccionar el de coste=0.1 con un porcentaje alto, muy similar al otro de C=1; y con el kernel "sigmoid" se confirma que no mejora los resultados ni con los diferentes costes ni con la constante de regularización.

En resumen, en la *Tabla 19* se disponen los resultados obtenidos para una mejor lectura del caso clínico *INFARTO DE MIOCARDIO.*

**Tabla 19.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.5

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 1354 (*) | 1352 (*) | 1355 (*) | 1354 (*) | 1352 (*) | 1355 (*) |
| Radial | 1842 (*) | 1970 (87,22%) | 1941 (88,10%) | - | - | - |
| Polynomial | 1189 (83,95%) | 1199 (86,44%) | 1202 (*) | - | - | - |
| Sigmoid | 1346 (*) | 1346 (*) | 1346 (*) | 1346 (*) | 1346 (*) | 1346 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En pocas palabras, según los resultados mostrados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Infarto de Miocardio), consiguiendo un número de vectores soportes (**1941**) adecuados para el estudio con una tasa de clasificación del **88,10%** y con un coste del 10.

Asimismo, se aprecia que teóricamente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**1199**) menor, pero manteniendo una representación algo más baja en ambos niveles (**86,44%**) según el tipo de coste seleccionado (C=1).

Por tanto, ambos modelos de kernels pueden ser empleados para futuros análisis, pero el *mejor es el modelo de tipo radial con coste 10 dando 1941 SVMs,* como bien se puede visualizar en la *Figura 34.*

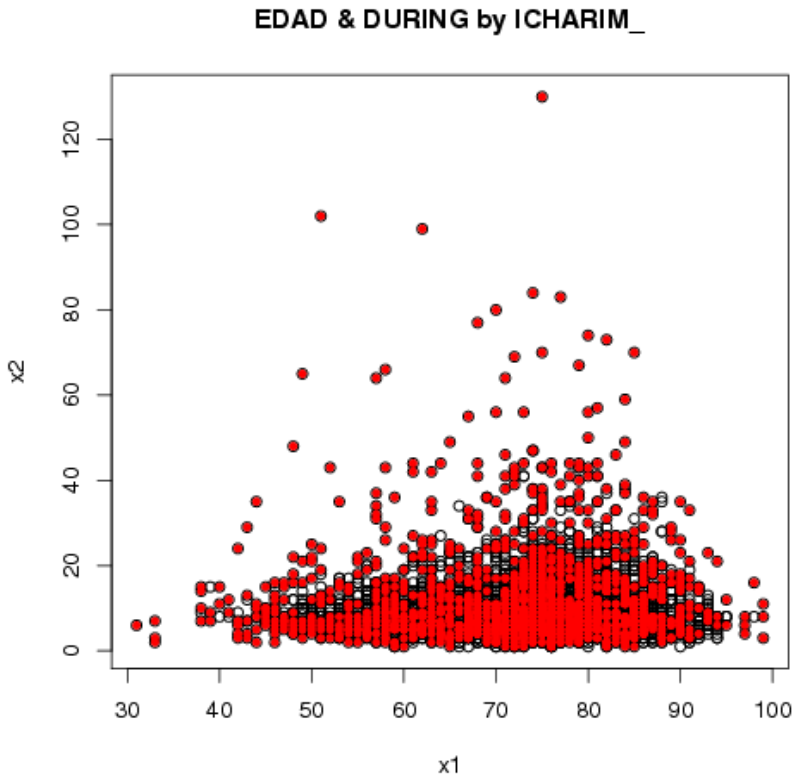



**Figura 34.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ICHARIM – Caso 4.5

*4.6. Nefropatía*

En esta misma pauta, para este caso los resultados también muestran que no existe ajuste de tipo lineal, por lo que se aplican los diferentes kernels (*radial, polinomial* y *sigmoidal)* con el fin de mostrar mejora en el ajuste del modelo.

Por ello, los resultados obtenidos muestran que el kernel "radial" es el mejor con coste=10 con buena clasificación de aciertos; el kernel "polinomial" es mejor con coste=0.1 con una clasificación bastante aceptable que presenta un porcentaje alto de aciertos, aunque se podría seleccionar el de coste=10 con un porcentaje alto, muy similar al otro con C=0.1; y con el kernel "sigmoid" se vuelve a ratificar que no mejora resultados ni con costes ni con la constante de regularización.

En resumen, a continuación en la *Tabla 20* se disponen los resultados obtenidos para una mejor lectura del caso clínico *NEFROPATÍA.*

**Tabla 20.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.6

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 980 (*) | 985 (*) | 984 (*) | 980 (*) | 985 (*) | 984 (*) |
| Radial | 1461 (*) | 1690 (90,77%) | 1684 (91,43%) | - | - | - |
| Polynomial | 900 (90,58%) | 885 (*) | 894 (90,50%) | - | - | - |
| Sigmoid | 974 (*) | 974 (*) | 974 (*) | 974 (*) | 974 (*) | 974 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

De esta manera, por los resultados obtenidos se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Nefropatía), consiguiendo un número de vectores soportes (**1684**) adecuados para el estudio con una tasa de clasificación del **91,43%** y con un coste del 10.

También, se aprecia que aparentemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**900**) menor, pero manteniendo una representación algo más baja en ambos niveles (**90,58%**) según el tipo de coste seleccionado (C=0.1).

Por lo que, ambos modelos de kernels pueden ser empleados para futuros análisis, pero el *mejor es el modelo de tipo radial con coste 10 dando 1684 SVMs,* como bien se puede visualizar en la *Figura 35.*

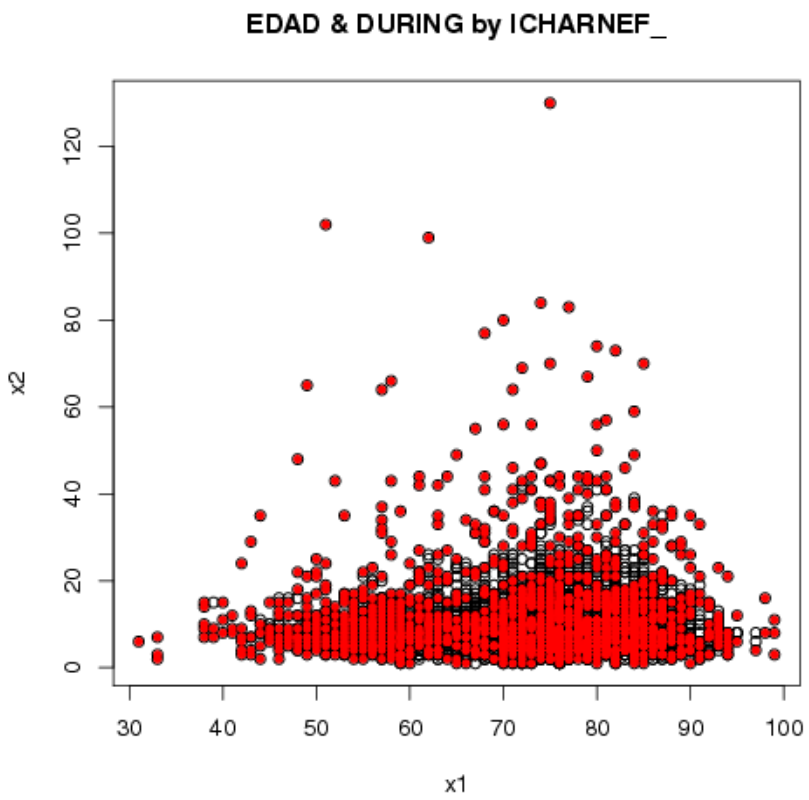



**Figura 35.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ICHARNEF – Caso 4.6

*4.7. Tumor sólido*

Para este caso aplicado, los resultados previos también indican que los datos no se ajustan al tipo lineal y se modifican los diferentes kernels para mejorarlo.

Por tanto, a la vista de los resultados obtenidos, se tiene que el kernel "radial" es el mejor con coste=10 con buena clasificación y un alto porcentaje de aciertos; el kernel "polinomial" es mejor con coste=0.1 con una clasificación bastante aceptable que presenta un porcentaje alto de aciertos, aunque se podría seleccionar el de coste=1 con el mismo porcentaje, pero con mayor número de SVMs; y con el kernel "sigmoid" se muestra que no existe mejora ni con diferentes costes ni con la constante de penalización.

A continuación, en la *Tabla 21* se disponen los resultados obtenidos para una mejor lectura del caso clínico *TUMOR SÓLIDO.*

**Tabla 21.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.7

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 1354 (*) | 1358 (*) | 1355 (*) | 1354 (*) | 1358 (*) | 1355 (*) |
| Radial | 1819 (*) | 1998 (87,23%) | 1965 (88,12%) | - | - | - |
| Polynomial | 1178 (87,04%) | 1182 (87,04%) | 1179 (85,52%) | - | - | - |
| Sigmoid | 1344 (*) | 1344 (*) | 1344 (*) | 1344 (*) | 1344 (*) | 1344 (*) |

* Valor porcentual insignificante por estar mal clasificados debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En conclusión, por lo que como muestran los resultados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Tumor Sólido), consiguiendo un número de vectores soportes (**1965**) adecuados para el estudio con una tasa de clasificación del **88,12%** y con un coste del 10.

En esta misma línea, se aprecia que posiblemente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**1178**) menor, pero manteniendo una representación algo baja en ambos niveles (**87,04%**) según el tipo de coste seleccionado (C=0.1).

Y del mismo modo que los casos anteriores, ambos modelos de kernels pueden ser empleados para futuros análisis, pero el *mejor es el modelo de tipo radial con coste 10 dando 1965 SVMs,* como bien se puede visualizar en la *Figura 36.*

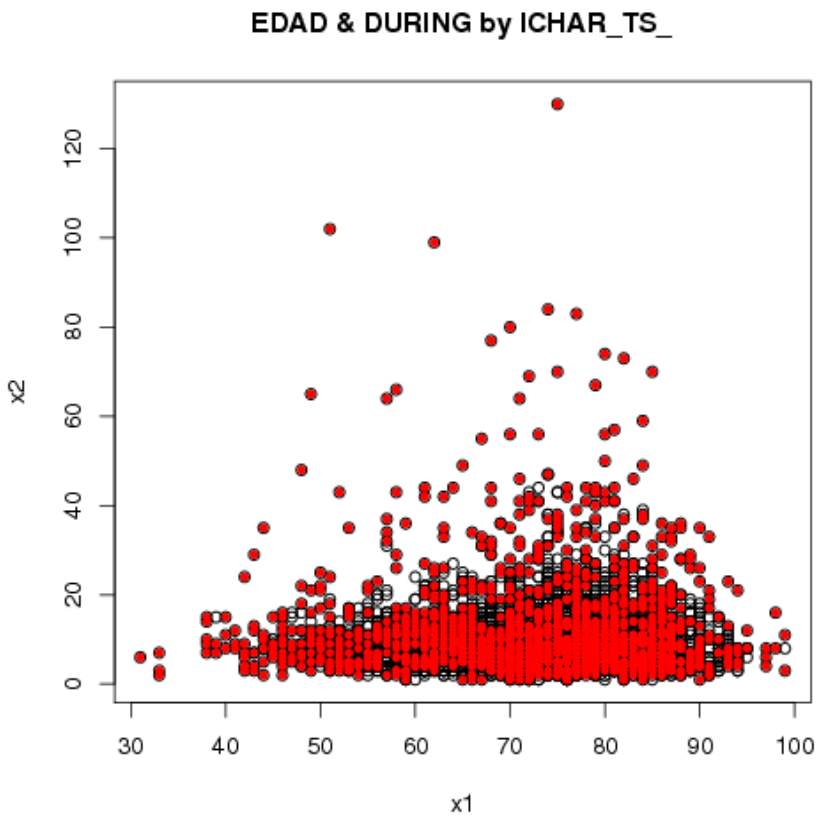



**Figura 36.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by ICHAR_TS – Caso 4.7

*4.8. Edemas maleolares*

Para finalizar con este procedimiento de vectores soportes, en este caso particular los resultados también continúan mostrando que no existe un ajuste de tipo lineal y se necesita aplicar otros kernels (*radial, polinomial* y *sigmoidal)* con el fin de mejorar el ajuste del modelo.

Dicho esto, los resultados obtenidos muestran que el kernel "radial" es el mejor con coste=10 con buena clasificación y un alto porcentaje de aciertos; el kernel "polinomial" es mejor con coste=10 con una clasificación bastante aceptable y que también mantiene un alto porcentaje de aciertos; y el kernel "sigmoid" solo se puede ratificar que no mejora los resultados ni con los diferentes costes ni con la constante de regularización.

En resumen, en la *Tabla 22* se disponen los resultados obtenidos para una mejor lectura del caso clínico *EDEMAS MALEOLARES.*

**Tabla 22.** SVMs con tasa de clasificación mediante la selección de distintos Kernels ajustados – Caso 4.8

| KERNELS | SVM (Tasa clasificación) | | | | | |
|---|---|---|---|---|---|---|
| | C=0.1 | C=1 | C=10 | C=0.1/Coef0=1 | C=1/Coef0=1 | C=10/Coef0=1 |
| Linear | 2719 (*) | 2724 (*) | 2719 (*) | 2719 (*) | 2724 (*) | 2719 (*) |
| Radial | 3135 (*) | 3153 (75,67%) | 3063 (77,95%) | - | - | - |
| Polynomial | 1929 (47,78%) | 1996 (*) | 2055 (73,74%) | - | - | - |
| Sigmoid | 2700 (*) | 2700 (*) | 2700 (*) | 2700 (*) | 2700 (*) | 2700 (*) |

* Valor porcentual insignificante por estar mal clasificado debido a la selección del kernel y el ajuste de parámetros. [-] No aplica.

En este sentido, según los resultados mostrados se puede comprobar que el modelo de **kernel radial** es perfecto para la relación de las variables (edad del individuo y duración de ingreso hospitalario por Edemas Maleolares), consiguiendo un número de vectores soportes (**3063**) adecuados para el estudio con una tasa de clasificación del **77,95%** y con un coste del 10.

Y de la misma manera, se aprecia que supuestamente el modelo de **kernel polinomial** se adapta algo mejor a esta relación, obteniendo un número de vectores soportes (**2055**) menor, pero manteniendo una representación algo baja en ambos niveles (**73,74%**) según el tipo de coste seleccionado (C=10).

Por lo tanto, se puede decir que ambos modelos de kernels pueden ser empleados para futuros análisis, pero el *mejor es el modelo de tipo radial con coste 10 dando 3063 SVMs,* como bien se puede visualizar en la *Figura 37.*

A modo de resumen, desde el punto de vista clínico, el planteamiento de todas estas nuevas variables incorporadas al análisis que están relacionadas con el seguimiento de los pacientes son satisfactoriamente buenas para ayudar a mejorar la interpretación de los resultados clínicos, y por consiguiente, la del agrupamiento de perfiles, que de alguna manera también ayudar a encontrar un enfoque más específico sobre los datos, en concreto, en cuanto a la selección de grupos según las patologías asociadas, que se sabe que tienen un papel fundamental en el avance de la enfermedad

junto a otras características inherentes del propio paciente, pudiendo resumir la información analizada en los vectores soportes obtenidos sin perder datos relevantes de los diferentes casos explorados en este estudio.

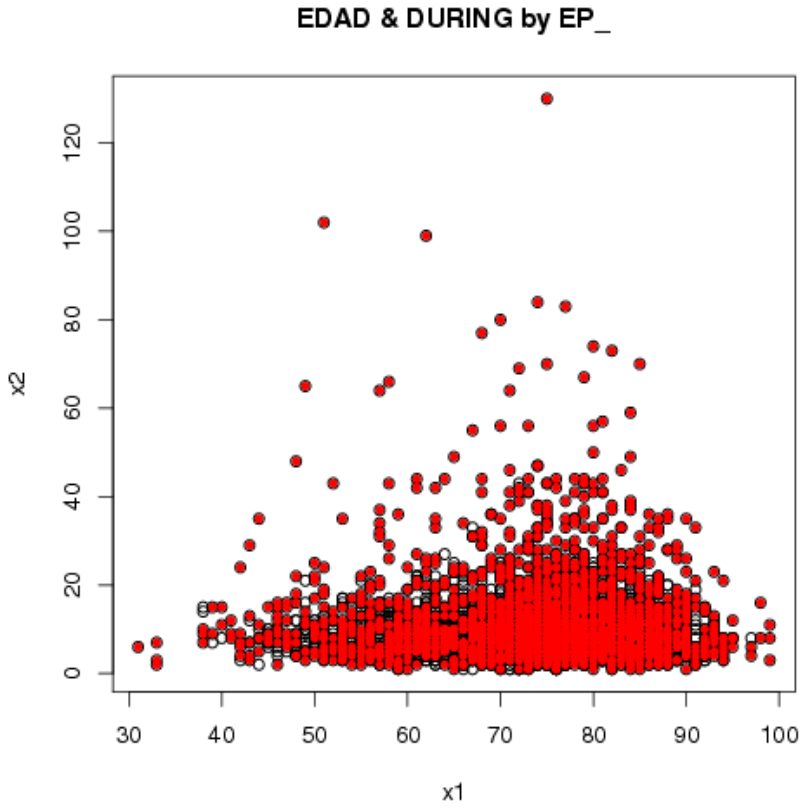



**Figura 37.** SVMs de EDAD & DURING (C=10 ; Kernel=radial) by EP – Caso 4.8

## IV. 4. Conclusiones

A la vista de los resultados obtenidos, el objetivo de este estudio ha sido el intentar verificar como están distribuidos los pacientes según las diferentes patologías y poder detectar que grupos de perfiles asociados a ellos son los que se definen como tales. Es decir, averiguar los tipos de enfermedades relacionadas entre sí para un mismo paciente, determinando los distintos perfiles clínicos, que puede ser como guía de lo estándar en la práctica clínica con el fin de detectar rápidamente al sujeto y poder intervenir de forma precisa con el correspondiente tratamiento, mejorando el estado de salud y la calidad de vida del mismo durante el periodo de la enfermedad, que esta puede ser corta o larga según el desarrollo de la misma dentro del propio paciente.

Se ha detectado que la alta dimensión de la base de datos en estudio complica el enfoque del análisis que se pretende, pero se ha encontrado una salida bastante eficaz con respecto a este aspecto, llevando directamente al tipo de análisis aplicado como son los SVMs y los Kernels.

Estos vectores soportes seleccionados como óptimos, pueden ayudar a profundizar de una manera más sencilla en el planteamiento que se ha perseguido, mostrando una muestra más pequeña y asequible de la información para ser posteriormente extrapolada a la población general sin pérdida alguna de la misma y con el mismo valor que la original.

En este sentido, como se ha podido observar en el análisis de los resultados, estos arrojan diferentes visualizaciones dependiendo del paciente que se esta analizando por lo que se pueden resumir las salidas obtenidas en los siguientes puntos:

- La relación de las variables estudiadas en cada uno de los casos se encuentra entre el 68 - 96% en ambos niveles, obteniéndose relativamente un número menor de vectores soportes, teniendo en cuenta la alta dimensión de la base de datos.

- Los modelos kernels que mejor se ajuntan a los datos son el radial y el polinomial ofreciendo un resultado más asequible con el menor coste posible para ayudar en la interpretación clínica.

- El modelo polinomial presenta un ajuste bastante bueno, pero a la hora de ejecutarlo muestra un límite en el número de iteración siendo su coste computacional algo más complejo.

- Cuando se trabaja con datos reales se aprecia una diferencia significativa en el tratamiento de la información y se encuentran problemas que con datos simulados no se percibe, siendo el análisis de estos últimos, mucho más sencillo y rápido.

Desde un punto de vista más clínico, los resultados destacables hasta el momento se pueden agrupar en los siguientes ítems:

- La edad de los participantes influye en la duración del ingreso hospitalario siendo predominante en el sexo masculino.

- El ser persona fumadora puede ser un signo adicional en el avance de la enfermedad contribuyendo a empeorar el estado del paciente.

- La variable exitus esta claramente identificada con la edad y la duración del ingreso hospitalario, donde se puede afirmar que cuanto más larga sea la estancia del paciente y más avanzada sea la edad del mismo, más posibilidades tendrá para que el evento sea positivo.

- Las variables de patologías tienen mayor impacto dependiendo de la gravedad de la enfermedad en la que se este analizando, destacando con mayor presición en aquellos pacientes con una avanzada edad y una duración del ingreso hospitalario bastante larga, donde visualmente se puede apreciar que la dispersión de los datos en la nube de puntos tiende a la derecha, que equivale a estancias más profundas y hacia arriba, que indica que a mayor edad se tiende a tener peor estado de salud por diversos factores externos y otras muchas por motivos propios.

En conclusión, se puede decir que las variables estudiadas guardan una relación entre ellas y que el signo de la patología debe ser analizada en profundidad como variable separadora (clase) para determinar la influencia de la misma sobre el participante, como se ha podido mostrar, puesto que la información en conjunto, en muchas ocasiones no es posible apreciar la influencia de cada una de ellas.

También, se puede decir que los mejores modelos que se adaptan a estos datos son el radial y el polinomial. Y que los SVMs seleccionados pueden ser la submuestra de estudio para mejorar los

resultados tanto visualmente como clínicamente, pudiendo extrapolarse más adelante a la muestra general.

Además, a la vista de los resultados analizados con estas variables de interés al estudio se pueden incluir también los datos espirométricos de las pruebas asociadas a la espirometría, los datos antropométricos y los datos de la exploración física, con la finalidad de ayudar a mejorar la interpretación clínica y enfocar algo mejor el objetivo planteado, que como se comento puede ser otra alternativa a la técnica de clasificación que se ha empleado en el capítulo III de esta tesis doctoral.

En este sentido, se ha podido verificar que esta última información no cambia en absoluto el agrupamiento de perfiles alcanzado, donde se contribuyó a unir a los pacientes de una manera diferente según características afines entre ellos y separándolos en otros segmentos distintos entre los grupos formados, y con esta técnica se confirma que los datos pueden resumirse bastante con los SVM aplicando el ajuste a los parámetros y sin perder la esencia de ellos, con el fin de mostrar la misma información en un espacio más reducido si se compara en términos de dimensión, manteniendo las propias características del original.

Dicho esto, si se desea extender este estudio analítico con la alternativa clasificatoria de SVM, es necesario optar por otros kernels más complejos, por la optimización de los parámetros mediante el método de validación cruzada (*cross*) usando la función genérica "*tune ()*" del paquete *e1071* de R (Meyer et al., 2021), o también seguir ajustando con la constante de penalización para mejorar el modelo.

No obstante, está claro que trabajar con datos simulados es más sencillo y menos costoso desde un punto computacional, puesto que pueden ser más moldeables y menos problemáticos a la hora de implementar los algoritmos, que cuando se están analizando datos reales que se ve incrementada la complejidad de una forma exponencial y los inconvenientes se multiplican bastante más que cuando se está realizando una práctica de laboratorio que los datos se diseñan a medida para un caso en concreto muy diferente a una situación real que estos son poco manejables y muy diferentes de un individuo a otro, necesitando encontrar una técnica que pueda abarcar todo esto y dar la mejor solución que represente la información analizada sin dejar datos relevantes por el camino que puedan aportar algún valor extra a la decisión final.

# CONCLUSIONES FINALES

# DEL ESTUDIO

## 1. Conclusiones finales de esta investigación

A modo de resumen final, se puede decir que los resultados de esta investigación nos ha permitido concluir las siguientes afirmaciones según la información analizada y revisada hasta este momento:

- El problema de la imputación de datos faltantes (missing values) sigue siendo un gran desafío en los estudios de investigación, pero se ha podido explorar la existencia de multitud de técnicas de imputación disponibles según el tipo de variables del conjunto de datos para no perder la calidad relacional del registro original. En este caso particular la opción del MICE fue una solución bastante acertada, evitando sesgos mediante la reducción de la incertidumbre en los valores perdidos y aumentando la eficiencia de los datos con el fin de obtener una mayor calidad y mejor información de nuestro registro, pero se recuerda de la existencia de diversos mecanismos para apaliar la mala calidad de los datos que desafortunadamente limitan el uso de las distintas fuentes de datos sanitarias y en general por esta misma cuestión descrita.

- Uno de los segundos aspectos más frecuentes en los registros es el problema de la maldición de la dimensionalidad, que también se ha podido mostrar y comparar varios métodos de aplicabilidad para paliar este problema, y al mismo tiempo se han mencionado otros procedimientos de la competencia para futuros análisis, ya que se sabe que el análisis PCA en este caso ha dado buenos resultados siendo una de las mejores técnicas de reducción dimensional, pero tiene ciertas limitaciones en algunos casos, puesto que es un método que solo usa combinaciones lineales de las variables originales, y a veces se puede llegar a perder mucha información.

- El abordaje sobre la búsqueda de patrones afines mediante diversos métodos de aprendizaje supervisado y no supervisado, se ha podido resolver através del análisis cluster o por conglomerado junto al soporte adicional de otras técnicas como el de correspondencia (CA) y de decisión (DT), originando un número óptimo de clutsters con tres grupos de perfiles clínicos finales muy similares entre sí y distintos entre ellos, que presentan características y propiedades similares, ayudando en la toma de decisiones con el fin de mejorar ciertos aspectos en los procedimientos clínicos y asistenciales para pacientes con un estado de salud bastante deteriorado, por malos

hábitos de salud y por una avanzada edad, y que además suele empeorar el episodio principal diagnosticado, provocando que resalten otras comorbilidades que están en el umbral intermedio sin dar su aparición hasta que se alcance ese limite extremo donde no existe retorno sin proceder con medidas algo trasticas que provoquen un pico de mejoría en muy poco tiempo. Por eso, se espera que el avance de la tecnología pueda ayudar en estos temas de salud, con el fin de poder generar modelos más personalizados, predictivos y preventivos en un futruo, sirviendo como base central para dar una solución factible y en un tiempo bastante reducido donde se pueda tomar decisiones más precisas y eficientes.

- Y por último, el enfoque de los vectores soporte (SVM) mediante los diferentes tipos de Kernels (lineal, radial, polinomial y sigmoidal) utilizados para la adaptación de los datos, junto a los diferentes parámetros de coste y de regularización o penalización para correguir el ajuste del modelo, ha resultado ser una buena alternativa para buscar y representar la información relevante en otra muestra más pequeña a través de un número óptimo de vectores sin perder los datos esenciales del caso analizado, pudiendo extrapolar los resultados obtenidos de la misma para la población general. Asimismo, para este caso particular, se ha detectado que los modelos kernels que mejor se ajustan a los datos son el radial y el polinomial ofreciendo un resultado más asequible con el menor coste posible ayudando a la interpretación clínica. Aunque el polinomial presenta un ajuste bastante bueno su coste computacional algo más complejo. No obstante, se ha confirmado que existe una relación bastante alta entre las variables estudiadas que oscila entre el 68 - 96% en ambos niveles de la categoría clase, pero aun así se ha alcanzado un número menor de vectores soportes, teniendo en cuenta la alta dimensión de la base de datos. Y al mismo tiempo, este análisis solo ha venido a confirmar un poco más lo que ya se conocía, que estos datos podían agruparse formando perfiles afines, siendo necesario trabajarlo en un espacio más reducido, pero manteniendo las características del original y que trabajar con datos simulados es mucho más sencillo y menos costoso desde un punto de vista computacional, puesto que los datos pueden ser más moldeables y menos problemáticos a la hora de implementar algoritmos, que cuando se están analizando datos reales, donde el panorama es totalmente diferente y la complejidad es mayor, por eso es una buena alternativa los avances y las transformaciones experimentadas a nivel informático puesto que ayudaran a mejorar estas problemáticas de implementación y de tiempo.

Es evidente, que la tecnología que se esta desarrollando en todas las áreas de inventigación, en especial en el campo de la salud, donde se esta fomentando el crecimiento por apdaptar y mejorar las aplicaciones informáticas con el fin de ofrecer una mejor atención y calidad a los ciudadados y pacientes, intentando que todos estos programas de utilidad sean cercanos a ellos, proporcionando un soporte adaptado y de fácil acceso para una continua utilización en el futuro con la finalidad de mejorar todos los aspecto necesarios en su vida cotidiana centrado en el paciente para fomentar sus cuidados y disminuir sus demandas especificas por el desconocimiento de estas herramientas disponibles para mejor su bienestar y enfocar su atención en lo que verdaderamente importante, que es una salud de calidad y una vida saludable que le proporcione beneficios en su día a día.

Dicho esto, como todos sabemos el futuro del avance digital esta cambiando nuestra vida y la sociedad en general y se espera que estos algoritmos implementados (con aprendizaje supervisado o no supervisado) dé ese valor añadido a todas nuestras herramientas de trabajo rutinarias, con un plus de mejora tecnológica en focado en mejorar decisiones basados en diagnosticos, tratamientos y procedimientos más efectivos con una información personalizada y preventiva basados en los datos y los resultados predictivos.

En definitva, poder ayudar en todos los procesos de decisión, dando soporte de calidad y validez científica a los estudios de investigación desarrollados con la máxima precisión posible a través de los datos almacenados en nuestros repositorios o instituciones, con la finalidad de dar respuestas rápidas y eficaces para que sirvan como medio de actuación y de reduccción del tiempo de respuesta, en los momentos más críticos o cruciales en los que la vida del paciente este en peligro, adaptando y mejorando la tecnología en estos puntos de inflexión que a corto y largo plazo, darán mejores cuidados que se reflejara en una buena calidad de vida y de hábitos saludables para el individuo (paciente) en general.

Y finalmente, la aportación que se ofrece con el desarrollo de esta investigación es pretender mostrar que la alta dimensión ya no es un problema para estudios multicéntricos, donde al aumentar la dimensión con la existencia de multitud de variables y registros, el volumen del espacio aumenta exponencialmente provocando que los datos sean muy dispersos, lo que se precisa fundamentalmente un abordaje para esta problemática a priori con el fin de obtener una información relevante y de calidad. Para ello, se pueden aplicar diferentes técnicas multivariantes efectivas, que garantizan una buena reducción dimensional manteniendo la esencial de la información de los datos originales y al mismo tiempo se reducen costes computacionales y de tiempo de reacción para dar respuestas

automáticas a casos similares, donde será casi necesario implementar estas aplicaciones en todos los ámbitos de investigación, puesto que nos acercamos a una globalización digital potente, que podrá abordar con exactitud una amplia variedad de temáticas tecnológicas y de investigación con la finalidad de presentar resultados significativos y de calidad con un gran valor para la sociedad en general.

Precisamente, este planteamiento primario, será muy necesario en poco tiempo con la nueva iniciativa de la Unión Europea (UE), en el que se ha propuesto la creación del Espacio Europeo de Datos Sanitarios (EEDS) para todos los países miembros, lo que generara un volumen inmenso de datos sanitarios que requerirán de técnicas más sofisticadas, (como son las de reducción de dimensionalidad, de imputación múltiple de valores faltantes o las de análisis de clasificación mixta de algoritmos de aprendizaje supervisado y no supervisado), para destacar la información esencial e indispensable que puedan ayudar a la toma de decisiones (Comisión Europea, 2022).

Y en paralelo, dar a conocer que existen algoritmos que pueden clasificar a los distintos pacientes de origen generando diferentes grupos muy similares entre ellos internamente, que pueden ser la base para otros de propiedades o características iguales donde se requiera un diagnóstico, tratamiento o procedimiento clínico, evitando realizar en la práctica asistencial o rutinaria multitud de gestiones, que ya de por sí son necesarias para conocer la situación general del paciente y tomar decisiones al respecto, pero se puede reducir todo este proceso asistencial teniendo información de grupos donde compararlos con propiedades idénticas que se pueden extrapolar al resto sin aplicar ningún otro procedimiento de rutina muy específico fuera de lo convencional, para ayudar en el soporte a la decisión final, y por consiguiente, mejorar la calidad de vida de los pacientes, en especial a los que presentan patologías crónicas, donde el tiempo es el peor aliado, puesto que deteriora el estado de salud significativamente y a su vez, hace que presenten otras comorbilidades que agravan su situación global, precisando mejores cuidados clínicos y atenciones personalizadas para poder obtener una mejor calidad de vida.

## 2. Contribuciones a esta tesis doctoral a futuro

Existen **otros artículos en proceso de desarrollo** relacionados con esta tesis doctoral, que por motivos de tiempo de permanencia y forma en estos estudios de doctorado, desafortunadamente y sintiéndolo mucho no podrán incluirse como parte del aval de esta tesis doctoral antes de la fecha de su depósito y registro legal.

Por ello, esta **pendiente de publicación** el capítulo III (*Clasificación grupal para la identificación y búsqueda de patrones afines mediante diferentes técnicas multivariantes*) y en un futuro no muy lejado también el capítulo IV (*Caso experimental con datos simulados y reales mediante SVM y métodos Kernel*).

## 3. Otras investigaciones relacionadas con el ámbito de estudio

En paralelo, también se han realizado otras investigaciones relacionadas con las técnicas no paramétricas, específicamente con el *Estimador Polinomial Local para datos reales y simulados*, método que realiza un ajuste polinomial con las observaciones que caen en la banda, corrigiendo de forma automática los efectos frontera para ajustarse al máximo posible a ellos a través de la curva del estimador polinomial local, que queda determinado por tres parámetros fundamentales: el ancho de banda, la función núcleo y el grado p en la aplicación real, y para el caso simulado por: el tamaño muestral, el tipo de dominio o rejilla y la función de tendencia para la simulación de los datos y la estimación del modelo, y en ambos casos, con el fin de facilitar un mejor modelado de los datos y una buena interpretación de los mismos. Para ello, se han **desarrollados dos capítulos de libro** con este tema, siendo **autora principal** en cada uno de ellos, como se indica en las siguientes referencias:

**Boukichou-Abdelkader, N.**, Montero-Alonso, M.Á., Muñoz-García, A. & Canário, P. N. (2014). REGRESIÓN NO PARAMÉTRICA: ESTIMADOR POLINOMIAL LOCAL. En: *Modelación Matemática de Fenómenos del Medio Ambiente y la Salud (III).* Red Iberoamericana de Estudios Cuantitativos Aplicados – RIDECA. Capítulo 5, 46-52. ISBN: 84-616-7997-0.

**Boukichou-Abdelkader, N.**, Montero-Alonso, M.Á., Muñoz-García, A. & Canário, P. N. (2015). EXPERIMENTOS DE SIMULACIÓN: ESTIMADOR POLINOMIAL LOCAL. En: *Experiencias en la Modelación de la toma de decisiones en salud humana, medio ambiente y desarrollo humano (I).* Red Iberoamericana de Estudios Cuantitativos Aplicados – RIDECA. Capítulo 6, 138-146. ISBN: 978-84-606-5638-8.

Asimismo, se ha desarrollado otro estudio enfocado en el área de investigación clínica y farmacológica, guardando relación directa con el contenido de esta Tesis Doctoral, concretamente el tema abordado es sobre la *Prevención de Diabetes y Enfermedades Cardiovasculares (estudio ePREDICE, que es un* ensayo clínico aleatorizado con fármacos y estilos de vida*)*, donde he sido participe desde su inicio y **soy co-autora de la siguiente aportación científica**, pudiéndose consultar en esta dirección.

Gabriel, R., **Boukichou-Abdelkader, N.**, Acosta, T., Gilis-Januszewska, A., Gómez-Huelgas, R., Makrilakis, K., et al. (2020). Early prevention of diabetes microvascular complications in people with hyperglycaemia in Europe. ePREDICE randomized trial. Study protocol, recruitment and selected baseline data. *PLoS ONE,* **15** (4): e0231196. DOI: https://doi.org/10.1371/journal.pone.0231196

# REFERENCIAS

[1] Alsaber, A. R., Pan, J. and Al-Hurban, A. (2021). Handling Complex Missing Data Using Random Forest Approach for an Air Quality Monitoring Dataset: A Case Study of Kuwait Environmental Data (2012 to 2018). *International Journal Of Environmental Research and Public Health,* **18** (3), 1333.

[2] Amat Rodrigo, J. (2017a). *Análisis de Componentes Principales y t-SNE.* Available at https://www.cienciadedatos.net/documentos/35_principal_component_analysis#t-SNE

[3] Amat Rodrigo, J. (2017b). *Otros métodos de reducción de dimensionalidad.* Available at https://www.cienciadedatos.net/documentos/35_principal_component_analysis#otros-metodos-de-reduccion-de-dimensionalidad

[4] Amat Rodrigo, J. (2017c). *Clustering y heatmaps: aprendizaje no supervisado.* Available at https://www.cienciadedatos.net/documentos/37_clustering_y_heatmaps

[5] Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, **68** (3), 337–404.

[6] Ayers, B., Sandholm, T., Gosev, I., Prasad, S. and Kilic, A. (2021). Using machine learning to improve survival prediction after heart transplantation. *Journal of Cardiac Surgery*, **36** (11), 4113–4120.

[7] Beck, M. W. (2013). *Visualizing neural networks in R, update.* Available at https://beckmw.wordpress.com/2013/11/14/visualizing-neural-networks-in-r-update/

[8] Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics.* Springer Science+Business Media, New York.

[9] Beysolow II, T. (2017). *Introduction to deep learning using R: A step-by-step guide to learning and implementing deep learning models using R. Machine Learning Example Problems, Chapter 10, 171–194.* Apress, Berkeley, CA.

[10] Bhalla, D. (2015). *Weight Of Evidence (WOE) and Information Value (IV) explained.* Available at. https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html

[11] Blazek, K., Van Zwieten, A., Saglimbene, V. and Teixeira-Pinto, A. (2021). A practical guide to multiple imputation of missing data in nephrology. *Kidney International,* **99** (1), 68–74.

[12] Blázquez-Sánchez, N., Rivas-Ruiz, F., Bueno-Fernández, S., Arias-Santiago, S., Fernández-Morano, M.T. and De Troya-Martín, M. (2020). Validation of a Questionnaire Designed to Study Knowledge, Attitudes, and Habits Related to Sun Exposure Among Young Adults: The CHACES Questionnaire. *Actas Dermo-Sifiliográficas*, **111** (7), 579–589.

[13] Bosco Mendoza Vega, J. (2018). *Árboles de decisión con R. Clasificación*. Available at https://rpubs.com/jboscomendoza/arboles_decision_clasificacion

[14] Bougeard, S. and Dray, S. (2018). "Supervised Multiblock Analysis in R with the ade4 Package." *Journal of Statistical Software*, **86** (1), 1–17.

[15] Boukichou-Abdelkader, N., Montero-Alonso, M.Á. and Muñoz-García, A. (2022). Different Routes or Methods of Application for Dimensionality Reduction in Multicenter Studies Databases. *Mathematics*, **10** (5), 696.

[16] Boutros, P. C. and Okey, A. B. (2005). Unsupervised pattern recognition: an introduction to the whys and wherefores of clustering microarray data. *Brief Bioinform.* **6** (4), 331–343.

[17] Bouzid, Z., Faramand, Z., Gregg, R. E., Frisch, S. O., Martin-Gill, C., Saba, S., Callaway, C., Sejdić, E. and Al-Zaiti, S. (2021). In Search of an Optimal Subset of ECG Features to Augment the Diagnosis of Acute Coronary Syndrome at the Emergency Department. *Journal of the American Heart Association*, **10** (3), e017871.

[18] Cheney, W. (2001). *Analysis for Applied Mathematics***.** Springer Science+Business Media, New York.

[19] Chia, K., Fischer, I., Thomason, P., Graham, H. K. and Sangeux, M. (2020). A Decision Support System to Facilitate Identification of Musculoskeletal Impairments and Propose Recommendations Using Gait Analysis in Children With Cerebral Palsy. *Frontiers in Bioengineering and Biotechnology*, **8**, 529415.

[20] Choubey, D. K., Kumar, M., Shukla, V., Tripathi, S., and Dhandhania, V. K. (2020). Comparative Analysis of Classification Methods with PCA and LDA for Diabetes**.** *Current Diabetes Reviews*, **16** (8), 833–850.

[21] Comisión Europea (2022). *Unión Europea de la Salud: Un espacio europeo de datos sanitarios para las personas y la ciencia.* Available at https://ec.europa.eu/commission/presscorner/detail/es/ip_22_2711

[22] Cucker, F. and Zhou, D. X. (2007). *Learning Theory. An Approximation Theory Viewpoint.* Cambridge University Press, New York**.**

[23] De La Fuente Fernández, S. (2011). *Análisis de Correspondencias Simples y Múltiples. Fac. Ciencias Económicas y Empresariales. UAM*. Available at https://www.fuenterrebollo.com/Economicas /ECONOMETRIA/REDUCIR-DIMENSION/CORRESPONDENCIAS/correspondencias.pdf

[24] De Leeuw, J., Mair, P. and Groenen, P. J. F. (2017). *Categorical principal component analysis (PRINCALS). Package Gifi: Multivariate Analysis with Optimal Scaling.* R Foundation for Statistical Computing, Vienna, Austria. Available at https://search.r-project.org/CRAN/refmans/Gifi/html/princals.html

[25] Delisle Nyström, C., Barnes, J. D. and Tremblay, M. S. (2018). An exploratory analysis of missing data from the Royal Bank of Canada (RBC) Learn to Play - Canadian Assessment of Physical Literacy (CAPL) project. *BMC Public Health*, **18** (Suppl 2), 1046.

[26] Deng, L. and Wang, Y. (2021). Hybrid diffusion tensor imaging feature-based AD classification. *Journal of X-ray Science and Technology,* **29** (1), 151–169.

[27] Dollfus, S. and Petit, M. (1995). Análisis de componentes principales de la PANSS y la SANS-SAPS en esquizofrenia: Su estabilidad en una fase aguda. *European Psychiatry* (Ed. Española), **2**(4), 219–230.

[28] Dray, S., Dufour, A. and Chessel, D. (2007). "The ade4 Package - II: Two-Table and K-Table Methods." R News, **7** (2), 47–52.

[29] El Boujnouni, H., Rahouti, M. and El Boujnouni, M. (2021). Identification of SARS-CoV-2 origin: Using Ngrams, principal component analysis and Random Forest algorithm. *Informatics in Medicine Unlocked*, **24**, 100577.

[30] Evgeniou, T., Pontil, M. and Poggio, T. (2000). Regularization Networks and Support Vector Machines. *Advances in computational mathematics*, **13** (1), 1–50.

[31] Faquih, T., Van Smeden, M., Luo, J., Le Cessie, S., Kastenmüller, G., Krumsiek, J., Noordam, R., Van Heemst, D., Rosendaal, F. R., Van Hylckama Vlieg, A., Willems Van Dijk, K. and Mook-Kanamori, D. O. (2020). A Workflow for Missing Values Imputation of Untargeted Metabolomics Data**.** *Metabolites,* **10** (12), 486.

[32] Feng, S., Hategeka, C. and Grépin, K. A. (2021). Addressing missing values in routine health information system data: an evaluation of imputation methods using data from the Democratic Republic of the Congo during the COVID-19 pandemic. *Population Health Metrics,* **19** (1), 44.

[33] Ferguson, K. K., Yu, Y., Cantonwine, D. E., Mcelrath, T. F., Meeker, J. D. and Mukherjee, B. (2018). Foetal ultrasound measurement imputations based on growth curves versus multiple imputation chained equation (MICE). *Paediatric and Perinatal Epidemiology,* **32** (5), 469–473.

[34] Fernández-Crehuet, J. M., Rosales-Salas, J. and De Ramos, S. (2019). State of health in the European Union: A European Health Index. *Journal of Healthcare Quality Research*, **34** (6), 308–313.

[35] Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis. Theory and Practice.* Statistics. Springer-Verlag.

[36] Fratello, M., Cattelani, L., Federico, A., Pavel, A., Scala, G., Serra, A. and Greco, D. (2022). Unsupervised Algorithms for Microarray Sample Stratification. *Methods Mol Biol.* **2401**, 121–146.

[37] Franchuk, V. V., Mikhaylichenko, B. V. and Franchuk, M. V. (2020). Primenenie metoda dereva reshenii v sudebno-meditsinskoĭ ékspertnoĭ praktike pri analize 'vrachebnykh del' [Application of the decision tree method in forensic-medical practice in the analysis of 'doctors proceedings']. *Sudebno-meditsinskaia ekspertiza*, **63** (1), 9–14.

[38] Gheondea-Eladi, A. (2019). Patient decision aids: a content analysis based on a decision tree structure. *BMC Med Inform Decis Mak.* **19** (1), 137.

[39] Gil Martínez, C. (2018). *Análisis de Componentes Principales (PCA),* Available at https://rpubs.com/Cristina_Gil/PCA

[40] González, I. and Déjean, S. (2021). *CCA: Canonical Correlation Analysis*. R package version 1.2.1. Available at https://CRAN.R-project.org/package=CCA

[41] González, I., Déjean, S., Martin, P. G. P. and Baccini, A. (2008). CCA: An R Package to Extend Canonical Correlation Analysis. *Journal of Statistical Software*, **23** (12), 1–14.

[42] González, J. and Muñoz, A. (2013). Functional analysis techniques to improve similarity matrices in discrimination problems. *J. Multivar. Anal.* **120**, 120–134.

[43] Granville, V. (2019). *How to Automatically Determine the Number of Clusters in your Data - and more.* Available at https://www.datasciencecentral.com/profiles/blogs/how-to-automatically-determine-the-number-of-clusters-in-your-dat

[44] Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems*. **17**, 107–145.

[45] Hamel, L. H. (2009). *Knowledge Discovery with Support Vector Machines.* John Wiley and Sons. Hoboken, New Jersey.

[46] Hanko, M., Grendár, M., Snopko, P., Opšenák, R., Šutovský, J., Benčo, M., Soršák, J., Zeleňák, K. and Kolarovszki, B. (2021). Random Forest-Based Prediction of Outcome and Mortality in Patients with Traumatic Brain Injury Undergoing Primary Decompressive Craniectomy. *World Neurosurgery*, **148**, e450–e458.

[47] Harrell Jr, F. E. and Dupont, C. (2021). *Hmisc: Harrell Miscellaneous.* R package version 4.5-0. Available at https://CRAN.R-project.org/package=Hmisc

[48] Holger Diedrich, M.A. (2015). *LLE: Locally linear embedding.* R package version 1.1. Available at https://cran.r-project.org/web/packages/lle/lle.pdf

[49] Honaker, J., King, G. and Blackwell, M. (2011). Amelia II: A Program for Missing Data. *Journal of Statistical Software*, **45** (7), 1–47.

[50] Hong, S. and Lynn, H. S. (2020). Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction. *BMC Medical Research Methodology*, **20** (1), 199.

[51] IBM Corporation (2021). *Modelos de árboles de decisión (decision tree models).* Available at https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=trees-decision-tree-models

[52] Im, E. O., Yi, J. S. and Chee, W. (2021). A decision tree analysis on multiple factors related to menopausal symptoms. *Menopause*. **28** (7), 772–786.

[53] Ispirova, G., Eftimov, T. and Seljak, B. K. (2020). Evaluating missing value imputation methods for food composition databases**.** *Food and Chemical Toxicology: An International Journal Published for The British Industrial Biological Research Association*, **141**, 111368.

[54] Jaya Lakshmi, B., Shashi, M. and Madhuri, K. B. (2020). A rough set based subspace clustering technique for high dimensional data. *Journal of King Saud University - Computer and Information Sciences*. **32** (3), 329–334.

[55] Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A.,* **374**, 20150202.

[56] Jopia, H. (2019). *smbinning: Scoring Modeling and Optimal Binning*. R package version 0.9. Available at https://CRAN.R-project.org/package=smbinning

[57] Karacan, I., Sennaroglu, B. and Vayvay, O. (2020). Analysis of life expectancy across countries using a decision tree. *East Mediterr Health J*. **26** (2), 143–151.

[58] Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004). kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software,* **11** (9), 1–20.

[59] Karthe (2016). *Tutorial on 5 Powerful R Packages used for imputing missing value.* Available at https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

[60] Kassambara, A. and Mundt, F. (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses.* R package version 1.0.7. Available at https://CRAN.R-project.org/package=factoextra

[61] Kraemer, G. (2022). *"dimRed": A Framework for Dimensionality Reduction.* The R Journal. Available at https://cran.r-project.org/web/packages/dimRed/dimRed.pdf

[62] Kraemer, G., Reichstein, M. and Mahecha, M. D. (2018). "dimRed and coRanking-Unifying Dimensionality Reduction in R." *The R Journal*, **10** (1), 342–358. coRanking version 0.2.5.

[63] Krijthe, J. H. (2015). *Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation*. Available at https://github.com/jkrijthe/Rtsne

[64] Kuroda, M., Mori, Y., Masaya, I. and Sakakihara, M. (2013). Alternating least squares in nonlinear principal components. *WIREs Comput Stat*, **5**, 456–464.

[65] Larsen, K. (2015). *Data Exploration with Weight of Evidence and Information Value in R.* Available at https://multithreaded.stitchfix.com/blog/2015/08/13/weight-of-evidence/

[66] Lê, S., Josse, J. and Husson, F. (2008). FactoMineR: An R Package for Multivariate Analysis. *Journal of Statistical Software*, **25** (1), 1–18.

[67] Legorreta, D. (2015). *Máquina de soporte Vectorial (SVM-Support Vector Machine).* Available at https://dlegorreta.wordpress.com/2015/04/07/maquina-de-soporte-vectorial-svm-sopport-vector-machine/

[68] Lenz, M., Schulz, A., Koeck, T., Rapp, S., Nagler, M., Sauer, M., Eggebrecht, L., Ten Cate, V., Panova-Noeva, M., Prochaska, J. H., Lackner, K. J., Münzel, T., Leineweber, K., Wild, P. S. and Andrade-Navarro, M. A. (2020). Missing value imputation in proximity extension assay-based targeted proteomics data. *PloS One,* **15** (12), e0243487.

[69] Li, G., Wang, C., Zhang, D. and Yang, G. (2021). An Improved Feature Selection Method Based on Random Forest Algorithm for Wind Turbine Condition Monitoring. *Sensors,* **21** (16), 5654.

[70] Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News*. **2** (3), 18–22.

[71] Linting, M. (2007). *Nonparametric inference in nonlinear principal components analysis: Exploration and beyond.* Available at https://hdl.handle.net/1887/12386

[72] Liu, W., Yuan, K. and Ye, D. (2008). Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis. *J Biomed Inform.* **41** (4), 602–606.

[73] López-Campos, J. L., Almagro, P., Gómez, J. T., Chiner, E., Palacios, L., Hernández, C., Navarro, M. D, Molina, J., Rigau, D., Soler-Cataluña, J. J, Calle, M., G-Cosío, B., Casanova, C. and Miravitlles, M. (2021). Actualización de la Guía Española de la EPOC (GesEPOC): Comorbilidades, automanejo y cuidados paliativos. *Archivos de Bronconeumología.* Available at https://doi.org/10.1016/j.arbres.2021.08.002

[74] López, R. F. and Fernández, J. M. F. (2008). *Las Redes Neuronales Artificiales. Metodología y Análisis de Datos en Ciencias Sociales.* Netbiblo.

[75] López Cano, E. (2018). *Análisis de correspondencias con R: aplicación a datos de encuestas.* Available at https://emilopezcano.github.io/seminario_urjc_2018/readme.html

[76] Lowie, T., Callens, J., Maris, J., Ribbens, S. and Pardon, B. (2021). Decision tree analysis for pathogen identification based on circumstantial factors in outbreaks of bovine respiratory disease in calves. *Preventive Veterinary Medicine*. **196**, 105469.

[77] Luo, Q., Egger, S., Yu, X. Q., Smith, D. P. and O'Connell, D. L. (2017). Validity of using multiple imputation for "unknown" stage at diagnosis in population-based cancer registry data. *PloS One,* **12** (6), e0180033.

[78] Luo, Y., Szolovits, P., Dighe, A. S. and Baron, J. M. (2018). 3D-MICE: integration of cross-sectional and longitudinal imputation for multi-analyte longitudinal clinical data**.** *Journal of the American Medical Informatics Association*, **25** (6), 645–653.

[79] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2021). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.2. Available at https://CRAN.R-project.org/package=cluster

[80] Mair, P. and De Leeuw, J. (2019). *Gifi: Multivariate Analysis with Optimal Scaling*. R package version 0.3-9. Available at https://CRAN.R-project.org/package=Gifi

[81] Manisera, M., Van Der Kooij, A. J. and Dusseldorp, E. (2010). Identifying the Component Structure of Satisfaction Scales by Nonlinear Principal Components Analysis. *Quality Technology and Quantitative Management*, **7** (2), 97–115.

[82] Marston, L., Carpenter, J. R., Walters, K. R., Morris, R. W., Nazareth, I. and Petersen, I. (2010). Issues in multiple imputation of missing data for large general practice clinical databases. *Pharmacoepidemiology and Drug Safety,* **19** (6), 618–626.

[83] Martín de Diego, I., Muñoz, A. and Moguerza, J. M. (2010). Methods for the combination of kernel matrices within a support vector framework. *Mach. Learn.* **78** (1-2), 137–174.

[84] Martínez De Lejarza, I. and Esparducer (2013). *Árboles de clasificación y regression*. Available at https://www.uv.es/mlejarza/actuariales/tam/arbolesdecision.pdf

[85] Martínez Heras, J. (2020). *Árboles de Decisión con ejemplos en Python. Árboles de Decisión para Clasificación*. Available at https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/#Arboles_de_Decision_para_Clasificacion

[86] Martos, G., Muñoz, A. and González, J. (2014). Generalizing the Mahalanobis distance via density kernels. *Intell. Data Anal.* **18**, S19–S31.

[87] Mera-Gaona, M., Neumann, U., Vargas-Canas, R. and López, D. M. (2021). Evaluating the impact of multivariate imputation by MICE in feature selection. *PloS One,* **16** (7), e0254720.

[88] Meulman, J. J., Van Der Kooij, A. J. and Heiser, W. J. (2004). Principal components analysis with nonlinear optimal scaling transformations for ordinal and nominal data. *SAGE Publications, Inc*. *In The SAGE handbook of quantitative methodology for the social sciences,* 50–71.

[89] Meulman, J. J., Van Der Kooij, A. J. and Babinec, A. (2002). New Features of Categorical Principal Components Analysis for Complicated Data Sets, Including Data Mining. *In: Gaul W., Ritter G. (eds) Classification, Automation, and New Media. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg*, 207–217.

[90] Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. and Leisch, F. (2021). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071).* TU Wien. R package version 1.7-7. Available at https://CRAN.R-project.org/package=e1071

[91] Miri, H. H., Hassanzadeh, J., Khaniki, S. H., Akrami, R. and Sirjani, E. B. (2020). Accuracy of Five Multiple Imputation Methods in Estimating Prevalence of Type 2 Diabetes based on STEPS Surveys. (MICE-PMM). *Journal of Epidemiology and Global Health,* **10** (1), 36–41.

[92] Mirzal, A. (2020). Statistical Analysis of Microarray Data Clustering using NMF, Spectral Clustering, Kmeans and GMM. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* **21**, pp. Epub ahead of print.

[93] Moguerza, J. M., Muñoz, A. and Martos Venturini, G. (2020). Support Vector Regression. Enciclopedia Wiley de Estadística. *Wiley StatsRef: Statistics Reference Online*, 1–8.

[94] Moguerza, J. M., Muñoz, A. and Psarakis, S. (2007). *Monitoring Nonlinear Profiles Using Support Vector Machines.* Springer-Verlag Berlin Heidelberg, 574–583.

[95] Moguerza, J. M. and Muñoz, A. (2006a). Support Vector Machines with Applications. *Statistical Science*, **21** (3), 322–336.

[96] Moguerza, J. M. and Muñoz, A. (2006b). Rejoinder to Support Vector Machines with Applications. *Statistical Science*, **21** (3), 358–362.

[97] Molina Molina, Ó. and Espinosa de los Monteros Pérez, E. (2010). Rotación en Análisis de Componentes Principales Categórico: un caso práctico. *Metodología de Encuestas*, **12**, 63–88.

[98] Muñoz, A. and González, J. (2010). Representing functional data using support vector machines. *Pattern Recognit. Lett.* **31** (6), 511–516.

[99] Muñoz, A., Hernández, N., Moguerza, J. M. and Martos, G. (2018). Combining Entropy Measures for Anomaly Detection. *Entropy*. **20** (9), 698.

[100] Muñoz, A., Moguerza, J. M. and Martos Venturini, G. (2019). Support Vector Machines. Enciclopedia Wiley de Estadística. *Wiley StatsRef: Statistics Reference Online*, 1–8.

[101] Navarro-Mateu, F., Garriga-Puerto, A. and Sánchez-Sánchez, J. A. (2010). Análisis de las alternativas terapéuticas del trastorno de pánico en atención primaria mediante un árbol de decisión [Tree decision analysis of the therapeutic alternatives for Panic Disorders in Primary Care]. *Aten Primaria*. **42** (2), 86–94. Spanish.

[102] Nenadic, O. and Greenacre, M. (2007). Correspondence Analysis in R, with two-and three-dimensional graphics: The ca package. *Journal of Statistical Software*. **20** (3), 1–13.

[103] Neto, J. P. (2013). *Support Vector Machines. Introduction*. Available at http://www.di.fc.ul.pt/~jpn/r/svm/svm.html#introduction

[104] Niño-Ramírez, S., Jaramillo-Arroyave, D., Ardila, O. and Guevara-Casallas, L. G. (2021). Reducing the heterogeneity in hepatocellular carcinoma. A cluster analysis based on clinical variables in patients treated at a quaternary care hospital. *Rev Gastroenterol Mex (Engl Ed).* **86** (4), 356–362. English, Spanish.

[105] Oksanen, J., Simpson, G. L., Guillaume Blanchet, F., et al. (2022). *vegan: Community Ecology Package*. R package version 2.6-2. Available at https://CRAN.R-project.org/package=vegan

[106] Orellana Alvear, J. (2018). *Árboles de decisión y Random Forest. Árboles de Decisión - Parte I.* Available at https://bookdown.org/content/2031/arboles-de-decision-parte-i.html

[107] Ortiz, M. T. and González, F. (2015). *Curso Estadística Multivariada. Mecanismos de faltantes (MCAR, MAR, MNAR).* Available at https://est-mult.netlify.app/index.html

[108] Ortiz-Gonçalves, B., Perea-Pérez, B., Labajo González, E., Albarrán Juan, E. and Santiago-Sáez, A. (2018). Tipologías de los madrileños ante la etapa final de la vida mediante un análisis de clusters [Typologies of Madrid's citizens (Spain) at the end-of-life: cluster analysis]. *Spanish. Gac Sanit.* **32** (4), 346–351.

[109] Parsai, T. and Kumar, A. (2021). Weight-of-evidence (WOE) process for assessing human health risk of mixture of metal oxide nanoparticles and corresponding ions in aquatic matrices. *Chemosphere,* **263**, 128289.

[110] Pasha, A. and Latha, P. H. (2020). Bio-inspired dimensionality reduction for Parkinson's disease (PD) classification. *Health Information Science and Systems*, **8** (1), 13.

[111] Pedregosa *et al. (*2011a). *Scikit-learn: Machine Learning en Python. Unsupervised_learning.* JMLR,12, 2825–2830. Available at https://scikit-learn.org/stable/modules/clustering.html#

[112] Pedregosa *et al. (*2011b). *Scikit-learn: Machine Learning en Python. Supervised_learning.* JMLR,12, 2825–2830. Available at https://scikit-learn.org/stable/supervised_learning.html

[113] Pinheiro, L., Pereira, M., Fernandez, M. P., Filho, F., De Abreu, W. and Pinheiro, P. (2021). Application of Data Mining Algorithms for Dementia in People with HIV/AIDS. *Computational and Mathematical Methods in Medicine*, **2021**, 4602465.

[114] Pozo-Rodríguez, F., Alvarez, C. J., Castro-Acosta, A., Melero Moreno, C., Capelastegui, A., Esteban, C., Hernández Carcereny, C., López-Campos, J. L., Izquierdo Alonso, J. L., López Quílez, A., Agustí, A. and Grupo AUDIPOC ESPAÑA (2010). Clinical audit of patients admitted to hospital in Spain due to exacerbation of COPD (AUDIPOC study): method and organisation. *Arch Bronconeumol,* **46** (7), 349–357.

[115] Prabhakaran, S. (2016). *Feature Selection Approaches (Random Forest, Information Value, Others).* Available at http://r-statistics.co/Variable-Selection-and-Importance-With-R.html

[116] Quantsignals (2012). *Learning Kernels SVM.* Available at http://www.r-bloggers.com/learning-kernels-svm/

[117] R Core Team (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

[118] Rajaguru, H. and S R, S. C. (2019). Analysis of Decision Tree and K-Nearest Neighbor Algorithm in the Classification of Breast Cancer. *Asian Pac J Cancer Prev*. 20 (12), 3777–3781.

[119] Ramsay, J. O., Graves, S. and Hooker, G. (2020). *fda: Functional Data Analysis*. R package version 5.1.9. Available at https://CRAN.R-project.org/package=fda

[120] Ramsay, J. O. and Silverman, B. W. (1997). *Functional Data Analysis.* Springer. New York.

[121] Rangel, J., Perea, J., De-Pablos-Heredero, C., Espinosa-García, JA., Mujica, PT., Feijoo, M., Barba, C. and García, A. (2020). Structural and Technological Characterization of Tropical Smallholder Farms of Dual-Purpose Cattle in Mexico. *Animals (Basel)*. **10** (1), 86.

[122] Revelle, W. (2021). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston, Illinois, USA. R package version 2.1.6. Available at URL: https://CRAN.R-project.org/package=psych

[123] Revelle, W. (2008). *fa.parallel: Scree plots of data or correlation matrix compared to random "parallel" matrices*. Available at https://www.rdocumentation.org/packages/psych/versions/1.0-58/topics/fa.parallel

[124] Rokach, L. and Maimon, O. (2007). Minería de datos con árboles de decisión. Teoría y Aplicaciones. *Serie en Percepción de Máquinas e Inteligencia Artificial.* Chapters 1, 6 and 10. **69**, 264.

[125] Rossiter, D. G. (2021). *Nonlinear Principal Components Analysis: Multivariate Analysis with Optimal Scaling (MVAOS)*. Available at http://www.css.cornell.edu/faculty/dgr2/_static/files/R_html/NonlinearPCA.html

[126] Salvador, R., Verdolini, N., Garcia-Ruiz, B., Jiménez, E., Sarró, S., Vilella, E., VIETA, E., Canales-Rodríguez, E. J., Pomarol-Clotet, E. and Voineskos, A. N. (2020). Multivariate Brain Functional Connectivity Through Regularized Estimators. *Frontiers in Neuroscience,* **14**, 569540.

[127] Sánchez Pantigoso, C. F. (2019). *Análisis de componentes. Ejemplo del Uso de PCA en R.* Available at https://rpubs.com/Csanchez15/551258

[128] Sancho Caparrini, F. (2020). *Algoritmos de Clustering. Density-Based Spatial Clustering of Applications with Noise (DBSCAN).* Available at http://www.cs.us.es/~fsancho/?e=230

[129] Santana, E. (2015). Machine Learning con R. *Ejemplos de Machine Learning y Data Mining con R - Imputar con Regresion Lineal*. Available at https://apuntes-r.blogspot.com/2015/05/imputar-con-regresion-lineal.html

[130] SAS Institute Inc. (2018). *SAS Campus Drive.* Cary, North Carolina 27513, USA. All rights reserved. Copyright. Available at http://www.sas.com/

[131] Saucedo Mendieta, L. A. (2019). *Tarea PCA-Cluster*. Available at https://rpubs.com/LuisSaucedo/497651

[132] Saxe, K. (2002). *Beginning Functional Analysis.* Springer-Verlag Berlin Heidelberg, New York.

[133] Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O. and Hemingway, H. (2014). Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using MICE: A CALIBER Study. *American Journal of Epidemiology*, **179** (6), 764–774.

[134] Siuly, S. and Li, Y. (2015). Designing a robust feature extraction method based on optimum allocation and principal component analysis for epileptic EEG signal classification**.** *Computer Methods and Programs in Biomedicine*, **119** (1), 29–42.

[135] Slade, E. and Naylor, M. G. (2020). A fair comparison of tree-based and parametric methods in multiple imputation by chained equations (MICE). *Statistics in Medicine,* **39** (8), 1156–1166.

[136] Stekhoven, D. J. (2022). *missForest: Nonparametric Missing Value Imputation using Random Forest*. R package version 1.5.

[137] Stekhoven, D. J. and Buehlmann, P. (2012). MissForest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, **28** (1), 112–118.

[138] Su, Y.-S., Gelman, A., Hill, J. and Yajima, M. (2011). Multiple Imputation with Diagnostics (mi) in R: Opening Windows into the Black Box. *Journal of Statistical Software*, **45** (2), 1–31.

[139] SPSS Inc. (2008). *CATPCA is available in SPSS® Statistics Professional Edition or the Categories option.* SPSS Statistics para Windows, versión 17.0. Chicago. Available at http://www.spss.com/

[140] Tăuţan, A. M., Rossi, A. C., De Francisco, R. and Ionescu, B. (2020). Dimensionality reduction for EEG-based sleep stage detection: comparison of autoencoders, principal component analysis and factor analysis. *Biomedizinische Technik/Biomedical Engineering,* **66** (2), 125–136.

[141] Theano Team (2015). *Tutorial, Deep Learning. Release 0.1***.** LISA lab, University of Montreal. Copyright Theano Development Team.

[142] Therneau, T. and Atkinson, B. (2019). *rpart: Recursive Partitioning and Regression Trees.* R package version 4.1-15. Available at https://CRAN.R-project.org/package=rpart

[143] Thioulouse, J., Dray, S., Dufour, A., Siberchicot, A., Jombart, T. and Pavoine, S. (2018). *Multivariate Analysis of Ecological Data with ade4*. Springer. Available at URL: https://doi.org/10.1007/978-1-4939-8850-1

[144] Tobón, Á., Rueda, J., Cáceres, D. H., Mejía, G. I., Zapata, E. M., Montes, F., Ospina, A., Fadul, S., Paniagua, L. and Robledo, J. (2020). Adverse treatment outcomes in multidrug resistant tuberculosis go beyond the microbe-drug interaction: Results of a multiple correspondence analysis. *Biomedica.* **40** (4), 616–625.

[145] Vaissie, P., Monge, A. and Husson, F. (2021). *Factoshiny: Perform Factorial Analysis from 'FactoMineR' with a Shiny Application.* R package version 2.4. Available at https://CRAN.R-project.org/package=Factoshiny

[146] Valencia-Aguirre, J., Daza-Santacoloma, G., Acosta, C. D. and Castellanos-Domínguez, G. (2010). Comparación de métodos de reducción de dimensión basados en análisis por localidades. *Rev. Tecno Lógicas*, **25**, 131–150.

[147] Van Buuren, S. and Groothuis-Oudshoorn, K. (2011). Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, **45** (3), 1–67.

[148] Vaquerizo, R. (2014). *Análisis y Decisión. Representación de redes neuronales en R.* Available at http://analisisydecision.es/?s=Redes+neuronales

[149] Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). UNA ENCUESTA DE ALGORITMOS DE CONJUNTO DE AGRUPACIÓN EN CLÚSTERES. *Revista Internacional de Reconocimiento de Patrones e Inteligencia Artificial*. **25** (3), 337–372.

[150] Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Fourth Edition. Springer, New York. ISBN 0-387-95457-0. Available at https://www.stats.ox.ac.uk/pub/MASS4/

[151] Villagarcía, T. and Muñoz, A. (1997). Imputación de datos censurados mediante redes neuronales: una aplicación a la EPA. *Cuadernos Economicos de ICE*, **63**, 193–204.

[152] Wang, L., Zhu, L., Jiang, J., Wang, L. and Ni, W. (2021a). Decision tree analysis for evaluating disease activity in patients with rheumatoid arthritis. *J Int Med Res*. **49** (10), 3000605211053232.

[153] Wang, X., Zhai, M., Ren, Z., Ren, H., Li, M., Quan, D., Chen, L. and Qiu, L. (2021b). Exploratory study on classification of diabetes mellitus through a combined Random Forest Classifier. *BMC Medical Informatics and Decision Making*, **21** (1), 105.

[154] Wickham, H. (2021). *Visualización de datos usando el paquete "ggplot2"*. Available at https://es.r4ds.hadley.nz/visualizaci%C3%B3n-de-datos.html

[155] Wurst, K. E., Sumner, K. M., Stanislaus, D., Powell, M. and Cunnington, M. (2020). A model for human and animal data integration: Weight of evidence (WOE) strategy. *Birth Defects Research*, **112** (18), 1505–1512.

[156] Yang, L., Wu, H., Jin, X., Zheng, P., Hu, S., Xu, X., Yu, W. and Yan, J. (2020). Study of cardiovascular disease prediction model based on random forest in eastern China**.** *Scientific Reports*, **10** (1), 5245.

[157] You, K. (2022). *Rdimtools: Dimension Reduction and Estimation Methods*. R package version 1.0.9. Available at https://CRAN.R-project.org/package=Rdimtools

[158] Zahid, F. M. and Heumann, C. (2019). Multiple imputation with sequential penalized regression. *Statistical Methods in Medical Research*, **28** (5), 1311–1327.

[159] Zhang, Z. (2016). Multiple Imputation for time series data with Amelia package. *Annals of Translational Medicine*, **4** (3), 56.

[160] Zheng, C. H., Huang, D. S., Zhang, L. and Kong, X. Z. (2009). Tumor clustering using nonnegative matrix factorization with gene selection. *IEEE Trans Inf Technol Biomed.* **13** (4), 599–607.

# ANEXOS. Acrónimos, definición de variables y scripts con software R

## Anexo A. Definición de variables

| | |
|---|---|
| EDAD | Edad del paciente (años) |
| SEXO | Sexo del paciente (Male/Female) |
| HT_ | Hábito tabáquico (fumador) |
| DURING | Duración del ingreso en el centro hospitalario (días) |
| ALP | Altura del paciente (metros) |
| PPKG | Peso del paciente (kg) |
| IMC | Índice de Masa Corporal |
| TARS | Tensión Arterial Sistólica (mmHg) |
| TARD | Tensión Arterial Diastólica (mmHg) |
| TURA | Temperatura del paciente ($^0$C) |
| FRE | Frecuencia Respiratoria (resp./min) |
| FCA | Frecuencia Cardíaca (lat/min) |
| FEV1 | Volumen espiratorio forzado en el primer segundo |
| FEV1P | FEV1 espirometría en % del teórico |
| FVC | Capacidad vital forzada |
| FVCP | FVC espirometría en % del teórico |
| FEVCVF_ | Espirometría previa o alta |
| ESPIROMETRIA_PA_ | Pacientes que tienen la espirometría previa al ingreso o al alta |
| INGRESOS_ | ¿Ha tenido el paciente algún ingreso hospitalario por cualquier motivo en los 90 días posteriores? |
| SV_ | ¿Recibió soporte ventilatorio en cualquier momento del ingreso? |
| EXACER_90DIAS | Pacientes que reingresaron por exacerbación de EPOC a los 90 días respecto a la fecha de ingreso |
| Reing_EXAC | Reingresos por exacerbación de EPOC |
| MUERTOS_90DIAS | Pacientes que murieron a los 90 días |
| EXITUS | Fallecimiento del paciente durante todo el periodo de ingreso |
| ICHARICC_ | Insuficiencia cardíaca congestiva |
| CCVSDM_ | Comorbilidad Cardiovascular |
| ICHAR_DM_ | Diabetes Mellitus |
| EV_ | Enfermedad Vascular |
| ICHARECV_ | Enfermedad cerebro vascular |
| ICHAREVP_ | Enfermedad vascular periférica |
| ICHARIM_ | Infarto de miocardio |
| ICHARNEF_ | Nefropatía |
| ICHAR_TS_ | Tumor sólido |
| EP_ | Edemas maleolares |

# Anexo B. Acrónimos

| | |
|---|---|
| PCA | Análisis de componentes principales o Principal Component Analysis |
| FPCA | Análisis de componentes principales funcionales |
| FDA | Análisis funcional de datos |
| APS-REM | Análisis paralelo con datos simulados y remuestreo de datos |
| RF | Bosque aleatorio |
| RF&IV | Bosque aleatorio según el índice de Gini y el valor de la información según el peso de la evidencia |
| PMM | Coincidencia de medias predictiva |
| MVN | Distribución normal multivariante |
| EPOC | Enfermedad pulmonar obstructiva crónica |
| eEPOC | Exacerbación de la enfermedad pulmonar obstructiva crónica |
| EMB | Expectativa-Maximización con Bootstrapping |
| MAR | Falta al azar |
| OOB | Fuera de bolsa (error OOB) |
| mi | Imputación múltiple con diagnóstico |
| MICE | Imputación múltiple por ecuaciones encadenadas |
| AI o IA | Artificial intelligence o Inteligencia artificial |
| WOE | Peso de la evidencia |
| IV | Valor de la información |
| ML | Machine Learning o aprendizaje autómatico |
| RL | Regresión Logística |
| SVM | Support Vector Machine o Máquinas de Vectores Soportes |
| CA | Correspondence Analysis o Análisis de Correspondencias |
| DT | Decision Tree o Árboles de Decisión por clasificación |
| KNN | K-Nearest-Neighbor o método de los K vecinos más cercano |
| ROC | Área bajo la curva |
| ECM | Error Cuadrático Medio |
| $R^2$ | Coeficiente de determinación |
| N(0,1) | Distribución Normal con media cero y desviación típica uno |
| UPGMA | Unweighted Pair Group Method with Arithmetic mean o método de grupos de pares no ponderados con media aritmética |
| WPGMA | Weighted Pair Group Method with Arithmetic mean o método de grupos de pares ponderada con media aritmética |
| WPGMC | Weighted Pair Group Method with Centroid o método de grupos de pares ponderados con el centroide |
| UPGMC | Unweighted Pair Group Method with Centroid o método de grupos de pares no ponderados con el centroide |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise o algoritmo de Clustering espacial basado en la densidad de aplicaciones con ruido |
| CH | Índice de Calinski-Harabasz |
| PAM | Particionamiento Alrededor de los Medoides |
| AF | Análisis Factorial |

| | |
|---|---|
| AD | Análisis Discriminante |
| ACS | Análisis de Correspondencias Simple |
| ACM | Análisis de Correspondencias Múltiple |
| RKHS | Reproducing Kernel Hilbert Spaces o Espacio de Hilbert del Núcleo Reproductor |
| ADF | Análisis de Datos Funcionales |
| SVMs | Vectores Soportes |

# Anexo C. Scripts con software R

*R code – Capítulos I II III IV*

**######################################## CAPITULO I ########################################**

*### EXPLORACIÓN Y PREPROCESAMIENTO PREVIO PARA CONOCER LOS DATOS SELECCIONADOS*

**# SINTAXIS INICIAL – RESUMEN ESTADÍSTICO**

```
#LISTA INICIAL DE PAQUETES que vamos a usar en estos primeros desarrollos
paquetes <- c("e1071", "fda", "splines", "Matrix", "MASS", “nnet”, “mice”, "lattice", "VIM", “cluster”, ”ade4”,
"graphics", "broom", "Rcpp", "robustbase", "sp", "curl" , "haven" , "colorspace", "grid", "data.table",
"lmtest", "zoo", “base” )
#Crea un vector lógico si están instalados o no y s.i hay al menos uno no instalado los instala
instalados <- paquetes %in% installed.packages()
if(sum(instalados == FALSE) > 0) {
  install.packages(paquetes[!instalados])
}
lapply(paquetes,require,character.only = TRUE)
datos<-read.table(file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_220616.txt", header=T)
#EDAD_: Edad estatificada en grupos; Menor que 68 -> (<Q1) ; Entre [68-80] -> (=Q1-Q3) ; Mayor que 80 -> (>Q3);
mean(datos$EDAD)   #[1] 73.39262
median(datos$EDAD) #[1] 75
quantile(datos$EDAD,probs = c(0.25,0.75)) #25% 75%  68  80
EDAD_<-cut(datos$EDAD,breaks=c(0,67,80,99))
table(EDAD_) #EDAD_         (0,67] (67,80] (80,99]     1287    2604    1287
sum(table(EDAD_)) #[1] 5178
#DURING_: Duración ingreso hosp. en grupos; Menor que 6 -> (<Q1) ; Entre [6-12] -> (=Q1-Q3) ; Mayor que 12 ->
(>Q3);
mean(datos$DURING)#[1] 9.955195
median(datos$DURING)#[1] 8
quantile(datos$DURING,probs = c(0.25,0.75)) #25% 75%   6  12
DURING_<-cut(datos$DURING,breaks=c(0,5,12,130))
table(DURING_) #DURING_   (0,5]   (5,12] (12,130]     1278     2776     1124
sum(table(DURING_)) #[1] 5178
```

**# SINTAXIS INICIAL – GRÁFICOS - BOX PLOT**

```
datos<-read.table(file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_191216.txt",
header=T)
boxplot(datos$EDAD,main="Edad del paciente")
plot(table(datos$EDAD_),col = "grey",lwd = 10)
plot(datos$SEXO,main="Sexo del paciente")
plot(datos$HT_,main="Habito tabáquico")
boxplot(datos$DURING,main="Duración del ingreso hospitalario")
table(datos$DURING_)
plot(table(datos$DURING_),col = "grey",lwd = 10)
ALP=as.numeric(datos$ALP)
summary(ALP) # Min. 1st Qu. Median Mean 3rd Qu. Max.  1.00  1.00  1.00   10.47   23.00   51.00
```

```
boxplot(ALP, main="Altura del paciente")
PPKG=as.numeric(datos$PPKG)
summary(PPKG) # Min. 1st Qu. Median Mean 3rd Qu. Max.  1.00  1.00  1.00   26.74   66.00   99.00
boxplot(PPKG, main="Peso del paciente")
IMC=as.numeric(datos$IMC)
boxplot(IMC, main="IMC del paciente")
TARS=as.numeric(datos$TARS)
boxplot(TARS, main="Tensión Arterial Sistólica")
TARD=as.numeric(datos$TARD)
boxplot(TARD, main="Tensión Arterial Diastólica")
TURA=as.numeric(datos$TURA)
boxplot(TURA, main="Temperatura del paciente")
FRE=as.numeric(datos$FRE)
boxplot(FRE, main="Frecuencia respiratoria")
FCA=as.numeric(datos$FCA)
boxplot(FCA, main="Frecuencia cardiaca")
FEV1P=as.numeric(datos$FEV1P)
boxplot(FEV1P, main="FEV1 espirometría")
FVCP=as.numeric(datos$FVCP)
boxplot(FVCP, main="FVC espirometría")
FEVCVF_=as.numeric(datos$FEVCVF_)
boxplot(FEVCVF_, main="Espirometría previa o alta")
plot(table(datos$ESPIROMETRIA_PA_),col = "grey",lwd = 15,main="Espirometría realizada    en preingreso o al
alta")
# SINTAXIS INICIAL – ANALIZAR EL TIPO DE IMPUTACIÓN APLICABLE A ESTE DATASET CON  NNET y MICE
# SINTAXIS INICIAL – MÉTODO DE IMPUTACIÓN RED NEURONAL – NNET
# PRUEBAS IMPUTACIÓN RED NEURONAL - NNET
library("e1071")
library("fda")
library("splines")
library("Matrix")
library("MASS")
library(nnet)
#datos<-read.table(file="C:/Users/Nisa B/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_191216.txt", header=T)
datos<-read.table(file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_150317.txt", header=T)
names(datos)
# [1] "NUMERO_AUDITORIA" "EDAD"          "SEXO"          "HT_"           "DURING"        "ALP"
# [7] "PPKG"          "IMC"           "TARS"          "TARD"          "TURA"          "FRE"
#[13] "FCA"           "FEV1P"         "FVCP"          "FEVCVF_"       "ESPIROMETRIA_PA_" "INGRESOS_"
#[19] "SV_"           "EXACER_90DIAS"   "Reing_EXAC"     "MUERTOS_90DIAS"  "EXITUS"        "ICHARICC_"
#[25] "CCVSDM_"       "ICHAR_DM_"     "EV_"           "ICHARECV_"     "ICHAREVP_"     "ICHARIM_"
#[31] "ICHARNEF_"     "ICHAR_TS_"     "EP_"           "SCORE_PAT_"    "EDAD_"         "DURING_"
ind = sample(5178,2500)
datos2 = datos[,c(1:36)]
summary(datos2)
etiq = datos2[,6]  #  ALP: Altura del paciente (metros)
etiq6 = as.numeric(etiq)
datos.train = datos2[-ind,]
```

```
datos.test  = datos2[ind,]
ytrain6 = etiq6[-ind]
ytest6  = etiq6[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
datos.train = as.numeric(datos.train)
datos.test = as.numeric(datos.test)
Tdatos.train = scale(datos.train) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test = scale(datos.test)   # Normalizar N(0,1) los datos de Test
Tdatos.train6 = scale(ytrain6) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test6 = scale(ytest6)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train6)
sd(Tdatos.train6)
summary(Tdatos.test6)
sd(Tdatos.test6)
show(Tdatos.train6)
nnet.fit6 <- nnet(Tdatos.train6 ~ datos.train , size=2, linout=T,trace=F)
#Ponemos la respuesta entre 0 y 1 dividiendo por 50 ya que por defecto los pesos salen entre -0.5 y 0.5.
nnet.fit6 <- nnet(Tdatos.train6/50 ~ datos.train , size=2, linout=T,trace=F)
nnet.fit6
#a 1-2-1 network with 7 weights
#inputs: ytrain6
#output(s): Tdatos.train6/50
#options were - linear output units
nnet.predict6 <- predict(nnet.fit6)*50
nnet.predict6
mean((nnet.predict6-Tdatos.train6)^2) # [1] 3.462665e-05
plot(Tdatos.train6, nnet.predict6, main="Neural network predictions vs altura", xlab="Transf_Altura")
show(Tdatos.test6)
nnet.fit6t <- nnet(Tdatos.test6 ~ ytest6 , size=2, linout=T,trace=F)
#Ponemos la respuesta entre 0 y 1 dividiendo por 50 ya que por defecto los pesos salen entre -0.5 y 0.5
nnet.fit6t <- nnet(Tdatos.test6/50 ~ ytest6 , size=2, linout=T,trace=F)
nnet.fit6t
#a 1-2-1 network with 7 weights
#Inputs: ytest6
#output(s): Tdatos.test6/50
#options were - linear output units
nnet.predict6t <- predict(nnet.fit6t)*50
nnet.predict6t
mean((nnet.predict6t-Tdatos.test6)^2) # [1] 5.295515e-05
plot(Tdatos.test6, nnet.predict6t, main="Neural network predictions vs altura", xlab="Transf_Altura")
etiq = datos[,7]     # PPKG: Peso del paciente (kg)
etiq7 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain7 = etiq7[-ind]
ytest7  = etiq7[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
```

```
Tdatos.train7 = scale(ytrain7) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test7 = scale(ytest7)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train7)
sd(Tdatos.train7)
summary(Tdatos.test7)
sd(Tdatos.test7)
show(Tdatos.train7)
nnet.fit7 <- nnet(Tdatos.train7/50 ~ ytrain7 , size=2, linout=T,trace=F)
nnet.fit7
#a 1-2-1 network with 7 weights
#inputs: ytrain7
#output(s): Tdatos.train7/50
#options were - linear output units
nnet.predict7 <- predict(nnet.fit7)*50
nnet.predict7
mean((nnet.predict7-Tdatos.train7)^2) #[1] 7.297735e-05
plot(Tdatos.train7, nnet.predict7, main="Neural network predictions vs peso", xlab="Transf_Peso")
show(Tdatos.test7)
nnet.fit7t <- nnet(Tdatos.test7/50 ~ ytest7 , size=2, linout=T,trace=F)
nnet.fit7t
#a 1-2-1 network with 7 weights
#inputs: ytest7
#output(s): Tdatos.test7/50
#options were - linear output units
nnet.predict7t <- predict(nnet.fit7t)*50
nnet.predict7t
mean((nnet.predict7t-Tdatos.test7)^2) #[1] 0.0001048142
plot(Tdatos.test7, nnet.predict7t, main="Neural network predictions vs peso", xlab="Transf_Peso")
etiq = datos[,8]     # IMC: Índice de Masa Corporal
etiq8 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain8 = etiq8[-ind]
ytest8  = etiq8[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train8 = scale(ytrain8) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test8 = scale(ytest8)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train8)
sd(Tdatos.train8)
summary(Tdatos.test8)
sd(Tdatos.test8)
show(Tdatos.train8)
nnet.fit8 <- nnet(Tdatos.train8/50 ~ ytrain8 , size=2, linout=T,trace=F)
nnet.fit8
#a 1-2-1 network with 7 weights
#inputs: ytrain8
#output(s): Tdatos.train8/50
```

```
#options were - linear output units
nnet.predict8 <- predict(nnet.fit8)*50
nnet.predict8
mean((nnet.predict8-Tdatos.train8)^2) #[1] 8.689667e-05
plot(Tdatos.train8, nnet.predict8, main="Neural network predictions vs IMC", xlab="Transf_IMC")
show(Tdatos.test8)
nnet.fit8t <- nnet(Tdatos.test8/50 ~ ytest8 , size=2, linout=T,trace=F)
nnet.fit8t
#a 1-2-1 network with 7 weights
#inputs: ytest8
#output(s): Tdatos.test8/50
#options were - linear output units
nnet.predict8t <- predict(nnet.fit8t)*50
nnet.predict8t
mean((nnet.predict8t-Tdatos.test8)^2) #[1] 9.89924e-05
plot(Tdatos.test8, nnet.predict8t, main="Neural network predictions vs IMC", xlab="Transf_IMC")
etiq = datos[,9]     # TARS: Tensión Arterial Sistólica
etiq9 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain9 = etiq9[-ind]
ytest9  = etiq9[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train9 = scale(ytrain9) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test9 = scale(ytest9)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train9)
sd(Tdatos.train9)
summary(Tdatos.test9)
sd(Tdatos.test9)
show(Tdatos.train9)
nnet.fit9 <- nnet(Tdatos.train9/50 ~ ytrain9 , size=2, linout=T,trace=F)
nnet.fit9
#a 1-2-1 network with 7 weights
#inputs: ytrain9
#output(s): Tdatos.train9/50
#options were - linear output units
nnet.predict9 <- predict(nnet.fit9)*50
nnet.predict9
mean((nnet.predict9-Tdatos.train9)^2) #[1] 9.027036e-05
plot(Tdatos.train9, nnet.predict9, main="Neural network predictions vs TARS", xlab="Transf_TARS")
show(Tdatos.test9)
nnet.fit9t <- nnet(Tdatos.test9/50 ~ ytest9 , size=2, linout=T,trace=F)
nnet.fit9t
#a 1-2-1 network with 7 weights
#inputs: ytest9
#output(s): Tdatos.test9/50
#options were - linear output units
```

```
nnet.predict9t <- predict(nnet.fit9t)*50
nnet.predict9t
mean((nnet.predict9t-Tdatos.test9)^2) #[1] 0.001235706
plot(Tdatos.test9, nnet.predict9t, main="Neural network predictions vs TARS", xlab="Transf_TARS")
etiq = datos[,10]    # TARD: Tensión Arterial Diastólica
etiq10 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain10 = etiq10[-ind]
ytest10  = etiq10[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train10 = scale(ytrain10) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test10 = scale(ytest10)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train10)
sd(Tdatos.train10)
summary(Tdatos.test10)
sd(Tdatos.test10)
show(Tdatos.train10)
nnet.fit10 <- nnet(Tdatos.train10/50 ~ ytrain10 , size=2, linout=T,trace=F)
nnet.fit10
#a 1-2-1 network with 7 weights
#inputs: ytrain10
#output(s): Tdatos.train10/50
#options were - linear output units
nnet.predict10 <- predict(nnet.fit10)*50
nnet.predict10
mean((nnet.predict10-Tdatos.train10)^2) #[1] 9.24249e-05
plot(Tdatos.train10, nnet.predict10, main="Neural network predictions vs TARD", xlab="Transf_TARD")
show(Tdatos.test10)
nnet.fit10t <- nnet(Tdatos.test10/50 ~ ytest10 , size=2, linout=T,trace=F)
nnet.fit10t
#a 1-2-1 network with 7 weights
#inputs: ytest10
#output(s): Tdatos.test10/50
#options were - linear output units
nnet.predict10t <- predict(nnet.fit10t)*50
nnet.predict10t
mean((nnet.predict10t-Tdatos.test10)^2) #[1] 9.283705e-05
plot(Tdatos.test10, nnet.predict10t, main="Neural network predictions vs TARD", xlab="Transf_TARD")
etiq = datos[,11]    # TURA: Temperatura del paciente (0C)
etiq11 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain11 = etiq11[-ind]
ytest11  = etiq11[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train11 = scale(ytrain11) # Normalizar N(0,1) los datos de entrenamiento
```

```
Tdatos.test11 = scale(ytest11)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train11)
sd(Tdatos.train11)
summary(Tdatos.test11)
sd(Tdatos.test11)
show(Tdatos.train11)
nnet.fit11 <- nnet(Tdatos.train11/50 ~ ytrain11 , size=2, linout=T,trace=F)
nnet.fit11
#a 1-2-1 network with 7 weights
#inputs: ytrain11
#output(s): Tdatos.train11/50
#options were - linear output units
nnet.predict11 <- predict(nnet.fit11)*50
nnet.predict11
mean((nnet.predict11-Tdatos.train11)^2) #[1] 8.200462e-05
plot(Tdatos.train11, nnet.predict11, main="Neural network predictions vs TURA", xlab="Transf_TURA")
show(Tdatos.test11)
nnet.fit11t <- nnet(Tdatos.test11/50 ~ ytest11 , size=2, linout=T,trace=F)
nnet.fit11t
#a 1-2-1 network with 7 weights
#inputs: ytest11
#output(s): Tdatos.test11/50
#options were - linear output units
nnet.predict11t <- predict(nnet.fit11t)*50
nnet.predict11t
mean((nnet.predict11t-Tdatos.test11)^2) #[1] 8.724976e-05
plot(Tdatos.test11, nnet.predict11t, main="Neural network predictions vs TURA", xlab="Transf_TURA")
etiq = datos[,12]    # FRE: Frecuencia Respiratoria
etiq12 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain12 = etiq12[-ind]
ytest12  = etiq12[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train12 = scale(ytrain12) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test12 = scale(ytest12)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train12)
sd(Tdatos.train12)
summary(Tdatos.test12)
sd(Tdatos.test12)
show(Tdatos.train12)
nnet.fit12 <- nnet(Tdatos.train12/50 ~ ytrain12 , size=2, linout=T,trace=F)
nnet.fit12
#a 1-2-1 network with 7 weights
#inputs: ytrain12
#output(s): Tdatos.train12/50
#options were - linear output units
```

```
nnet.predict12 <- predict(nnet.fit12)*50
nnet.predict12
mean((nnet.predict12-Tdatos.train12)^2) #[1] 9.189622e-05
plot(Tdatos.train12, nnet.predict12, main="Neural network predictions vs FRE", xlab="Transf_FRE")
show(Tdatos.test12)
nnet.fit12t <- nnet(Tdatos.test12/50 ~ ytest12 , size=2, linout=T,trace=F)
nnet.fit12t
#a 1-2-1 network with 7 weights
#inputs: ytest12
#output(s): Tdatos.test12/50
#options were - linear output units
nnet.predict12t <- predict(nnet.fit12t)*50
nnet.predict12t
mean((nnet.predict12t-Tdatos.test12)^2) #[1] 0.000330491
plot(Tdatos.test12, nnet.predict12t, main="Neural network predictions vs FRE", xlab="Transf_FRE")
etiq = datos[,13]    # FCA: Frecuencia Cardíaca
etiq13 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain13 = etiq13[-ind]
ytest13  = etiq13[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train13 = scale(ytrain13) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test13 = scale(ytest13)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train13)
sd(Tdatos.train13)
summary(Tdatos.test13)
sd(Tdatos.test13)
show(Tdatos.train13)
nnet.fit13 <- nnet(Tdatos.train13/50 ~ ytrain13 , size=2, linout=T,trace=F)
nnet.fit13
#a 1-2-1 network with 7 weights
#inputs: ytrain13
#output(s): Tdatos.train13/50
#options were - linear output units
nnet.predict13 <- predict(nnet.fit13)*50
nnet.predict13
mean((nnet.predict13-Tdatos.train13)^2) #[1] 8.594073e-05
plot(Tdatos.train13, nnet.predict13, main="Neural network predictions vs FCA", xlab="Transf_FCA")
show(Tdatos.test13)
nnet.fit13t <- nnet(Tdatos.test13/50 ~ ytest13 , size=2, linout=T,trace=F)
nnet.fit13t
#a 1-2-1 network with 7 weights
#inputs: ytest13
#output(s): Tdatos.test13/50
#options were - linear output units
nnet.predict13t <- predict(nnet.fit13t)*50
```

```
nnet.predict13t
mean((nnet.predict13t-Tdatos.test13)^2) #[1] 0.000130541
plot(Tdatos.test13, nnet.predict13t, main="Neural network predictions vs FCA", xlab="Transf_FCA")
etiq = datos[,14]    # FEV1P: FEV1 espirometría en % del teórico
etiq14 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain14 = etiq14[-ind]
ytest14  = etiq14[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train14 = scale(ytrain14) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test14 = scale(ytest14)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train14)
sd(Tdatos.train14)
summary(Tdatos.test14)
sd(Tdatos.test14)
show(Tdatos.train14)
nnet.fit14 <- nnet(Tdatos.train14/50 ~ ytrain14 , size=2, linout=T,trace=F)
nnet.fit14
#a 1-2-1 network with 7 weights
#inputs: ytrain14
#output(s): Tdatos.train14/50
#options were - linear output units
nnet.predict14 <- predict(nnet.fit14)*50
nnet.predict14
mean((nnet.predict14-Tdatos.train14)^2) #[1] 1.16667e-05
plot(Tdatos.train14, nnet.predict14, main="Neural network predictions vs FEV1", xlab="Transf_FEV1")
show(Tdatos.test14)
nnet.fit14t <- nnet(Tdatos.test14/50 ~ ytest14 , size=2, linout=T,trace=F)
nnet.fit14t
#a 1-2-1 network with 7 weights
#inputs: ytest14
#output(s): Tdatos.test14/50
#options were - linear output units
nnet.predict14t <- predict(nnet.fit14t)*50
nnet.predict14t
mean((nnet.predict14t-Tdatos.test14)^2) #[1] 7.00789e-05
plot(Tdatos.test14, nnet.predict14t, main="Neural network predictions vs FEV1", xlab="Transf_FEV1")
etiq = datos[,15]    # FVCP: FVC espirometría en % del teórico
etiq15 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain15 = etiq15[-ind]
ytest15  = etiq15[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train15 = scale(ytrain15) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test15 = scale(ytest15)   # Normalizar N(0,1) los datos de Test
```

```
summary(Tdatos.train15)
sd(Tdatos.train15)
summary(Tdatos.test15)
sd(Tdatos.test15)
show(Tdatos.train15)
nnet.fit15 <- nnet(Tdatos.train15/50 ~ ytrain15 , size=2, linout=T,trace=F)
nnet.fit15
#a 1-2-1 network with 7 weights
#inputs: ytrain15
#output(s): Tdatos.train15/50
#options were - linear output units
nnet.predict15 <- predict(nnet.fit15)*50
nnet.predict15
mean((nnet.predict15-Tdatos.train15)^2) #[1] 7.361987e-05
plot(Tdatos.train15, nnet.predict15, main="Neural network predictions vs FVC", xlab="Transf_FVC")
show(Tdatos.test15)
nnet.fit15t <- nnet(Tdatos.test15/50 ~ ytest15 , size=2, linout=T,trace=F)
nnet.fit15t
#a 1-2-1 network with 7 weights
#inputs: ytest15
#output(s): Tdatos.test15/50
#options were - linear output units
nnet.predict15t <- predict(nnet.fit15t)*50
nnet.predict15t
mean((nnet.predict15t-Tdatos.test15)^2) #[1] 9.018382e-05
plot(Tdatos.test15, nnet.predict15t, main="Neural network predictions vs FVC", xlab="Transf_FVC")
etiq = datos[,16]    # FEVCVF_: Espirometría previa o alta
etiq16 = as.numeric(etiq)
datos.train = datos2[-ind,]
datos.test  = datos2[ind,]
ytrain16 = etiq16[-ind]
ytest16  = etiq16[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Tdatos.train16 = scale(ytrain16) # Normalizar N(0,1) los datos de entrenamiento
Tdatos.test16 = scale(ytest16)   # Normalizar N(0,1) los datos de Test
summary(Tdatos.train16)
sd(Tdatos.train16)
summary(Tdatos.test16)
sd(Tdatos.test16)
show(Tdatos.train16)
nnet.fit16 <- nnet(Tdatos.train16/50 ~ ytrain16 , size=2, linout=T,trace=F)
nnet.fit16
#a 1-2-1 network with 7 weights
#inputs: ytrain16
#output(s): Tdatos.train16/50
#options were - linear output units
nnet.predict16 <- predict(nnet.fit16)*50
```

```
nnet.predict16
mean((nnet.predict16-Tdatos.train16)^2) #[1] 9.177834e-05
plot(Tdatos.train16, nnet.predict16, main="Neural network predictions vs FEVCVF", xlab="Transf_FEVCVF")
show(Tdatos.test16)
nnet.fit16t <- nnet(Tdatos.test16/50 ~ ytest16 , size=2, linout=T,trace=F)
nnet.fit16t
#a 1-2-1 network with 7 weights
#inputs: ytest16
#output(s): Tdatos.test16/50
#options were - linear output units
nnet.predict16t <- predict(nnet.fit16t)*50
nnet.predict16t
mean((nnet.predict16t-Tdatos.test16)^2) #[1] 9.743745e-05
plot(Tdatos.test16, nnet.predict16t, main="Neural network predictions vs FEVCVF", xlab="Transf_FEVCVF")
#Gráficos de la Figura 1 sobre la red imputados se insertan al texto.
# CONTINUAMOS SINTAXIS para el método NNET Y MICE mediante meth='sample' and meth='pmm'
#LISTA INICIAL DE PAQUETES que vamos a usar en los desarrollos
paquetes <- c("e1071", "fda", "splines", "Matrix", "MASS", “nnet”, “mice”, "lattice", "VIM", “cluster”, ”ade4”,
"graphics", "broom", "Rcpp", "robustbase", "sp", "curl" , "haven" , "colorspace", "grid", "data.table",
"lmtest", "zoo", “base” )
#Crea un vector lógico si están instalados o no y s.i hay al menos uno no instalado los instala
instalados <- paquetes %in% installed.packages()
if(sum(instalados == FALSE) > 0) {
  install.packages(paquetes[!instalados])
}
lapply(paquetes,require,character.only = TRUE)
#Descargado e instalado nueva versión R-3.5.1-win.exe superior a R-3.2.1 para gráficos nuevos
d<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_180617.txt",
header=T)
View (d)
names(d)
d2=d
View (d2)
names(d2)
View(d)
names(d)
#Eliminamos la variable Auditoria de la base llamada "d2"
d2 <- d2[ ,!colnames(d2)=="NUMERO_AUDITORIA"]
indx = sapply(d2,is.factor)
indx[indx==1]
d2[indx] = lapply(d2[indx], function(x) as.numeric(as.character(x)))
str(d2)
View(d2)
summary(d2)
#Eliminamos variables IMC y FEVCVF_ de base llamada "d2" para imputar y después calcular para SVM
d2 <- d2[ ,!colnames(d2)=="IMC" ]
d2 <- d2[ ,!colnames(d2)=="FEVCVF_" ]
```

```
View(d2)
names(d2)
#Permanecer variables sin missings de la base llamada "d2a" para pegar después a la imputada
d2a <- d2[,c("EDAD", "SEXO", "HT_", "DURING", "ESPIROMETRIA_PA_", "INGRESOS_", "SV_", "EXACER_90DIAS",
"Reing_EXAC", "MUERTOS_90DIAS", "EXITUS", "ICHARICC_", "CCVSDM_", "ICHAR_DM_", "EV_", "ICHARECV_",
"ICHAREVP_", "ICHARIM_", "ICHARNEF_", "ICHAR_TS_", "EP_", "SCORE_PAT_")]
View(d2a)
write.table(d2a, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110618.txt",col.names=TRUE)
datos_d2a<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110618.txt",header=T)
View(datos_d2a)
#Permanecer variables solo missings de base llamada "d3" para imputar y después calcular para SVM
d3 <- d2[,c("ALP","PPKG","TARS","TARD","TURA","FRE","FCA","FEV1P","FVCP")]
View(d3)
names(d3) #[1] "ALP"   "PPKG"  "TARS"  "TARD"  "TURA"  "FRE"   "FCA"   "FEV1P" "FVCP"
ind = sample(5178,2500)
datos = d3[,c(1:9)]
View(datos)
etiq5 = datos[,1]  #  ALP: Altura del paciente (metros)
str(etiq5)
View(etiq5)
#Para cada valor de ALP  "etiq5" la sentencia "is.na" devuelve T (true) si hay missing y F (false) si es dato.
ind.na = is.na(etiq5)
View(ind.na)
#Ver los datos sin valores faltantes en ALP,  variable Altura "etiq5" al eliminar "auditoria" y guardada "ind.na"
datos2 = datos[ind.na==F,]
dim(datos2) #[1] 1948   9
View(datos2)
#A la vista del visor se observa que hay otras variables relacionadas que tienen valores faltantes "NA"
#Ver el número de casos completos en toda la base de datos "datos2"
sum(complete.cases(datos2)) #[1] 991
#Visto resultado tenemos más casos 991 en toda la base "datos2" en vez 967 de antes pero es debido al quitar
IMC y FEVCVF_
datos3 = datos2[complete.cases(datos2)==T,]
dim(datos3) #[1] 991  9  #antes 991 31 variables
View(datos3)
# SINTAXIS para aplicar inicialmente imputación con NNET sin ver otros métodos, pero MICE parece ser la opción.
d.train = datos[-ind,]
d.test  = datos[ind,]
dim(d.train) #[1] 2678   9  #antes 2678 31 variables
dim(d.test)  #[1] 2500   9  #antes 2500 31 variables
Xtrain5 = etiq5[-ind]
Xtest5  = etiq5[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Td.train = scale(d.train) # Normalizar N(0,1) los datos de entrenamiento
Td.test = scale(d.test)   # Normalizar N(0,1) los datos de Test
Td.train5 = scale(Xtrain5) # Normalizar N(0,1) los datos de entrenamiento
Td.test5 = scale(Xtest5)   # Normalizar N(0,1) los datos de Test
```

```
summary(Td.train5)
sd(na.omit(Td.train5))
summary(Td.test5)
sd(na.omit(Td.test5))
etiq5c = datos3[,1]  #  ALP: Altura del paciente (metros) en “datos3” que es completa
View(etiq5c)
etiq5cc = scale(etiq5c)
View(etiq5cc)
summary(etiq5cc)
sd(etiq5cc) #[1] 1
#Eliminamos la variable ALP de la base llamada "datos3 antes de ejecutar la red"
datos4 <- datos3[ ,!colnames(datos3)=="ALP"]
View(datos4)
d.mat=as.matrix(datos4)
nnet.fit5 <- nnet((etiq5cc) ~ d.mat, subset= -ind , size=2,linout=T,trace=F)
nnet.fit5
#a 8-2-1 network with 21 weights
#inputs: d.matPPKG d.matTARS d.matTARD d.matTURA d.matFRE d.matFCA d.matFEV1P d.matFVCP
#output(s): (etiq5cc)
#options were - linear output units
nnet.predict5 <- predict(nnet.fit5,newdata =d.mat,subset = ind)
nnet.predict5
mean((nnet.predict5-(etiq5cc))^2) #[1] 0.9994475
plot(etiq5cc, nnet.predict5, main="Neural network predictions vs Altura")
#Grafico no conseguido tras realizar varias modificaciones, se intenta mejorar sin éxito y aplicamos técnica MICE
View(etiq5c)
View(datos4)
View(d.mat)
nnet.fit5 <- nnet((etiq5c) ~ d.mat, subset= -ind , size=2,linout=T,trace=F)
nnet.fit5
nnet.predict5 <- predict(nnet.fit5,newdata =d.mat,subset = ind)
nnet.predict5
mean((nnet.predict5-(etiq5c))^2) #[1] 0.005422943
plot(etiq5c, nnet.predict5, main="Neural network predictions vs Altura")# Grafico no adecuado – Errado hay que
mejorarlo
d.mat=as.matrix(datos4)
nnet.fit7 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.001)
nnet.fit7
#a 30-15-1 network with 481 weights
#inputs: d.matEDAD d.matSEXO d.matHT_ d.matDURING d.matPPKG d.matTARS d.matTARD d.matTURA d.matFRE
#d.matFCA d.matFEV1P d.matFVCP #d.matESPIROMETRIA_PA_ d.matINGRESOS_ d.matSV_ d.matEXACER_90DIAS
#d.matReing_EXAC d.matMUERTOS_90DIAS d.matEXITUS #d.matICHARICC_ d.matCCVSDM_ d.matICHAR_DM_
#d.matEV_ d.matICHARECV_ d.matICHAREVP_ d.matICHARIM_ d.matICHARNEF_ #d.matICHAR_TS_ d.matEP_
#d.matSCORE_PAT_
# output(s): (etiq5cc * 50)
#options were - linear output units  decay=0.001
nnet.predict7 <- predict(nnet.fit7,newdata =d.mat,subset = ind)
```

```
nnet.predict7
mean((nnet.predict7-(etiq5cc))^2) #[1] 441.4833
plot(etiq5cc, nnet.predict7, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict7/50, main="Neural network predictions vs Altura")
d.mat=as.matrix(datos4)
nnet.fit9 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=2,linout=T,trace=F)
nnet.fit9
nnet.predict9 <- predict(nnet.fit9,newdata =d.mat,subset = ind)
nnet.predict9
mean((nnet.predict9-(etiq5cc))^2)#[1] 6.316459
mean((nnet.predict9-(etiq5cc))^2) #[1] 1.175427
plot(etiq5cc, nnet.predict9, main="Neural network predictions vs Altura")
#plot(etiq5cc/50, nnet.predict9, main="Neural network predictions vs Altura")
#plot(etiq5cc*50, nnet.predict9, main="Neural network predictions vs Altura")
nnet.fit8 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=10,linout=T,trace=F)
nnet.fit8
nnet.predict9 <- predict(nnet.fit9,newdata =d.mat,subset = ind)
nnet.predict9
#mean((nnet.predict9-(etiq5cc))^2)#[1] 455.8166
#mean((nnet.predict9-(etiq5cc/50))^2)#[1] 469.313
mean((nnet.predict9-(etiq5cc))^2)[1] 1.175427
#plot(etiq5cc, nnet.predict9, main="Neural network predictions vs Altura")
#plot(etiq5cc, nnet.predict9/50, main="Neural network predictions vs Altura")
d.mat=as.matrix(datos4)
nnet.fit7 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.001)
nnet.fit7
nnet.predict7 <- predict(nnet.fit7,newdata =d.mat,subset = ind)
nnet.predict7
mean((nnet.predict7-(etiq5cc))^2) #[1] 441.4833
plot(etiq5cc, nnet.predict7, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict7/50, main="Neural network predictions vs Altura")
#Continuar probando el paramentro "decay" hasta ajustarlo, por ejemplo probar 5^-5 por atinar más antes de pasar a otro.
```

**# SINTAXIS para MODIFICAR - Pruebas método *"nnet y ajuste decay"***

```
Datos_d2a_pmm<-
read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_180918.txt",header=T)
etiq5 = datos[,1]  #  ALP: Altura del paciente (metros)
str(etiq5)
View(etiq5)
#para cada valor de ALP  "etiq5" la sentencia "is.na" devuelve T (true) si hay missing y F (false) si es dato.
ind.na = is.na(etiq5)
View(ind.na)
#Ver los datos sin valores faltantes en ALP,  variable Altura "etiq5" al eliminar "auditoria" y guardada "ind.na"
datos2 = datos[ind.na==F,]
dim(datos2)
View(datos2)
sum(complete.cases(datos2))
datos3 = datos2[complete.cases(datos2)==T,]
```

```
dim(datos3)
View(datos3)
d.train = datos[-ind,]
d.test  = datos[ind,]
dim(d.train)
dim(d.test)
Xtrain5 = etiq5[-ind]
Xtest5  = etiq5[ind]
set.seed(500) # Semilla para iniciar el generador aleatorio
Td.train = scale(d.train) # Normalizar N(0,1) los datos de entrenamiento
Td.test = scale(d.test)   # Normalizar N(0,1) los datos de Test
Td.train5 = scale(Xtrain5) # Normalizar N(0,1) los datos de entrenamiento
Td.test5 = scale(Xtest5)   # Normalizar N(0,1) los datos de Test
summary(Td.train5)
sd(na.omit(Td.train5))
summary(Td.test5)
sd(na.omit(Td.test5))
etiq5c = datos3[,1]  #  ALP: Altura del paciente (metros) en “datos3” que es completa
View(etiq5c)
etiq5cc = scale(etiq5c)
View(etiq5cc)
summary(etiq5cc)
sd(etiq5cc)
datos4 <- datos3[ ,!colnames(datos3)=="ALP"]
View(datos4)
d.mat=as.matrix(datos4)
nnet.fit5 <- nnet((etiq5cc) ~ d.mat, subset= -ind , size=2,linout=T,trace=F)
nnet.fit5
#a 8-2-1 network with 21 weights
#inputs: d.matPPKG d.matTARS d.matTARD d.matTURA d.matFRE d.matFCA d.matFEV1P d.matFVCP
#output(s): (etiq5cc)
#options were - linear output units
nnet.predict5 <- predict(nnet.fit5,newdata =d.mat,subset = ind)
nnet.predict5
mean((nnet.predict5-(etiq5cc))^2)
plot(etiq5cc, nnet.predict5, main="Neural network predictions vs Altura")
nnet.fit8 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00001)
nnet.fit8
nnet.predict8 <- predict(nnet.fit8,newdata =d.mat,subset = ind)
nnet.predict8
mean((nnet.predict8-(etiq5cc))^2)
plot(etiq5cc, nnet.predict8, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict8/50, main="Neural network predictions vs Altura")
nnet.fit9 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-5)
nnet.fit9
nnet.predict9 <- predict(nnet.fit9,newdata =d.mat,subset = ind)
nnet.predict9
```

```
mean((nnet.predict9-(etiq5cc))^2)
plot(etiq5cc, nnet.predict9, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict9/50, main="Neural network predictions vs Altura")
nnet.fit6 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-4)
nnet.fit6
nnet.predict6 <- predict(nnet.fit6,newdata =d.mat,subset = ind)
nnet.predict6
mean((nnet.predict6-(etiq5cc))^2)
plot(etiq5cc, nnet.predict6, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict6/50, main="Neural network predictions vs Altura")
nnet.fit4 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-3)
nnet.fit4
nnet.predict4 <- predict(nnet.fit4,newdata =d.mat,subset = ind)
nnet.predict4
mean((nnet.predict4-(etiq5cc))^2)
plot(etiq5cc, nnet.predict4, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict4/50, main="Neural network predictions vs Altura")
nnet.fit3 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-8)
nnet.fit3
nnet.predict3 <- predict(nnet.fit3,newdata =d.mat,subset = ind)
nnet.predict3
mean((nnet.predict3-(etiq5cc))^2)
plot(etiq5cc, nnet.predict3, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict3/50, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-12)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict2/50, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-15)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict2/50, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=8^-15)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict2/50, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=4^-15)
nnet.fit2
```

```
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
plot(etiq5cc, nnet.predict2/50, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=9^-18)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=9^-10)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=9^-7)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00019)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0004)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0006)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0007)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
```

```
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00065)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00061)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00063)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00065)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00069)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00067)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00066)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000667)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
```

```
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000066)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0000066)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000065)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000067)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0000666)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=-0.0000666)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=-0.00006)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00006)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
```

```
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000065)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.00065)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000661)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000663)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000665)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.000664)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=8^-3
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=8^-9
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
```

```
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=6^-9
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5^-9
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=3^-9
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5e-4
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5e-6
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=10,linout=T,trace=F, decay=5e-8
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=10,linout=T,trace=F, decay=5e-5
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=5,linout=T,trace=F, decay=5e-5
nnet.fit2
```

```
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5e-7
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5e7)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=5e6)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=2e6)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=e6)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=15,linout=T,trace=F, decay=0.0006)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=19,rang=0.1, decay=5e-4, maxit=2000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=19,rang=0.1, decay=5e-5, maxit=2000, linout=T,trace=F)
```

```
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=20,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=25,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=28,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=30,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=30,rang=0.1, decay=5e-4, maxit=4000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=30,rang=0.1, decay=5e-4, maxit=2500, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=40,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
```

```
nnet.fit2 <- nnet((etiq5cc*50) ~ d.mat, subset= -ind , size=35,rang=0.1, decay=5e-4, maxit=3000, linout=T,trace=F)
nnet.fit2
nnet.predict2 <- predict(nnet.fit2,newdata =d.mat,subset = ind)
nnet.predict2
mean((nnet.predict2-(etiq5cc))^2)
plot(etiq5cc, nnet.predict2, main="Neural network predictions vs Altura")
#NOTA. Las pruebas no alcanzan lo perseguido en resultados antes y después a la imputación. Descartar NNET y aplicar MICE.
#Gráficos de la Figura 1 sobre la red imputados se insertan al texto.
#Apuntar que el error sale bastante alto y los graficos visualmente malos ni ajustando los pesos. Ver MICE
```

**# SINTAXIS para el método MICE mediante meth='sample' and meth='pmm'**

```
d<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_180617.txt",
header=T)
View (d)
d2=d
View (d2)
View(d)
names(d)
d2 <- d2[ ,!colnames(d2)=="NUMERO_AUDITORIA"]
indx = sapply(d2,is.factor)
indx[indx==1]
d2[indx] = lapply(d2[indx], function(x) as.numeric(as.character(x)))
str(d2)
View(d2)
summary(d2)
#Eliminamos variables IMC y FEVCVF_ de la base llamada "d2" para imputar y después se calculan para SVM
d2 <- d2[ ,!colnames(d2)=="IMC" ]
d2 <- d2[ ,!colnames(d2)=="FEVCVF_" ]
View(d2)
names(d2)
#Permanecer variables sin missings de la base llamada "d2a" para pegar después a la imputada
d2a <- d2[,c("EDAD", "SEXO", "HT_", "DURING", "ESPIROMETRIA_PA_", "INGRESOS_", "SV_", "EXACER_90DIAS",
"Reing_EXAC", "MUERTOS_90DIAS", "EXITUS", "ICHARICC_", "CCVSDM_", "ICHAR_DM_", "EV_", "ICHARECV_",
"ICHAREVP_", "ICHARIM_", "ICHARNEF_", "ICHAR_TS_", "EP_", "SCORE_PAT_")]
View(d2a)
write.table(d2a, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110618.txt",col.names=TRUE)
datos_d2a<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110618.txt",header=T)
View(datos_d2a)
#Permanecer variables solo missings de la base llamada "d3" para imputar y después se calculan para SVM
d3 <- d2[,c("ALP","PPKG","TARS","TARD","TURA","FRE","FCA","FEV1P","FVCP")]
View(d3)
names(d3) #[1] "ALP"   "PPKG"  "TARS"  "TARD"  "TURA"  "FRE"   "FCA"   "FEV1P" "FVCP"
ind = sample(5178,2500)
datos = d3[,c(1:9)]
View(datos)
str(datos)
md.pattern(datos) # vemos los datos faltantes y el total completo en el dataset
#library(VIM)
```

```
plot_aggr <- aggr(datos, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(datos), cex.axis=.8,
gap=2, ylab=c("Histogram of missing data","Pattern"))
imp_mice <- mice(datos, m=9, maxit=50, meth='sample', seed=500)
imp_mice
complete_datos <- complete(imp_mice)
complete_datos
summary(complete_datos) # Vemos los datos completados tras la imputación
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 93.00
# Mean   :1.641   Mean   : 74.85   Mean   :136.4   Mean   : 75.04   Mean   :36.78   Mean   :24.26   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#     FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 43.00   Median : 63.00
# Mean   : 45.02   Mean   : 64.88
# 3rd Qu.: 55.00   3rd Qu.: 77.00
# Max.   :150.00   Max.   :150.00
summary(datos) # Vemos los datos originales de la base para comparar con la salida
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.0   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.0   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.0   Median :135.0   Median : 75.00   Median :36.60   Median :24.00   Median : 94.00
# Mean   :1.642   Mean   : 74.8   Mean   :136.3   Mean   : 75.06   Mean   :36.78   Mean   :24.27   Mean   : 94.58
# 3rd Qu.:1.690   3rd Qu.: 84.0   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.0   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
# NA's   :3230    NA's   :3216    NA's   :593     NA's   :593      NA's   :759     NA's   :2726    NA's   :812
#     FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 42.00   Median : 64.00
# Mean   : 44.97   Mean   : 65.07
# 3rd Qu.: 55.00   3rd Qu.: 77.00
# Max.   :150.00   Max.   :150.00
# NA's   :1973     NA's   :2125
#A la vista de las salidas. Si comparamos los originales y los imputados no hay mucha
#diferencia excepto un punto -/+ en la mediana en las variables imputadas: TARD, FCA y FVCP y
#en FEV1P (+), un punto más. Podemos decir que la imputación MICE se puede dar por correcta.
#Vemos gráfico missing para verificar que están todas las variables a cero y completas.
plot_aggr_imp <- aggr(complete_datos, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(complete_datos), cex.axis=.8, gap=2, ylab=c("Histogram of missing data","Pattern"))
#Vemos todos los gráficos imputados y observados para verificarlos si la linea es ascendente.
plot(datos$ALP, complete_datos$ALP, main="MICE vs Altura")
plot(datos$PPKG, complete_datos$PPKG, main="MICE vs Peso")
plot(datos$TARS, complete_datos$TARS, main="MICE vs Tensión Sistolica")
plot(datos$TARD, complete_datos$TARD, main="MICE vs Tensión Diastolica")
plot(datos$TURA, complete_datos$TURA, main="MICE vs Temperatura")
```

```
plot(datos$FRE, complete_datos$FRE, main="MICE vs Frecuencia")
plot(datos$FCA, complete_datos$FCA, main="MICE vs Frecuencia Cardiaca")
plot(datos$FEV1P, complete_datos$FEV1P, main="MICE vs FEV1P")
plot(datos$FVCP, complete_datos$FVCP, main="MICE vs FVCP")
#Guardamos la base imputada para pegarla a la general y sacar los perfiles y sensibilidad
complete_datos
View(complete_datos)
write.table(complete_datos, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",col.names=TRUE)
datos_imp<-
read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",header=T)
View(datos_imp)
```

**# SINTAXIS con el otro método predictive mean matching "pmm"**

```
#Anotar que para variables categoricas se usa el metodo logist regression "logreg", aquí todas son numericas
imppmm_mice <- mice(datos, m=9, maxit=50, meth='pmm', seed=500)
imppmm_mice
complete_pmm_datos <- complete(imppmm_mice)
complete_pmm_datos
write.table(complete_pmm_datos, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/pmm_mice_110618.txt",col.names=TRUE)
summary(complete_pmm_datos) # Ver datos completos tras imputación "pmm"
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.650   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 94.00
# Mean   :1.643   Mean   : 75.02   Mean   :136.3   Mean   : 75.01   Mean   :36.78   Mean   :24.28   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#    FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.0
# 1st Qu.: 32.00   1st Qu.: 51.0
# Median : 42.00   Median : 63.0
# Mean   : 45.08   Mean   : 64.8
# 3rd Qu.: 55.00   3rd Qu.: 77.0
# Max.   :150.00   Max.   :150.0
# Vemos los datos completados tras la imputación "sample" y revisamos "pmm"
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 93.00
# Mean   :1.641   Mean   : 74.85   Mean   :136.4   Mean   : 75.04   Mean   :36.78   Mean   :24.26   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#    FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 43.00   Median : 63.00
# Mean   : 45.02   Mean   : 64.88
# 3rd Qu.: 55.00   3rd Qu.: 77.00
# Max.   :150.00   Max.   :150.00
#Vemos todos los gráficos imputados y observados para verificarlos si la linea es ascendente.
plot(datos$ALP, complete_pmm_datos$ALP, main="MICE vs Altura")
```

```
plot(datos$PPKG, complete_pmm_datos$PPKG, main="MICE vs Peso")
plot(datos$TARS, complete_pmm_datos$TARS, main="MICE vs Tensión Sistolica")
plot(datos$TARD, complete_pmm_datos$TARD, main="MICE vs Tensión Diastolica")
plot(datos$TURA, complete_pmm_datos$TURA, main="MICE vs Temperatura")
plot(datos$FRE, complete_pmm_datos$FRE, main="MICE vs Frecuencia")
plot(datos$FCA, complete_pmm_datos$FCA, main="MICE vs Frecuencia Cardiaca")
plot(datos$FEV1P, complete_pmm_datos$FEV1P, main="MICE vs FEV1P")
plot(datos$FVCP, complete_pmm_datos$FVCP, main="MICE vs FVCP")
#Perfecto. Los gráficos del “pmm” salen igual que el método “sample” (línea ascendente)
#Pegar las bases imputadas por cada método a la base “d2a” original y calcular “IMC” y “FEVCVF_”
datos_d2a<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110618.txt",header=T)
View(datos_d2a)
datos_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",header=T)
View(datos_imp)
datos_pmm<-
read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/pmm_mice_110618.txt",header=T)
View(datos_pmm)
#Pegamos las bases “d2a” original y base imputada “simple” y calcular “IMC” y “FEVCVF_”
Datos_d2a_imp = cbind(datos_d2a,datos_imp)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
write.table(Datos_d2a_imp,file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_110618.txt",col.names=TR
UE)
Datos_d2a_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_110618.txt",header=T)
View(Datos_d2a_imp)
#Pasar a numérico las variables para poder calcular “IMC” y “FEVCVF_”
indx = sapply(Datos_d2a_imp,is.numeric)
indx[indx==1]
Datos_d2a_imp[indx] = lapply(Datos_d2a_imp[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_imp)
#IMC es peso, expresado en kilos, entre la estatura, en metros, elevada al cuadrado (kg/m2)
View(Datos_d2a_imp)
View(Datos_d2a_imp$PPKG)
mean(Datos_d2a_imp$PPKG)
View(Datos_d2a_imp$ALP)
mean(Datos_d2a_imp$ALP)
IMC=(Datos_d2a_imp$PPKG)/((Datos_d2a_imp$ALP)^2)
View(IMC)
IMC=round(IMC,2)
View(IMC)
Datos_d2a_imp = cbind(Datos_d2a_imp,IMC)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
#FEVCVF_ es el cociente entre la FEV1P y la FVCP multiplicado por 100
View(Datos_d2a_imp)
View(Datos_d2a_imp$FEV1P)
mean(Datos_d2a_imp$FEV1P)
```

```
View(Datos_d2a_imp$FVCP)
mean(Datos_d2a_imp$FVCP)
FEVCVF_=((Datos_d2a_imp$FEV1P)/(Datos_d2a_imp$FVCP))*100
View(FEVCVF_)
FEVCVF_=round(FEVCVF_,2)
View(FEVCVF_)
Datos_d2a_imp = cbind(Datos_d2a_imp,FEVCVF_)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
#Guardamos la base final con las variables calculadas
write.table(Datos_d2a_imp,
file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_180618.txt",col.names=TRUE)
Datos_d2a_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_180618.txt",header=T)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
#Eliminar "SCORE_PAT_" calculada porque no procede, genera ruido en la salida del analisis de perfiles.
Datos_d2a_imp <- Datos_d2a_imp[ ,!colnames(Datos_d2a_imp)=="SCORE_PAT_" ]
View(Datos_d2a_imp)
#Pegamos las bases “d2a” original y base imputada “pmm” y calcular “IMC” y “FEVCVF_”
Datos_d2a_pmm = cbind(datos_d2a,datos_pmm)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
write.table(Datos_d2a_pmm,
file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_110618.txt",col.names=TRUE)
Datos_d2a_pmm<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_110618.txt",header=T)
View(Datos_d2a_pmm)
#Pasar a numérico las variables para poder calcular “IMC” y “FEVCVF_”
indx = sapply(Datos_d2a_pmm,is.numeric)
indx[indx==1]
Datos_d2a_pmm[indx] = lapply(Datos_d2a_pmm[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_pmm)
#IMC es peso, expresado en kilos, entre la estatura, en metros, elevada al cuadrado (kg/m2)
View(Datos_d2a_pmm)
View(Datos_d2a_pmm$PPKG)
mean(Datos_d2a_pmm$PPKG)
View(Datos_d2a_pmm$ALP)
mean(Datos_d2a_pmm$ALP)
IMC=(Datos_d2a_pmm$PPKG)/((Datos_d2a_pmm$ALP)^2)
View(IMC)
IMC=round(IMC,2)
View(IMC)
Datos_d2a_pmm = cbind(Datos_d2a_pmm,IMC)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
#FEVCVF_ es el cociente entre la FEV1P y la FVCP multiplicado por 100
View(Datos_d2a_pmm)
View(Datos_d2a_pmm$FEV1P)
mean(Datos_d2a_pmm$FEV1P)
```

```
View(Datos_d2a_pmm$FVCP)
mean(Datos_d2a_pmm$FVCP)
FEVCVF_=((Datos_d2a_pmm$FEV1P)/(Datos_d2a_pmm$FVCP))*100
View(FEVCVF_)
FEVCVF_=round(FEVCVF_,2)
View(FEVCVF_)
Datos_d2a_pmm = cbind(Datos_d2a_pmm,FEVCVF_)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
#Guardamos la base final con las variables calculadas
write.table(Datos_d2a_pmm,
file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_180618.txt",col.names=TRUE)
Datos_d2a_pmm<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_180618.txt",header=T)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
#Eliminar "SCORE_PAT_" calculada porque no procede,genera ruido en la salida del analisis de perfiles.
Datos_d2a_pmm <- Datos_d2a_pmm [ ,!colnames(Datos_d2a_pmm)=="SCORE_PAT_" ]
View(Datos_d2a_pmm)
#Descargado e instalado nueva versión R-3.5.1-win.exe superior a R-3.2.1 para gráficos nuevos
d<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_180617.txt", header=T)
View (d)
names(d)
d2=d
View (d2)
names(d2)
d2 <- d2[ ,!colnames(d2)=="NUMERO_AUDITORIA"]
indx = sapply(d2,is.factor)
indx[indx==1]
d2[indx] = lapply(d2[indx], function(x) as.numeric(as.character(x)))
str(d2)
View(d2)
summary(d2)
#Eliminamos variables IMC y FEVCVF_ de la base llamada "d2" para imputar y después se calculan para SVM
d2 <- d2[ ,!colnames(d2)=="IMC" ]
d2 <- d2[ ,!colnames(d2)=="FEVCVF_" ]
View(d2)
names(d2)
#Eliminamos variable SCORE_PAT_ de la base llamada "d2" por generar ruido en la salida del análisis
de perfiles y ya no es necesaria
d2 <- d2[ ,!colnames(d2)=="SCORE_PAT_" ]
View(d2)
names(d2)
#Permanecer variables sin missings de la base llamada "d2a" para pegar después a la imputada
d2a <- d2[,c("EDAD", "SEXO", "HT_", "DURING", "ESPIROMETRIA_PA_", "INGRESOS_", "SV_", "EXACER_90DIAS",
"Reing_EXAC", "MUERTOS_90DIAS", "EXITUS", "ICHARICC_", "CCVSDM_", "ICHAR_DM_", "EV_", "ICHARECV_",
"ICHAREVP_", "ICHARIM_", "ICHARNEF_", "ICHAR_TS_", "EP_")]
View(d2a)
```

```
write.table(d2a, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110918.txt",col.names=TRUE)
datos_d2a<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110918.txt",header=T)
View(datos_d2a)
#Permanecer variables solo missings de la base llamada "d3" para imputar y después se calculan para SVM
d3 <- d2[,c("ALP","PPKG","TARS","TARD","TURA","FRE","FCA","FEV1P","FVCP")]
View(d3)
names(d3) #[1] "ALP"   "PPKG"  "TARS"  "TARD"  "TURA"  "FRE"   "FCA"   "FEV1P" "FVCP"
ind = sample(5178,2500)
datos = d3[,c(1:9)]
# SINTAXIS para el método MICE y se ejecuta mediante meth='sample' and meth='pmm'
View(datos)
str(datos)
md.pattern(datos) # vemos los datos faltantes y el total completo en el dataset
#library(VIM)
plot_aggr <- aggr(datos, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(datos), cex.axis=.8,
gap=2, ylab=c("Histogram of missing data","Pattern"))
imp_mice <- mice(datos, m=9, maxit=50, meth='sample', seed=500)
imp_mice
complete_datos <- complete(imp_mice)
complete_datos
summary(complete_datos) # Vemos los datos completados tras la imputación
#      ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 93.00
# Mean   :1.641   Mean   : 74.85   Mean   :136.4   Mean   : 75.04   Mean   :36.78   Mean   :24.26   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#     FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 43.00   Median : 63.00
# Mean   : 45.02   Mean   : 64.88
# 3rd Qu.: 55.00   3rd Qu.: 77.00
# Max.   :150.00   Max.   :150.00
summary(datos) # Vemos los datos originales de la base para comparar con la salida
#      ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.0   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.0   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.0   Median :135.0   Median : 75.00   Median :36.60   Median :24.00   Median : 94.00
# Mean   :1.642   Mean   : 74.8   Mean   :136.3   Mean   : 75.06   Mean   :36.78   Mean   :24.27   Mean   : 94.58
# 3rd Qu.:1.690   3rd Qu.: 84.0   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.0   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
# NA's   :3230    NA's   :3216    NA's   :593     NA's   :593      NA's   :759     NA's   :2726    NA's   :812
#     FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 42.00   Median : 64.00
# Mean   : 44.97   Mean   : 65.07
# 3rd Qu.: 55.00   3rd Qu.: 77.00
```

```
# Max.   :150.00   Max.   :150.00
# NA's   :1973     NA's   :2125
#A la vista de las salidas. Si comparamos los originales y los imputados no hay mucha
#diferencia excepto un punto -/+ en la mediana en las variables imputadas: TARD, FCA y FVCP y
#en FEV1P (+), un punto más. Podemos decir que la imputación MICE se puede dar por correcta.
#Vemos gráfico missing para verificar que están todas las variables a cero y completas.
plot_aggr_imp <- aggr(complete_datos, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
labels=names(complete_datos), cex.axis=.8, gap=2, ylab=c("Histogram of missing data","Pattern"))
#Vemos todos los gráficos imputados y observados para verificarlos si la linea es ascendente.
plot(datos$ALP, complete_datos$ALP, main="MICE vs Altura")
plot(datos$PPKG, complete_datos$PPKG, main="MICE vs Peso")
plot(datos$TARS, complete_datos$TARS, main="MICE vs Tensión Sistolica")
plot(datos$TARD, complete_datos$TARD, main="MICE vs Tensión Diastolica")
plot(datos$TURA, complete_datos$TURA, main="MICE vs Temperatura")
plot(datos$FRE, complete_datos$FRE, main="MICE vs Frecuencia")
plot(datos$FCA, complete_datos$FCA, main="MICE vs Frecuencia Cardiaca")
plot(datos$FEV1P, complete_datos$FEV1P, main="MICE vs FEV1P")
plot(datos$FVCP, complete_datos$FVCP, main="MICE vs FVCP")
#Guardamos la base imputada para pegarla a la general y sacar los perfiles y sensibilidad
complete_datos
View(complete_datos)
write.table(complete_datos, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",col.names=TRUE)
datos_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",header=T)
View(datos_imp)
# SINTAXIS con el otro método predictive mean matching “pmm”
#Anotar que para variables categoricas se usa el metodo logist regression “logreg”, aquí todas son numericas
imppmm_mice <- mice(datos, m=9, maxit=50, meth='pmm', seed=500)
imppmm_mice
complete_pmm_datos <- complete(imppmm_mice)
complete_pmm_datos
write.table(complete_pmm_datos, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/pmm_mice_110618.txt",col.names=TRUE)
summary(complete_pmm_datos) # Ver datos completos tras imputación “pmm”
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.650   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 94.00
# Mean   :1.643   Mean   : 75.02   Mean   :136.3   Mean   : 75.01   Mean   :36.78   Mean   :24.28   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#    FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.0
# 1st Qu.: 32.00   1st Qu.: 51.0
# Median : 42.00   Median : 63.0
# Mean   : 45.08   Mean   : 64.8
# 3rd Qu.: 55.00   3rd Qu.: 77.0
# Max.   :150.00   Max.   :150.0
# Vemos los datos completados tras la imputación “sample” y revisamos “pmm”
#     ALP             PPKG            TARS            TARD            TURA            FRE             FCA
# Min.   :1.400   Min.   : 30.00   Min.   : 68.0   Min.   : 33.00   Min.   :34.00   Min.   :10.00   Min.   : 19.00
```

```
# 1st Qu.:1.600   1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 65.00   1st Qu.:36.00   1st Qu.:20.00   1st Qu.: 81.00
# Median :1.640   Median : 74.00   Median :135.0   Median : 74.00   Median :36.60   Median :24.00   Median : 93.00
# Mean   :1.641   Mean   : 74.85   Mean   :136.4   Mean   : 75.04   Mean   :36.78   Mean   :24.26   Mean   : 94.44
# 3rd Qu.:1.690   3rd Qu.: 84.00   3rd Qu.:150.0   3rd Qu.: 83.00   3rd Qu.:37.20   3rd Qu.:28.00   3rd Qu.:105.00
# Max.   :1.980   Max.   :150.00   Max.   :245.0   Max.   :170.00   Max.   :40.00   Max.   :58.00   Max.   :180.00
#     FEV1P           FVCP
# Min.   : 20.00   Min.   : 23.00
# 1st Qu.: 32.00   1st Qu.: 51.00
# Median : 43.00   Median : 63.00
# Mean   : 45.02   Mean   : 64.88
# 3rd Qu.: 55.00   3rd Qu.: 77.00
# Max.   :150.00   Max.   :150.00
#Vemos todos los gráficos imputados y observados para verificarlos si la linea es ascendente.
plot(datos$ALP, complete_pmm_datos$ALP, main="MICE vs Altura")
plot(datos$PPKG, complete_pmm_datos$PPKG, main="MICE vs Peso")
plot(datos$TARS, complete_pmm_datos$TARS, main="MICE vs Tensión Sistolica")
plot(datos$TARD, complete_pmm_datos$TARD, main="MICE vs Tensión Diastolica")
plot(datos$TURA, complete_pmm_datos$TURA, main="MICE vs Temperatura")
plot(datos$FRE, complete_pmm_datos$FRE, main="MICE vs Frecuencia")
plot(datos$FCA, complete_pmm_datos$FCA, main="MICE vs Frecuencia Cardiaca")
plot(datos$FEV1P, complete_pmm_datos$FEV1P, main="MICE vs FEV1P")
plot(datos$FVCP, complete_pmm_datos$FVCP, main="MICE vs FVCP")
#Perfecto. Los gráficos del “pmm” salen igual que el método “sample” (línea ascendente).
#Pegar las bases imputadas por cada método a la base “d2a” original y calcular “IMC” y “FEVCVF_”
datos_d2a<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/d2a_110918.txt",header=T)
View(datos_d2a)
datos_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/imp_mice_110618.txt",header=T)
View(datos_imp)
datos_pmm<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/pmm_mice_110618.txt",header=T)
View(datos_pmm)
#Pegamos las bases “d2a” original y base imputada “simple” y calcular “IMC” y “FEVCVF_”
Datos_d2a_imp = cbind(datos_d2a,datos_imp)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
write.table(Datos_d2a_imp, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_110918.txt",col.names=TRUE)
Datos_d2a_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_110918.txt",header=T)
View(Datos_d2a_imp)
#Pasar a numérico las variables para poder calcular “IMC” y “FEVCVF_”
indx = sapply(Datos_d2a_imp,is.numeric)
indx[indx==1]
Datos_d2a_imp[indx] = lapply(Datos_d2a_imp[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_imp)
#IMC es peso, expresado en kilos, entre la estatura, en metros, elevada al cuadrado (kg/m2)
View(Datos_d2a_imp)
View(Datos_d2a_imp$PPKG)
mean(Datos_d2a_imp$PPKG)
View(Datos_d2a_imp$ALP)
mean(Datos_d2a_imp$ALP)
IMC=(Datos_d2a_imp$PPKG)/((Datos_d2a_imp$ALP)^2)
```

```
View(IMC)
IMC=round(IMC,2)
View(IMC)
Datos_d2a_imp = cbind(Datos_d2a_imp,IMC)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
#FEVCVF_ es el cociente entre la FEV1P y la FVCP multiplicado por 100
View(Datos_d2a_imp)
View(Datos_d2a_imp$FEV1P)
mean(Datos_d2a_imp$FEV1P)
View(Datos_d2a_imp$FVCP)
mean(Datos_d2a_imp$FVCP)
FEVCVF_=((Datos_d2a_imp$FEV1P)/(Datos_d2a_imp$FVCP))*100
View(FEVCVF_)
FEVCVF_=round(FEVCVF_,2)
View(FEVCVF_)
Datos_d2a_imp = cbind(Datos_d2a_imp,FEVCVF_)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
#Guardamos la base final con las variables calculadas
write.table(Datos_d2a_imp, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_180918.txt",col.names=TRUE)
Datos_d2a_imp<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_180918.txt",header=T)
View(Datos_d2a_imp)
str(Datos_d2a_imp)
indx = sapply(Datos_d2a_imp,is.numeric)
indx[indx==1]
Datos_d2a_imp[indx] = lapply(Datos_d2a_imp[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_imp)
#Pegamos las bases “d2a” original y base imputada “pmm” y calcular “IMC” y “FEVCVF_”
View(datos_pmm)
Datos_d2a_pmm = cbind(datos_d2a,datos_pmm)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
write.table(Datos_d2a_pmm, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_110918.txt",col.names=TRUE)
Datos_d2a_pmm<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_110918.txt",header=T)
View(Datos_d2a_pmm)
#Pasar a numérico las variables para poder calcular “IMC” y “FEVCVF_”
indx = sapply(Datos_d2a_pmm,is.numeric)
indx[indx==1]
Datos_d2a_pmm[indx] = lapply(Datos_d2a_pmm[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_pmm)
# IMC es peso, expresado en kilos, entre la estatura, en metros, elevada al cuadrado (kg/m2)
View(Datos_d2a_pmm)
View(Datos_d2a_pmm$PPKG)
mean(Datos_d2a_pmm$PPKG)
View(Datos_d2a_pmm$ALP)
mean(Datos_d2a_pmm$ALP)
IMC=(Datos_d2a_pmm$PPKG)/((Datos_d2a_pmm$ALP)^2)
```

```
View(IMC)
IMC=round(IMC,2)
View(IMC)
Datos_d2a_pmm = cbind(Datos_d2a_pmm,IMC)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
#FEVCVF_ es el cociente entre la FEV1P y la FVCP multiplicado por 100
View(Datos_d2a_pmm)
View(Datos_d2a_pmm$FEV1P)
mean(Datos_d2a_pmm$FEV1P)
View(Datos_d2a_pmm$FVCP)
mean(Datos_d2a_pmm$FVCP)
FEVCVF_=((Datos_d2a_pmm$FEV1P)/(Datos_d2a_pmm$FVCP))*100
View(FEVCVF_)
FEVCVF_=round(FEVCVF_,2)
View(FEVCVF_)
Datos_d2a_pmm = cbind(Datos_d2a_pmm,FEVCVF_)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
#Guardamos la base final con las variables calculadas
write.table(Datos_d2a_pmm, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_180918.txt",col.names=TRUE)
Datos_d2a_pmm<-read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_pmm_180918.txt",header=T)
View(Datos_d2a_pmm)
str(Datos_d2a_pmm)
indx = sapply(Datos_d2a_pmm,is.numeric)
indx[indx==1]
Datos_d2a_pmm[indx] = lapply(Datos_d2a_pmm[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_pmm)
View(Datos_d2a_pmm)
View(Datos_d2a_imp)
```

*#BASE FINAL IMPUTADA CON MICE para ANÁLISIS COMPROBADA (Datos_d2a_imp_180918)*

**# SINTAXIS SEGUNDA – ANALIZAR BASE DE DATOS TRAS LA IMPUTACIÓN REALIZADA**

**# RESUMEN DE RESULTADOS EPIDEMIOLÓGICOS-CLÍNICOS (TABLA 1)**

```
d<-read.table(file="C:/Users/NisaB/Desktop/NBTS/Datos_d2a_imp_180918.txt", header=T)
View(d)
summary(d)
# $EDAD          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   31.00   68.00   75.00   73.39   80.00   99.00
# $DURING        Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  1.000   6.000   8.000   9.955  12.000 130.000
# $ALP           Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   1.400   1.600   1.640   1.641   1.690   1.980
# $PPKG          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   30.00   65.00   74.00   74.85   84.00  150.00
# $TARS          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    68.0   120.0   135.0   136.4   150.0   245.0
# $TARD          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   33.00   65.00   74.00   75.04   83.00  170.00
# $TURA          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  34.00   36.00   36.60   36.78   37.20   40.00
# $FRE           Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   10.00   20.00   24.00   24.26   28.00   58.00
# $FCA           Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   19.00   81.00   93.00   94.44  105.00  180.00
# $FEV1P         Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   20.00   32.00   43.00   45.02   55.00  150.00
# $FVCP          Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   23.00   51.00   63.00   64.88   77.00  150.00
# $IMC           Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  11.24   23.87   27.39   27.88   31.25   57.16
# $FEVCVF_       Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   16.55   53.70   68.66   72.67   85.87  300.00
```

```
table(d$SEXO)                    #        0   1    4526  652
table(d$HT_)                     #        0   1     906 4272
table(d$ESPIROMETRIA_PA_)        #        0   1    1644 3534
table(d$INGRESOS_)               #        0   1    3343 1835
table(d$SV_)                     #        0   1    4596  582
table(d$EXACER_90DIAS)           #        0   1    3793 1385
table(d$Reing_EXAC)              #        0   1    3751 1427
table(d$MUERTOS_90DIAS)          #        0   1    4873  305
table(d$EXITUS) #muerto;vivo     #        0   1     259 4919
table(d$ICHARICC_)               #        0   1    4058 1120
table(d$CCVSDM_)                 #        0   1    2961 2217
table(d$ICHAR_DM_)               #        0   1    3844 1334
table(d$EV_)                     #        0   1    3588 1590
table(d$ICHARECV_)               #        0   1    4620  558
table(d$ICHAREVP_)               #        0   1    4422  756
table(d$ICHARIM_)                #        0   1    4505  673
table(d$ICHARNEF_)               #        0   1    4691  487
table(d$ICHAR_TS_)               #        0   1    4506  672
table(d$EP_)                     #        0   1    3828 1350
sd(d$EDAD)                       #       [1] 10.08
sd(d$DURING)                     #       [1] 7.82
sd(d$ALP)                        #       [1] 0.08
sd(d$PPKG)                       #       [1] 15.55
sd(d$IMC)                        #       [1] 6.03
sd(d$TARS)                       #       [1] 23.85
sd(d$TARD)                       #       [1] 13.84
sd(d$TURA)                       #       [1] 0.82
sd(d$FRE)                        #       [1] 6.62
sd(d$FCA)                        #       [1] 18.50
sd(d$FEV1P)                      #       [1] 16.82
sd(d$FVCP)                       #       [1] 19.21
sd(d$FEVCVF_)                    #       [1] 28.48
```

########################################## **CAPITULO II** ##########################################

**# SINTAXIS PARA LA REDUCCIÓN DIMENSIONAL – DISITNTOS MÉTODOS MEDIANTE ACP, APS-REM y RF&IV**
**# SINTAXIS ANALISIS DE COMPONENTES PRINCIPALES**

```
# ACPrincipales en base imputada final "Datos_d2a_imp_180918" con método "simple"
library(ade4)
summary(Datos_d2a_imp) # datos crudos sin estandarizar
summary(Datos2_d2a_imp) # estandarizados, escalas distintas en datos crudos.
# Vemos matriz de correlaciones para ver si estas son altas, ya que esta es una de las hipótesis para el análisis ACP.
m.cor<-cor(Datos2_d2a_imp)
View(m.cor)
#coorplot indica color correlación que existe, azul=correlacion positiva y rojo=correl.negativa
library(corrplot) # corrplot 0.84 loaded
corrplot (m.cor)
```

```
#library(PerformanceAnalytics)  # paquete no posible hay que descargar versión R4.1.0
#Este comando "chart.Correlation" da gráficas de dispersión y calcula correlaciones entre variables añadiéndolas a
#estos gráficos independientes entre cada variable
library(PerformanceAnalytics)
library(xts)
library(e1071)
library(graphics)
library(PerformanceAnalytics)
chart.Correlation(m.cor,histogram = F ,pch = 19)
#Gráfico visual de la matrix de correlaciones completa.
#Utilizaremos el gráfico de Sedimentación que muestra la cantidad optimas de componentes a tomar en la data,
#siendo los valores por encima de la linea de 1.0 los más aceptables.
# SINTAXIS ANALISIS PARALELO (APS-REM) Y ACP
library(psych)
scree(m.cor,main ="Gráfico de Sedimentación")
scree(Datos2_d2a_imp,main ="Gráfico de Sedimentación")
#También podemos hacer un Análisis Paralelo con la funcion "fa.parallel" y generar un gráfico de apoyo
#con el cual verificamos los resultados dados por el gráfico de sedimentación
fa.parallel(m.cor,fa="pc") # datos escalados, advierte usar datos crudos con el análisis paralelo
fa.parallel(Datos2_d2a_imp,fa="pc") # Análisis Paralelo verificando que son 12 comp.
# Parallel analysis suggests that the number of factors =  NA  and the number of components =  12
#Utilizamos la función dudi.pca y así se generará el análisis y la gráfica de los autovalores (scannf=T)
acp<-dudi.pca(df=Datos2_d2a_imp, scannf=T, nf=2)
# Select the number of axes: 2
#help(dudi.pca)
#acp1<-dudi.pca(df=Datos2_d2a_imp, scannf=F, nf=2) #scannf=F no muestra el gráfico autovalores
#acp1
acp
acp$eig # mostramos los 32 autovalores observando que mayores 1 tenemos las 12 componentes
# [1] 3.245778693 2.911244128 1.981098258 1.744384127 1.601285304 1.472005418 1.387449620
# [8] 1.280648739 1.235331979 1.143754033 1.096753063 1.041321596 0.982012431 0.963989042
# [15] 0.950422871 0.920224866 0.909853707 0.873784865 0.846464235 0.829847464 0.806125025
# [22] 0.792853983 0.733963670 0.720077786 0.596593217 0.407131594 0.218106563 0.182586128
# [29] 0.058477511 0.042573076 0.019593800 0.004263208
summary(acp)
acp12<-dudi.pca(df=Datos2_d2a_imp, scannf=F, nf=12) # modificamos (nf=12) para ver las 12 componentes.
acp12
summary(acp12)
acp12$eig
View(acp12$c1)
acp12$c1
#Actualizamos sintaxis más adelante para mostrar presentaciones más actuales
#Reinicio para ver todas, ya que con dudi.pca muestra solo las 5 primeras de las 32 variables.
componentes_all<-prcomp(Datos2_d2a_imp, scale=TRUE,center = TRUE)
componentes_all
summary(componentes_all) # autovalores 1 y prop.acumulada (variación total explicada) >70%
#Projected inertia (%) = Proportion of Variance #Cumulative projected inertia (%) = Cumulative Proportion
```

```
plot(componentes_all)
biplot(componentes_all, scale=0.5)
#Calculamos las contribuciones (absoluta y relativa) para ver la importancia de las mismas.
#acpi1<-inertia.dudi(acp, row.inertia=F, col.inertia=T) #variables-col # acpi1
#acpi2<-inertia.dudi(acp, row.inertia=T, col.inertia=F) #pacientes-row # acpi2
acpi<-inertia.dudi(acp, row.inertia=T, col.inertia=T) # vemos solo 2 componentes
# acpi
# s.label(acp$li) # Representamos los puntos anteriores para ver la gráfica
# acp$co # Vemos las columnas igual que las filas
# acp$c1
# s.label(acp$co)
#Contribuciones absolutas y relativas a la inercia de filas (o de columnas)
# acpi
# biplot(acp$co, acp$li) # Mostrar representación conjunta de filas y columnas
# s.corcircle(acp$li)  # Gráfico de correlaciones de las variables (FILAS)
# s.corcircle(acp$co) # Gráfico de correlaciones de las variables (COLUMNAS)
acpi12<-inertia.dudi(acp12, row.inertia=T, col.inertia=T) # vemos las 12 componentes finales
acpi12
acpi12_col<-inertia.dudi(acp12, row.inertia=F, col.inertia=T)
acpi12_col
acpi12_row<-inertia.dudi(acp12, row.inertia=T, col.inertia=F)
acpi12_row
# SINTAXIS ACPrincipales
acp$eig # mostramos los 32 autovalores observando que mayores 1 tenemos las 12 componentes
summary(componentes_all) # autovalores > 1 y prop.acumulada (variación total explicada) >70%
#Calculamos las contribuciones (absoluta y relativa) para ver la importancia de las mismas.
acpi12_col<-inertia.dudi(acp12, row.inertia=F, col.inertia=T)
acpi12_col #Como hemos realizado antes con las kmedias, sacamos directamente ACP
#cluster2_kmedias<-kmeans(Datos2_d2a_imp,4)
#cluster2_kmedias
acp_m=princomp(Datos2_d2a_imp)
acp_m
comp=predict(acp_m)[,1:2]
comp
km2=kmeans(comp,4)
km2
# K-means clustering with 4 clusters of sizes 2205, 590, 1527, 856
# Within cluster sum of squares by cluster:
# [1] 1420.0506 1020.3151 2430.9849  533.4728
# (between_SS / total_SS =  85.2 %)
plot(comp,col=km2$cluster2) #cada cluster de un color
points(km2$centers,col=1:4, pch=8,cex=2) #medias de cluster en las componentes
text(comp[,1],comp[,2],labels=rownames(Datos2_d2a_imp),col=km2$cluster2) #etiquetas nombres
plot(comp,col=km2$cluster2) #cada cluster de un color
points(km2$centers,col=1:4, pch=8,cex=2) #medias de los cluster en las componentes
text(comp[,1],comp[,2],labels=colnames(Datos2_d2a_imp),row=km2$cluster2) #etiquetas nombres
#Anotar que este grafico es el mismo que el biplot anterior y que se ve mejor que este de las kmedias del acp.
```

**# SINTAXIS PCA**

```
library(Factoshiny)
library(shiny)
library(FactoInvestigate)
prueba<-Factoshiny(d)
library(jquerylib)
install.packages("jquerylib")
library(shinyjqui)
install.packages("Factoshiny")
prueba<-Factoshiny(d)
res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
res.HCPC<-HCPC(res.PCA,nb.clust=3,kk=100,consol=FALSE,graph=FALSE)
plot.HCPC(res.HCPC,choice='tree',title='Hierarchical tree')
plot.HCPC(res.HCPC,choice='map',draw.tree=FALSE,title='Factor map')
plot.HCPC(res.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,title='Hierarchical tree on the factor map')
#res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
#res.HCPC<-HCPC(res.PCA,nb.clust=3,kk=100,consol=FALSE,graph=FALSE)
summary(res.HCPC)
res.PCA<-PCA(d,graph=FALSE)
plot.PCA(res.PCA,choix='var')
plot.PCA(res.PCA)
summary(res.PCA)
dimdesc(res.PCA)
res.PCA<-PCA(d,ncp=12,graph=FALSE)
plot.PCA(res.PCA,choix='var')
plot.PCA(res.PCA)
```

**# SINTAXIS ANALISIS CON RF&IV**

```
#Descargado e instalado nueva versión R-4.1.0-win.exe para actualizar salidas de análisis
d<-read.table(file="C:/Users/NisaB/Desktop/NBTS/Datos_d2a_imp_180918.txt", header=T)
View(d)
nombres<-names(d)
nombres
install.packages(tidyverse)
install.packages(rpart)
install.packages(rpart.plot)
install.packages(caret)
library(tidyverse)
library(rpart)
library(rpart.plot)
library(caret)
library(lattice)
summary(d)
str(d)
lapply(d,summary)
data.frame(colSums(is.na(d))) # missing completados.
dim(d) #[1] 5178   32
```

```
class(d) #[1] "data.frame"
indx = sapply(d,is.numeric) # transformamos a numerico las variables
indx[indx==1]
d[indx] = lapply(d[indx], function(x) as.numeric(as.character(x)))
str(d)
table(d$SEXO)
table(d$HT_)
table(d$ESPIROMETRIA_PA_)
table(d$INGRESOS_)
table(d$SV_)
table(d$EXACER_90DIAS)
table(d$Reing_EXAC)
table(d$MUERTOS_90DIAS)
table(d$EXITUS)
table(d$ICHARICC_)
table(d$CCVSDM_)
table(d$ICHAR_DM_)
table(d$EV_)
table(d$EV_)
table(d$ICHARECV_)
table(d$ICHAREVP_)
table(d$ICHARIM_)
table(d$ICHARNEF_)
table(d$ICHAR_TS_)
table(d$EP_)
d1 <-
d[,c("EDAD","DURING","ALP","PPKG","TARS","TARD","TURA","FRE","FCA","FEV1P","FVCP","IMC","FEVCVF_")]
names(d1)
randomForest(d)
#Call: randomForest(x = d)
#Type of random forest: unsupervised      Number of trees: 500      No. of variables tried at each split: 5
dpre_rf <- randomForest(d,mtry=2,ntree=50, importance = T)
dpre_rf
#Call: randomForest(x = d, ntree = 50, mtry = 2, importance = T)
#Type of random forest: unsupervised      Number of trees: 50
#No. of variables tried at each split: 2
importance(dpre_rf)
#                    1          2 MeanDecreaseAccuracy MeanDecreaseGini
# EDAD           9.143841  0.626080028          8.7733250       171.82978
# SEXO           5.478142  0.184612240          4.7420128        31.62448
# HT_            6.300911  0.797291197          5.6817315        36.84869
# DURING         1.511749  0.113384308          1.1442697       140.31719
# ESPIROMETRIA_PA_  6.217206  0.391951300          4.9162779        39.30613
# INGRESOS_        9.867001  1.214510884          9.9874045       196.81619
# SV_            4.435471  0.550175234          4.4291134        23.62432
# EXACER_90DIAS    10.324053  1.893885773         10.4303938       164.83231
# Reing_EXAC       9.646944  0.603461373          9.7687767       242.19560
```

```
# MUERTOS_90DIAS    5.236488 -0.717146783          4.3573722        22.99425
# EXITUS           7.817225  0.797672386          7.3099288        22.95276
# ICHARICC_        8.052441 -1.268385910          7.8028230        87.92548
# CCVSDM_         10.280063  1.180660427         10.2902921       209.72344
# ICHAR_DM_        6.461731  0.758599509          5.0503529        36.56857
# EV_             12.318730  0.005347732         12.5918006       194.26879
# ICHARECV_        9.621123 -0.581298165          8.5942876        78.67005
# ICHAREVP_       10.634662 -1.362293635         10.0308798        90.87474
# ICHARIM_        10.838686  0.357297475         11.0400014        85.35488
# ICHARNEF_        5.257814  0.197897255          5.0012959        25.97349
# ICHAR_TS_        2.094592  2.201231129          2.7284566        23.82815
# EP_              5.865147  0.318439420          4.9547959        39.31523
# ALP              5.900174  0.294388858          5.6057318       158.79108
# PPKG             7.651681  1.397470646          7.8844279       224.86344
# TARS             6.359958 -1.161851039          5.4241493       166.60532
# TARD             7.960946  1.830312410          8.1212066       157.85915
# TURA             1.740259  0.286460519          1.5069481       129.89077
# FRE              1.166290 -0.075725184          0.8819195       128.07834
# FCA              4.229378  0.998361602          4.1784895       158.19131
# FEV1P            9.979118 -0.192817197          9.8023887       206.09781
# FVCP             8.985454 -0.061255696          8.3811332       192.07075
# IMC              7.135210 -0.243645418          7.1303868       212.88434
# FEVCVF_         10.105388  1.225651900         10.2032513       233.34052
class(importance(dpre_rf)) #[1] "matrix" "array"
dimp_rf <- importance(dpre_rf)[,4]
dimp_rf <- data.frame(VARIABLE = names(dimp_rf), IMP_RF = dimp_rf)
dimp_rf <- dimp_rf %>% arrange(desc(IMP_RF)) %>% mutate(RANKING_RF = 1:nrow(dimp_rf))
dimp_rf
#                   VARIABLE    IMP_RF RANKING_RF
# Reing_EXAC          Reing_EXAC 242.19560         1
# FEVCVF_              FEVCVF_ 233.34052         2
# PPKG                 PPKG 224.86344         3
# IMC                  IMC 212.88434         4
# CCVSDM_               CCVSDM_ 209.72344         5
# FEV1P                FEV1P 206.09781         6
# INGRESOS_            INGRESOS_ 196.81619         7
# EV_                  EV_ 194.26879         8
# FVCP                 FVCP 192.07075         9
# EDAD                 EDAD 171.82978         10
# TARS                 TARS 166.60532         11
# EXACER_90DIAS      EXACER_90DIAS 164.83231         12
# ALP                  ALP 158.79108         13
# FCA                  FCA 158.19131         14
# TARD                 TARD 157.85915         15
# DURING               DURING 140.31719         16
# TURA                 TURA 129.89077         17
# FRE                  FRE 128.07834         18
```

```
# ICHAREVP_             ICHAREVP_  90.87474        19
# ICHARICC_             ICHARICC_  87.92548        20
# ICHARIM_              ICHARIM_  85.35488        21
# ICHARECV_             ICHARECV_  78.67005        22
# EP_                   EP_  39.31523        23
# ESPIROMETRIA_PA_ ESPIROMETRIA_PA_  39.30613        24
# HT_                   HT_  36.84869        25
# ICHAR_DM_             ICHAR_DM_  36.56857        26
# SEXO                  SEXO  31.62448        27
# ICHARNEF_             ICHARNEF_  25.97349        28
# ICHAR_TS_             ICHAR_TS_  23.82815        29
# SV_                   SV_  23.62432        30
# MUERTOS_90DIAS     MUERTOS_90DIAS  22.99425        31
# EXITUS                EXITUS  22.95276        32
names<-names(d)
names
lista<-c(dimp_rf$VARIABLE[1:12])
lista
# [1] "Reing_EXAC"    "FEVCVF_"      "PPKG"         "IMC"          "CCVSDM_"
# [6] "FEV1P"         "INGRESOS_"    "EV_"          "FVCP"         "EDAD"
# [11] "TARS"         "EXACER_90DIAS"
dtemp<-mutate(d)
View(dtemp)
help("smbinning")
class(dtemp)
str(dtemp)
lista<-names(d)
dimp_iv <- smbinning.sumiv(dtemp[c(lista,'SEXO')],y="SEXO")
  |-------------------------------------------------| 100%
View(dimp_iv)
dimp_iv <- dimp_iv %>% mutate(Ranking = 1:nrow(dimp_iv), IV = ifelse(is.na(.$IV),0,IV)) %>% select(-Process)
View(dimp_iv)
names(dimp_iv) <- c('VARIABLE','IMP_IV','RANKING_IV')
names(dimp_iv)
# [1] "VARIABLE"   "IMP_IV"     "RANKING_IV"
View(dimp_iv)
dimp_iv
#          VARIABLE IMP_IV RANKING_IV
# 21            ALP 0.2265         1
# 1            EDAD 0.2186         2
# 28          FEV1P 0.0487         3
# 29           FVCP 0.0310         4
# 22           PPKG 0.0244         5
# 2            HT_ 0.0000         6
# 3          DURING 0.0000         7
# 4  ESPIROMETRIA_PA_ 0.0000         8
```

```
# 5        INGRESOS_ 0.0000         9
# 6             SV_ 0.0000        10
# 7     EXACER_90DIAS 0.0000        11
# 8       Reing_EXAC 0.0000        12
# 9    MUERTOS_90DIAS 0.0000        13
# 10          EXITUS 0.0000        14
# 11       ICHARICC_ 0.0000        15
# 12        CCVSDM_ 0.0000         16
# 13       ICHAR_DM_ 0.0000        17
# 14            EV_ 0.0000        18
# 15       ICHARECV_ 0.0000        19
# 16       ICHAREVP_ 0.0000        20
# 17        ICHARIM_ 0.0000        21
# 18       ICHARNEF_ 0.0000        22
# 19       ICHAR_TS_ 0.0000        23
# 20            EP_ 0.0000        24
# 23           TARS 0.0000        25
# 24           TARD 0.0000        26
# 25           TURA 0.0000        27
# 26            FRE 0.0000        28
# 27            FCA 0.0000        29
# 30            IMC 0.0000        30
# 31         FEVCVF_ 0.0000        31
# 32          SEXO.1 0.0000        32
lista_IV<-c(dimp_iv$VARIABLE[1:5])
lista_IV
# [1] "ALP"   "EDAD"  "FEV1P" "FVCP"  "PPKG"
dimp_final <- inner_join(dimp_rf,dimp_iv,by='VARIABLE') %>%
+ select(VARIABLE,IMP_RF,IMP_IV,RANKING_RF,RANKING_IV) %>%
+ mutate(RANKING_TOT = RANKING_RF + RANKING_IV) %>%
+     arrange(RANKING_TOT)
dimp_final   # TABLA FINAL  RF&IV
#          VARIABLE    IMP_RF IMP_IV RANKING_RF RANKING_IV RANKING_TOT
# 1            PPKG 224.86344 0.0244         3         5          8
# 2           FEV1P 206.09781 0.0487         6         3          9
# 3            EDAD 171.82978 0.2186        10         2         12
# 4       Reing_EXAC 242.19560 0.0000         1        12         13
# 5            FVCP 192.07075 0.0310         9         4         13
# 6             ALP 158.79108 0.2265        13         1         14
# 7        INGRESOS_ 196.81619 0.0000         7         9         16
# 8         CCVSDM_ 209.72344 0.0000         5        16         21
# 9     EXACER_90DIAS 164.83231 0.0000        12        11         23
# 10          DURING 140.31719 0.0000        16         7         23
# 11            EV_ 194.26879 0.0000         8        18         26
# 12            HT_  36.84869 0.0000        25         6         31
# 13 ESPIROMETRIA_PA_  39.30613 0.0000        24         8         32
# 14         FEVCVF_ 233.34052 0.0000         2        31         33
```

```
# 15            IMC 212.88434 0.0000         4        30         34
# 16       ICHARICC_  87.92548 0.0000        20        15         35
# 17           TARS 166.60532 0.0000        11        25         36
# 18       ICHAREVP_  90.87474 0.0000        19        20         39
# 19            SV_  23.62432 0.0000        30        10         40
# 20           TARD 157.85915 0.0000        15        26         41
# 21       ICHARECV_  78.67005 0.0000        22        19         41
# 22        ICHARIM_  85.35488 0.0000        21        21         42
# 23            FCA 158.19131 0.0000        14        29         43
# 24       ICHAR_DM_  36.56857 0.0000        26        17         43
# 25           TURA 129.89077 0.0000        17        27         44
# 26   MUERTOS_90DIAS  22.99425 0.0000        31        13         44
# 27            FRE 128.07834 0.0000        18        28         46
# 28         EXITUS  22.95276 0.0000        32        14         46
# 29            EP_  39.31523 0.0000        23        24         47
# 30       ICHARNEF_  25.97349 0.0000        28        22         50
# 31       ICHAR_TS_  23.82815 0.0000        29        23         52
View(dimp_final)
cor(dimp_final$IMP_RF,dimp_final$IMP_IV)
#Correlación baja cerca a 0 es muy baja, malisima  # [1] 0.2065059
#No tiene sentido comparar ambos metodos en este caso porque cada uno tiene su particularidad.
names
lista
# [1] "Reing_EXAC"    "FEVCVF_"      "PPKG"         "IMC"          "CCVSDM_"
# [6] "FEV1P"         "INGRESOS_"    "EV_"          "FVCP"         "EDAD"
# [11] "TARS"         "EXACER_90DIAS"
lista_IV
# [1] "ALP"   "EDAD"  "FEV1P" "FVCP"  "PPKG"
lista_final<-append(lista,lista_IV[1])
lista_final
# [1] "Reing_EXAC"    "FEVCVF_"      "PPKG"         "IMC"          "CCVSDM_"
# [6] "FEV1P"         "INGRESOS_"    "EV_"          "FVCP"         "EDAD"
# [11] "TARS"         "EXACER_90DIAS"     "ALP"
d_final <- d
names(d_final)
dim(d_final)  #[1] 5178   32
lista_final<-c(lista_final)
lista_final
d_final<-d_final[c(lista_final)]
names(d_final)
dim(d_final)  #[1] 5178   13
View(d_final)
```

########################################  **CAPITULO III**  ########################################

**# SINTAXIS PARA IDENTIFICAR GRUPOS – DISITNTOS MÉTODOS MEDIANTE CLUSTER, AC y DT**

**# SINTAXIS ANALISIS CLUSTER en base imputada final "Datos_d2a_imp_180918" con método "simple"**

```
#LISTA INICIAL DE PAQUETES que vamos a usar en estos desarrollos
paquetes <- c("e1071", "fda", "splines", "Matrix", "MASS", “nnet”, “mice”, "lattice", "VIM", “cluster”, ”ade4”,
"graphics", "broom", "Rcpp", "robustbase", "sp", "curl" , "haven" , "colorspace", "grid", "data.table",
"lmtest", "zoo", “base” )
#Crea un vector lógico si están instalados o no y s.i hay al menos uno no instalado los instala
instalados <- paquetes %in% installed.packages()
if(sum(instalados == FALSE) > 0) {
  install.packages(paquetes[!instalados])
}
lapply(paquetes,require,character.only = TRUE)

Datos_d2a_imp<-
read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/Datos_d2a_imp_180918.txt",header=T)
View(Datos_d2a_imp)
names(Datos_d2a_imp)
View(Datos_d2a_imp)
names(Datos_d2a_imp)
str(Datos_d2a_imp)
indx = sapply(Datos_d2a_imp,is.numeric)
indx[indx==1]
Datos_d2a_imp[indx] = lapply(Datos_d2a_imp[indx], function(x) as.numeric(as.character(x)))
str(Datos_d2a_imp)
#Tipificamos porque los datos son muy dispares antes de ejecutar el cluster
library(cluster)
Datos2_d2a_imp<-scale(Datos_d2a_imp)
View(Datos2_d2a_imp)
#Seleccionamos como medida la distancia euclídea
distancia<-dist(Datos2_d2a_imp, method="euclidean")
View(as.matrix(distancia))
distancia1=as.matrix(distancia)
View(distancia1)
write.table(distancia1, file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/distancia1_180920.txt",col.names=TRUE)
distancia1<-
read.table(file="C:/Users/NisaB/Desktop/1_Trabajo_UC3M_JUNIO_v2/distancia1_180920.txt",header=T)
View(distancia1)
#Usamos diferentes métodos para elegir el mejor:(1)Linkage Simple,(2)Completo,(3)Media y (4)Ward)
cluster1<-hclust(distancia,method="complete")
cluster2<-hclust(distancia,method="single")
cluster3<-hclust(distancia,method="ave")
cluster4<-hclust(distancia,method="ward.D")
#Vemos los 4 métodos y mejoramos las salidas de los gráficos
plot(cluster1)
plot(cluster2)
plot(cluster3)
plot(cluster4)
#cluster1<-hclust(distancia,method="complete")
plot(cluster1,cex=0.6, hang=-1)
plot(cluster1,col = "dark green", lwd = 1, cex=0.01,hang=-2.1)
```

```
plot(cluster1,col = "dark green", lwd = 1, cex=0.01,hang=-3.1)
plot(cluster1,col = "dark green", lwd = 1, cex=0.01,hang=-6.1)
plot(cluster1,col = "dark green", lwd = 1, cex=0.01,hang=0)
plot(cluster1,col = "dark green", lwd = 1, cex=0.0001,hang=0)
plot(cluster1,col = "dark green", lwd = 1, cex=0.0001,hang=0,main = "Cluster Dendrogram (Linkage Complete)")
#cluster2<-hclust(distancia,method="single")
plot(cluster2,cex=0.6, hang=-1)
plot(cluster2,col = "dark blue", lwd = 1, cex=0.01,hang=0.9,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.01,hang=0.1,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.01,hang=-6,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.01,hang=-0.1,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.001,hang=0,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.01,hang=0,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd = 1, cex=0.0001,hang=0,main = "Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd =1,cex=0.00001,hang=0,ylim="",main="Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd =1,cex=0.3,hang=0,main="Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd =1,cex=0.05,hang=0,main="Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd =1,cex=0.00000001,hang=0,main="Cluster Dendrogram (Linkage Single)")
plot(cluster2,col = "dark blue", lwd =1,cex=0.0000001,hang=0,main = "Cluster Dendrogram (Linkage Single)")
#cluster3<-hclust(distancia,method="ave")
plot(cluster3,cex=0.6, hang=-1)
plot(cluster3,col = "dark orange", lwd = 1, cex=0.6,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.3,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.1,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.0000001,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.001,hang=-6,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.001,hang=-2,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.001,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.05,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.01,hang=0,main = "Cluster Dendrogram (Average)")
plot(cluster3,col = "dark orange", lwd = 1, cex=0.0000001,hang=0,main = "Cluster Dendrogram (Average)")
#cluster4<-hclust(distancia,method="ward.D")
plot(cluster4,cex=0.6, hang=-1)
plot(cluster4,col = "dark red", lwd = 1, cex=0.6,hang=-1,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.1,hang=5,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.1,hang=-5,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.1,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.1,hang=-9,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.1,hang=-3,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=1,hang=-3,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0000000000001,hang=-3,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0000000000001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00010,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00010,hang=0.10,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00010,hang=0.0001,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00010,hang=-0.0001,main = "Cluster Dendrogram (Ward.D)")
```

```
plot(cluster4,col = "dark red", lwd = 1, cex=0.00010000,hang=0.0001,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0001,hang=0.0001,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.000001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.01,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.0000000000000001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.000000001,hang=0,main = "Cluster Dendrogram (Ward.D)")
plot(cluster4,col = "dark red", lwd = 1, cex=0.00001,hang=0,main = "Cluster Dendrogram (Ward.D)")
# Calculamos los coeficientes cofenéticos para decidir cuál de los métodos es mejor.
# Primero se calculan las matrices cofenéticas de los cuatro métodos.
# Después, la correlación de estas con la matriz de distancias, en este caso, euclídea.
co11<-cophenetic(cluster1)
co12<-cophenetic(cluster2)
co13<-cophenetic(cluster3)
co14<-cophenetic(cluster4)
cor(distancia,co11)  #cluster1 (distancia,method="complete") # Distancia máxima [1] 0.5339575
cor(distancia,co12)  #cluster2 (distancia,method="single")      # Distancia mínima  [1] 0.7608009
cor(distancia,co13)  #cluster3 (distancia,method="ave")         # Distancia media    [1] 0.7369236
cor(distancia,co14)  #cluster4 (distancia,method="ward.D")                                [1] 0.2906593
print(cluster1) #(Distance:euclidean, method = "complete")
print(cluster2) #(Distance:euclidean, method = "single")
print(cluster3) #(Distance:euclidean, method = "ave")
print(cluster4) #(Distance:euclidean, method = "ward.D")
#Realizamos cortes (k) en los cluster para mejorar visualización y detectar si podemos reducir los grupos a 4 ó 5
View(Datos2_d2a_imp)
# indicamos k=20 grupos de cluster
t1<-cutree(cluster1,k=20)
# plot(t1)
library(factoextra)
library(ggplot2)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t1))
t2<-cutree(cluster2,k=20)
#plot(t2)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t2))
t3<-cutree(cluster3,k=20)
#plot(t3)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t3))
t4<-cutree(cluster4,k=20)
#plot(t4)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t4))
# indicamos k=15 grupos de cluster
t11<-cutree(cluster1,k=15)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t11))
t22<-cutree(cluster2,k=15)
```

```
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t22))
t33<-cutree(cluster3,k=15)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t33))
t44<-cutree(cluster4,k=15)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t44))
# indicamos k=10 grupos de cluster
t111<-cutree(cluster1,k=10)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t111))
t222<-cutree(cluster2,k=10)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t222))
t333<-cutree(cluster3,k=10)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t333))
t444<-cutree(cluster4,k=10)
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t444))
# indicamos k=5 grupos de cluster
t1111<-cutree(cluster1,k=5) # Se visualizan muchos grupos - Dist.Max
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t1111))
t2222<-cutree(cluster2,k=5) # Se visualizan 4 grupos -  Dist.Minima
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t2222))
t3333<-cutree(cluster3,k=5) # Se visualizan 5 grupos con la media
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t3333))
t4444<-cutree(cluster4,k=5) # Se visualizan muchos grupos con Ward
fviz_cluster(list(data=Datos2_d2a_imp,cluster=t4444))
#Gráficos variando Kmedias se insertan al texto.
# Realizamos método de Kmeans para mejorar visualización de cluster con grupos 4 ó 5 detectados en los cluster 2 y 3
#Usar Kmeans indicando sólo número de grupos 4 o 5, detectado cluster 2 y 3 mejores según coef.cofenetico
Datos2_d2a_imp<-scale(Datos_d2a_imp)
View(Datos2_d2a_imp)
cluster2_kmedias<-kmeans(Datos2_d2a_imp,4)
cluster2_kmedias
# K-means clustering with 4 clusters of sizes 2416, 259, 1376, 1127
# Within cluster sum of squares by cluster:
# [1] 54123.766  8481.109 37941.984 35900.748
# (between_SS / total_SS =  17.6 %)
View(Datos2_d2a_imp)
cluster3_kmedias<-kmeans(Datos2_d2a_imp,5)
cluster3_kmedias
# K-means clustering with 5 clusters of sizes 816, 2269, 618, 407, 1068
# Within cluster sum of squares by cluster:
# [1] 17384.77 50580.77 19712.67 12850.50 33778.71
# (between_SS / total_SS =  18.9 %)
#Posiblemente optimzar algo más, nos ayudaremos con el analisis de correspondencia y chequeamos nuevamente.
# Ver estos 4 grupos mediante la ayuda del ACP y luego ver AC
# Cluster 5 desde PCA
res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
res.HCPC<-HCPC(res.PCA,nb.clust=5,consol=FALSE,graph=FALSE)
```

```
plot.HCPC(res.HCPC,choice='tree',title='Hierarchical tree')
plot.HCPC(res.HCPC,choice='map',draw.tree=FALSE,title='Factor map')
plot.HCPC(res.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,title='Hierarchical tree on the factor map')
# Cluster 4 desde PCA
res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
res.HCPC<-HCPC(res.PCA,nb.clust=4,consol=FALSE,graph=FALSE)
plot.HCPC(res.HCPC,choice='tree',title='Hierarchical tree')
plot.HCPC(res.HCPC,choice='map',draw.tree=FALSE,title='Factor map')
plot.HCPC(res.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,title='Hierarchical tree on the factor map')
# Cluster 3 desde PCA
res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
res.HCPC<-HCPC(res.PCA,nb.clust=3,consol=FALSE,graph=FALSE)
plot.HCPC(res.HCPC,choice='tree',title='Hierarchical tree')
plot.HCPC(res.HCPC,choice='map',draw.tree=FALSE,title='Factor map')
plot.HCPC(res.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,title='Hierarchical tree on the factor map')
res.PCA<-PCA(d,ncp=Inf, scale.unit=FALSE,graph=FALSE)
res.HCPC<-HCPC(res.PCA,nb.clust=3,consol=FALSE,graph=FALSE)
summary(res.HCPC)
```

**# SINTAXIS CLUSTER PARA OPTIMIZAR NUMERO DE GRUPOS OPTIMOS**

```
d<-read.table(file="C:/Users/NisaB/Desktop/NBTS/Datos_d2a_imp_180918.txt", header=T)
View(d)
datos <- scale(d)
library(factoextra)
#Loading required package: ggplot2
library(ggplot2)
fviz_nbclust(x = datos, FUNcluster = kmeans, method = "wss", k.max = 15, diss = get_dist(datos, method = "euclidean"), nstart = 50)
# Figure 1. method = " euclidean " para dataset numérico completo y sin missing
library(purrr)
#Attaching package: 'purrr'
#The following object is masked from 'package:maps':map
library(maps)
#Esta función aplica el algoritmo kmeans y devuelve la suma total de cuadrados internos
calcular_totwithinss <- function(n_clusters, datos, iter.max=1000, nstart=50){
+   cluster_kmeans <- kmeans(centers = n_clusters, x = datos, iter.max = iter.max,
+                            nstart = nstart)
+   return(cluster_kmeans$tot.withinss)
+ }
# Se aplica esta función para diferentes valores de k
total_withinss <- map_dbl(.x = 1:15, .f = calcular_totwithinss, datos = datos)
total_withinss
# [1] 165664.0  151488.4   141124.5   136447.6  132469.7   128982.6   125930.9   122952.5
# [9] 120106.1  118287.2   116324.0   114925.6  113100.4   112144.4   111063.4
data.frame(n_clusters = 1:15, suma_cuadrados_internos = total_withinss) %>%
```

```
+   ggplot(aes(x = n_clusters, y = suma_cuadrados_internos)) +
+     geom_line() +
+     geom_point() +
+     scale_x_continuous(breaks = 1:15) +
+     labs(title = "Evolución de la suma total de cuadrados intra-cluster") +
+     theme_bw()
# Figure 3. es la misma de la Fig.1 pero de otro modo calculado y salen mismos resultados
library(cluster)
library(factoextra)
fviz_nbclust(x = datos, FUNcluster = pam, method = "wss", k.max = 15,
+             diss = dist(datos, method = "manhattan"))
# Figure 2. misma anterior, pero para cuando existen valores perdidos method = "manhattan" para missing
dataset
# library(factoextra)
# datos <- scale(d)
View(datos)
fviz_nbclust(x = datos, FUNcluster = kmeans, method = "silhouette", k.max = 15)+labs(title="Optimal number of
clusters")
#Etiqueta modificada para el tiutlo en español en el gráfico de óptimo
fviz_nbclust(x = datos, FUNcluster = kmeans, method = "silhouette", k.max = 15)+labs(title="Número óptimo de
clusters")
# Figura 4. La silueta de la curva indica 3 clusters óptimo como mejor separación de agrupamiento de los datos.
# NOTA. La última salida confirma lo que se observaba al inicio, el óptimo son 3 clusters, luego hay 3 grupos finales.
# SINTAXIS ANALISIS CA
library(FactoMineR)
library(ade4)
corres11 <- CA(d)
corres11
# Results of the Correspondence Analysis (CA)**
# The row variable has  5178  categories; the column variable has 32 categories
# The chi square of independence between the two variables is equal to 280212.3 (p-value =  0 ).
# The results are available in the following objects:
#   name            description
# 1  "$eig"          "eigenvalues"
# 2  "$col"          "results for the columns"
# 3  "$col$coord"     "coord. for the columns"
# 4  "$col$cos2"      "cos2 for the columns"
# 5  "$col$contrib"   "contributions of the columns"
# 6  "$row"          "results for the rows"
# 7  "$row$coord"     "coord. for the rows"
# 8  "$row$cos2"      "cos2 for the rows"
# 9  "$row$contrib"   "contributions of the rows"
# 10 "$call"         "summary called parameters"
# 11 "$call$marge.col" "weights of the columns"
# 12 "$call$marge.row" "weights of the rows"
corres1 <- CA(d)
summary(corres1)
```

```
# Call: CA(X = d)     #The chi square of independence between the two variables is equal to 280212.3 (p-value =  0
).
# Eigenvalues
# Rows (the 10 first)        Iner*1000    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
# Columns (the 10 first)  Iner*1000    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
max.porc <- max(1/(dim(d)-1)*100)
max.porc   # [1] 3.225806
library(factoextra)
fviz_screeplot(corres1)+geom_hline(yintercept = max.porc, linetype = 2, color = "red")+labs(title="Gráfico de
sedimentación", x = "Dimensiones",  y = "Porcentaje de variabilidad explicada")
library(factoextra)
fviz_screeplot(corres11)+geom_hline(yintercept=max.porc,linetype=2,color="red")+labs(title="Gráfico de
sedimentación",x="Dimensiones",y="Porcentaje de variabilidad explicada")
fviz_ca_row(corres1, col.row = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE)
fviz_ca_row(corres11, col.row = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE)
fviz_ca_row(corres1, col.row = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE)
fviz_ca_col(corres1, col.col = "cos2",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE)
fviz_contrib(corres11, choice = "row", axes = 1:2)
fviz_contrib(corres11, choice = "col", axes = 1:2)
fviz_contrib(corres1, choice = "row", axes = 1:2)
fviz_contrib(corres1, choice = "col", axes = 1:2)
corrplot(corres1$row$contrib, is.corr = FALSE)
# Este gráfico visualiza la contribución atributo-variable para cada dimensión por separado
corrplot(corres1$col$contrib, is.corr = FALSE)
ellipseCA(corres1)
#corres2 <- MCA(d,  method = "Burt", na.method = "average") # Multiple Correspondence Analysis (MCA)
str(d)
class(d)
help("MCA")
corres2 <- MCA(d, method="Burt",na.method="average")
corres2 <- MCA(d, method="Indicator")
corres2 <- MCA(d, method="Burt",na.method="average",quanti.sup=13, quali.sup=19)
#corres2 <- MCA(d,quanti.sup=13, quali.sup=2:3&5:21, method="Burt",na.method="average")
#CORRESPONDENCE ANALYSIS (CA)
res.CA<-CA(d,graph=FALSE)
plot.CA(res.CA)
res1.CA<-CA(d,graph=FALSE)
summary(res1.CA)
res2.CA<-CA(d,graph=FALSE)
ellipseCA(res2.CA,ellipse=c('col'))
res3.CA<-CA(d,graph=FALSE)
plot.CA(res3.CA)
```

**# SINTAXIS ANÁLISIS DE CLASIFICACIÓN POR ÁRBOLES DE DECISIÓN - DT**

```
d<-read.table(file="C:/Users/nboukichou/NISA_RIOJASALUD/CursoR-Rstudio/Datos_d2a_imp_180918.txt",
header=T)
View(d)
```

```
dim(d) #[1] 5178   32
d_final <- d
names(d_final)
dim(d_final)  #[1] 5178   32
lista_final<-c(lista_final)
lista_final
d_final<-d_final[c(lista_final)]
names(d_final)
dim(d_final)  #[1] 5178   13
View(d_final)
set.seed(500)
d_entrenamientof <- sample_frac(d_final, .7)
d_pruebaf <- setdiff(d_final, d_entrenamientof)
View(d_entrenamientof)
View(d_pruebaf)
arbol_1f <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof)
arbol_1f
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 825.0621 0.3503448
# 2) Reing_EXAC< 0.5 2642 255.8232 0.1086298 *
#  3) Reing_EXAC>=0.5 983   0.0000 1.0000000 *
rpart.plot(arbol_1f)
prediccion_1f <- predict(arbol_1f, newdata = d_pruebaf, type = "class")
### prediccion_1f <- predict(arbol_1f,d_pruebaf)
set.seed(1000)
d_entrenamientof_2 <- sample_frac(d_final, .7)
d_pruebaf_2 <- setdiff(d_final, d_entrenamientof_2)
arbol_2f <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof_2)
arbol_2f
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 826.2549 0.3514483
#  2) Reing_EXAC< 0.5 2653 267.6223 0.1138334 *
#  3) Reing_EXAC>=0.5 972   0.0000 1.0000000 *
rpart.plot(arbol_2f)
### prediccion_2f <- predict(arbol_2f,d_pruebaf_2)
set.seed(1500)
d_entrenamientof_3 <- sample_frac(d_final, .7)
d_pruebaf_3 <- setdiff(d_final, d_entrenamientof_3)
arbol_3f <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof_3)
arbol_3f
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
```

```
# 1) root 3625 822.6499 0.3481379
#  2) Reing_EXAC< 0.5 2647 253.5293 0.1072913 *
#  3) Reing_EXAC>=0.5 978   0.0000 1.0000000 *
rpart.plot(arbol_3f)
### prediccion_3f <- predict(arbol_3f,d_pruebaf_3)
set.seed(500)
d_entrenamiento <- sample_frac(d, .7)
d_prueba <- setdiff(d, d_entrenamiento)
View(d_entrenamiento)
View(d_prueba)
arbol_1 <- rpart(formula = SEXO ~ ., data = d_entrenamiento)
arbol_1
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 388.10650 0.12193100
#  2) HT_>=0.5 2984 234.03750 0.08579088
#    4) EDAD>=64.5 2348 110.26920 0.04940375 *
#    5) EDAD< 64.5 636 109.18240 0.22012580
#    10) ALP>=1.585 490  70.90612 0.17551020 *
#     11) ALP< 1.585 146  34.02740 0.36986300 *
#   3) HT_< 0.5 641 132.02810 0.29017160
#    6) ALP>=1.585 469  83.66738 0.23240940 *
#    7) ALP< 1.585 172  42.52907 0.44767440 *
rpart.plot(arbol_1)
prediccion_1 <- predict(arbol_1, newdata = d_prueba, type = "class")
#prediccion_1
#confusionMatrix(prediccion_1, d_prueba[["SEXO"]])
## prediccion_1 <- predict(arbol_1,d_prueba)
## prediccion_1
## View(prediccion_1)
set.seed(1000)
d_entrenamiento_2 <- sample_frac(d, .7)
d_prueba_2 <- setdiff(d, d_entrenamiento_2)
arbol_2 <- rpart(formula = SEXO ~ ., data = d_entrenamiento_2)
arbol_2
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 408.32110 0.12937930
#   2) HT_>=0.5 2975 236.45180 0.08705882
#    4) EDAD>=63.5 2390 118.46230 0.05230126 *
#    5) EDAD< 63.5 585 103.30600 0.22905980
#     10) ALP>=1.595 425  57.79765 0.16235290 *
#     11) ALP< 1.595 160  38.59375 0.40625000 *
#   3) HT_< 0.5 650 142.15380 0.32307690
#    6) ALP>=1.585 474  92.97257 0.26793250 *
```

```
#     7) ALP< 1.585 176  43.85795 0.47159090
rpart.plot(arbol_2)
### prediccion_2 <- predict(arbol_2, newdata = d_prueba_2, type = "class")
### confusionMatrix(prediccion_2, d_prueba_2[["SEXO"]])
set.seed(1500)
d_entrenamiento_3 <- sample_frac(d, .7)
d_prueba_3 <- setdiff(d, d_entrenamiento_3)
arbol_3 <- rpart(formula = SEXO ~ ., data = d_entrenamiento_3)
arbol_3
# n= 3625
# node), split, n, deviance, yval
#       * denotes terminal node
# 1) root 3625 397.88970 0.12551720
#   2) HT_>=0.5 2986 235.70800 0.08640322
#     4) EDAD>=64.5 2331 113.82240 0.05148005 *
#     5) EDAD< 64.5 655 108.92520 0.21068700
#       10) ALP>=1.585 515  72.30291 0.16893200 *
#       11) ALP< 1.585 140  32.42143 0.36428570 *
#   3) HT_< 0.5 639 136.26600 0.30829420
#     6) ALP>=1.515 579 116.23140 0.27806560
#       12) EDAD< 87.5 518  96.87452 0.24903470 *
#       13) EDAD>=87.5 61  15.21311 0.52459020 *
#     7) ALP< 1.515 60  14.40000 0.60000000 *
rpart.plot(arbol_3)
### prediccion_3 <- predict(arbol_3, newdata = d_prueba_3, type = "class")
### confusionMatrix(prediccion_3, d_prueba_3[["SEXO"]])
set.seed(500)
d_entrenamiento <- sample_frac(d, .7)
d_prueba <- setdiff(d, d_entrenamiento)
arbol_1 <- rpart(formula = HT_ ~ ., data = d_entrenamiento)
arbol_1
# n= 3625
# node), split, n, deviance, yval
#       * denotes terminal node
# 1) root 3625 527.65350 0.8231724
#   2) SEXO>=0.5 442 107.72850 0.5791855
#     4) EDAD>=72.5 226  47.93363 0.3053097 *
#     5) EDAD< 72.5 216  25.10648 0.8657407 *
#   3) SEXO< 0.5 3183 389.95920 0.8570531
#     6) ESPIROMETRIA_PA_< 0.5 991 171.71540 0.7769929 *
#     7) ESPIROMETRIA_PA_>=0.5 2192 209.02010 0.8932482 *
rpart.plot(arbol_1)
set.seed(1000)
d_entrenamiento_2 <- sample_frac(d, .7)
d_prueba_2 <- setdiff(d, d_entrenamiento_2)
arbol_2 <- rpart(formula = HT_ ~ ., data = d_entrenamiento_2)
arbol_2
```

```
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 520.510300 0.8262069
#   2) SEXO>=0.5 474 113.839700 0.5991561
#     4) EDAD>=77.5 168  28.285710 0.2142857 *
#     5) EDAD< 77.5 306  47.006540 0.8104575
#      10) EDAD>=64.5 145  32.758620 0.6551724 *
#      11) EDAD< 64.5 161   7.602484 0.9503106 *
#   3) SEXO< 0.5 3151 378.559200 0.8603618
#     6) ESPIROMETRIA_PA_< 0.5 983 170.763000 0.7761953 *
#     7) ESPIROMETRIA_PA_>=0.5 2168 197.675300 0.8985240 *
 rpart.plot(arbol_2)
set.seed(1500)
d_entrenamiento_3 <- sample_frac(d, .7)
d_prueba_3 <- setdiff(d, d_entrenamiento_3)
arbol_3 <- rpart(formula = HT_ ~ ., data = d_entrenamiento_3)
arbol_3
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 526.35970 0.8237241
#  2) SEXO>=0.5 455 111.70550 0.5670330
#    4) EDAD>=73.5 218  42.09633 0.2614679 *
#    5) EDAD< 73.5 237  30.53165 0.8481013 *
#  3) SEXO< 0.5 3170 380.37100 0.8605678
#    6) ESPIROMETRIA_PA_< 0.5 943 163.78790 0.7762460 *
#    7) ESPIROMETRIA_PA_>=0.5 2227 207.03910 0.8962730 *
rpart.plot(arbol_3)
# SINTAXIS AJUSTADA DEL ARBOL PARA MOSTRARLO COMPLETO
View(d)
library(tidyverse)
library(lattice)
str(d)
indx = sapply(d,is.numeric) # transformamos a numerico las variables
indx[indx==1]
d[indx] = lapply(d[indx], function(x) as.numeric(as.character(x)))
str(d)
table(d$INGRESOS_)   # 0    1         3343 1835
table(d$HT_)             #  0    1          906 4272
table(d$SEXO)          #  0    1        4526  652
d_final <- d
View(d_final)
names(d_final)
dim(d_final)  #[1] 5178   32
set.seed(500)
d_entrenamientof <- sample_frac(d_final, .7)
```

```
d_pruebaf <- setdiff(d_final, d_entrenamientof)
View(d_entrenamientof)
View(d_pruebaf)
arbol_1f <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof)
arbol_1f
# n= 3625
# node), split, n, deviance, yval
#      * denotes terminal node
# 1) root 3625 825.06210 0.35034480
#  2) Reing_EXAC< 0.5 2642 255.82320 0.10862980
#    4) MUERTOS_90DIAS< 0.5 2559 224.77140 0.09730363 *
#    5) MUERTOS_90DIAS>=0.5 83  20.60241 0.45783130 *
#  3) Reing_EXAC>=0.5 983   0.00000 1.00000000 *
rpart.plot(arbol_1f)
arbol_1ff <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof,control=rpart.control(cp=0.001))
arbol_1ff
#n= 3625
#node), split, n, deviance, yval
#      * denotes terminal node
  1) root 3625 825.0621000 0.35034480
    2) Reing_EXAC< 0.5 2642 255.8232000 0.10862980
      4) MUERTOS_90DIAS< 0.5 2559 224.7714000 0.09730363
        8) EXITUS< 0.5 181   0.0000000 0.00000000 *
        9) EXITUS>=0.5 2378 222.9272000 0.10470980
         18) ICHAREVP_< 0.5 2027 173.8135000 0.09472126
           36) TURA< 36.25 618  36.5388300 0.06310680 *
           37) TURA>=36.25 1409 136.3861000 0.10858770
             74) TARD< 72.5 616  41.7126600 0.07305195
              148) FEVCVF_>=30.665 605  38.2214900 0.06776860 *
              149) FEVCVF_< 30.665 11   2.5454550 0.36363640 *
             75) TARD>=72.5 793  93.2913000 0.13619170
              150) FRE>=23.5 449  42.8686000 0.10690420
                300) TARS>=131 325  22.2276900 0.07384615 *
                301) TARS< 131 124  19.3548400 0.19354840
                  602) DURING< 10.5 90  10.4000000 0.13333330
                   1204) FEVCVF_>=55.65 68   4.6323530 0.07352941 *
                   1205) FEVCVF_< 55.65 22   4.7727270 0.31818180
                     2410) FEVCVF_< 40.595 8   0.0000000 0.00000000 *
                     2411) FEVCVF_>=40.595 14   3.5000000 0.50000000 *
                  603) DURING>=10.5 34   7.7647060 0.35294120
                   1206) FVCP< 67.5 24   3.9583330 0.20833330 *
                   1207) FVCP>=67.5 10   2.1000000 0.70000000 *
              151) FRE< 23.5 344  49.5348800 0.17441860
                302) DURING< 9.5 230  27.5478300 0.13913040
                  604) FEVCVF_< 86.915 177  16.1694900 0.10169490 *
                  605) FEVCVF_>=86.915 53  10.3018900 0.26415090
                   1210) FEVCVF_>=101.665 20   0.0000000 0.00000000 *
                   1211) FEVCVF_< 101.665 33   8.0606060 0.42424240
                     2422) TURA>=36.8 18   3.1111110 0.22222220 *
                     2423) TURA< 36.8 15   3.3333330 0.66666670 *
```

```
          303) DURING>=9.5 114  21.1228100 0.24561400
            606) TARS< 147.5 65   8.4615380 0.15384620
             1212) ALP>=1.595 48   2.8125000 0.06250000 *
             1213) ALP< 1.595 17   4.1176470 0.41176470 *
            607) TARS>=147.5 49  11.3877600 0.36734690
             1214) FEVCVF_>=87.905 13   0.9230769 0.07692308 *
             1215) FEVCVF_< 87.905 36   8.9722220 0.47222220
              2430) IMC< 29.74 22   4.7727270 0.31818180
               4860) EDAD< 77.5 13   1.6923080 0.15384620 *
               4861) EDAD>=77.5 9   2.2222220 0.55555560 *
              2431) IMC>=29.74 14   2.8571430 0.71428570 *
      19) ICHAREVP_>=0.5 351  47.7435900 0.16239320
        38) FCA>=67.5 332  41.0602400 0.14457830
          76) IMC< 39.62 318  36.4528300 0.13207550
           152) FEV1P< 45.5 163  11.1165600 0.07361963 *
           153) FEV1P>=45.5 155  24.1935500 0.19354840
            306) FCA>=104.5 38   1.8947370 0.05263158 *
            307) FCA< 104.5 117  21.2991500 0.23931620
             614) FCA< 85.5 51   5.2941180 0.11764710 *
             615) FCA>=85.5 66  14.6666700 0.33333330
              1230) FEVCVF_>=96.985 23   2.6086960 0.13043480 *
              1231) FEVCVF_< 96.985 43  10.6046500 0.44186050
               2462) FVCP>=78.5 19   3.6842110 0.26315790 *
               2463) FVCP< 78.5 24   5.8333330 0.58333330
                4926) TURA< 36.75 16   3.9375000 0.43750000 *
                4927) TURA>=36.75 8   0.8750000 0.87500000 *
          77) IMC>=39.62 14   3.4285710 0.42857140 *
        39) FCA< 67.5 19   4.7368420 0.47368420 *
     5) MUERTOS_90DIAS>=0.5 83  20.6024100 0.45783130
      10) FVCP< 66.5 37   7.2972970 0.27027030
        20) DURING< 17.5 30   4.8000000 0.20000000
          40) ICHARNEF_< 0.5 22   1.8181820 0.09090909 *
          41) ICHARNEF_>=0.5 8   2.0000000 0.50000000 *
        21) DURING>=17.5 7   1.7142860 0.57142860 *
      11) FVCP>=66.5 46  10.9565200 0.60869570
        22) CCVSDM_< 0.5 22   5.0909090 0.36363640 *
        23) CCVSDM_>=0.5 24   3.3333330 0.83333330 *
    3) Reing_EXAC>=0.5 983   0.0000000 1.00000000 *
rpart.plot(arbol_1ff)
set.seed(1000)
d_entrenamientof_2 <- sample_frac(d_final, .7)
d_pruebaf_2 <- setdiff(d_final, d_entrenamientof_2)
arbol_2f <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof_2)
arbol_2ff <- rpart(formula = INGRESOS_ ~ ., data = d_entrenamientof_2,control=rpart.control(cp=0.001))
arbol_2f
arbol_2ff
rpart.plot(arbol_2f)
rpart.plot(arbol_2ff)
# INGRESO 100 % en la de entrenamiento - la muestra total de ingresos es el 35%
# 73% NO REINGRESAN_EXAC SE MANTIENEN VIVOS (71%) PREDISPOSICION A MUERTE EN UN 2%
# 27%  REINGRESAN_EXAC PREDISPOSICION A GRAVEDAD SE MANTIENE POR OTROS FACTORES
```

```
arbol_3f <- rpart(formula = SEXO ~ ., data = d_entrenamientof)
arbol_3f
# n= 3625
# node), split, n, deviance, yval
#       * denotes terminal node
# 1) root 3625 388.10650 0.12193100
#   2) HT_>=0.5 2984 234.03750 0.08579088
#     4) EDAD>=64.5 2348 110.26920 0.04940375 *
#     5) EDAD< 64.5 636 109.18240 0.22012580
#      10) ALP>=1.585 490  70.90612 0.17551020 *
#      11) ALP< 1.585 146  34.02740 0.36986300 *
#   3) HT_< 0.5 641 132.02810 0.29017160
#     6) ALP>=1.585 469  83.66738 0.23240940 *
#     7) ALP< 1.585 172  42.52907 0.44767440 *
arbol_3ff <- rpart(formula = SEXO ~ ., data = d_entrenamientof_2,control=rpart.control(cp=0.001))
arbol_3ff
# n= 3625
# node), split, n, deviance, yval
#       * denotes terminal node
  1) root 3625 412.0204000 0.13075860
    2) HT_>=0.5 2995 257.0698000 0.09482471
      4) EDAD>=60.5 2555 155.2149000 0.06497065
        8) ALP>=1.545 2344 118.3340000 0.05332765
         16) EDAD>=69.5 1792  67.2656200 0.03906250
           32) ALP>=1.585 1568  50.2755100 0.03316327
             64) ESPIROMETRIA_PA_>=0.5 1073  23.4631900 0.02236719 *
             65) ESPIROMETRIA_PA_< 0.5 495  26.4161600 0.05656566
              130) TARS< 140.5 339  11.5752200 0.03539823
                260) EDAD< 89.5 319   8.7460820 0.02821317 *
                261) EDAD>=89.5 20   2.5500000 0.15000000
                  522) TARD>=61 11   0.0000000 0.00000000 *
                  523) TARD< 61 9   2.0000000 0.33333330 *
              131) TARS>=140.5 156  14.3589700 0.10256410
                262) PPKG< 63.5 33   0.0000000 0.00000000 *
                263) PPKG>=63.5 123  13.9187000 0.13008130
                  526) ICHAREVP_< 0.5 97   8.1649480 0.09278351
                    1052) TARS>=145.5 82   4.6951220 0.06097561 *
                    1053) TARS< 145.5 15   2.9333330 0.26666670 *
                  527) ICHAREVP_>=0.5 26   5.1153850 0.26923080
                    1054) FVCP>=55 14   0.9285714 0.07142857 *
                    1055) FVCP< 55 12   3.0000000 0.50000000 *
           33) ALP< 1.585 224  16.5535700 0.08035714
             66) FVCP>=34.5 217  13.9631300 0.06912442
              132) FCA< 105.5 172   6.7151160 0.04069767
                264) FEV1P< 55.5 128   1.9687500 0.01562500 *
                265) FEV1P>=55.5 44   4.4318180 0.11363640
                  530) TARD< 78 30   0.0000000 0.00000000 *
                  531) TARD>=78 14   3.2142860 0.35714290 *
              133) FCA>=105.5 45   6.5777780 0.17777780
                266) FCA>=109 36   2.7500000 0.08333333
```

```
          532) EDAD>=72.5 29   0.0000000 0.00000000 *
          533) EDAD< 72.5 7   1.7142860 0.42857140 *
        267) FCA< 109 9   2.2222220 0.55555560 *
      67) FVCP< 34.5 7   1.7142860 0.42857140 *
   17) EDAD< 69.5 552  49.5199300 0.09963768
     34) INGRESOS_>=0.5 204   6.7598040 0.03431373 *
     35) INGRESOS_< 0.5 348  41.3793100 0.13793100
       70) ICHAR_DM_>=0.5 81   2.8888890 0.03703704 *
       71) ICHAR_DM_< 0.5 267  37.4157300 0.16853930
        142) FCA>=92.5 154  15.8961000 0.11688310
          284) FRE< 19.5 32   0.0000000 0.00000000 *
          285) FRE>=19.5 122  15.3442600 0.14754100
            570) FRE>=22.5 84   7.2380950 0.09523810
              1140) FVCP< 54 30   0.0000000 0.00000000 *
              1141) FVCP>=54 54   6.8148150 0.14814810
                2282) IMC< 31.74 47   4.4680850 0.10638300
                  4564) ALP< 1.685 35   0.9714286 0.02857143 *
                  4565) ALP>=1.685 12   2.6666670 0.33333330 *
                2283) IMC>=31.74 7   1.7142860 0.42857140 *
            571) FRE< 22.5 38   7.3684210 0.26315790
              1142) DURING< 5.5 14   0.0000000 0.00000000 *
              1143) DURING>=5.5 24   5.8333330 0.41666670
                2286) FEV1P< 36.5 11   1.6363640 0.18181820 *
                2287) FEV1P>=36.5 13   3.0769230 0.61538460 *
        143) FCA< 92.5 113  20.5486700 0.23893810
          286) FVCP< 85.5 92  13.8587000 0.18478260
            572) FCA< 79.5 36   0.9722222 0.02777778 *
            573) FCA>=79.5 56  11.4285700 0.28571430
              1146) IMC>=33.63 9   0.0000000 0.00000000 *
              1147) IMC< 33.63 47  10.5531900 0.34042550
                2294) FEV1P< 49 34   6.1176470 0.23529410
                  4588) FRE>=21 19   1.7894740 0.10526320 *
                  4589) FRE< 21 15   3.6000000 0.40000000 *
                2295) FEV1P>=49 13   3.0769230 0.61538460 *
          287) FVCP>=85.5 21   5.2380950 0.47619050
            574) ALP>=1.64 13   2.7692310 0.30769230 *
            575) ALP< 1.64 8   1.5000000 0.75000000 *
 9) ALP< 1.545 211  33.0331800 0.19431280
  18) FRE< 42 203  29.6157600 0.17733990
    36) EDAD>=77.5 73   4.6575340 0.06849315
      72) TARD>=59.5 63   1.9365080 0.03174603 *
      73) TARD< 59.5 10   2.1000000 0.30000000 *
    37) EDAD< 77.5 130  23.6076900 0.23846150
      74) FEV1P< 73 120  19.2000000 0.20000000
        148) TARD>=53.5 112  16.4285700 0.17857140
          296) FVCP< 55.5 40   1.9000000 0.05000000 *
          297) FVCP>=55.5 72  13.5000000 0.25000000
            594) PPKG>=53.5 64  10.3593800 0.20312500
              1188) TARS< 165.5 57   7.5789470 0.15789470
                2376) DURING< 12.5 41   3.6097560 0.09756098
                  4752) FCA>=82 33   0.9696970 0.03030303 *
                  4753) FCA< 82 8   1.8750000 0.37500000 *
```

```
          2377) DURING>=12.5 16   3.4375000 0.31250000 *
         1189) TARS>=165.5 7   1.7142860 0.57142860 *
        595) PPKG< 53.5 8   1.8750000 0.62500000 *
      149) TARD< 53.5 8   2.0000000 0.50000000 *
     75) FEV1P>=73 10   2.1000000 0.70000000 *
   19) FRE>=42 8   1.8750000 0.62500000 *
 5) EDAD< 60.5 440  86.3545500 0.26818180
 10) ALP>=1.595 326  52.0398800 0.19938650
   20) ICHAR_DM_>=0.5 59   2.8474580 0.05084746 *
   21) ICHAR_DM_< 0.5 267  47.6030000 0.23220970
     42) TARD>=92.5 34   0.9705882 0.02941176 *
     43) TARD< 92.5 233  45.0300400 0.26180260
       86) FEV1P< 55.5 183  31.2568300 0.21857920
       172) FEVCVF_>=54.775 102  12.7941200 0.14705880
         344) TURA< 36.55 49   0.9795918 0.02040816 *
         345) TURA>=36.55 53  10.3018900 0.26415090
           690) FEVCVF_< 72.785 35   4.2857140 0.14285710
           1380) FCA< 106 21   0.9523810 0.04761905 *
           1381) FCA>=106 14   2.8571430 0.28571430 *
           691) FEVCVF_>=72.785 18   4.5000000 0.50000000 *
       173) FEVCVF_< 54.775 81  17.2839500 0.30864200
         346) TARD>=69.5 55   8.8000000 0.20000000
           692) FCA>=89 43   4.4186050 0.11627910
           1384) FEVCVF_< 49.685 33   0.9696970 0.03030303 *
           1385) FEVCVF_>=49.685 10   2.4000000 0.40000000 *
           693) FCA< 89 12   3.0000000 0.50000000 *
         347) TARD< 69.5 26   6.4615380 0.53846150
           694) ALP>=1.655 16   3.7500000 0.37500000 *
           695) ALP< 1.655 10   1.6000000 0.80000000 *
       87) FEV1P>=55.5 50  12.1800000 0.42000000
       174) FCA< 110.5 40   8.7750000 0.32500000
         348) TURA< 36.65 23   2.6086960 0.13043480 *
         349) TURA>=36.65 17   4.1176470 0.58823530 *
       175) FCA>=110.5 10   1.6000000 0.80000000 *
   11) ALP< 1.595 114  28.3596500 0.46491230
     22) ESPIROMETRIA_PA_< 0.5 24   3.3333330 0.16666670
       44) TURA< 37.35 17   0.9411765 0.05882353 *
       45) TURA>=37.35 7   1.7142860 0.42857140 *
     23) ESPIROMETRIA_PA_>=0.5 90  22.3222200 0.54444440
       46) FEV1P< 32.5 20   3.7500000 0.25000000
         92) PPKG>=61 13   0.9230769 0.07692308 *
         93) PPKG< 61 7   1.7142860 0.57142860 *
       47) FEV1P>=32.5 70  16.3428600 0.62857140
         94) TURA>=36.65 29   7.1724140 0.44827590
           188) FCA< 100.5 18   3.6111110 0.27777780 *
           189) FCA>=100.5 11   2.1818180 0.72727270 *
         95) TURA< 36.65 41   7.5609760 0.75609760
           190) FCA>=100.5 13   3.2307690 0.46153850 *
           191) FCA< 100.5 28   2.6785710 0.89285710 *
3) HT_< 0.5 630 132.6984000 0.30158730
  6) EDAD< 81.5 395  69.4936700 0.22784810
  12) FRE< 34.5 366  59.6311500 0.20491800
```

```
  24) FEV1P< 49.5 237  29.8312200 0.14767930
    48) FCA>=79.5 193  18.7150300 0.10880830
      96) ICHARICC_< 0.5 149  11.0335600 0.08053691
       192) FEVCVF_>=47.53 117   4.7863250 0.04273504 *
       193) FEVCVF_< 47.53 32   5.4687500 0.21875000
         386) IMC< 21.99 12   0.0000000 0.00000000 *
         387) IMC>=21.99 20   4.5500000 0.35000000
           774) FCA< 93.5 7   0.0000000 0.00000000 *
           775) FCA>=93.5 13   3.2307690 0.53846150 *
      97) ICHARICC_>=0.5 44   7.1590910 0.20454550
       194) TARS< 137 23   0.9565217 0.04347826 *
       195) TARS>=137 21   4.9523810 0.38095240
         390) IMC>=23.93 14   2.3571430 0.21428570 *
         391) IMC< 23.93 7   1.4285710 0.71428570 *
    49) FCA< 79.5 44   9.5454550 0.31818180
      98) TURA>=36.85 11   0.0000000 0.00000000 *
      99) TURA< 36.85 33   8.0606060 0.42424240
       198) TARS>=128 20   4.2000000 0.30000000
         396) FRE< 23 9   0.8888889 0.11111110 *
         397) FRE>=23 11   2.7272730 0.45454550 *
       199) TARS< 128 13   3.0769230 0.61538460 *
  25) FEV1P>=49.5 129  27.5969000 0.31007750
    50) ALP>=1.495 120  23.9250000 0.27500000
     100) TURA< 36.15 38   3.5789470 0.10526320
       200) PPKG>=62 31   0.9677419 0.03225806 *
       201) PPKG< 62 7   1.7142860 0.42857140 *
     101) TURA>=36.15 82  18.7439000 0.35365850
       202) IMC< 36.43 75  16.3200000 0.32000000
         404) FRE>=27.5 23   2.6086960 0.13043480
           808) FCA>=89 13   0.0000000 0.00000000 *
           809) FCA< 89 10   2.1000000 0.30000000 *
         405) FRE< 27.5 52  12.5192300 0.40384620
           810) FRE< 19.5 17   2.4705880 0.17647060 *
           811) FRE>=19.5 35   8.7428570 0.51428570
            1622) ALP< 1.705 28   6.8571430 0.42857140
              3244) TURA>=37.4 8   0.8750000 0.12500000 *
              3245) TURA< 37.4 20   4.9500000 0.55000000
                6490) FCA< 97.5 11   2.5454550 0.36363640 *
                6491) FCA>=97.5 9   1.5555560 0.77777780 *
            1623) ALP>=1.705 7   0.8571429 0.85714290 *
       203) IMC>=36.43 7   1.4285710 0.71428570 *
    51) ALP< 1.495 9   1.5555560 0.77777780 *
 13) FRE>=34.5 29   7.2413790 0.51724140
   26) PPKG< 85.5 21   5.1428570 0.42857140
     52) IMC>=29.375 8   0.8750000 0.12500000 *
     53) IMC< 29.375 13   3.0769230 0.61538460 *
   27) PPKG>=85.5 8   1.5000000 0.75000000 *
7) EDAD>=81.5 235  57.4468100 0.42553190
 14) FCA< 109.5 198  46.3434300 0.37373740
   28) FRE>=22.5 108  20.7407400 0.25925930
     56) FEV1P>=33.5 82  12.2561000 0.18292680
      112) EDAD< 90.5 69   7.8260870 0.13043480
```

```
      224) TARD< 84.5 55  3.7090910 0.07272727 *
      225) TARD>=84.5 14  3.2142860 0.35714290 *
    113) EDAD>=90.5 13  3.2307690 0.46153850 *
   57) FEV1P< 33.5 26  6.5000000 0.50000000
    114) DURING>=7.5 15  3.3333330 0.33333330 *
    115) DURING< 7.5 11  2.1818180 0.72727270 *
  29) FRE< 22.5 90  22.4888900 0.51111110
   58) IMC< 31.005 67  16.4477600 0.43283580
    116) TARS< 133.5 32  6.0000000 0.25000000
     232) FEV1P>=34.5 25  3.3600000 0.16000000
      464) PPKG>=64 17  0.9411765 0.05882353 *
      465) PPKG< 64 8  1.8750000 0.37500000 *
     233) FEV1P< 34.5 7  1.7142860 0.57142860 *
    117) TARS>=133.5 35  8.4000000 0.60000000
     234) EXACER_90DIAS>=0.5 8  1.5000000 0.25000000 *
     235) EXACER_90DIAS< 0.5 27  5.6296300 0.70370370
      470) TARS>=156.5 12  3.0000000 0.50000000 *
      471) TARS< 156.5 15  1.7333330 0.86666670 *
   59) IMC>=31.005 23  4.4347830 0.73913040
    118) DURING< 7.5 9  2.2222220 0.44444440 *
    119) DURING>=7.5 14  0.9285714 0.92857140 *
 15) FCA>=109.5 37  7.7297300 0.70270270
  30) ICHAR_DM_< 0.5 24  5.8333330 0.58333330
   60) PPKG< 84.5 17  4.1176470 0.41176470 *
   61) PPKG>=84.5 7  0.0000000 1.00000000 *
  31) ICHAR_DM_>=0.5 13  0.9230769 0.92307690 *
rpart.plot(arbol_3f)
rpart.plot(arbol_3ff)
# Mujeres 12% - NO APPLICABLE POCO COHERENTE
# 82% FUMADORAS MAYORES DE 65 AÑOS (65%) PREDISPOSICION A GRAVEDAD EN UN 18%
# 18% NO FUMADORAS PREDISPOSICION A GRAVEDAD EN UN 5%
arbol_4f <- rpart(formula = HT_ ~ ., data = d_entrenamientof)
arbol_4ff <- rpart(formula = HT_ ~ ., data = d_entrenamientof_2,control=rpart.control(cp=0.001))
arbol_4f
arbol_4ff
rpart.plot(arbol_4f)
rpart.plot(arbol_4ff)
# Fumadores 82% , 1=MUJERES ; 0=HOMBRES
# MUJERES 12% FUMADORAS MAYORES DE 73 AÑOS PREDISPOSICION A GRAVEDAD EN UN 6%
# HOMBRES 88% FUMADORES SIN ESPIROMETRIA (27%) PREDISPOSICION A GRAVEDAD EN UN 60%

######################################## CAPITULO IV ########################################

# SINTAXIS DE DATOS REALES – SVM CLASIFICACIÓN Y KERNELS
library("e1071")
library("fda")
library("splines")
library("Matrix")
datos<-read.table (file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_220616.txt",
header=T)
```

```
#Dataset usado antes de aplicar imputación por ser la idea inicial, pero en el desarrollo no la mejor para este caso.
datos
datos$SEXO
datos$EDAD
summary (datos$EDAD) # Min.  1st Qu.  Median  Mean   3rd Qu.  Max.   31.00   68.00   75.00   73.39   80.00   99.00
table (datos$SEXO) # HOMBRES   MUJERES    4526     652
x<-datos$SEXO
y<-datos$EDAD
plot(x,y,main="SEXO & EDAD")
datos$EDAD
datos$DURING
summary (datos$DURING) # Min. 1st Qu.  Median  Mean 3rd Qu.  Max.   1.00   6.00   8.00   9.955  12.00 130.00
# SINTAXIS CASO 1.- FUMADORES Y NO FUMADORES
table(datos$HT_)  # NO_FUMADOR     SI_FUMADOR       906      4272
x1<-datos$EDAD
x2<-datos$DURING
plot(x1,x2, main="EDAD & DURING")
x<-(matriz<-c(x1,x2))
y<-datos$HT_
clases<-c(datos$HT_)
plot(x1,x2,type="n", main="EDAD & DURING by HT")
points(x1[clases=="NO_FUMADOR"],x2[clases=="NO_FUMADOR"],col=2)
points(x1[clases=="SI_FUMADOR"],x2[clases=="SI_FUMADOR"],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV # [1] 1833
summary(model)
#Call: svm(formula = as.factor(clases) ~ coord, type = "C-classification",     kernel = "linear", cross = 2, scale = F)
#Parameters:   SVM-Type:  C-classification  SVM-Kernel:  linear       cost:  1       gamma:  0.5
#Number of Support Vectors:   1833 ( 927 906 )  Number of Classes:   2   Levels:   1 2    2-fold cross-validation on
training #data: Total Accuracy: 82.5029  Single Accuracies:  81.49865 83.50715
predict(model,coord)
predichos = as.numeric(predict(model,coord))
tab<-table(predichos,clases)
tab
classAgreement(tab)  #$diag  [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV  # [1] 1828
model$cost  # [1] 10
predict(model,coord)
predichos = as.numeric(predict(model,coord))
tab<-table(predichos,clases)
tab
```

```
classAgreement(tab)  #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV  # [1] 1827
model$cost  # [1] 0.1
predict(model,coord)
predichos = as.numeric(predict(model,coord))
tab<-table(predichos,clases)
classAgreement(tab)
```

**# SINTAXIS AJUSTAR CONSTANTE DE REGULARIZACIÓN**

```
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV # [1] 1833
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#               clases
# predichos    1    2
#          2  906 4272
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1833
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=0.5)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1833
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=10)
points(model$SV,col=2,pch=20)
```

```
model$tot.nSV #[1] 1833
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F, coef0=0.1,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=10,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1828
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=0.5,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1828
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
```

```
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F, coef0=0.5,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab) #$diag [1] 0.174971
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0 =0.3,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=0.8,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=10,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1827
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=100,cost=100)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1826
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
```

```
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=50,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1828
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2396
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#                clases
# predichos    1    2
#           1   52    7
#           2  854 4265
classAgreement(tab) # $diag [1] 0.8337196  $kappa [1] 0.08826507
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2327
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#           1  145   30
#           2  761 4242
classAgreement(tab) #$diag [1] 0.8472383  $kappa [1] 0.2243274
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2331
summary(model)
pred<-predict(model,coord)
```

```
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1458
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1473
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#        1  292 2469
#        2  614 1803
classAgreement(tab) #$diag [1] 0.4045964  $kappa [1] -0.1415094
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1464
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#        1  210  421
#        2  696 3851
classAgreement(tab) #$diag [1] 0.7842796  $kappa [1] 0.1513361
#NOTA.- Kernel=polinomial el mejor con coste=0.1, clasificación bastante buena y con un % alto aunque también se podría
#seleccionar el de coste=10 con un % medio y con un grado de concordancia (Kappa) negativo (relación inversa).
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
points(model$SV,col=2,pch=20)
```

```
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
```

```
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se #ha conseguido mejorarlo.
# SINTAXIS MEJORANDO el Ajuste con el CAMBIO de parámetros
library("e1071")
library("fda")
library("splines")
library("Matrix")
datos<-read.table (file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_220616.txt", header=T)
table(datos$HT_)
x1<-datos$EDAD
x2<-datos$DURING
plot(x1,x2, main="EDAD & DURING")
x<-(matriz<-c(x1,x2))
y<-datos$HT_
clases<-c(datos$HT_)
plot(x1,x2,type="n", main="EDAD & DURING by HT")
points(x1[clases=="NO_FUMADOR"],x2[clases=="NO_FUMADOR"],col=2)
points(x1[clases=="SI_FUMADOR"],x2[clases=="SI_FUMADOR"],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,sigma=0.05)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2396
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#       1   52    7
#       2  854 4265
classAgreement(tab)  #$diag  [1] 0.8337196  $kappa  [1] 0.08826507
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1,sigma=0.05)
```

```
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2331
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10, sigma=0.05)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2327
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#       1  145   30
#       2  761 4242
classAgreement(tab)  #$diag  [1] 0.8472383  $kappa  [1] 0.2243274
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,sigma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2396
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#       1   52    7
#       2  854 4265
classAgreement(tab)  #$diag  [1] 0.8337196  $kappa  [1] 0.08826507
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1,sigma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2331
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
```

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10, sigma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2327
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#        1  145   30
 #       2  761 4242
classAgreement(tab) #$diag  [1] 0.8472383  $kappa  [1] 0.2243274
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,sigma=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2396
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#                clases
# predichos    1    2
#        1   52    7
#        2  854 4265
classAgreement(tab) $diag  [1] 0.8337196  $kappa  [1] 0.08826507
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1,sigma=1)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 2331
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10, sigma=1)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 2327
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
```

```
# predichos    1    2
#        1  145   30
#        2  761 4242
classAgreement(tab)  #$diag  [1] 0.8472383  $kappa  [1] 0.2243274
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,sigma=10)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 2396
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#       clases
# predichos    1    2
#        1   52    7
#        2  854 4265
classAgreement(tab)  #$diag  [1] 0.8337196  $kappa  [1] 0.08826507
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1,sigma=10)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 2331
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10, sigma=10)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 2327
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#       clases
# predichos    1    2
#        1  145   30
#        2  761 4242
classAgreement(tab)  #$diag  [1] 0.8472383  $kappa  [1] 0.2243274
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10, sigma=10,  lambda=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2327
summary(model)
```

```
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#        1  145   30
#        2  761 4242
classAgreement(tab)  #$diag  [1] 0.8472383  $kappa  [1] 0.2243274
#NOTA.- A la vista de los resultados parece que NO se ha podido mejorar el modelo RADIAL cuando se le introduce #los parámetros sigma y lambda por cada coste C.
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,degree=0)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=10,degree=0)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=0.1,degree=0)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,degree=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1828
summary(model)
```

```
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=10,degree=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1828
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=0.1,degree=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1826
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,degree=2)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1449
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=10,degree=2)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1403
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
```

```
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="polynomial",cross=2,scale=F,cost=0.1,degree=2)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1570
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- A la vista de los resultados parece que NO se ha podido mejorar el modelo polynomial al disminuir el grado #del polinomio por lo que el mejor modelo sería el de grado 3 y coste 0.1 como se puede apreciar.
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,gamma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="sigmoid",cross=2,scale=F,cost=10,gamma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,gamma=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,gamma=1)
points(model$SV,col=2,pch=20)
```

```
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,gamma=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,gamma=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,gamma=0.001)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="sigmoid",cross=2,scale=F,cost=10,gamma=0.001)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
```

```
classAgreement(tab)
plot(coord, main="EDAD & DURING by HT")
model<-svm(as.factor(clases)~coord,type="C-
classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,gamma=0.001)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1812
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
```

**# SINTAXIS CASO 2.- EXITUS**

```
datos<-read.table
(file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_220715.txt",header=T)
table(datos$EXITUS) #MUERTO (True)  VIVO (False)    259        4919
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$EXITUS)
plot(x1,x2,type="n", main="EDAD & DURING by EXITUS")
points(x1[clases=="VIVO"],x2[clases=="VIVO"],col=2)
points(x1[clases=="MUERTO"],x2[clases=="MUERTO"],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 526
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 529
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 528
```

```
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=lineal ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 526
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 529
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 528
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1290
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#       clases
```

```
# predichos    1    2
#        1   15    0
#        2  244 4919
classAgreement(tab) #$diag  [1] 0.9528776   $kappa  [1] 0.1045853
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1275
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#         clases
# predichos    1    2
#        1   70   11
#        2  189 4908
classAgreement(tab)  #$diag  [1] 0.961375   $kappa  [1] 0.3974032
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1051
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 489
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 470
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
```

```
tab
#        clases
# predichos    1    2
#        1    0   14
#        2  259 4905
classAgreement(tab)  #$diag  [1] 0.9472769   $kappa  [1] -0.00515664
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 493
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    1    2
#        1   10   29
#        2  249 4890
classAgreement(tab)  #$diag  [1] 0.9463113   $kappa  [1] 0.05473844
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación bastante buena y con un % alto aunque también se
#podría seleccionar el de coste=0.1 con un % bueno, muy similar al de coste 10.
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
```

```
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by EXITUS")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 518
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
# SINTAXIS CASO 3.- PRUEBA DE ESPIROMETRÍA REALIZADA Y NO REALIZADA
datos<-read.table
(file="C:/Users/Usuario/Desktop/Trabajo_UC3M_JUNIO_v2/INICIAL_UC3M_selección_220715.txt",header=T)
table(datos$ESPIROMETRIA_PA_)   # 0(NO)    1(SI)    1644      3534
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ESPIROMETRIA_PA_)
plot(x1,x2,type="n", main="EDAD & DURING by ESPIROMETRIA")
points(x1[clases==0],x2[clases==0],col=2)
```

```
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3320
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3326
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3323
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3320
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3326
summary(model)
```

```
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3323
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se
ha #conseguido mejorarlo.
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3598
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    0    1
#        1  261   77
#        2 1383 3457
classAgreement(tab)  #$diag  [1] 0.7180379   $kappa  [1] 0.1739147
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV  #[1] 3470
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    0    1
#        1  448  168
#        2 1196 3366
classAgreement(tab)  #$diag  [1] 0.7365778  $kappa  [1] 0.2701369
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3645
```

```
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2195
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    0    1
#           1    7   15
#           2 1637 3519
classAgreement(tab)  #$diag  [1] 0.6809579   $kappa  [1] 1.823663e-05
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2134
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    0    1
#           1    8   15
#           2 1636 3519
classAgreement(tab)  #$diag  [1] 0.681151  $kappa  [1] 0.0008443097
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2177
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
#        clases
# predichos    0    1
#           1  906 2509
```

```
#         2  738 1025
classAgreement(tab)  #$diag  [1] 0.3729239   $kappa  [1] -0.1233371
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación bastante buena y con un % alto aunque también se podría seleccionar el de coste=1 con un % alto casi similar al otro y el de coste=0.1 tiene un % bajo aunque tenga menos SVMs.
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
```

```
predichos = as.numeric(pred)
tab<-table(predichos,clases)
classAgreement(tab)
plot(coord, main="EDAD & DURING by ESPIROMETRIA")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3288
summary(model)
pred<-predict(model,coord)
predichos = as.numeric(pred)
tab<-table(predichos,clases)
tab
classAgreement(tab)
```

*#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se #ha conseguido mejorarlo.*

**# SINTAXIS CASO 4.- PATOLOGÍAS DETECTADAS EN EL PARTICIPANTE**

**#4.1.- INSUFICIENCIA CARDÍACA CONGESTIVA**

```
table(datos$ICHARICC_) # 0(NO)    1(SI)     4058        1120
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHARICC_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHARICC_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHARICC_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2317
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2380
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2276
```

*#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2317
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2380
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2276
```

*#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se ha #conseguido mejorarlo.*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 2846
tab
#         clases
# predichos    0    1
```

```
#        1 4054 1072
#        2    4   48
classAgreement(tab) #$diag  [1] 0.7921978   $kappa  [1] 0.06394478
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2746
tab
#        clases
# predichos    0    1
#        1 4001  936
#        2   57  184
classAgreement(tab) #$diag  [1] 0.8082271  $kappa  [1] 0.2098627
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2775
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 1740
tab
#        clases
# predichos    0    1
#        1 3768 1092
#        2  290   28
classAgreement(tab)  #$diag  [1] 0.7331016  $kappa  [1] -0.06272265
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1774
tab
#        clases
# predichos    0    1
#        1 3759  999
#        2  299  121
classAgreement(tab)  #$diag  [1] 0.7493241  $kappa  [1] 0.04439952
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1766
tab
#        clases
# predichos    0    1
#        1 3325  867
#        2  733  253
classAgreement(tab)  #$diag  [1] 0.6910004   $kappa  [1] 0.04731109
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación bastante buena y con un % alto aunque también se podría
#seleccionar el de coste=1 con un % alto casi similar al otro y el de coste=0.1 tiene un % bajo aunque tenga un poco menos
SVMs.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 2240
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2240
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2240
```

```
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2240
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2240
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2240
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
```

**#4.2.- COMORBILIDAD CARDIOVASCULAR**

```
table(datos$CCVSDM_)   0(NO)    1(SI)     2961          2217
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$CCVSDM_)
plot(x1,x2,type="n", main="EDAD & DURING by CCVSDM_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by CCVSDM_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 4505
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 4483
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 4478
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 4505
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 4483
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 4478
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se
ha #conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 4478
tab
#        clases
# predichos    0    1
#        1 2465 1302
#        2  496  915
classAgreement(tab)  #$diag  [1] 0.6527617   $kappa  [1] 0.2569447
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 4271
tab
#        clases
```

```
# predichos    0    1
#        1 2369 1075
#        2  592 1142
classAgreement(tab)  #$diag  [1] 0.678061  $kappa  [1] 0.3240475
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV  #[1] 4636
tab
#       clases
# predichos    0    1
#        1 2905 2101
#        2   56  116
classAgreement(tab)  #$diag  [1] 0.5834299  $kappa  [1] 0.03778974
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 2546
tab
#        clases
# predichos    0    1
#        1 2898 2142
#        2   63   75
classAgreement(tab)  #$diag  [1] 0.5741599  $kappa  [1] 0.0142293
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2499
tab
#        clases
# predichos    0    1
#        1 2178 1381
#       2  783  836
classAgreement(tab)  #$diag  [1] 0.582078  $kappa  [1] 0.1165997
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2445
tab
#       clases
# predichos    0    1
#        1 2053 1404
#        2  908  813
classAgreement(tab)  #$diag  [1] 0.5534956  $kappa  [1] 0.06179553
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación buena y con un % medio.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 4434
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 4434
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 4434
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 4434
```

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 4434
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 4434
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se #ha conseguido mejorarlo.
```

**#4.3.- DIABETES MELLITUS**

```
table(datos$ICHAR_DM_)  # 0(NO)    1(SI)     3844      1334
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHAR_DM_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHAR_DM_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHAR_DM_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2694
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2696
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2693
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2694
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2696
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2693
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se ha #conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 3165
tab
#        clases
# predichos    0    1
#        1 3837 1268
#        2    7   66
classAgreement(tab)  #$diag  [1] 0.7537659   $kappa  [1] 0.06892595
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 3074
tab
#        clases
# predichos    0    1
#        1 3782 1108
#        2   62  226
```

```
classAgreement(tab)  #$diag  [1] 0.774044  $kappa  [1] 0.2060293
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 3138
NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 1966
tab
#        clases
# predichos    0    1
#        1 3654 1257
#        2  190   77
classAgreement(tab)  #$diag  [1] 0.7205485   $kappa  [1] 0.01122463
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV  #[1] 2030
tab
#        clases
# predichos    0    1
#        1 2532  943
#        2 1312  391
classAgreement(tab) #$diag  [1] 0.5645037  $kappa  [1] -0.0442147
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1967
tab
#        clases
# predichos    0    1
#        1 3842 1330
#        2    2    4
classAgreement(tab)  #$diag  [1] 0.7427578   $kappa  [1] 0.0036715
#NOTA.- Kernel=polinomial el mejor con coste=0.1, clasificación bastante buena y con un % alto aunque también
se #podría seleccionar el de coste=1 con un % alto casi similar al otro.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 2668
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2668
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2668
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2668
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2668
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2668
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
```

**#4.4.- ENFERMEDAD VASCULAR ABARCANDO LAS DE CEREBRO VASCULAR Y LAS DE VASCULAR PERIFÉRICA**

```
table(datos$EV_)   #  0(NO)    1(SI)     3588    1590
```

```
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$EV_)
plot(x1,x2,type="n", main="EDAD & DURING by EV_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by EV_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 3217
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV  #[1] 3210
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 3210
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 3217
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 3210
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 3210
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se
ha conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 3592
tab
#        clases
# predichos    0    1
#        1 3551 1437
#        2   37  153
classAgreement(tab)  #$diag  [1] 0.7153341   $kappa  [1] 0.1138174
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 3458
tab
#        clases
# predichos    0    1
#        1 3427 1188
#        2  161  402
classAgreement(tab)  #$diag  [1] 0.7394747   $kappa  [1] 0.2535583
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 3564
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 2216
tab
#        clases
```

```
# predichos   0   1
#       1 1251  629
#       2 2337  961
classAgreement(tab)  #$diag  [1] 0.427192   $kappa  [1] -0.03612931
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2245
tab
#        clases
# predichos   0   1
#       1 3446 1487
#       2  142  103
classAgreement(tab)  #$diag  [1] 0.6853998  $kappa  [1] 0.03296842
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2220
tab
#        clases
# predichos   0   1
#       1  225   58
#       2 3363 1532
classAgreement(tab)  #$diag  [1] 0.3393202   $kappa  [1] 0.01661512
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación bastante buena y con un % medio aunque también
se #podría seleccionar el de coste=1 aunque tenga un % bajo pero tiene menos SVMs.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 3180
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 3180
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 3180
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 3180
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 3180
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 3180
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
```

**#4.5.- INFARTO DE MIOCARDIO**

```
table(datos$ICHARIM_)  # 0(NO)    1(SI)    4505        673
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHARIM_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHARIM_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHARIM_")
```

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1352
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1355
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1354
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 1352
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 1355
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 1354
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se
ha #conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 1970
tab
#       clases
# predichos   0   1
#         1 4505  662
#         2   0  11
classAgreement(tab)  #$diag  [1] 0.8721514   $kappa  [1] 0.0281008
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1941
tab
#       clases
# predichos   0   1
#         1 4491  602
#         2  14  71
classAgreement(tab)  #$diag  [1] 0.8810351   $kappa  [1] 0.162935
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1842
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 1199
tab
#       clases
# predichos   0   1
#       1 4472  669
#       2  33   4
classAgreement(tab)  #$diag  [1] 0.8644264   $kappa  [1] -0.002310167
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1202
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1189
```

```
tab
#        clases
# predichos    0    1
#        1 4343  669
#        2  162    4
classAgreement(tab)  #$diag  [1] 0.8395133   $kappa  [1] -0.04416796
#NOTA.- Kernel=polinomial el mejor con coste=1, clasificación bastante buena y con un % alto aunque también se
#podría seleccionar el de coste=0.1 con un % alto casi similar al otro.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 1346
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1346
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1346
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 1346
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 1346
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 1346
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se ha conseguido mejorarlo.
#4.6.- NEFROPATÍA
table(datos$ICHARNEF_)  # 0(NO)    1(SI)     4691      487
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHARNEF_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHARNEF_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHARNEF_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 985
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 984
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 980
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 985
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 984
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 980
```

*#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se ha conseguido mejorarlo.*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 1690
tab
#        clases
# predichos    0    1
#        1 4691  478
#        2    0    9
classAgreement(tab)  #$diag  [1] 0.9076864   $kappa  [1] 0.03298975
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1684
tab
#        clases
# predichos    0    1
#        1 4683  436
#        2    8   51
classAgreement(tab)  #$diag  [1] 0.9142526   $kappa  [1] 0.1699412
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1461
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 885
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 894
tab
#        clases
# predichos    0    1
#        1 4686  487
#        2    5    0
classAgreement(tab)  #$diag  [1] 0.9049826   $kappa  [1] -0.001915282
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 900
tab
#        clases
# predichos    0    1
#          1 4690  487
#          2    1    0
classAgreement(tab)  #$diag  [1] 0.9057551   $kappa  [1] -0.0003856067
#NOTA.- Kernel=polinomial el mejor con coste=0.1, clasificación buena y con un % alto aunque también se podría seleccionar el de coste=10 con un % alto casi similar al otro.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 974
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 974
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 974
```

```
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 974
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 974
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 974
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se #ha conseguido mejorarlo.
```

**#4.7.- TUMOR SÓLIDO**

```
table(datos$ICHAR_TS_)   # 0(NO)    1(SI)     4506      672
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHAR_TS_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHAR_TS_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHAR_TS_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 1358
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1355
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1354
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 1358
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 1355
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 1354
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se ha #conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 1998
tab
#        clases
# predichos    0    1
#          1 4506  661
#          2    0   11
classAgreement(tab)  #$diag  [1] 0.8723445   $kappa  [1] 0.02814819
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1965
tab
#        clases
```

```
# predichos    0    1
#         1 4498  607
#         2    8   65
classAgreement(tab)  #$diag  [1] 0.8812283  $kappa  [1] 0.1529534
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1819
#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 1182
tab
#        clases
# predichos    0    1
#       1 4506  671
#       2    0    1
classAgreement(tab)  #$diag  [1] 0.8704133   $kappa  [1] 0.00258709
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1179
tab
#        clases
# predichos    0    1
#       1 4426  670
 #      2   80    2
classAgreement(tab)  #$diag  [1] 0.8551564   $kappa  [1] -0.0235888
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1178
tab
#        clases
# predichos    0    1
#       1 4506  671
#       2    0    1
classAgreement(tab)  #$diag  [1] 0.8704133   $kappa  [1] 0.00258709
#NOTA.- Kernel=polinomial el mejor con coste=0.1, clasificación bastante buena y con un % alto aunque también
se #podría seleccionar el de coste=1 con el mismo % pero con mayor número de SVMs.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 1344
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 1344
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1344
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 1344
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 1344
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 1344
```

*#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se #ha conseguido mejorarlo.*

**#4.8.- EDEMAS MALEOLARES**

```
table(datos$EP_)  # 0(NO)    1(SI)     3828     1350
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$EP_)
plot(x1,x2,type="n", main="EDAD & DURING by EP_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by EP_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 2724
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2719
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2719
```

*#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2724
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2719
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2719
```

*#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se ha #conseguido mejorarlo.*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV #[1] 3153
tab
#        clases
# predichos    0    1
#        1 3794 1226
#        2   34  124
classAgreement(tab)  #$diag  [1] 0.7566628  $kappa  [1] 0.1161697
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 3063
tab
#        clases
# predichos    0    1
#        1 3762 1076
#        2   66  274
classAgreement(tab)  #$diag  [1] 0.7794515   $kappa  [1] 0.2450642
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 3135
```

*#NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto*

```
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV #[1] 1996
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2055
tab
#        clases
# predichos    0    1
#        1 3790 1322
#        2   38   28
classAgreement(tab)  #$diag [1] 0.7373503  $kappa  [1] 0.01562348
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 1929
tab
#        clases
# predichos    0    1
#        1 1911  787
#        2 1917  563
classAgreement(tab)  #$diag  [1] 0.4777907   $kappa  [1] -0.06589442
#NOTA.- Kernel=polinomial el mejor con coste=10, clasificación bastante buena y un alto %.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV #[1] 2700
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV #[1] 2700
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV #[1] 2700
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV #[1] 2700
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV #[1] 2700
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV #[1] 2700
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
# SINTAXIS SVM - KERNELs reactivación procedimiento para comprobar tras BD Completa
table(datos$EV_)
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$EV_)
plot(x1,x2,type="n", main="EDAD & DURING by EV_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by EV_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV
```

```
table(datos$ICHAR_DM_)
x1<-datos$EDAD
x2<-datos$DURING
clases<-c(datos$ICHAR_DM_)
plot(x1,x2,type="n", main="EDAD & DURING by ICHAR_DM_")
points(x1[clases==0],x2[clases==0],col=2)
points(x1[clases==1],x2[clases==1],col=4)
coord<-cbind(x1,x2)
plot(coord, main="EDAD & DURING by ICHAR_DM_")
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F)
points(model$SV,col=2,pch=20)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1)
model$tot.nSV
#NOTA.- Kernel=linear ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,coef0=1)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV
#NOTA.- Kernel=linear ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1) se
#ha conseguido mejorarlo.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F)
model$tot.nSV
tab
classAgreement(tab)
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=10)
model$tot.nSV
tab
classAgreement(tab)
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="radial",cross=2,scale=F,cost=0.1)
model$tot.nSV
NOTA.- Kernel=radial el mejor con coste=10, clasificación buena y con un % alto
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F)
model$tot.nSV
tab
classAgreement(tab)
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=10)
model$tot.nSV
tab
classAgreement(tab)
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="polynomial",cross=2,scale=F,cost=0.1)
model$tot.nSV
tab
```

```
classAgreement(tab)
#NOTA.- Kernel=polinomial el mejor con coste=0.1, clasificación bastante buena y con un % alto aunque también
se #podría seleccionar el de coste=1 con un % alto casi similar al otro.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1)
model$tot.nSV
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor. Se intenta ajustar por la constante de regularización.
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,coef0=1)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=10,coef0=1)
model$tot.nSV
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="sigmoid",cross=2,scale=F,cost=0.1,coef0=1)
model$tot.nSV
#NOTA.- Kernel=sigmoid ninguno de los costes es el mejor pero tampoco con el ajuste de regularización (coef0=1)
se #ha conseguido mejorarlo.
#model<-svm(SEX~., data=d_entrenamientof, kernel="linear", cost=0.1, scale=TRUE)
View(d_entrenamientof)
install.packages("e1071")
library(e1071)
View(d)
str(d)
model<-svm(d$SEX~ ., data =d, kernel="linear", cost=0.1, scale=TRUE)
model
# Call: svm(formula = d$SEX ~ ., data = d, kernel = "linear", cost = 0.1, scale = TRUE)
# Parameters:   SVM-Type:  eps-regression
# SVM-Kernel:  linear       cost:  0.1       gamma:  0.03125      epsilon:  0.1
# Number of Support Vectors:  143
cor(predict(model,d),d$SEXO)  # [1] 1
plot(cor(predict(model,d),d$SEXO))
rpart.plot(arbol_4ff)
model<-svm(SEXO~., data=d_entrenamientof, kernel="linear", cost=0.1, scale=TRUE)
model
# Call: svm(formula = SEXO ~ ., data = d_entrenamientof, kernel = "linear", cost = 0.1, scale = TRUE)
# Parameters:  SVM-Type:  eps-regression
# SVM-Kernel:  linear      cost:  0.1     gamma:  0.03225806    epsilon:  0.1
# Number of Support Vectors:  1101
cor(predict(model,d_entrenamientof),d_entrenamientof$SEXO)  # [1] 0.08038576
### plot(model)
d<-read.table(file="C:/Users/NisaB/Desktop/NBTS/Datos_d2a_imp_180918.txt", header=T)
View(d)
dim(d)
nombres<-names(d)
indx = sapply(d,is.numeric) # transformamos a numerico las variables
indx[indx==1]
```

```
d[indx] = lapply(d[indx], function(x) as.numeric(as.character(x)))
str(d)
summary(d)
# SINTAXIS DE DATOS SIMULADOS - SVM Y KERNLAB
library("e1071")
library("fda")
x<-c(rnorm(100,0,1),rnorm(100,5,1))
y<-c(rnorm(100,0,1),rnorm(100,5,1))
clases<-c(rep(1,100),rep(2,100))
plot(x,y,type="n")
points(x[clases==1],y[clases==1],col=2)
points(x[clases==2],y[clases==2],col=4)
coord<-cbind(x,y)
plot(coord)
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",scale=F)
points(model$SV,col=2,pch=20)
model$SV
    coordx          coordy
# 21      1.467914          2.675001
# 140     1.808059          5.703508
# 154     3.892939          2.738039
summary(model)
#Call: svm(formula = as.factor(clases) ~ coord, type = "C-classification",     kernel = "linear", scale = F)
#Parameters:   SVM-Type:  C-classification   SVM-Kernel:  linear       cost:  1      gamma:  0.5
#Number of Support Vectors:  3    ( 1 2 )   Number of Classes:  2   Levels:  1 2
model$tot.nSV #[1] 3
predichos = as.numeric(predict(model,coord))
table(predichos,clases)
#        clases
# predichos   1   2
#           1 100   0
#           2   0 100
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",scale=F,cost=0.1)
plot(coord)
points(model$SV,col=2,pch=20)
model$tot.nSV #[1] 6
model<-svm(as.factor(clases)~coord,type="C-classification",kernel="linear",scale=F,cost=1)
plot(coord)
points(model$SV,col=2,pch=20)
library("kernlab")
kfunction <- function(linear =0, quadratic=0)
 {
  k <- function (x,y)
 {
    linear*sum((x)*(y)) + quadratic*sum((x^2)*(y^2))
  }
  class(k) <- "kernel"
```

```
  k
 }
n = 25
a1 = rnorm(n)
a2 = 1 - a1 + 2* runif(n)
b1 = rnorm(n)
b2 = -1 - b1 - 2*runif(n)
x = rbind(matrix(cbind(a1,a2),,2),matrix(cbind(b1,b2),,2))
y <- matrix(c(rep(1,n),rep(-1,n)))
svp <- ksvm(x,y,type="C-svc",C = 100, kernel=kfunction(1,0),scaled=c())
plot(c(min(x[,1]), max(x[,1])),c(min(x[,2]), max(x[,2])),type='n',xlab='x1',ylab='x2')
title(main='Linear Separable Features')
ymat <- ymatrix(svp)
points(x[-SVindex(svp),1], x[-SVindex(svp),2], pch = ifelse(ymat[-SVindex(svp)] < 0, 2, 1))
points(x[SVindex(svp),1], x[SVindex(svp),2], pch = ifelse(ymat[SVindex(svp)] < 0, 17, 16))
w <- colSums(coef(svp)[[1]] * x[SVindex(svp),])
b <- b(svp)
abline(b/w[2],-w[1]/w[2])
abline((b+1)/w[2],-w[1]/w[2],lty=2)
abline((b-1)/w[2],-w[1]/w[2],lty=2)
```

*######################################## SCRIPT FINALIZADO ####################################*