Sistemas de recomendación flexibles dependientes de usuario

Bartolomé Ortiz Viso

Supervisoras:

Prof. María Amparo Vila Miranda

Prof. María José Martín Bautista

Programa de Doctorado en Tecnologías de la Información y la Comunicación

Universidad de Granada

Marzo, 2024

UNIVERSIDAD DE GRANADA

User-dependent Flexible Recommender Systems

Bartolomé Ortiz Viso

Supervisors:

Prof. María Amparo Vila Miranda

Prof. María José Martín Bautista

Ph.D. Program in Information and Communication Technologies

University of Granada

Marzo, 2024

Editor: Universidad de Granada. Tesis Doctorales
Autor: Bartolomé Ortiz Viso
ISBN: 978-84-1195-299-6
URI: https://hdl.handle.net/10481/92324

Dedicatoria

A mis padres y a mi hermana, porque ni una sola coma de este texto habría sido posible sin vosotros. Os quiero mucho.

Gracias.

Agradecimientos

Terminar una tesis nunca es un esfuerzo individual, más bien un esfuerzo (y sufrimiento) colectivo. No quiero dejar pasar la ocasión, por tanto, de agradecer a aquellas personas que han puesto su granito de arena en este proyecto (son muchas, pero intentaré ser breve):

Gracias a mis directoras, Amparo y María José, para quienes no tengo más que buenas palabras. Mis inicios en el mundo académico no fueron sencillos, pero gracias a vosotras pude encontrar un grupo de magníficos profesionales y personas donde me sentí acogido y valorado. Gracias por darme la oportunidad y guiarme en este camino.

A mis compañeros y compañeras de grupo, los "*ugritaiers*", por las horas de sufrimiento compartido pero sobre todo por todos esos momentos de team-building lejos de las pantallas de ordenador.

Al mejor despacho posible, el *DB3 goloso*, y a lo que construimos durante esos años de tesis. Irrepetible.

A los Fluxions, al ATPATC, y a quienes los hicieron posibles, porque de todos los proyectos en los que me embarqué durante la tesis, en ellos fue donde me sentí más vivo.

A Mise y Oliva, simplemente por estar y querer sin condición alguna.

Y por último, a mis *chochus*. Gracias por esos paseos y por esos cafés. Gracias por estar ahí y aliviar la carga cuando pesaba demasiado.

Acknowledgment

During the elaboration of this thesis, this work was supported by the following research projects and grants:

- 10.13039/100010661-European Union (Stance4Health) (Grant Number: 816303)
- 10.13039/501100004837-Ministerio de Ciencia e Innovación funded by MCIN (Ministerio de Ciencia e Innovación)/AEI (Agencia estatal de Investigacion)/10.13039/501100011033 and by ERDF (European Regional Development Fund) A way of making Europe (Grant Number: PID2021-123960OB-I00)
- 10.13039/501100004837-MCIN (Ministerio de Ciencia e Innovación)/AEI (Agencia estatal de Investigacion)/10.13039/501100011033
- European Union NextGenerationEU/PRTR (Plan de Recuperación, Transformación y Resiliencia) (Grant Number: TED2021-129402B-C21)

I would also want to acknowledge the institutions and professionals that welcome me:

- University of Granada (Spain), specially the Research Centre for Information and Communications Technologies (CITIC-UGR) where most of this thesis was made and is full of great professionals.
- University of Aarhus (Denmark) where I did my research residence in 2022. Thanks Darius, Violeta, Liisa and Klaus for your valuable time.

Abstract

Recommender systems are ubiquitous in today's technological landscape, integral to various services. Their prevalence is driven by the massive flow of information on the internet, and without effective filtering, user experiences would suffer. As powerful tools, recommender systems facilitate optimal interaction with the digital world by extracting relevant information tailored to individual needs. Despite their widespread use, recommender systems are also commercial tools, prompting extensive research in recommendation techniques and algorithms. This fast-paced research area holds the potential for significant impacts on our lives, raising concerns about their influence on both physical and psychological aspects.

In the academic domain, recommender systems serve diverse applications, with a primary focus on assisting users in media service selection or product choices in e-commerce. The evolution of these systems, fueled by advances like neural networks, expanded their applicability beyond traditional domains to areas such as news and e-learning. Despite advancements, challenges persist in effectively applying recommender systems, particularly in complex scenarios influenced by contextual factors and diverse objectives. The need for personalized recommendations, considering multiple items, their order, and additional contextual factors like health considerations and expert opinions, presents an intriguing problem lacking a clear resolution strategy.

This thesis aims to address these complex recommendation scenarios by developing new methodologies and software tools. The combination of various elements, textual features, expert sources, and user preferences forms the forefront of this recommendation problem, requiring innovative solutions. With this purpose, we propose an initial approach

to recommendation systems in complex environments, highlighting the differences between them and their potential applications in a novel classification. In this classification, we emphasize the source of complexity in various scenarios by collecting studies that have explored these environments.

Once this classification is completed, we proceed to present our approach to the problem. In our case, it involves a dual approach: firstly, we use those strong constraints to obtain combinations of items that can be recommendations. These solutions are created through a genetic algorithm that evaluates different designated constraints to find multiple composite solutions. Furthermore, the stochastic nature of the system allows us not only to satisfy numerous constraints but also to have great diversity and adaptability. Once these initial solutions are obtained, in the second phase of our system, we refine our recommendation. This second phase typically focuses on user preferences, which are less restrictively profound than the previous ones and allow for greater variability.

This approach is tested in the domains of nutrition and podcasts. The former is an application within the European project Stance4Health, utilizing the initial prototype of our system. In the podcast application, we offer a less strict type of recommendation than the first, allowing for greater enhancement in line with user preferences. The second system is configured as a Python package, enabling broader replicability and the use of our approach in various scenarios.

Subsequently, our approach delves into understanding user interests from a psychological perspective in recommendations and identifies what is necessary for users to follow recommendations. This is particularly relevant in the case of health-based nutrition recommendations, as users may not notice an immediate beneficial effect from the system, but the long-term effects are positive. To that end, we conduct a theoretical study on parameters that increase user engagement and subsequently evaluate it in the health application Stance4Health. From the research done in this field, we find that justifications, by enhancing the explainability of the system, can make users perceive our recommendations as more useful and interesting. However, obtaining these justifications can be costly. To address this, we propose a supervised algorithm that filters documents from nutrition experts to build an evidence-based database, enabling nutritional justifications based on the evaluation of our recipe.

Finally, we provide an estimate of the greenhouse gas emissions generated by our approach throughout the thesis, with data allowing estimation per individual execution.

Keywords: Recommender systems, knowledge-based recommender systems, genetic algorithms, trustworthy systems, explainable systems.

Resumen

Los sistemas de recomendación son omnipresentes en el panorama tecnológico actual, apareciendo en la mayoria de aplicaciones que usamos hoy en dia. Su prevalencia se debe a las enormes cantidades de información que se vierten a internet dia a dia y a como, sin un filtrado efectivo, la experiencia del usuario sería totalmente inoperativa. Los sistemas de recomendación facilitan la interacción óptima con el mundo digital al extraer información relevante de los elementos con los cuales interactuamos y facilitandonos el acceso a aquellos que es mas probable que nos interesen segun las necesidades individuales. A pesar de su uso generalizado, los sistemas de recomendación también son herramientas comerciales, lo que ha impulsado una amplia investigación en técnicas y algoritmos de recomendación. Esta área de investigación tiene el potencial de tener impactos significativos en nuestras vidas, generando preocupaciones sobre su influencia tanto en aspectos físicos como psicológicos.

En el ámbito académico, los sistemas de recomendación tienen diversas aplicaciones, centrándose principalmente en ayudar a los usuarios en la selección de servicios o productos en comercio electrónico. La evolución de estos sistemas, impulsada por avances como las redes neuronales, amplió su aplicabilidad más allá de los dominios tradicionales a áreas como noticias y canciones o películas. A pesar de estos avances, aun hay desafíos en la aplicación efectiva de los sistemas de recomendación, especialmente en escenarios complejos influenciados por factores contextuales y objetivos diversos. La necesidad de recomendaciones personalizadas, considerando múltiples elementos, su orden y factores contextuales adicionales como consideraciones de salud y opiniones de expertos, plantea un problema intrigante que carece de una estrategia clara de resolución.

Esta tesis tiene como objetivo abordar estos escenario de recomendacion denominados complejos mediante el desarrollo de nuevas metodlogias y herramientas de software. La combinación de varios elementos, características textuales, fuentes de expertos y preferencias del usuario constituye la vanguardia de este problema de recomendación, requiriendo soluciones innovadoras. Con este propósito, proponemos un enfoque inicial para sistemas de recomendación en entornos complejos, destacando las diferencias entre ellos y sus posibles aplicaciones en una clasificación novedosa. En esta clasificación, resaltamos la fuente de complejidad en diversos escenarios mediante la recopilación de estudios que han explorado estos entornos.

Una vez completada esta clasificación, procedemos a presentar nuestro enfoque para el problema. En nuestro caso, implica un enfoque dual:primeramente utilizamos aquellas restricciones fuertes para obtener combinaciones de items que puedan ser recomendaciones. Estas soluciones se crean mediante un algoritmo genetico que evalua las diferentes restricciones designadas con el fin de encontrar multiples soluciones compuestas. Ademas, la estocasticidad del sistema nos permite no solo satisfacer numerosas restricciones, si no tener una gran diversidad y adaptabilidad. Una vez obtenidas estas soluciones iniciales, en la segunda fase de nuestro sistema, refinamos nuestra recomendación. Esta segunda fase se centra habitualmente en las prefrencias del usuario, que son de un menor calado restrictivo que las anteriores y admiten una mayor variabilidad .

Este enfoque se prueba en los ámbitos de nutrición y podcasts. El primero es una aplicación dentro del proyecto europeo Stance4Health, utilizando el prototipo inicial de nuestro sistema. En la aplicación de podcasts, ofrecemos un tipo de recomendación menos estricta que la primera, permitiendo una mayor mejora en línea con las preferencias del usuario. El segundo sistema se configura a partir de un paquete de Python, permitiendo una replicabilidad más amplia y el uso de nuestro enfoque en diversos escenarios.

Posteriormente, nuestro enfoque profundiza en comprender los intereses del usuario desde una perspectiva psicológica en las recomendaciones e identifica lo necesario para que los usuarios sigan las recomendaciones. Esto es particularmente relevante en el caso de las recomendaciones de nutrición basadas en la salud, ya que el usuario puede no notar un

efecto beneficioso inmedito por el sistema, pero a la larga los efectos son positivos. Con ese fin, realizamos un estudio teórico sobre parámetros que aumentan la participación del usuario y posteriormente lo evaluamos en la aplicación de salud Stance4Health. A partir de la literatura, encontramos que las justificaciones, al mejorar la explicabilidad del sistema, pueden hacer que los usuarios perciban nuestras recomendaciones como más útiles e interesantes. Sin embargo, obtener estas justificaciones puede ser costoso. Para abordar esto, proponemos un algoritmo supervisado que filtra documentos de expertos en nutrición para construir una base de datos basada en evidencia, permitiendo justificaciones nutricionales basadas en la evaluación de nuestra receta.

Finalmente, proporcionamos una estimación de las emisiones de gases de efecto invernadero generadas por nuestro enfoque a lo largo de toda la tesis, con datos que permiten la estimación por ejecución individual.

Palabras clave: Sistemas de recomendación, sistemas de recomendación basados en conocimiento, algoritmos genéticos, sistemas trustworthy, Sistemas explicables.

Contents

List of Figures

Chapter 1

Introduction

“Nothing is more difficult, and therefore more precious, than to be able to decide."
— Napoleon Bonaparte, -

Recommender systems are one of the most widespread technologies in our society, to the extent that it is challenging for us to recognise an technological service that does not incorporate them, regardless of the context in which it is situated. There is a reason behind this fact: currently, vast amounts of information are poured onto the internet, and the absence of an effective information filtering system would result in a poor user experience. Therefore, recommender systems represent one of the most powerful tools for achieving a satisfactory interaction with the digital world, as they assist us in extracting the maximum amount of relevant information for our needs.

This, in turn, does not hide the fact that recommender systems are also one of the most widely used commercial tools to date. Both factors have led to an initial acceleration in research around recommendation techniques and algorithms, concurrently with the development of new and improved data-sets to challenge the models and metrics that have been developed. Consequently, we find ourselves in a fast-moving research area that strives to improve its results, with a potentially enormous impact on our lives.

Moreover, this has also, although more recently, raised concerns about the power of these systems to affect our lives, both on physical and psychological sides. As a result there has been an increase in the number of studies where ethics plays a fundamental role, developing to adapt to the situations and risks posed by recommender systems.

1.1 Motivation

The applications of recommender systems are numerous and highly diverse, particularly in the academic domain. However, it is relevant to note that the canonical application of such systems typically has two very clear and common objectives: assisting the user in choosing from a catalog of media services (typically music, series, or movies) and/or aiding the user in finding specific products in an e-commerce setting. These scenarios have been an ongoing challenge since the early days of recommender systems, with a variety of articles focused on providing increasingly better results, creating new datasets for comparison, or developing new metrics that more accurately represent the advantages of one algorithm over another.

These results marked a key moment in the history of this field because researchers realized that these algorithms, through changes, transformations, and thematic adaptations, could expand and be useful in many different areas. Thus, this type of system expanded into areas such as news and e-learning. Simultaneously, aided by the advent of neural networks, there was significant progress in the capabilities of these systems to consider an increasingly diverse array of information sources for the same task. An example of this is the transition from systems recommending books based on shared authors with the user's history to systems recommending books by extracting opinions from the textual analysis of user reviews and comparing them with similar demographic sectors. Being able to extract and compute more and better information from users, items, and their context allowed the effective application of these systems to new areas, not only recommending individual items but also groups of items for one or more users (something we will expand upon in Chapter 3).

Despite all this development, there are still areas where, although these systems have been applied, they have not done so effectively or efficiently. This is mainly because these are situations with a significant contextual load, influenced by many factors, and whose objectives are not limited to a single item. The peculiarity of these situations is that, for a successful recommendation, not just one item from our database is sufficient; we need several. Moreover, whenever we recommend multiple items, the order in which they are presented can be important (for instance, recommending a movie trilogy should follow a specific order).

In addition to these characteristics, it is important to consider that in many situations, it's not only the user's preferences that need to be taken into account. We've discussed context as a limiting factor that can bring about notable changes in user preferences (for example, recommending a sleep podcast should not occur during lunchtime, even though we know it's the user's favorite). However, Contextual factors are only one of the possible disruptors that can affect user preferences in a specific situation. In other situations where health should play a fundamental role in the recommendation, we could consider expert's opinions as important as user preferences for the recommendation process. This modifications should not be consider in total opposition to user preferences, but they can be changes to fulfill user's goals (a user may not love to eat vegetables, but they must be recommended if the user wants to follow a healthy diet).

All these mentioned characteristics pose a very interesting recommendation situation that, to date, lacks a clear resolution strategy. The combination of different items, the inclusion of textual features and expert sources, and the interaction with user preferences constitute this recommendation problem, residing at the forefront of what we know is possible. Hence, there arises a need for new tools and approaches capable of dealing with this situation and delivering novel results.

1.2 Hypotheses

At this point, we are shaping the type of scenario or issue that we aim to address during this thesis. From that analysis we are going to propose three pivotal hypotheses. These hypotheses encapsulate the key points we intend to analyze and challenge throughout the course of this thesis. Each hypothesis will connect, in the following section, to one or more objectives. With this conceptual relation we aim to offer an easy-to-follow path that explain our steps in this research project and the conclusions that contribute to a better understanding on which hypothesis are true and in what sense they can be assured.

- **Hypothesis 1 ($\mathcal{H}1$)**: In complex situations (i.e., characterized by a significant presence of contextual factors and parameters influencing the outcome), the ideal recommendation will seldom involve suggesting a single item to a user. Therefore, we will need a method to incorporate that information into a recommendation that simultaneously utilizes multiple items.

- **Hypothesis 2 ($\mathcal{H}2$)**: In such situations, the recommendations rules that must be adhered to at all costs should be treated differently than other less strict parameters, regardless of their origin or nature. These would allow the system to focus using with different strategies to excel at fulfilling most of them.

- **Hypothesis 3 ($\mathcal{H}3$)**: Lastly, in situations with several factors, user adherence to our recommendations should come by providing mechanisms for explainability and trust. If the user better comprehends our system and understands the rationale behind our recommendations, we will generate, at the very least, increased trust in the system and, over the long term, greater user adherence.

1.3 Objectives

Based on the hypotheses outlined in the previous sections and the forthcoming analyses in Chapters 2 and 3, we formulate a series of distinct objectives for this thesis. These objectives are structured as components of a research project aimed at developing innovative recommendation systems tailored for complex scenarios. In addition to the three primary objectives, we have outlined supplementary sub-goals that aid in delineating the thesis structure and the subsequent steps in our research. These objectives are interlinked with the aforementioned hypotheses, serving as the schema that will guide the remainder of this thesis. Thus, the identified objectives are:

- **Objective 1 ($\mathcal{O}1$):** To analyze the different recommendation systems employed to address the recommendation problem in complex situations and elucidate the challenges encountered. This objective will be fulfilled in Chapters 2 and 3, and contain the following sub-goals:
  - $\mathcal{O}1.1$: To define the scenarios considered complex and provide a mathematical description of the recommendation problem they pose. This step will narrow our broad recommendation problem to specific situations.
  - $\mathcal{O}1.2$: To delve into scientific literature pertaining to these complex problems and explore conceptual strategies employed to address them. This objective would improve our understanding and the actual limitations of recommendation systems and discard or reinforce any approach stated in $\mathcal{H}1$ and $\mathcal{H}2$.
- **Objective 2 ($\mathcal{O}2$):** To develop a recommendation system for complex situations capable of balancing multiple objectives dependent on users and external sources (experts and contextual factors). This objective will comprehend the construction of the algorithm and its successful application to different fields. This objective will be fulfilled in Chapters 4, 5, 6 and 7 of this thesis. Moreover, it constrains the following sub-goals:
  - $\mathcal{O}2.1$: To build information-aggregation mechanisms enabling the creation of complex items from simpler ones following specific rules. This objective aim to improve our understanding of $\mathcal{H}1$.
  - $\mathcal{O}2.2$: To develop adjustment algorithms allowing recommendations to be modified based on soft rules.
  - $\mathcal{O}2.3$: To utilize the aforementioned algorithms to build a recommendation system capable of generating personalized recommendations for different users.

Both $\mathcal{O}2.1$ and $\mathcal{O}2.2$ will be aligned in this final step, aiming to add insights on $\mathcal{H}2$.

- **Objective 3 ($\mathcal{O}3$):** To investigate and enhance user adherence to recommendation systems by improving their explainability in multi-objective situations. We will deep-in in the current theory of technology acceptance and explain the explainability approaches in Chapter 8. Inside we have marked the following goals:
  - $\mathcal{O}3.1$: To translate system rules into linguistically describable statements in natural language.
  - $\mathcal{O}3.2$: To generate explanations and justifications based on the rules used to generate recommendations, whether strict or not.
  - $\mathcal{O}3.3$: To build a secondary system utilizing the previous results to concurrently provide unsupervised natural language justifications for recommendations. This final step will let us extract conclusions about our approach and validate the ideas of $\mathcal{H}3$.

After outlining the objectives of the thesis, the remainder of the document proceeds as follows. In Chapters 2 and 3, we delve into the state of the art of recommendation systems and the techniques employed to generate recommendations, building upon the specifics outlined in each objective. Through this exploration, we elucidate the strategies employed in complex scenarios with multiple factors. Subsequently, in Chapters 4 through 6, we introduce our recommendation algorithm, providing a theoretical description supplemented by two applications in distinct domains in the subsequent chapters. In Chapter 7 we introduce the python package we created along with the application to widespread the usage of our approach. Moving on to Chapter 8, we address the psychological approach to recommendation systems and discuss the utilization of an explicability-focused approach to enhance user engagement. Lastly, we offer a description of the climate impact of the thesis to end with the conclusions from the study and outline avenues for future development.

1.4 Associated publications

During the completion of this thesis, the following academic works have been published and collaborated on:

1.4.1 Conference papers

- **Ortiz Viso, B.**, Evolutionary Approach in Recommendation Systems for Complex Structured Objects. in RecSys 2020 - 14th ACM Conference on Recommender Systems 776–781 (2020). doi:10.1145/3383313.3411455.

- **Ortiz-Viso, B.**, Martın-Bautista, M. J. & Vila, M.-A. Sistemas de recomendacion evolutivos para objetos con estructuras complejas. in Actas del XXI Congreso de Tecnologías y Lógica Fuzzy (ESTYLF'22) Escuela de Ingeniería Industrial y Aeroespacial Campus de la Fábrica de Armas (2022).

- **Ortiz-Viso, B.**, Fernandez-Basso, C., Gómez-Sánchez, J. & Martin-Bautista, M. J. "Health Is the Real Wealth": Unsupervised Approach to Improve Explainability in Health-Based Recommendation Systems. in Flexible Query Answering Systems (eds. Larsen, H. L. et al.) 234–246 (Springer Nature Switzerland, Cham, 2023). doi:10.1007/978 − 3 − 031 − 42935 − 4_19.

1.4.2 Journal papers

First author:

- **Ortiz-Viso, B.**, Morales-Garzón, A., Martin-Bautista, M. J. & Vila, M.-A. Evolutionary Approach for Building, Exploring and Recommending Complex Items With Application in Nutritional Interventions. IEEE Access 11, 65891–65905 (2023).

Collaborator:

- *Hinojosa-Nogueira, D., ***Ortiz-Viso, B.** .et al. Stance4Health Nutritional APP: A Path to Personalized Smart Nutrition. Nutrients 15, 276 (2023). *(Equal Contribution)

- Hinojosa-Nogueira, D., Perez-Burillo, S., Navajas-Porras, B., **Ortiz-Viso, B**. et al. Development of an unified food composition database for the european project "Stance4Health". Nutrients 13, 4206 (2021).

1.5 Graphical Summary

To conclude the introduction, a timeline connecting key scientific advances related to the presented recommendation model is illustrated in the graphical summary (Figure 1.1). The journey began with an analysis of the state of the art, leading to the design and validation of a theoretical approach presented at the RecSys conference. Subsequently, the development of the first prototype and its application commenced. The nutritional prototype was then utilized in the S4H nutritional trial, focusing on microbiome-related advancements.

During the trial, the author contributed to creating the nutritional database and co-authored the final app design, subsequently published. Following the trial, feedback was gathered, leading to the refinement of the model and a specific design for nutritional intervention, resulting in another publication. A field-agnostic package was then developed, allowing the approach to be applied across various fields, and integrated with other Python packages in the recommendation domain.

The feedback from the nutritional intervention emphasized the need for improvement in the algorithm's explanation system. This culminated in the final work of the thesis, a conference paper where the author developed an unsupervised method for generating recommendation justifications.

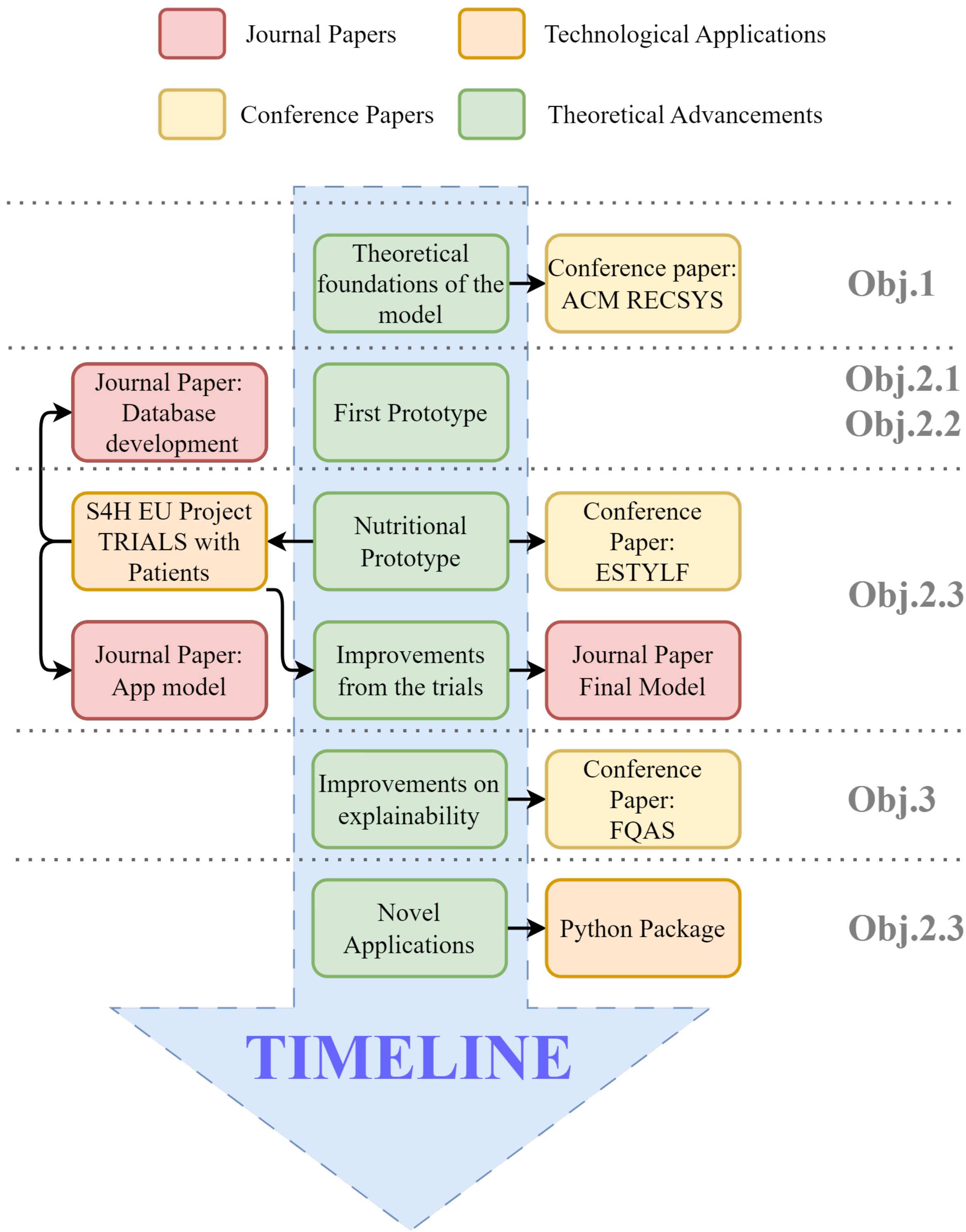

Figure 1.1: Illustration depicting the chronological progression of the thesis.

Chapter 2

History of the Recommendation systems

"The history of science, like the history of all human ideas, is a history of irresponsible dreams, of obstinacy, and of error."
Conjectures and Refutations - Karl Popper,*1960*

As discussed in the introduction, recommendation systems represent a commonly utilized technology in our society. However, while the intuitive idea is straightforward to grasp, in order to engage in a more sophisticated discussion about them (also accurately describe them and introduce the points where novel advancements have been made in this thesis), we require a theoretical framework that precisely defines what these systems are, what kind of problems they address, and what components constitute them. Throughout this section, we will address these questions, providing an exposition that elaborates on the concept of recommendation systems, the classical types of recommendations from a historical perspective, and the theoretical models of recommendation.

2.1 The recommendation problem

Recommender systems constitute a specific category of information filtering systems with the capability to select the most relevant items for a user from an extensive set. They achieve this by utilizing information about users, items, their interactions, and any other external data source.

To objectively discuss around its definition, we have created an extensive definition of the recommendation problem that serves as an formal definition as well as an introduction of the variables to be consider not only to classify but also to expose the results of this thesis:

Def 1.1 (Recommendation Problem): *Given a set of* ***Users*** $\mathcal{U}$ *and* ***Items*** $\mathcal{I}$*, create a function* $f(i, u|\mathcal{R}, \mathcal{C}) \longrightarrow p, \forall(i, u) \in (\mathcal{I}, \mathcal{U})$ *that associate each Item* i *with the probability* p *that a specific User* u*, finds it relevant, based on the interactions/ratings* $\mathcal{R}$ *between users and items, and the particular context,* $\mathcal{C}$ *in which they are located.*

Where **Item** is the general term used to denote what the system recommends. **User** is a generic term referring to the entity that has made the request (whether it be an individual person, a group of individuals, or any other stakeholder),**Rating** will be the interactions between Items and Users, and finally, **Context** is all the additional information taken into account for the prediction.

This definition is applicable to various scenarios where these systems operate, ranging from music (predicting which songs are more relevant to offer to users in a streaming service based on their likes) to nutrition (predicting which meals are more relevant to offer to users from a large dataset of recipes based on their goal). Before continuing, two clarifications should be made: the first related to the use of the word "relevant" instead of "favorable","desired" or "preferred" and the second concerns additional sources of information, the so called "Context".

Regarding the use of the term "relevant": it is deliberately chosen to convey a broader sense than just liked items. Recommender system are applied in multitude of different situations, and in some of them, the context may play a specific role in the recommendation, encompassing items that may not necessarily be desired or favorable but are pertinent to the user's needs. An applied example would be a food recommendation that is not strictly based on user preferences, but also (or maybe exclusively) on user goals i.e. lose weight.

As for the second point, regarding additional sources of information, recommender systems may draw insights not only from user-item interactions but also from a huge variety of external data sources. These can include demographic data, expert's opinions, behavioral data, or contextual information (i.e. weather, country, marital status, etc). All of them enhance the system's ability to generate more accurate and personalized recommendations.

In essence, what we are trying to express is that despite it first direct application, recommender systems extend beyond a "simple" matching of user preferences with items; they encapsulate a formal predictive framework that leverages diverse data sources to connect users with items they needed.

Next, we will delve further into the concepts of user, item, and context (where we will process the interactions between users and items and other additional sources of information). The handling of these concepts will be pivotal in the subsequent sections, where they will prove key factors in understanding the subdivisions of various recommendation models as well as the different types of recommendations we will propose.

2.1.1 Users

Users are the entities seeking recommendations, and thus, they constitute the central axis around which all recommendation systems are developed. The objective of our system involves comprehending, to some extent, as much as possible about users at different levels. This includes understanding their objectives, characteristics, and the context in which they are using our system. Of course, their preferences are also a crucial aspect, but it is noteworthy that these preferences fall under the broader category of characteristics.

All these varied sources of information are utilized to construct a profile of our user

[1]–[3], a user model. Within this model, we encode features accessible through demographic, biological, or historical data. Additionally, preferences and other relevant data generated from various sources are encoded. Examples of such information may include a user's purchase history, a generalization of tags associated with these items (as a type of processed feature), inclusion in a specific demographic group (with or without processing, i.e., young people vs. individuals with similar tastes), or a highly relevant process in recent years: extracting opinions about certain items based on user-written reviews.

2.1.2 Items

Items refer to all entities that we aim to recommend within our recommendation system. When addressing any recommendation problem, our objective revolves around gaining a deeper understanding of the type of recommendation we intend to make and the nature of the object under consideration. The depth of this understanding will depend on the approach and technique we choose to employ. However, the current trends [4], [5] emphasizes acquiring or generating features for our items to alleviate situations where the introduction of an item or user to our database yields too much uncertainty to recommend it effectively, both from data around items and users but also around the ontology concepts that links them. In short, using novel techniques to use additional knowledge to better represent the item and its context.

The initial consideration in dealing with items, nevertheless, involves locating them. This entails accessing a database containing a substantial quantity of items available for recommendation (as a low quantity would produce a recommendation either unspecific or trivial). Access to such data-sets is not always readily available, and their release for improved model comparisons remains a topic of intense debate [6]. Once we have the objects, we can determine whether the existing features are sufficient or if we should enhance their description in our system.

To achieve this, text and image processing techniques have represented a notable advancement in the field. These techniques enable us to obtain unsupervised information, allowing for improved classification and understanding of the items in our problem domain. Finally, another strategy to complement this information involves aggregating data from various sources concerning the same item. We will explore this option in the last of these subsections.

2.1.3 Ratings and other User-Item interactions

While we can extract a considerable amount of information from users and items, one of the significant advancements in this type of system came with the introduction of user interactions with items, both positive and negative [1]. Interactions constitute a fundamental pillar in many recommendation models and techniques [7], making them stand out amidst other contextual values. Typically, these interactions stem from data-sets that log users' interactions with items, aligning actions from the final selection of an item to the preceding interactions leading to this conclusion (including selection, evaluation, rating, removal from the shopping list, etc.).

Although these interactions can be diverse, we commonly categorize them into two types: implicit [8] and explicit interactions[1]. Explicit feedback occurs when users directly evaluate items, either through a numerical scale (ratings, stars), a binary option (like/dislike), or the ordering of a scale (item 1 before item 2). By definition, explicit feedback is the most accurate, as it represents the least manipulated information about the interactions between the user and the item.

In contrast, implicit interactions encompass all those surrounding the recommendation process that do not involve a direct evaluation by the user of the recommended items. Examples include clicks or viewing time. While these measures introduce more uncertainty than explicit feedback, they also raise interesting research questions, as their quality to establish conclusions is weaker and they tend to be noisier than explicit ones.

2.1.4 Context and Additional knowledge

We have aimed to underscore user-item interactions in a distinct category, given their longstanding and prevalent use in the recommendation process. In addition to these interactions, we currently have access to an extensive array of additional information sources that can be employed in the recommendation process.

These data sources contribute to understanding the context in which the recommendation occurs, considering several factors (e.g., visit time planed for a tourist recommendation [9]), those external to the user (e.g., meteorological conditions during the visit [10]), or factors that condition the recommendation (e.g., generating recommendations accounting the relations between the users in a group [11]) or may potentially alter user preferences (e.g., health recommendations after an illness [12]).

All these information sources serve to both constrain and augment the complexity of the problem while providing greater adaptability. The incorporation of many of these features is indeed the foundation of the knowledge-based recommendation along with other fields as multi-objective recommendation [13] that will be discussed in the next section.

2.2 Models of recommendation

Once recommendation systems and the data sources involved in the recommendation process are accurately defined, we proceed to provide a comprehensive overview of the current state of the art in recommendation systems [14]. We focus on recommendation models and recommendation techniques. In this section of models, we delve into the theoretical approaches employed in the recommendation problem, providing a comprehensive classification adapting seminal works in the field as [15].

We will refer to recommendation models as the theoretical frameworks that guide us in understanding which data sources are utilized in the recommendation process, which user-item's interactions are considered, and how they are taken into account. For example,

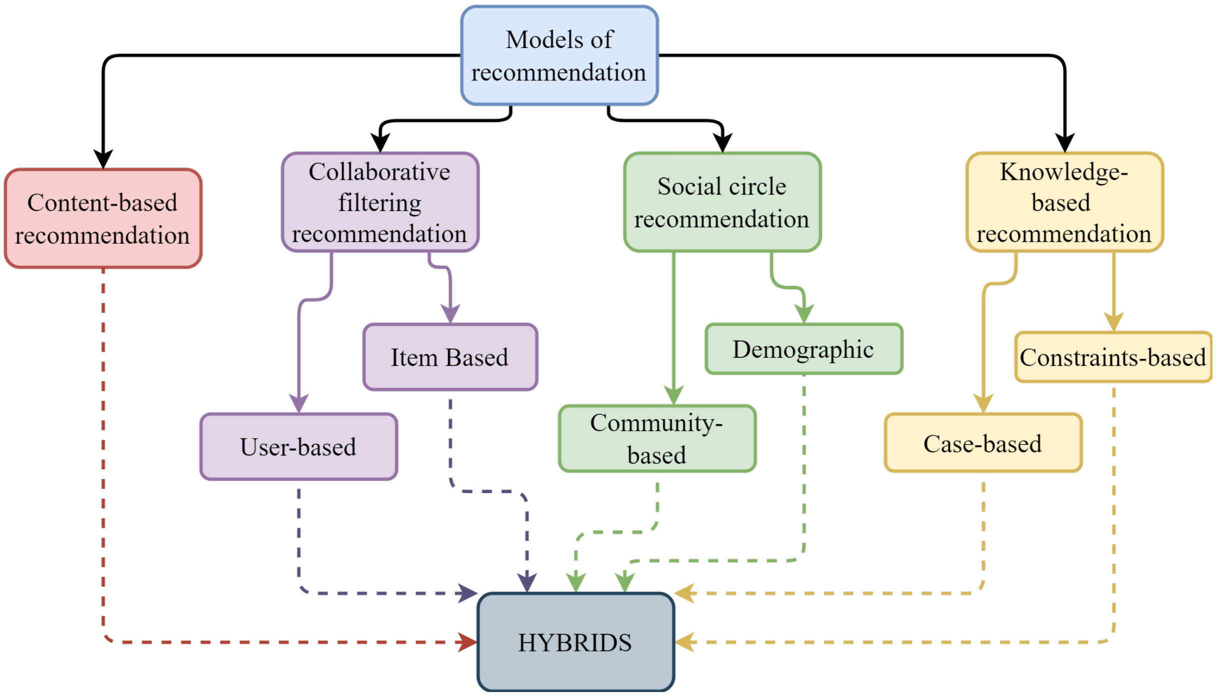

Figure 2.1: Models of recommendation and its relationships as described in this chapter.

the classification of recommendation models allows us to differentiate between content-based models (systems that use the characteristics of the items to be recommended) and demographic models (which take into account user demographic profiles and their relationship to the items). This classification is highly flexible and allows for a great deal of granularity or differentiation. Furthermore, it is a purely conceptual classification, not technological, which is why it has been chosen to demonstrate the interoperability of recommendation techniques (in the following subsection) and how models accommodate multiple approaches and applications. A summary of the models presented can be found in Figure 2.1.

2.2.1 Content-based recommendation

The first of the recommendation models or approaches we will explore is content-based filtering[16]. In this model, we focus exclusively on the descriptive features of items to construct a parametric representation of all items and users (taking into account which items they have interacted with). This type of system supports a broad approach, as the features used for recommendation can be intrinsic to the object or a description thereof (with tags), or they can be representations extracted through more recent information processing techniques such as natural language processing. Ultimately, the recommendation problem transforms into a metric problem where, given the user and item representations, we attempt to find the closest items using a predefined distance metric.

2.2.2 Collaborative filtering recommendation

Collaborative filtering [17] is a type of recommendation system that predicts a user's preferences or interests by leveraging the opinions and behaviors of a group of users. Instead of relying on explicit knowledge about items or users, collaborative filtering makes predictions based on the preferences of users with similar tastes. There are two main types of collaborative filtering:

- User-Based Collaborative Filtering: This approach recommends items based on the preferences of users who are similar to the target user. The system identifies users with similar tastes by analyzing their historical ratings and recommends items liked by those similar users.

- Item-Based Collaborative Filtering: In this approach, recommendations are made by identifying items that are similar to the ones a user has liked or rated with in the past. The system computes similarity between items based on user ratings and suggests items that obtain the highest predicted rating related to the user's historical preferences.

Collaborative filtering does not require a detailed understanding of the items or users; instead, it relies on the collective wisdom of the user community. Even all models has their drawbacks, collaborative filtering models are pone to have "cold start" problems, where new items or users have limited or no interaction history. As recommendations are based on the behavior of similar users or items, having none of those data sources may result in an under-recommended item or unreliable recommendation in the long run. This is why even they are effective, they are commonly hybridize, at least in some scenarios.

2.2.3 Social circle-based recommendation

Much of the recommendation systems exhibit selectivity in choosing which individuals to focus on for making recommendations. This selectivity is driven not only by the fact that individuals within our social group have a greater influence on us but also because individuals within our demographic range inherently possess a considerable amount of shared experiences and cultural references. This distinction allows the use of such social and demographic profiles to be categorized separately, often divided into two categories in works such as [7]. However, we have chosen to consider them as a classification based on social circles, aligning with the description by [18], with two intermediate categories: the proximity community to the individual and the social/demographic community of the individual.

- Community-Based Recommender Systems: Community-Based Recommender Systems recommend items based on the preferences of the user's friends and other social relationships [19], [20] as they usually supous a sources of more trustworthy recommendation. This also can lead to a more developed explainability system where recommendation are directly connected to user's friends to improve their adoption.

- Demographic Recommender Systems: Demographic Recommender Systems categorize users based on demographic attributes, proving beneficial when product information is limited, addressing scalability and cold-start problems. These systems utilize user attributes such as age, gender, and language to provide recommendations. Unlike content-based and collaborative-based filtering, demographic recommender systems yield fast results without requiring user ratings. However, one of the main challenge remains the privacy treatment of data, specially as novel regulatory laws are being developed.

2.2.4 Knowledge-based recommendation

Finally, we describe knowledge-based recommendation engines [21]. These systems need not only user preferences, but also a deeper knowledge of the application area. We have left this classification to the end because we believe that corresponds to the initial seed of the analysis presented in this work. It is in this kind of system that we can start to consider complexity as a way to describe them.

Knowledge-based recommendations are commonly divided into two basic categories [22].

1. On the one hand, we have those case-based recommendation engines [23]. In them, we adapt the recommendation according to the similarities that a certain user or case may have with those cases we previously have.

2. The second category are those engines that use rules or restrictions to make the recommendation[24].

Both approaches need a specific group of additional information, which usually implies a deeper knowledge of both scenarios and the constraints that represent the relationship between the most common recommendation elements (user and items).

Note that this approach could be made up of several simple rules, as well as items with few characteristics, so, the complexity of these models may vary among them. However, as stated in the previous paragraph when thinking about this type of systems and in particular in their applications within certain areas, the idea of complexity appears in a more or less naive way, understanding complexity as a combination of multiple options with several interconnected constraints and parameters. We will delve into this characterization in the next chapter as a starting point for our proposed system.

2.2.5 Hybrid systems

At this point, we understand that some questions about known recommender systems may arise. In the current research landscape, trying to strictly categorise a recommender system is nearly impossible. Most of the approaches have small characteristics or components may raise doubts about their classification, making it somewhat diffuse. This means that almost any application benefits from some hybridization between the models, especially outside of academia [25]. In fact, it is intuitive to think that as we approach applications with greater impact, in areas with greater complexity, this hybridization will be more pronounced. This does not diminish the presented classification; it makes it relevant due to its capacity to represent the characteristics of the systems in an atomic form.

Formally this issue is studied as Hybridization [26], [27], and refers to the combination of two or more models as described previously. The rationale behind this choice is to compensate for the limitations of one model with the advantages of another. A direct explanatory example would be using content-based recommendation to address the cold start problem in collaborative filtering approaches.

Several techniques can be implemented to fuse two or more systems, and these are thoroughly described in [15], [28]. In addition, several authors connect this hybridization with the contextual-aware models, as they usually improve already existing techniques while adding additional knowledge [29]. We could argue that this area is already a well established on its own. Even in it, we could explore more refined classification, as presented in [14] where points out differences between those engines that focus their sources on the location, time, or social context of the user.

2.3 Collaborative filtering techniques of recommendation

After reviewing the various recommendation models, our focus shifts to the computational techniques that enable the utilization of each information source to generate a recommendation (which, we should recall, is nothing more than a number prediction).

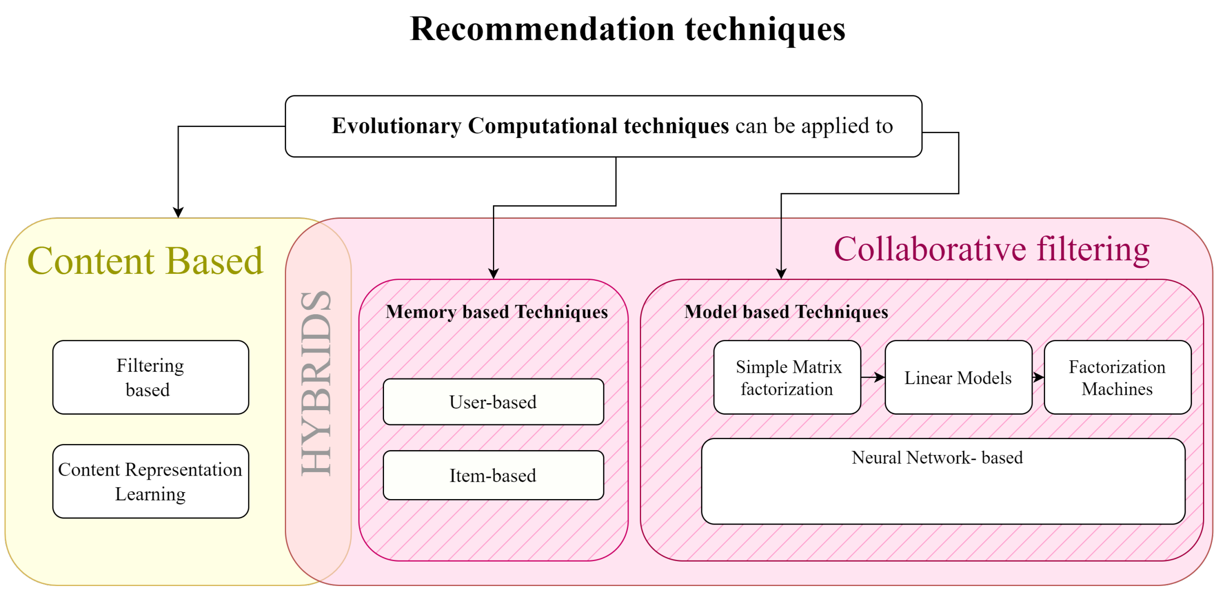

Figure 2.2: Summary of techniques

Subsequently, we offer a more technical perspective, delving into the principal computational techniques that have been and continue to be employed for generating recommendations, adhering to the outlined schemes. We specifically commence with the scheme presented in [30], but restructuring and expanding the sections to enhance their explanation and their relevance to this thesis.

Most of this techniques started associated with a specific model of recommendation, but most of them has gradually been used in hybrid approaches. As so, most of the techniques have been compared one to another and have bee hybridize with other when possible in the scientific literature. A summary diagram can be found in Figure 2.2.

Before starting, readers are advised to recall the notation followed in Section 2.1, as we will use it.

2.3.1 Memory based techniques

Memory-based techniques in recommender systems (also called neighbour-based) involve utilizing the direct observations and interactions of users to make personalized recommendations. These techniques rely on the stored history of user preferences and similarities between users or items, which often are computed from a group of users or items, which led us to have a K- nearest neighbour approach where k is the selected number of neighbours. They are often divided in user-based and item-based collaborative filtering.

- **User-based:** For a given item i, and a user u, a set of k-different users $V = v_1, \ldots, v_k$ is selected based on their similarity with the original user u, $w_{u,v}$ and they rating of i, r_v. The prediction function will be define as

  $$f(r) = \frac{\sum w_{u,v} r_v}{\sum w_{u,v}}$$

  where these rating can also be normalize around the maximum and minimum of the other users rating.

  In case the rating are more boolean oriented (like, dislikes or just likes) or we want to make that assumption, a similar approach can be made, changing the summarizing expression by a weighted delta, that is, one if the element is element is rated and 0 otherwise.

- **Item-based:** The item based procedure is similar for a regression approach. We want to get the predicted rated based on the similarity of the items already classified.

- **Similarity measures:** Both item and user-based approaches use similarity measures at some point to weigh the amount of information each rating contributes to the prediction. Several similarity approaches can be generated, but the two most commonly used are Pearson correlation and cosine similarity.

2.3.2 Model based techniques

Model-based techniques in recommender systems (also called latent-factor models) focus on creating predictive models by transforming user preferences and item characteristics into the same embedding space.

These models are constructed using machine learning algorithms that generalize patterns from observed data, allowing the system to make predictions for unseen user-item pairs. However, they also face the downside of sparsity in the interaction matrix, as well as the need to incorporate more contextual data than just interactions. Model-based approaches encompass low-rank models such as matrix factorization, linear models, factorization machines, and ultimately deep learning.

2.3.2.1 Matrix factorization

Matrix factorization models map users and items into a vector inside a shared latent factor space, x_{user} and x_{item}.The dimensionality of this space is a key parameter that determines the complexity and expressiveness of the model. In this space, user-item interactions are conceptualized as inner products, $< x_{user}, x_{item} >$. The latent space is composed of factors that store inherent characteristics of both products and users, deduced automatically without expert knowledge (can be considered as embedding representations of something close to a category). The predominant technique in this section is based on Singular Value Decomposition (SVD[31], although slightly difference, as classic SVD is undefined if interaction matrix is incomplete) where the rating, is predicted as

$$\hat{r}_{u,i} = \mu + b_{item} + b_{user} + < x_{user}, x_{item} >$$

Where μ refers to the overall average rating over all items and b_{item} and b_{user} refers to the observed deviation of the item and the user respectively from the average. SVD-based approach has already been adapted to multiple situations, from implicit feedback (SVD++ [32]) to time dependent recommendations. This approach can also be combined with other approaches for neural network implementations, using SVD or SVD++ as secondary step to obtain a low level representation from the embedding produced by the neural models [33].

What makes matrix factorization models particularly powerful is their ability to automatically deduce latent factors without requiring explicit expert knowledge about the characteristics of users or items, just using the interaction matrix.

2.3.2.2 Linear models

As already said, most of the advances in recommendation tend to use somewhat more data. Matrix models tend to focus on user-item interactions, but those can be scarce at first (cold start problem). To add additional information to the models, Linear models where design [30] as a way to start considering the side information, such as user demographic, historic behaviors, item attributes, contextual information.

The linear model $\hat{r} : R^p \to R$ over an input vector $x \in R^p$ is

$$\hat{r}(x) = \mu + \sum_{j=1}^{p} b_j x_j = \mu + \mathbf{b}^T \mathbf{x}$$

where $\mu \in R$ is the global bias and $b \in R^p$ a linear contribution of each input variable, e.g., b_j contains the effect for the variable x_j. In general we will use X as a collection of one hot encoding based on the categorical variables that describe our users and items. This encoding could vary from just the whole user-item dataset (thus user would hove a 1 in their id and 0 in the rest) to more sophisticated approaches joining user, item and implicit feedback.

Linear models, however, lacks the ability to learn feature interactions, and a common practice is to manually include pairwise feature interactions in its feature vector. Such method is hard to generalize to model high-order feature interactions or those never or rarely appear in the training data [34].

2.3.2.3 Factorization Machines

Factorization machines (FM) [34], [35] are a generalization of linear regression that include all pairwise interactions between the input variables. The general form of a Factorization Machine (FM) over a p-dimensional feature vector $\mathbf{x} \in \mathbb{R}^p$ is

$$\hat{r}(\mathbf{x}) := \mu + \sum_{j=1}^{p} b_j x_j + \sum_{j=1}^{p} \sum_{l>j} \langle \mathbf{v}_j, \mathbf{v}_l \rangle x_j x_l$$

where $V \in \mathbb{R}^{p \times f}$ are model parameters that are learned, and f is the embedding dimension.

The first part of an FM is equivalent to a linear regression, the second part contains the pairwise interactions $x_j x_l$ between all input coordinates. These values again can be form based on the user and item categories. But also can admit additional data, time constraint and historical data. Those vectors are usually form by one hot encoding if categorical or numerical otherwise (for example in ratings).

The evolution from this approach would be the Field Aware factorization machines [36]. Field-aware Factorization Machines groups features to several fields, and each feature has different latent vector for each field [30].

However despite all the novel techniques, the irruption of new forms of encoding has overwhelming surpass most of the shown approaches (in collaborative filtering). Those novel approaches for example AutoRec [37] in 2015, using autoencoders[38] surpassed all the previous attempts in rating prediction tasks. Autoencoders, and most widely, neural networks are now established as the state of the art in recommendation techniques.

2.4 Bioinspired recommendation techniques

2.4.1 Neural networks

Neural networks in recommender systems leverage deep learning architectures to model complex relationships and patterns in user-item interactions. These models can automatically learn hierarchical representations of features and capture nonlinear dependencies.

Since 2016 [30] more and more development has been made in this area and there are several reasons behind it rise[39].

First all of the previous techniques where based to some extent in linear o multi-linear models (from basic recommendation to factorization machine). Linearity was a huge assumption to be made in our kind of problems, specially when several recommendation strategies need different parameters with different nature. Deep learning and in particular neural network are notoriously able to work with nonlinear relation between variables as we can use activation function that are able to model these behaviours (i.e. relu or sigmoid).

Secondly, neural networks allow us to incorporate several sources of data to the recommendation process, but also are able to adjust in which way they should be taken into account. If have enough data, this aspect can reduce the handcrafting feature engineering process. In this section we will briefly introduce some of the techniques that evolve from the idea of applying neural networks to the recommendation problem, pointing out their approach and finally pointing out the tools that we will use as a secondary step in our proposal.

- Neural Collaborative Filtering. We first start with can be consider an natural extension from the traditional recommendation methods. First we can model the recommendation problem as a Two-tower neural network where users and items are embedded. Essentially their goal is to replace dot product with a MLP to enhance feature crossing capacity of the model.

  Neural Network Matrix Factorization (NNMF) [40] and Neural Collaborative Filtering (NCF) [41] are representative works of this approach.

  Let u and i denote the side information (e.g., user profiles and item features) or just the one-hot identifier of user u and item i. The scoring function is defined as follows:

  $$\hat{r}_{ui} = f(U^T \cdot s_{\text{user}_u}, V^T \cdot s_{\text{item}_i} | U, V, \theta), \tag{2.1}$$

  where U, V, and θ are parameters, and s_{user_u} and s_{item_i} represent the side information embeddings.

  Indeed we have augmented the generality of this problem, as traditional matrix factorization techniques can be viewed as a special case of NCF [42]. The whole network can be trained with weighted square loss (for explicit feedback) or binary cross-entropy loss (for implicit feedback).

- Wide & Deep Learning. The model comprises a wide learning component, akin to a single-layer perceptron acting as a generalized linear model, and a deep learning component represented by a multilayer perceptron [43].

  The fusion of wide and deep learning techniques allows the recommender system to balance memorization and generalization. The wide component excels at memorization, capturing direct features from historical data, while the deep component focuses on generalization by producing more abstract representations. This combination enhances both accuracy and diversity in recommendations.

- Deep Factorization Machines.Furthermore factorization machines can also experience a fusion with neural approaches. DeepFM [44] was the first to propose Deep Factorization models. This technique models high-order feature interactions using deep neural networks and low-order interactions with factorization machines. MLP leverages non-linear activations and deep structures to model high-order interactions. Indeed we are working with a kind of wide and deep model where we replace LR with FM as a new wide component.

  The input of DeepFM, denoted as x, is an m-fields data consisting of pairs (u, i) (identity and features of user and item). For simplicity, the outputs of FM and MLP are denoted as $y_{\mathrm{FM}}(x)$ and $y_{\mathrm{MLP}}(x)$, respectively. The prediction score is calculated by:

  $$\hat{r}_{ui} = \sigma(y_{\mathrm{FM}}(x) + y_{\mathrm{MLP}}(x)), \tag{2.2}$$

  where $\sigma(\cdot)$ is the sigmoid activation function. In contrast to wide and deep models, DeepFM does not require tedious feature engineering.

- Recurrent Neural Networks:

  The temporal evolution of users' preferences up to this point is a parameter we have not taken into account, partly due to the difficulty it would entail in modeling it within the system. Specifically, with this idea in mind, the application of

Recurrent Neural Networks (RNN) [45] begins to enhance the results of sequential recommendations.

One of the most notable works, utilizing the widely adopted Long Short-Term Memory (LSTM) Transformer model [46], is the Recurrent Recommender Network (RRN) [47]. This model employs two LSTM networks as foundational components to model the dynamic user state u_{ut} and item state v_{it}, aiming to capture the seasonal evolution of items and changes in user preferences over time.

Simultaneously, the model considers fixed properties such as user long-term interests and item static features, incorporating the stationary latent attributes of user u_u and item v_i. The predicted rating of item j given by user i at time t is defined as:

$$\hat{r}_{ui|t} = f(u_{ut}, v_{it}, u_u, v_i), \tag{2.3}$$

where u_{ut} and v_{it} are learned from LSTM, and u_u and v_i are learned by standard matrix factorization. The optimization objective is to minimize the square error between predicted and actual rating values.

- Deep Reinforcement Learning (DRL) introduces a temporal aspect to the recommendation process by incorporating the user's interactive behaviors in real time, allowing for adaptation to continuously changing environments [39]. Initial approaches, such as Exploration and Exploitation models like LinUCB [48], aim to enhance the model using real-time rewards.

  Subsequent studies have delved into this idea, with notable examples including DEERS (Deep Reinforcement Learning for Recommendation with Sequential Feedback) [49]. DEERS focuses on recommendation systems handling both negative and positive feedback in a sequential interaction setting. Another significant contribution is DRN (Deep Reinforcement Learning Network) [50], a framework based on deep Q-Learning for recommendation. DRN innovatively explores model parameters online using dueling bandit gradient descent, allowing it to adapt to dynamic changes in news content and user preferences. The model enhances recommendation diversity and incorporates return patterns of users to the service.

- Graph neural network based recommendations The interaction data generated by recommender systems can be represented as a bipartite graph where the nodes are users and items and the edges represent interactions between them. This schema offers a novel view on the recommendation problem and the tools we can use to deal with it.

  While we could talk here about network embedding only, Graph-based recommendations are not reduced to that. The tools for processing and extracting relevant data from graphs were highly boost with the apparition of graph neural networks GNN [51]. GNNs can model graph-structural relationships, allowing for more sophisticated analysis of user-item interactions. But they are not only used to model those interactions, as the ability of process knowledge graphs, so we can connect concepts and extract more meaningful connections between users and items. This connections along with the ability to introduce knowledge graph of all kinds can also be useful to produce better explainable recommendations. So graph-based recommendation covers not only the single task of network embedding but also all the tools based on graph neural networks that act as models that can be design for different task (within them, network embedding)[52]. The most prominent approaches for this task would be: the embedding methods to capture semantic relationships, the connection-based methods which use user-items connections and the propagation-based method that aims to fused both taking the representation of items and their connection to users.

  The two foundational works (and most influential) for the embeddings-based recommendation are are DeepWalk [53] and Node2vec [54]. Along with this embedding-base approach, the path-based methods also arised: build a user-item graph and leverage the connectivity patterns of the entity in the graph for recommendation. Combining both embedding and path-based method vi neural networks approaches has led to other propagation methods as RippleNet ([55] starts from selected seeding nodes and recursively propagates the embeddings from a node's neighbors to refine the node's embedding) or, KGAT (Knowledge Graph Attention Network, [56] adds user-item interaction links into graph and employs an attention mechanism to discriminate the importance of the neighbours) fast gain recognition and were wide-spreaded, generating a novel research field[52].

- Content Representation Learning

  Based on the exposition in the previous sections, it is clear that neural networks have represented a significant advancement in the field of recommendation systems. Their ability to encode information and translate it into vector spaces is highly powerful. These sections have provided an overview of various algorithms that can be employed in the recommendation process. However, in most cases, we are discussing an extension of collaborative filtering, where we predict the rating or the order of a potential recommendation.

  In this final subsection, which is most relevant to the thesis, we will explore another approach to neural networks: utilizing them in content-based recommendations.

  – Natural Language Processing (NLP) : The early models of Content-Based Recommender Systems (CBRS) relied on keyword-based approaches that employed simple term-counting [57]. However, these initial models encountered limitations in achieving a comprehensive understanding of textual content describing items and in encoding semantic relationships between terms. The ability to enhance the representation within our recommendation models, driven by advances in natural language processing, has been instrumental in improving these recommendations. We now have the capability to mine data and extract characteristics from every text, thereby enhancing both content-based and collaborative filtering approaches. In particular, collaborative filtering benefits from the ability to derive representations and classifications from text and reviews. However, in this work, we will focus on the content-based improvement based on embeddings.

Early textual approaches, such as bag of words, term frequency, and word2vec [58], were initially explored, but the results were not easily generalized as they heavily depended on specific terms. Despite the initial challenges, word2vec, when properly fine-tuned, outperformed other approaches. Following word2vec, several methods and advancements, including doc2vec, were proposed. Despite these early attempts, progress in neural networks enabled the incorporation of contextual information into the representation of embedded text.

Novel advances in text encoders like BERT or USE have played a significant role in handling ambiguity issues in content representation. These encoders also prove valuable for user and item profiling, whether through the aggregation of embeddings from interactions or descriptions of items. However, profiling users by aggregating embeddings from interactions or item descriptions remains an open research question [59]. Even there are works [60] that suggests that the inclusion of these encoders does not necessarily guarantee a better representation. Therefore, using their representation to enhance recommendations remains an interesting and active area of research. This is especially true even with the recent introduction of large language models [61].

Additionally, beside its applications to produce item embeddings, transformer models are a powerful architecture to produce recommendation for sequential data, like BERT [62], that was quickly applied in the field of sequential recommendation, in systems like BERT4Rec [63].

– Image processing through convolutional networks: Although not directly addressed in the thesis, as we do not generate recommendations based on or derived from photographs, our application (developed in Chapter 5) required a large number of images, which could be useful to consider in future steps (as in [64]). In any case, the ability to extract features from images and encode them also represents an interesting way to augment the information of the different items we aim to recommend. The representation achieve by deep neural networks-approach implied and noticeable improvement [65], although adding visual feature were not always improved the recommendation metrics [66] and can be consider quite dependant on the recommendation application we are dealing and the system processing the resulting embeddings (fashion applications as [67] are quite common). Also video benefited from convolutional feature extraction [68].

Similar to the previous case, the primary tool for extracting visual features, convolutional neural networks, can also be employed in the pure recommendation process. Examples of this include ConvNCF [69], which utilizes the outer product instead of the dot product to model user-item interaction patterns, or encoding graph structures to transform the recommendation problem into a link prediction one, as proposed by [70] for Pinterest.

2.4.2 Evolutionary algorithms

Evolutionary Computing (EC) [71] is an computational field designed to tackle complex optimization problems by emulating mechanisms observed in biological evolution. In the EC process, a set of artificial entities known as individuals (possible solution of the problem) are designed to explore the possible region of the problem defined, as the keep improving themselves in the search of optimal regions. In order to do so, a fitness function is defined, which is a metric between individuals (or their characteristics) and goal. This function is employed to optimize the problem as it is used to selected and prioritize which individuals or part of individuals will be taken into account. Furthermore, every step in the process (the selection, the competition, etc) can also be defined as a function that admit multiple forms and report multiple advantages and disadvantages.

Evolutionary algorithms have been proven useful to be used in different aspect or phases of the recommendation problem, often overlooked by the Neural networks trend. As stated in [72] evolutionary algorithms are used in balancing multiple quality metrics, group recommender systems, Multi-stakeholder recommender systems, Multi-task recommender systems and Clustering and rule based recommendation systems.

However for the purpose of this thesis and this section, we will focus on the multiple quality metrics, multi-stakeholder and multitask recommendation approaches and describe in which way EC is used in then, as they are the direct domain application of our approach. More specifically we will point out the role that Genetics algorithms plays in them, with examples selecting those that has a foundational meaning to this thesis. More examples in additional depth can be found in [72], [73].

2.4.2.1 Multiple metrics

When we think about specific recommendations, we often do so considering a particular metric that generates that recommendation. In other words, we create a possible evaluation of our system and use it to check if our algorithms behave as expected. For this purpose, we often construct a proximity metric with our results; in the case of recommendation systems, this is accuracy. However, two problems arise: this is just one measure of all possible ones, and even within accuracy, various approaches can be considered [74] (accuracy of ratings, predictions to the user, or even rankings). The second issue is that although this factor has traditionally been considered decisive for the user, it is not the only metric that ensures a good result. Other metrics such as novelty [75], serendipity [76], diversity [77], or item coverage [76] also contribute to the recommendation. This very same problem can be re-applied to different metrics defined to measure the similarity between users and items. Most of the time they have several data sources associated and how to combine them to improve the algorithm is debatable.

Given the plethora of metrics for evaluating and making recommendations, how can we harmonize them? On the one hand, we could use a scaling-based approach, where we aggregate scores into a common weighted function as proposed by [78]–[80]. This is the same approach that others have done to harmonize several characteristics in the profiling phase. Additionally, genetic algorithms have been used with relative success. Works such as [81] suggest the possibility of using an evolutionary approach to produce hybrid recommendation models. In this approach, the evolutionary algorithm optimizes the coefficients contributed by each recommendation system to optimize metrics, including novelty and accuracy (that can be considered competitive metrics). Similarly, other works such as [82] position the evolutionary algorithm after an initial recommendation, searching for combinations that add more diversity to the produced top items. These two approaches are particularly foundational in our thesis.

2.4.2.2 Multi-stakeholder recommender systems

Just as we have different metrics to evaluate systems, often various recommendation systems have different objectives for the stakeholders involved in the recommendation. At this point, consider, for instance, an e-commerce platform where we need to balance the functionality of the platform, that of the users, and that of the sellers, all with clear interests in the process.

In this context, genetic algorithms once again prove effective with a strategy we have seen before: the reordering of possible results, enhancing re-ranking implementations, for example, to optimize the fairness of the recommender [83]. Following a similar idea, the recommendation process itself [84] seeks a solution on the Pareto frontier, attempting to optimize an educational example where the utility of the student and the teacher is related through an equation dependent on a parameter calculated by an evolutionary algorithm.

Another interesting approach directly aims to select the output from three different methods: popularity, matrix factorization, and a fairness measure between interests [85]. This process links back to the previous one, as not only the fairness between stakeholders is needed but also diversity.

2.4.2.3 Multi-task recommender systems

The different stakeholders of the system can force it to act in different ways. Apart from this situation, there are also others where in the same recommendation process, we may have several objectives (or sub-problems) in mind. For example, we may seek a recommendation system that aims to optimize rating prediction but also the retrieval of sets of elements [86]. Alternatively, we may have recommendation as the primary objective and try to achieve some other additional goal, such as additional classification or the generation of explanations [87].

In this area, there is not extensive use of genetic algorithms; they are fundamentally based on the use of scalarization methods where a weighted sum is optimized according to the metrics of each problem. The works presented and the rest of [72] are interesting from the perspective of this thesis. Although works like [87] adopt a strategy of equal weight among features, others like [88] reevaluate those weights, and it is in them where we believe there could be an interesting approach to explainability. Once optimized or altered, these weights provide a lot of information about the system's functioning that can be conveyed to the user, as we will do in Chapter 8.

2.5 Summary and location of our proposal

Deep neural networks have emerged as highly valuable tools for the creation of innovative recommendation algorithms and the refinement of existing ones. Their utility extends to incorporating novel data sources, presenting potential vectors for recommendation across various techniques. While these networks offer notable scalability, a noteworthy drawback lies in their limited interpretability. Interestingly, despite their depth, several studies have highlighted that deeper neural networks do not always guarantee superior results in benchmark tests. This underscores the importance of remaining attentive to novel architectural developments as they unfold.

Turning back to the interpretability challenge, providing insightful explanations or conducting audits for neural-based recommender systems remains a complex task. In our approach, we sought to leverage the extraordinary potential of novel data encoding techniques, employing embedding and attention mechanisms within the content-based segment of our algorithm. This strategic incorporation aims not only to enhance recommendation accuracy but also to address the interpretability concerns associated with deep neural networks.

In addition to these advancements, recognizing the persistent need for both benchmarks and datasets, our method serves a secondary purpose: generating bundled datasets. Once rigorously validated, our datasets hold the potential to significantly contribute to the training of sequential models, further expanding the versatility and applicability of our approach in the realm of recommendation systems.

On the side of the evolutionary techniques, as stated in [73], there are several interesting points for further research on the application of Evolutionary Computing (EC) in recommendation systems, even when combined with more powerful deep learning models.

By employing EC computing techniques, we can address the combination of various parameters, particularly crucial in situations where multiple connections and characteristics must be considered (we will denote these as complex scenarios and further define them in the next chapter). In such scenarios, EC-based systems can obtain solutions that are improvable. We refer to these solutions as those that may satisfy a set of conditions but may not excel in them. This aspect should not be dismissed. Our focus will shift from generating these solutions to improving them, working with already satisfactory possible outcomes.

Specifically, for the application we are going to develop, the solution generation using EC algorithms will help us enhance the stochastic results of our systems. This is particularly useful when dealing with a large dataset and low expected preferences. The randomness in the selection (which can also be manipulated by pre-selecting items that fulfill a certain description) significantly increases the diversity of the models and will be a key factor in applying this system to nutrition.

The main downside, otherwise, will be the time expended on each computation. Being aware of it, our focus will also take a look on different parameter combinations, and change evaluation functions depending on expected complexity (multivariable problems vs. weighted scalarization).

Chapter 3

Recommending on complex scenarios

“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better."
— Edsger Wybe Dijkstra , *On the nature of Computing Science (1984)*

In the previous chapters we have delve into the different approaches for recommendations and the techniques that enable them. However most of them are compared and used in some basic recommendations, that is with films, books or other single items. As we progress in the research field more and more scenarios are being consider suitable for recommendation. Those application use (or should use) a huge range of parameters and deal with situations where there are multiple risk for a non-appropriate recommendation. Our goal with this section is explore this situations and classify them, as they are going to be the main application field of our model proposal. First we will define the two main characteristics of the system (fulfilling like that the Objective 1.1) and then we will present our classification of different scenarios, identifying our main application area (completing the Objective 1.2).

3.1 Complexity and Impact

Two of the characteristics that usually share recommendation systems in those multiple-parameter scenarios are complexity and impact. They are not entirely descriptive but serve to understand many of the problems that motivate the progress of these particular recommendation engines.

- **Impact:** With this adjective, we refer to items that can produce a long-lasting event or effect on the user's life. We point out this characteristic to emphasize that often the decisions that produce a greater expenditure of time, effort, or resources, in general, require more knowledge of the problem. They also need to be solved with a combination of characteristics evaluated in different ways. And finally, they often require a proper trade-off between the pros and cons of a decision. Of course, here we have decisions as to what we listen [89] to or read [90], and the impact of them could be huge (and not always good) [91]. But it is already common to have systems that interact to help us look at Wikipedia[92], what clothes to wear[93], what to eat[94], what to do when we are ill[95]. Those systems are directly affecting important aspects of our lives, our nutrition, the way we dress, the way we could find information, or even study.

- **Complexity:** Coupled with this high impact, we talk about complexity when the problems are composed by several layers of conceptual information that forces some relationships between all the elements of the recommendation process. These could be relationships between items, items and users, users and expert-based knowledge, users with other users and so on [96]–[98]. Those relationships can be diverse, simultaneous and evolve in time.

3.2 Classification

3.2.1 Recommendation Systems in complex scenarios

Although it is easy to understand how knowledge-based systems or constraint-based systems can involve problems of great complexity, the truth is that it is not clear that we can easily classify a certain system as a complex recommender system. Instead, it is the nature of the inputs and the outputs of the systems that really determine the complexity we are dealing with.

This is the main reason why some systems that recommend only single specific items could already be considered as complex scenarios, as they start to take into account numerous variables[99], or we use some of them with a computational demanding extraction process [100]. Think for example on the differences between recommending a certain music album because it is popular or because of the type of music which fits your current context and mood.

There is a great variety of options that raise the complexity of the problem, being one of the most remarkable those recommendations that either must take into account a group of users as input [101] or that must have a structure (either constant or modifiable by the user) as a menu [102]. The introduction of contextual variables also increases the complexity of our recommendation system. Since we must decide how we code those variables, what weight they will have in our system, and what relationship they have with both the items and the users. In addition, the nature of contextual inputs can be very diverse, from a temporal succession originated by the user's interaction with the information system (consecutive purchases [103]) to the location of the user or group of users, or even the problem we are dealing with to find the right solver [104].

On all these occasions the output of our recommendation system could be a single element or depending on the case, a ranking of items. Therefore, another source of complexity appears in our system. Even in the simplest case possible: the creation of a ranking of single elements to recommend, the way to do it, and the information we use for it, can transform the situation into a complex scenario.

Offering a ranking of recommendations implies, in its most basic meaning, providing a final result that has an ordered structure. To do this, we can surely use the item features, but we could use almost any of the characteristics mentioned in Section 3.3. Adding extra data sources can alter our ranking, making it richer, user-dependent, and changeable. And this leads us to a more complex, structured output, as can be seen in [105] and [106].

Often, recommending in complex environments implies offering a group recommendation. In many situations, [107] a single element may not be the result expected by the user. This set of objects to recommend may have a structure that depends on the user or follow fixed schemes, but our system must collect a large amount of information and fill it with the items. An academic path [108] falls into this category.

Perhaps the output should take into account external and sequential conditions, that modifies in real time the recommendations that are being offered. In that specific scenario, we are globally offering a set of recommendations, but from the user's point of view, they are individual items spaced in time, like a music session [109] or travel itinerary [110].

All the research works presented above are examples of Recommendation system for complex scenarios. Either independently in Input or Output or both, these characteristics are relevant to classify and better understand these types of systems. This will help to provide them with a greater and independent conceptual entity and highlight the problems faced by the recommendation engines in these situations.

In Section 3.3 and 3.4 we will formalize these differences by making a classification of the sources that can describe a Recommendation system in complex scenarios. Moreover, we will set the characteristics and subdivisions of each to understand not only the classification but also the current state of research and what problems we face when we seek our Recommendation system interact and help us in highly complex scenarios.

3.3 Input's complexity sources

The primary source of complexity arises from the input our system receives. While managing multiple items and users can increase the difficulty of our problem, it doesn't necessarily impact the complexity and depth of the recommendation. Instead, it is the nature of the data and the required sources that determine whether we encounter complexity.

Therefore, it is crucial to analyze whether our system processes additional information concerning the actual relationships between items, users, and their context. Handling this volume of information can be decisive in a complex scenario. Simultaneously, we must determine the optimal way to represent new data and integrate it with more common sources, such as users' preferences or lists of items.

In some cases, we may encounter a scarcity of samples for each item, utilizing diverse information sources with unique items or distinctive characteristics. However, an abundance of different information sources is not necessary. Imposing additional rules on the input can represent another type of complex scenario. Various rules and constraints can be as restrictive as dealing with a vast amount of data.

In conclusion, we want to highlight scenarios deemed complex due to the main source of information or the nature of the recommended items requiring special processing. Most of the earlier categories might include examples falling into this last one. As new data sources become part of recommendation systems, this category serves as a specific distinction between classic recommendation areas and emerging ones.

3.3.1 Multiple data sources combined

One of the first and most intuitive aspects of recommendations in complex scenarios is the large amount of information needed in many of these cases. But it is not only a question of size but also of data sources. In complex scenarios, it is often essential to be able to extract information from the items in different ways. Choosing which combinations will ultimately be needed for optimal performance is key to making these systems useful today.

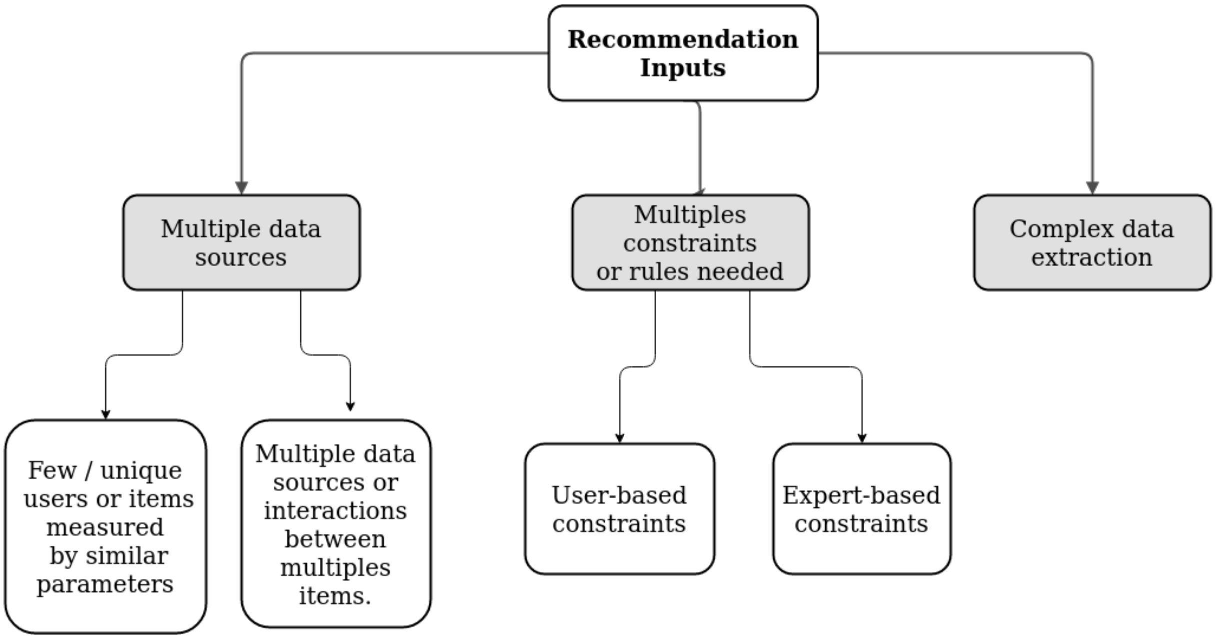

Figure 3.1: Sources of complexity in the input of a Recommendation System, divided and classified by their characteristics

3.3.1.1 Multiple data sources or interactions between multiples items

The first subcategory is also the most intuitive, encompassing recommendation systems that aggregate data from diverse contextual elements or select essential characteristics of items. For instance, works like [111] offer recommendations to users by combining economic information (via a decision tree) derived from process data, involved resources, task duration, and other relevant elements such as task frequencies.

These characteristics often originate from different data sources with varied origins. Contextual data, historical data on relationships and interactions, and the creation of additional labels are recurring themes in this category. Contextual information, particularly in domains like tourism, plays a pivotal role. Approaches vary based on the type of context emphasized and how it is incorporated into the system. Socially-aware systems, which analyze social trust and networks [98], and evaluations of contextual parameters, such as weather and season in processing textual reviews [112], represent different perspectives. Embedding heterogeneous information networks with added contextual information proves useful in cultural consumption [113].

It is noteworthy that the evolution of data labels is a dynamic aspect. Properly labeling different data sources and adapting item labels over time are critical. The field of multilabel learning, especially in streaming services, is highly active [114].

Analyzing multiple interactions among items or how users engage with them can introduce another layer of complexity. There's a recent shift from examining single sources of feedback (only positives or negatives) to processing multi-feedback at various levels. Deep neural networks [115], aggregated neural networks [116], and co-filtering routing mechanisms [117] are employed for this purpose.

The ongoing research focus extends to developing algorithms that effectively combine all necessary data sources. Addressing multiple constraints or relationships forms the basis of works like [118], where the problem's modeling establishes a hierarchy in the relationships considered in the recommendation process. Similarly, [119] proposes a comparable approach using genetic algorithms.

3.3.2 Few or unique users and items measured by similar parameters

This section focuses on systems that operate with a restricted or limited number of items, inherent to their nature. Recommending the best items for users becomes challenging when dealing with a small corpus of unmodifiable objects. In such scenarios, it is crucial to incorporate as much information as possible to understand and identify the user's needs.

Commonly, these systems adopt an approach of developing new sources of information to gain novel insights into the recommended items or the users employing the systems [120], [121]. Machine learning techniques are often employed to deduce these insights from larger datasets, which are then applied to the items within our dataset [122]. The auction system serves as a notable example in this category. Here, the primary focus lies in feature extraction modules from users' historical data and item data, enabling the co-embedding of users and items into a joint latent space [123], [124].

Within this category, there are situations where users can be recommended unique and valuable items. For instance, [125] functions as a recommendation system for paintings, integrating information from diverse sources such as the author's details and the work itself, using manually curated visual features and embeddings from neural networks. Moreover, works like [126], [127] leverage text mining models to unveil semantic connections in item descriptions.

Housing choices also fall under this section due to their significant impact and uniqueness. Approaches like [128] take into account not only the house itself but also factors like proximity to different mobility options. Similarly, [121] incorporates user energy consumption as an additional source of information for recommending energy plans.

3.3.3 Multiples constraints needed

As users, our preferences may be straightforward and adaptable to various variations. However, there are scenarios where, in addition to expressing our preferences, we may wish to impose fixed and unalterable restrictions on the system. These constraints have the power to entirely rule out certain items.

Many of these scenarios are inherently complex. Not only must we satisfy these rules, but we also need to comprehend their influence on the classification of new products. Furthermore, the diverse sources of constraints and their nature, such as expert constraints, context constraints, and user constraints, introduce several new approaches to address them.

In this section, a majority of the works leverage machine learning approaches to infer information or understand how constraints impact sets of items. However, depending on the level of restriction, these approaches might be insufficient to prevent undesirable recommendations [129]. To address this concern specifically, [130] employs a similarity measure-based approach to derive an equivalent set of constraints. This set can either generate potential recommendations or eliminate those that are incompatible with the given constraints.

3.3.3.1 User based constraints

When we accept a certain set of parameters as a constraint it became a metric of how good our system works. For that reason, in this section, there are two types of works. Those that apply machine learning techniques to better understand users' preferences and why some items are extremely non-desirable [131]. And those that select a set of constraints that can rule out several items in the dataset. Recent works show that a fusion of both strategies is needed to improve the results [129].

In certain cases the constraints that the system must face come strictly from the user. Within this Section, we find works such as [132] from the area of economics where the user's position and role determine the set of constraints that are imposed on the problem so that they reflect the user's objective. Therefore, a knowledge-based translation between user objectives and dataset parameters is needed or must be inferred by machine learning techniques.

Another important area of complex recommendation, such as diet, may have constraints such as those shown in [133] where the amount and type of ingredients that the user has at that instant is a strong constraint on the recommendation of his or her food. In these situations, the configuration of the recommendation is indeed a constraint, which rules out certain combinations of items [134].

Depending on the approaches, news recommendations are another rising trend that mostly focuses on what the user wants or does not want. Several works try to translate users' restrictions on news with knowledge-based graphs to uncover relevant entities from both a user's clicked history, pruning irrelevant items [135].

3.3.3.2 Expert Based constraints

Often, when we are in complex environments, as users we may ignore a large percentage of information that could be useful to us when making a decision. For that reason, being able to gather information from experts is a valuable asset (because they potentially have a deeper understanding on what to recommend or even why).

Incorporating this information into the system can be costly and restrictive, but it allows us to substantially improve our understanding of the system. Works like [136] or [137] bring this expert knowledge into relevance for complex scenarios such as improving student employability.

In certain situations, experts in an area may be considered to be those who have more experience in a specific area or are viewed as role models by other users, as in [138], [139] or [140].

3.3.4 Complex data extraction

Finally, we would like to highlight those situations in which, due to the novelty or complexity of the recommendation, we find ourselves in a territory with little scientific production or that is facing several challenges for the first time.

At first, coding movies, books, or other items may be simple if we restrict ourselves to simple and quantifiable characteristics. However, even if our goal is to recommend a single object by extracting a small amount of information from it, we could certainly be in a complex situation. It could be produced if this object can be very useful (high impact objects), needs to be adapted, or it is hard to describe it (computationally). We can highlight works on how to encode items as diverse as works of art through a neural network ([125]) or Wikipedia articles through text embeddings ([92]).

Knowledge graphs are another technique that has been used to help and model how different aspects of one item are preferred by the user. They are used to profile a user representation or encoding items in a most useful approach. News recommendation are benefited from this approach [141].

Many of the categories presented can be combined in different situations. It is common for systems to retrieve information from several sources to improve their performance. To exemplify these relationships, Table 1 shows which possible categories could be assigned to some cited works.

Other scenarios may require not only getting and embedding data from different sources but also improving the data, by eliminating confusing factors such as popularity or noise. For that purpose, casual inference is being used in analysis and filtering data [142]. However, recent works remark on the necessity of a dynamic perspective of this issue [143].

3.4 Output's complexity sources

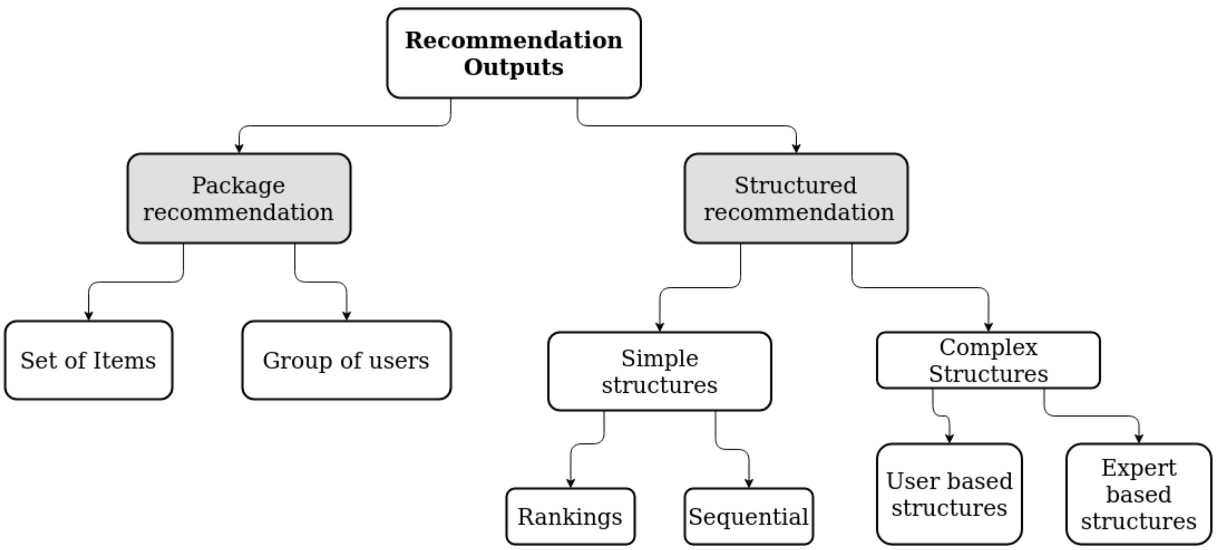

Figure 3.2: Sources of complexity in the output of a Recommendation System, divided and classified by their characteristics

Another major source of complexity comes from the output of the recommendation system in real-life scenarios. This source of complexity can be expressed in different ways (Figure 63.2), but in most complex situations the result of the recommendation is a set of items that hold relationships between them, the context (time-dependent) or the user.

In those scenarios, using datasets containing single items (or even pre-made sets) provides a limited ability for personalization. It is, therefore, necessary to alter and combine options, improve the level of conceptual complexity and, in some cases, create new bundles of items.

The necessity to offer items to groups of users also motivates the appearance of these complex outputs. However, for our output to be considered complex, it should be a real group recommendation (balancing the preferences of the members, aggregating them) not just a collection of independent recommendations.

In this Section, we also wanted to collect those recommendations that have a certain temporal continuity. These recommendations can be simple if they do not add to their analysis the evolution of the user. But in many complex situations, understanding how the user is evolving becomes an indispensable task for a proper recommendation in a complex scenario.

3.4.1 Package recommendation

The packaged/bundle recommendation (that is a set of items i no particular order) is the most intuitive type of recommendation if we are dealing with a scenario where single items cannot satisfy our users. Depending on the recipient of the recommendation we divide this category into two different types:

3.4.1.1 Item group

Recommending a set of items constitutes a solid area of research [107]. In this case, the objective of our recommendation is to offer a set of objects that satisfy different characteristics. This object could be built specifically for the users or be part of a prebuilt bundle. This scenario is one of the most common approaches by various e-commerce mechanics (that quite active in technical research today [144]) but also is expanding to other areas like health [145].

Due to the lack of pre-made user-bundle interactions, several works in this category deal with the problem of bundle creation from item's and user's data. Moreover, the implicit feedback from the bundles (instead of collecting explicit feedback from just their components) it is another source of new research. On both tasks, deep neural networks present a large percentage of the works, where especially graph neural networks [146] and deep attention networks [147] are being used to evaluate implicit relations and user's preferences in bundles, as they let the system embed bundles from items representations.

However, more classical techniques could deal with these scenarios as well. Matrix factorization [148] in which bundles of video games are modified due to the user preferences or evolutionary algorithms [149] that offers exercises and nutritional bundles to stay healthy. The main limitation is again the lack of data from bundles ([148] uses a bundle purchase dataset, but they are not really well spreaded) and deal with the users' feedback on them (or deduce users' feedback from their components).

Aside from recommendation engines, the creation and evaluation of item sets may constitute the middle step that could potentially produce better recommendations [150], helping to improve implicit feedback data from the users.

3.4.1.2 Group recommendation

Another option that requires a different approach is the recommendation of a set of objects whose target is a group of users [101]. This scenario is considered complex when we incorporate in our output the social context of the relationship between users. In group recommendation, we differentiate between models that aggregate user preferences or items scores (memory-based) and those which try to learn and simulate the group's decision process (model-based).

Even if demographic recommendation systems can be considered as a first approach, truth is that group feedback, user interactions and roles are the main challenges. Moreover, the lack of prior data from group recommendations is also a challenge, as the previous works were mainly focuses on the single-user recommendation.

Aggregating user and item data can be found in early works in the area as [151] or [152] where a latent factor approach is used to obtain a group profile aggregated with other latent individual user profiles.

Despite that, deep learning approaches are again used to embed item's and user's information, in order to generate model-based systems. Graph neural networks have been used with success in discovering user roles in the group [153], even when users suffer from a lack of data (new group members) [154]. Assessing the weight of every individual in the group has also been specifically treated with attention mechanisms (attention neural networks) [155]. Transfer learning has also been proved as a viable approach [156] when encoding both group and preferences through neural networks.

3.4.2 Structured recommendation

3.4.2.1 Simple structures

Although many recommendations are able to offer a set of items, sometimes a richer structure is needed. Item characteristics, context, or user preferences can imply a non-direct organization of the products due.

Having a ranking or order to some degree is the simplest choice because we recommend a set of objects, but we do it in such a way that over them there is a relationship that can be either of order or importance according to the meaning of the ranking that is shown. However, properly ranking in any kind of situation it is still an active research topic [157].

This type of recommendation also has a very powerful conceptual entity, and it is common to find works that focus specifically on it like [158] providing a ranking system that uses long-time preferences for music.

- **Time-dependent recommendation**

  In this category, we have also included those time-dependent recommendations. Sequence recommendation can be seen as a primitive structure that is time-dependent. Offering one item after another would have, in the end, a simple order structure. We highlight works like [110] that combines multiple sources of information to offer a sequence travel plan or [106] that use contextual information to offer a re-rank recommendation taking into account the user actions. On sequence recommendation, how we treat the sequence of items is crucial. Although Markov models and recurrent neural networks have proven to be successful, recent trends incorporate historical data with the contextual relationship between the objects, as their category [159]. To deeper analyze this relationship, casual inference [160] and knowledge graphs [161] has also been used with success.

3.4.2.2 Complex structures

This Section is centered on the recommendation of sets of elements that have a final structure. In this case, the structure has greater conceptual complexity than a ranking or order. In particular, sub-items, could belong to different categories within a structure without any of them being superior to the previous one.

This is one of the less-studied cases, as there are some approaches in bundle generation, but complex structures arise in very specific scenarios. E.g., diet recommendation, where some recipes could be recommended but only at certain hours. Furthermore, these scenarios always need to build those structured bundles from simpler items and inherit the difficulties in evaluation stated in the bundle recommendation.

We differentiate two types according to the origin of the structure (or structures) that the user can obtain:

- **Expert-based**

  The first case of this type of structure is when we use expert knowledge to build the structure (or set of possible structures) that can be selected or generated in the output.

  In this case, experts are usually able to identify which structures are the most appropriate or meet the requirements of the problem, allowing to minimize the number of possible parameters that can be personalized.

  Examples of this category are [162] that offer workouts and the diet advice that best suits them, based on their profile information, preferences, and declared purpose. Other diet recommendations as [95] can also appear in this category. Education could be another source of structured recommendation based on expert design (as some courses fit a specific schedule), where traditional techniques such as matrix factorization have been used [163].

- **User-based**

The second case of complex structures are those that can be chosen by the user due to their preferences.

In this recommendation, relationships between the items are crucial, but also between the user and the packages generated and the influence of all the individual items in a package. This information can be modeled with several approaches, individually or both at the same time as [164].

In this category, we can find [165] that can use the information of geotagged photos and trajectory pattern mining to offer an itinerary (which serves as a complex structure in this context) to the user.

Other works allow the user to choose the structure of their recommendation, filtering beforehand from those structures that experts believe can be discarded. In this approach, recommending a long-term diet fits and represents another scenario with interesting applications as shown in [166]. Based on that first work, we develop the main methodology of the thesis and the applications for nutrition and podcasts.

3.5 Application domains

Given the definition of complexity and impact, we can begin to understand those possible areas that can be considered as a complex scenario in which a recommendation system acts. Due to their nature, they can appear in many areas, particularly those areas that accumulate many of the characteristics presented in the classification: a large number of elements, difficulty in extracting relevant information, the combination of these elements in the output, the importance of the user context or constraints on the results. These areas tend to have many papers developing new strategies and technologies for each of these aspects, even combining some of them.

With the papers selected in our study as a starting point, we have described the main areas detected, analyzing what qualities made them a complex scenario.

3.5.1 Fashion

Fashion and our clothing are great examples of a recommendation domain that can range from a buying and selling situation (where we deal with a few factors or classic recommendation strategies), to a highly complex situation.

One of those situations can be dealing with a large set of subjective and contextual clothing characteristics: popularity, color, and style to highlight a few. For those cases understanding the characteristics of the clothes via convolutional neural networks has shown to be helpful [167] [67]. In other cases understanding clothing as a structured with interrelated items [93]. Adding clothing specific data sources as size ranges [168] or context[169] can also help to offer a detailed recommendation.

3.5.2 Business Intelligence and Economy

In general, areas related to economics and business management also offer an interesting challenge for RS. Investments, the choice of different businesses, or the performance of electronic sales pages are important sources of problems. In fact, they take advantage of the filtering information capability Recommendation system have. Outcomes in economic Recommendation system can vary a lot: small stock variations and purchases, packages of companies or funds invest in [170]. E-commerce also establishes a huge source of problems in this area, looking for a great user experience that leads to higher sales. Other examples of current applications are [171], [111].

3.5.3 Housing

Choosing a house to spend from a few months to a large part of your life can be a particularly important challenge for RS. Again, this is a situation with a multitude of possible factors to take into account and a potentially very high impact on our lives. In the recommendation of rental and sale of homes, we are again faced with an important contextual factor. The relationship of the house with the stores, centers, and the rest of the possibilities of the area is a determining factor for its acquisition and therefore, it is convenient to take it into account.

In this case, it is not usual to have great feedback from other users since we find practically unique elements in almost all its aspects (except perhaps, opinions about the city or the neighborhood in which they are). We can find works like [172] that tries to reduce biases by hybridization of CF and CB recommendations, and [173] with the aim to analyze and incorporate subjective housing preferences.

3.5.4 Tourism

Tourism is a great source of recommendation problems considered complex by the exponential amount of information that can be taken into account by our systems. Leaving aside the fact that we are, again, in a situation that can have a great impact on our lives (as would live 2 weeks in another house), the context becomes an essential tool.

The time of the year, the conditions of the place we are going to visit or why is this place relevant [174], are contextual elements of great importance. However, evaluating the feedback from other users and the visual information of their experiences can be an equally important source of information. For this reason, most recent approaches focus on getting social data thought natural language processing [112][175] and photos, seeking to build a powerful context-aware [110] tourism recommender.

3.5.5 Education

Education is another area that has great relevance in our lives. In this case, much of the complexity lies in how we try to understand and process the information in the courses and how it fits into the interests of the learner [176]. In addition, the rise of distance education has made it possible to access numerous courses, so we find at least two different paradigms depending on the situation. Find a recommendation that matches your interests from an extensive catalog [177] in online learning. Or find courses that match your interests or academic projections [178] with the limitations that a university or a closed list of universities implies. Besides, some new approaches centered on how to collaborate in academic research are appearing [177], opening other promising areas of application within the academic community.

3.5.6 Health and exercise

Finally, it is worth highlighting the power of these systems in medicine. Recommendation system are increasingly been applied to give personalized advice on health and fitness, taking into account users pathologies and how they alter the standard recommendations. Although we could include nutrition in this Section, food recommendation does not have to specifically take into account healthy patterns, and it has unique characteristics that deserve to be treated separately.

In terms of health and exercise, we can highlight the diet advice in [179] with the integration of many factors and the use of human microbiota. Moreover, [180] focuses on recommending physical activities to a very specific type of user (with arterial hypertension) to improve their status. In a similar line but using Recommendation system as a tool in medicine, [95] offers an interesting approach to analyze cases of heart disease helping doctors.

Finally, works as [97] stand out for its application in the field of mental health, an area that can be greatly benefited from this kind of personalized approaches.

3.5.7 Nutrition

Nutrition is another source of major recommendation problems that can be considered complex scenarios [181]. On the one hand, we can talk about individualized and unique recommendations, where the main source of complexity is obtaining information about the context and the dishes. Obtaining nutritional information about the dish through recipes or photos [182] and combining it with the user's tastes [64] would be an example.

From that, we could assess the problem of group recommendation in which we must manage more information and offer a different output. Finally, we have the most complete recommendation possible, where the user receives a daily or weekly menu or any periodicity in time, where synergies between the smaller items (dishes), more complex structures (daily intakes) and the expert recommendation may appear [183], [94]. To all these, we must add the need to find a balance (sometimes) between all the aspects described and the healthiness of the recommendation with its impact on the health of the individual.

In this type of applications, we find an interesting relationship between what the user likes and what they should consume. Unlike many recommendations based solely on preferences, adding dietary requirements introduces a complex environment where having different mechanisms (one for adaptation to the restrictions based on evolutionary algorithms) and another for refinement (based on content recommendation to improve the likeability of the menu) can make a significant difference, allowing us to deal with the diverse natures of many parameters. This is why nutrition will be one of the key objectives of our system, created to provide recommendations in these scenarios.

Chapter 4

Methodology

"Take a method and try it: If it fails, admit it frankly and try another. But above all, try something."
The New Deal- Franklin D. Roosevelt*, 1932*

In this Chapter, we are going to present our proposed algorithm (see Figure 4.1) for complex structured recommendations. During the text, some brief examples will appear to clarify the concepts discussed. We will expand these applications upon Chapters 5 and 6 .

It is worth noting that definitions in the next subsections can be adapted to fulfill the specific requirements of our problem. This flexibility is necessary due to the different nature of the problems approachable with this model. We offer a secondary figure 4.1 where all variables depicted in this chapter can be easily consulted.

4.1 Sources of data and main objective of the study

One of the crucial parts of any recommendation process is the acquisition of the necessary data. In our case, the data sources needed may vary in nature, extent and characteristics, but they always correspond to three essential elements in our system.

Table 4.1: Summary of the variables used in the model.

Variables	Explanation	Notation
Preliminary Items	Items that will form the structured recommended object	$\mathcal{I}$
User	Users' characteristics from those that will request a recommendation	$\mathcal{U}$
Additional data sources	Additional information (knowledge/expert - based)	$\mathcal{K}$
Structured built items	Items built by combinations of the items in $\mathcal{I}$ and the constraint from user and other knowledge	$\hat{\mathcal{I}} = \mathcal{L}(\mathcal{I},\mathcal{U},\mathcal{K})$
Parameters that intervene in the generation of the structure	Set of constraint that define the structure and the subitems we can choose from $\mathcal{I}$	$\mathcal{E}(\mathcal{U},\mathcal{K})$
Parameters of the genetic algorithm	Parameters related to the constraints that must be reached after the genetic algorithm	$\mathcal{V}(\mathcal{U},\mathcal{K})$
Pool of eligible candidates	Subset of definitive items where our genetic algorithm will search for optimization	$\mathcal{H} \subset \mathcal{I}$

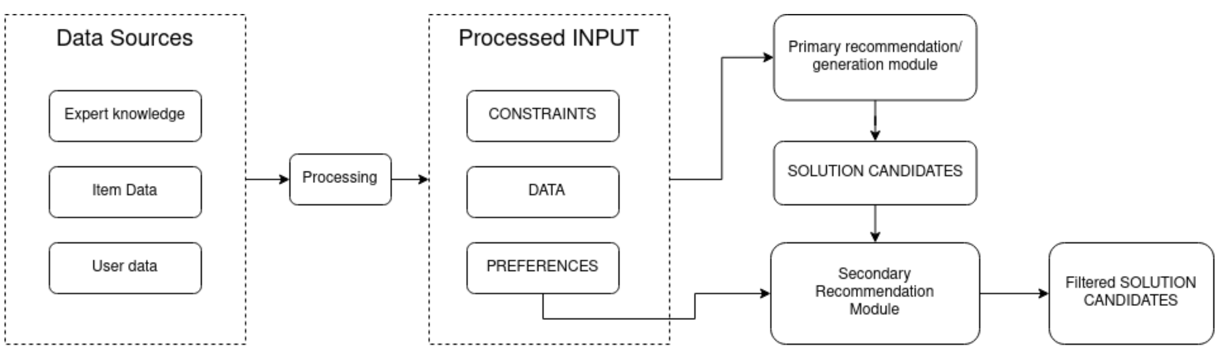

Figure 4.1: Basic scheme of our recommender system.

The first two, as in any recommendation, are the space of Items to recommend and the space of Users. We denote the spaces of these elements by $\mathcal{I}$ and $\mathcal{U}$ respectively. The elements of each of these spaces belong to p-dimensional, q-dimensional space and are encoded by an array of parameters, $i = [i_1, \ldots, i_p], \forall i \in \mathcal{I}$ and $u = [u_1, \ldots, u_q] \forall u \in \mathcal{U}$. Moreover, we need an additional source of information that contain expert knowledge $\mathcal{K}$

about our problem. This last source of data that can affect any phase of the recommendation, is formulated as a set of parameters or constraints denoted by $\mathcal{K} = \{k_1, \dots, k_l\}$. For example, a possible $k_1 \in \mathcal{K}$ could be a constraint that all selected items of $\mathcal{I}$ must fulfill. In both $\mathcal{U}$ and $\mathcal{K}$ we have all the information related to the structure of our future recommendation and the relationships between items (that is, for example number of possible subsections or relationships with items).

The final items recommended are not those elements of our initial data source $\mathcal{I}$. Items in $\mathcal{I}$ are just a lower conceptual unit with no structure and no relationship between its parts. In short they are the components with which we will make our final recommendation. In some scenarios, we may obtain a rich source of data that contains already-built complex objects. But in reality, this is far from usual, and not realistic at all dealing with personalized problems. From now on we will denote the elements of $\mathcal{I}$ as sub-items in order to differentiate them from the definitive Items.

Thus, our first goal is the generation of the definitive Items which built our final recommendation. In the next subsection we elaborate further on their generation, but we identify here that the set of these items obeys the form

$$\mathcal{L}(\mathcal{I}, \mathcal{U}, \mathcal{K}) = \{\{i^1, \dots, i^n\}, \dots, \{i^m, \dots, i^k\}\}$$

where i^x represents a certain $i \in \mathcal{I}$ for any superscript.

$\mathcal{L}(\mathcal{I}, \mathcal{U}, \mathcal{K})$ represents the combinations of different elements of $\mathcal{I}$ that are dependent on the sub-items in $\mathcal{I}$ which will form it , the constraints proposed by the user$\mathcal{U}$ and the characteristics proposed by other knowledge-based sources $\mathcal{K}$. For simplicity, we will denote it as $\hat{\mathcal{I}} = \mathcal{L}(\mathcal{I}, \mathcal{U}, \mathcal{K})$ from now on. Note that this set may not be a subset of the set of parts of $\mathcal{I}$, up to this point, unless otherwise stated, repetitions of elements of $\mathcal{I}$ could occur in it.

With these sets we can now define the objective of our system: Given an initial set $\mathcal{I} = \{1_i, \dots, n_i\}$ we want to obtain a subset of structured combinations of elements of $\mathcal{I}$ denoted $\hat{\mathcal{I}}$ for each user in $\mathcal{U}$, so that it optimizes the needs and some preferences of each user as well as the parameters from $\mathcal{K} = [k_1, \ldots, k_l]$.

To better understand this definition, we can think of a classic recommendation problem such as personalized nutrition. In this case, we would have the recipes as the elements of $\mathcal{I}$ with their respective quantities of ingredients. $\mathcal{U}$ would be the set of users of the application, each one with different characteristics such as weight, age, or what pathologies they may suffer from. In addition, $\mathcal{K}$ would be the set of all the parameters that a healthy diet needs to satisfy (percentage of kilocalories, fat, or seasonality, amongst others). The objects that will appear in $\hat{\mathcal{I}}$ will be groups of recipes (associated with a meal structure) depending on the user and the nutritional information we have. All of them will conform a daily menu recommendation.

4.2 Creation of the item space

The final item space $\hat{\mathcal{I}}$ is the space in which the logic of our recommender system operates. Obtaining a space of this complexity can be a hard and time-consuming task due to difficulties in finding and processing useful databases. For this reason, our system uses data with a lower conceptual level as bricks to build the final dataset.

This approach will favour the adaptation of the space $\hat{\mathcal{I}}$ to the initial information and to those parameters of $\mathcal{K}$ and $\mathcal{U}$ involved in this part.

This task works through an evolutionary algorithm whose objective is to obtain a subset of elements from $\hat{\mathcal{I}}$ that satisfies $\{u_{1+x}, \ldots, u_{q+x}\} \subset u$ (that is the set of parameters of the users that act as constraints over $\mathcal{I}$) and $\{k_1, \ldots, k_{1+n}\} \subset \mathcal{K}$ (the set of parameters of $\mathcal{K}$ that act as constraints over $\mathcal{I}$).

Within these parameters, there are two distinct types:

- On the one hand, we will denote $\mathcal{E}(\mathcal{U}, \mathcal{K})$ as those parameters that intervene in the generation of the structure. Although they affect characteristics such as convergence or the speed of the algorithm, they do not intervene directly in the evaluation of that candidate. We can think of them as parameters that only admit two values, whether they are in the final item or not, and therefore we will eliminate any element that does not have them or, if necessary, we will mark them as a condition to be fulfilled before evaluating the fitness of these objects. This step can also be considered as a preprocessing step if the constraint affect solely to the items that are considered in the creation of the bundles. However, these constraints can also affect to the whole structure. For example: certain elements are sufficiently spaced within the recommendation or the proportion of them follows a rule (as on per bundle).

- On the other hand, denoted as $\mathcal{V}(\mathcal{U}, \mathcal{K})$ are those parameters on which the fitness functions of the genetic algorithm act and which admit adjustments and variations during the generation process. An example of this type would be the calculation of the total kilocalories to be consumed by certain users, taking into account the sum of all the kilocalories in the recipes selected for a daily menu.

During the first part of the generation step, we will randomly create elements of $\hat{\mathcal{I}}$. This set, which we will call $\mathcal{H} \subset \hat{\mathcal{I}}$, verifies $\mathcal{E}(\mathcal{U}, \mathcal{K})$ by construction. We then need to describe the fitness function that evaluates which changes of the elements bring us closer to satisfying the parameters in $\mathcal{V}(\mathcal{U}, \mathcal{K})$.

For example: in a dietary advice recommender, this step should produce a menu by filling in the selected structure (such as breakfast, lunch, and dinner). This structure is part of the parameters $\mathcal{E}(\mathcal{U}, \mathcal{K})$. In this set, we can also have information about allergens or intolerances, which rule out which elements of $\mathcal{I}$ cannot be within the structure.

In $\mathcal{V}(\mathcal{U}, \mathcal{K})$ we have characteristics such as "expected total purchase price", "seasonality", "amount of an ingredient", or "amounts of different nutrients", all of them are that the recommendation should have. However their amount have different ranges of acceptability, i.e. total kilocalory consumption in the day may be more or less than 100 kilocalories without affecting the user if the daily average stays close to the objective.

4.2.1 Fitness function

In this chapter we will focus on those parameters that appear in the set $\mathcal{V}(\mathcal{U}, \mathcal{K})$. We denote $\mathcal{V}_{\mathcal{U}}$ as the set of parameters from the user personalization and $\mathcal{V}_{\mathcal{K}}$ as the set of parameters from alternative data sources. Then we define the fitness function of our genetic algorithm as:

$$\begin{cases} \mathcal{F}(x) = W_{\mathcal{V}_{\mathcal{U}}} f_{\mathcal{U}}(x) + W_{\mathcal{V}_{\mathcal{K}}} f_{\mathcal{K}}(x) \\ \text{for all } x \in \mathcal{H} \subset \hat{\mathcal{I}} \end{cases} \tag{4.1}$$

Where $W_{\mathcal{V}_{\mathcal{U}}}$ is the weight associated to user's parameters and $W_{\mathcal{V}_{\mathcal{K}}}$ is the weight associated to extra data sources and expert knowledge that is taken into account.

The functions $f_{\mathcal{U}}$ and $f_{\mathcal{K}}$ are defined:

$$\begin{cases} f_{\mathcal{U}}(x) = \sum_{\forall u \in \mathcal{V}_{\mathcal{U}}} w_u g_u(x) \\ f_{\mathcal{K}}(x) = \sum_{\forall k \in \mathcal{V}_{\mathcal{K}}} w_k g_k(x) \\ \text{for all } x \in \mathcal{H} \subset \hat{\mathcal{I}} \end{cases} \tag{4.2}$$

with

$$\begin{aligned} g_u(x) &= (\tfrac{v(optimal_u)}{optimal_u})(u - optimal_u)^2 \\ g_k(x) &= (\tfrac{v(optimal_k)}{optimal_k})(k - optimal_k)^2 \end{aligned} \tag{4.3}$$

in this case $optimal_{k,u}$ represents the optimal value to be reached by k, u. In addition, $v(optimal_{k,u})$ is defined as :

$$v(optimal_{k,u}) = optimal_{k,u} * C/(k, u_{range})^2 \tag{4.4}$$

Where k, u_{range} are the specific range of acceptance for k, u and C is the global optimum of the criterion. If we establish that our convergence criterion is acceptable when the value is less than 10, this function changes the scale of our acceptance function, in order to punish solutions that fall outside this interval. For example, for an objetive that should be as close as possible to 100, the original function would be: $g(x) = \frac{1}{100}(x - 100)^2$. With this formula, values of x between 69 and 131 fall below 10, however, if we want more control on its behaviour, we can use the Eq. 4.4 factor. We can use it to shift the scale of the function and produce that only values between 90 and 110 fall below 10, with $g(x) = \frac{1}{10}(x - 100)^2$. These functions allow us to treat a set of constraints that are expressed as ranges of values. Inside those ranges, the most optimal (with the smallest score) value will be the midpoint of that range. These ranges are a translation on what is defined by user's or expert's opinion. It is however important to notice how weights affect this additional factor, more useful when evaluating multivariable functions than scalarized ones.

For simplicity, we have omitted those parameters that act as modifiers, but, for completeness, we develop them here:

Some parameters in $\mathcal{K}$ o $\mathcal{U}$, do not generate modifications directly in $x \in \mathcal{I}$, they will affect other parameters as $g'_k : g(f_k(x))$, $g'_u : g(f_u(x))$ for $x \in \mathcal{H}$. An example of these parameters in the nutrition application would be a reduction on the limits of the daily cholesterol consumption due to the existence of a pathology.

In the discussion below, it is assumed that these parameters are applied in the functions and that, by construction, they have the form described above.

Thanks to the fitness function we can summarize the optimization problem describe in this subsection as follows:

$$\begin{cases} Min_{x\in\mathcal{H}} \|\mathcal{F}(x) = W_{\mathcal{V}_\mathcal{U}} f_\mathcal{U}(x) + W_{\mathcal{V}_\mathcal{K}} f_\mathcal{K}(x)\| \\ \text{where } \mathcal{H} \subset \hat{\mathcal{I}} \end{cases} \tag{4.5}$$

4.2.2 Other functions associated to the generation process

The rest of the associated functions that make up the evolutionary algorithm are often associated with different characteristics of the generation process. Thus, the classical crossover, mutation and selection functions are determined according to our objective concerning the variation of the solutions, the avoidance of local minima, and the structure of the elements of $\hat{\mathcal{I}}$. In our case they will be defined as follows:

- **Selection function:** The selection functions are based on the fitness function. These functions select those objects from $\hat{\mathcal{I}}$ with a higher score to be the candidates to expand their genome in the next generation. In situations where we are interested in a large variety, such as those presented in the paper, we have opted for a probabilistic function based on subsets of n elements. In our case, this function is able to produce high diversity. At the same time, it is capable of generating solutions with huge potential in the next steps (which can be obtained later with crossover and mutations).

- **Crossover function:** In our applications, the objects that make up the chromosome are directly related to the structure of the elements that are recommended. Thus, during the crossover, elements of $\mathcal{I}$ that are in a similar position relative to each other (and relative to each of their structures) that are part of $\hat{\mathcal{I}}$ can be swapped. The frequency and arrangement in which they do so will depend on what features we are looking for. In our work, we have opted for a classical crossover that splits two solutions into two different sections and exchanges one part with the other.

- **Mutation function:** Mutation functions alter elements of $\mathcal{I}$ within the structure. They are therefore an interesting parameter to adjust in terms of their form and action. These functions can act on concrete or general subsets of the final structure. For example, in a nutritional recommendation system, the structure has a series of subsets related to each meal and filled with elements of our base dataset $\mathcal{I}$. These elements also have a quantity associated with them. The mutation, in this case, could be done (with a given probability) on a block of food, half of the structure, a particular meal, or a particular size.

4.3 Solution refinement

The above procedure has allowed us to create a set of items adjusted to the selected parameters from the user and other additional sources of information. Furthermore, this set of solutions has a selected structure and it items preserve the relationships imposed during the generation process.

This set is also highly diverse. Its variability is not intrinsic to the model and is produced by the genetic algorithm. Parameters such us the stopping condition, the number of generations, the initialization of the population, the mutation rate or the types of mutation, can affect and generate different combinations. These features can also prevent the convergence of the algorithm to its global optimum, offering local sub-optimalities that can be refined in later phases and are sufficiently different from the previous ones.

We also make special emphasis on the initialization of the algorithm, as it can offer another customizable parameter and relegates much of the weight to the initial dataset. One of the downsides of these approaches is that it forces us to have a sufficiently well-annotated and diverse initial dataset or at least varied in those characteristics where we want to have huge variability.

At this point, thanks to all the previous conditions, we have a set of solutions that satisfy a series of imposed restrictions (to a greater or lesser extent, but always between the marked limits). This allows us to obtain a space of solutions or a subset of it, on which we can apply more specific algorithms of **RS**, but without depending on many constraints.

In the nutritional case, the solutions obtained so far meet the main characteristics associated with the healthiness of the model and the user's objectives, as these directly affect the final amount of nutrients. We can now focus on selecting from these alternatives, the one that best aligns with their preferences and tastes.

This improvement, using the same measures, will be done in three steps as described in Figure 4.2. First, we will have an evaluation phase where we analyze those elements that conform to our solutions. Through them, we will be able to know if any of the elements of $\mathcal{I}$ are marked by the user as non-preferential (or close to a non-preferential item). Notice that this may happen in this phase, since we are talking about preferences that may not be fulfilled for the sake of a recommendation adapted to the user's needs and constraints.

Once we have analyzed those elements of our structure that have room for improvement, we will look for elements in our database that are similar in the desired characteristics ($\mathcal{V}(\mathcal{U}, \mathcal{K})$), but that are not marked as non-preferred by the user (either because

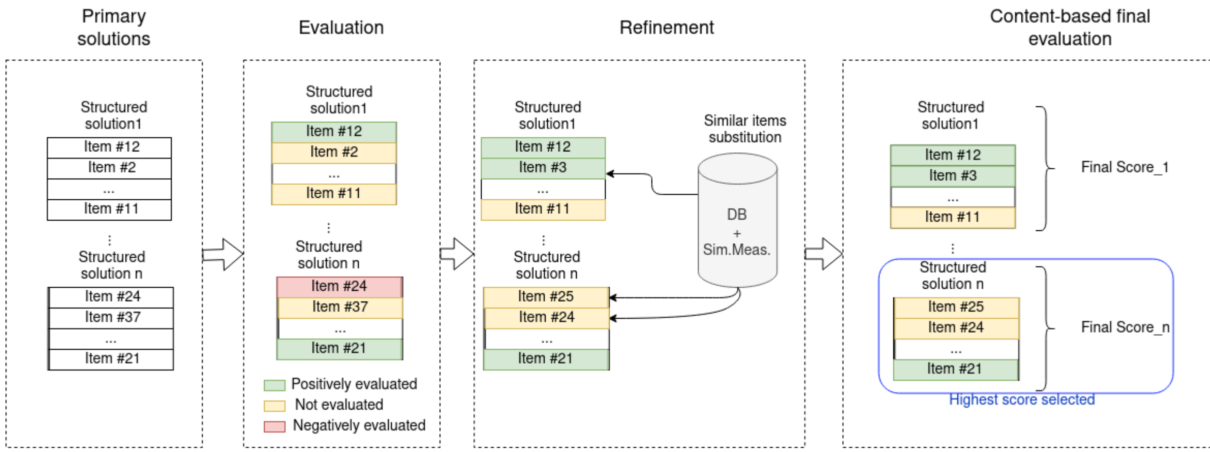

Figure 4.2: Solution refinement process.

they are marked as liked or because they lack a negative evaluation).

We make as many changes as possible (or up to a specific changing rate) without altering our initial structure and move on to the last evaluation.

Finally, we select one of the generated and refined recommendations. We compute the scores associated with the user's preferences (at the conceptual levels we have chosen) to give a weighted average of the final score of the menu. An application example can be found in Figure 5.1.

4.4 Content based system

For both the evaluation and refinement steps, a content-based system is the natural step to take. This is due to several characteristics:

- First, the objects we are recommending already have a high degree of conceptualization. They are composite objects with a lot of data and relationships between them, so an approach based on exploiting precisely that information is very relevant.

- The content-based system is robust to cold start, which may be the case of several items in our database.

- Finally, we must not forget that the starting dataset can be very diverse (or the system has diversity conditions when it is generated), which means that, if we generate the initial population on which to apply the second phase, we will find unique objects, which would cause a user/item matrix to be very sparse.

These characteristics, while relevant, can be addressed with other types of systems, or even with hybrid systems. On the other hand, in order to use other methods such as collaborative approaches, we would need another source of information in our system (i.e. a user database and their interactions). We discuss this type of improvements related to the problems studied in the following chapters.

4.4.1 Similarity Measures

This section is highly variable depending on the area of application. However, as we have already mentioned, due to the large amount of information we have on the products, several similarity measures associated with different aspects can be established.

Each element that makes up our final items has a set of characteristics, denoted as $i_1, \ldots, i_p$. These characteristics can range from specific values to multivalued or Boolean labels (possesses a quality or does not possess a quality).

This fact allows us to transform these elements into vectors of a multidimensional space on which to apply different metrics. These metrics can be evaluated from the user's own profile, or by collecting those elements that the user has rated positively. Which one we apply will depend on our objectives and the nature of our data.

In particular, given that we are working with items created from the combination of another set, the characteristics we use can be a very high dimensional space. Therefore, the similarity metrics used in natural language processing, adapted to this kind of similar nature, are applicable in our approach.

Moreover, as we have been pointing out throughout the text, the presence of bi-valued or Boolean variables may also be common. This is why if we assume that k is the number of features of the elements of $\mathcal{I}$ appearing in the sub-set of $\mathcal{H}$ selected for the recommendation process, we highlight two similarity measures:

- For those variables that encode information of a different nature, operations such as the cosine similarity provide a metric that works optimally with large amounts of data even though the data are sparse:

$$\begin{cases} Sim_{Cos}(x,y) = \frac{\sum_{i_j} x_{i_j} y_{i_j}}{\sqrt{\sum_{i_j} x_{i_j}^2}\sqrt{\sum_{i_j} y_{i_j}^2}} \\ i_j \text{ define the j-th parameter of x or y in } \mathcal{I} \end{cases} \tag{4.6}$$

.

- For those scores that are evaluated with a characteristic array composed of bi-valued variables, we can also use Jaccard's Similarity (or consequently, Jaccard distance), especially useful if we seek to design a metric where we detect items whose composition is very similar (whether it is a desirable quality or not). We could transform some parameters into a binary set or use a generalized version of the Jaccard index for real valued vectors:

$$Sim_{Jacc}(x,y) = J(x,y) = \frac{|x \cap y|}{|x| + |y| + |x \cap y|} \tag{4.7}$$

  If not only do we want to account for the number of similar features, but also if there are some features that are more important than others, and we want to reflect that through our metric, we can resort to a weighted Jaccard similarity as:

$$\text{Weighted Jaccard}(x,y) = \frac{\sum_i \min(x[i], y[i]) \cdot \delta(x[i], y[i])}{\sum_i \max(x[i], y[i]) \cdot \delta(x[i], y[i])}$$

  where $\delta(x,y)$ is the Kronecker delta function that will be 0 in those categories that we do not want to consider. The weights allow to emphasize or de-emphasize certain elements based on their importance in the similarity calculation. This is especially useful in scenarios where not all elements are equally relevant for the comparison (as a user profile where certain categories are more important than others).

Once again, we highlight that other measures of similarity or distance are easily usable, and this will depend on the nature of our data and objectives whether we use one or the other. Lastly this evaluation can be taken into account with the fitness evaluation from the first part as another important weighted factor. A final score, weighted sum of all the score used, is set as the final threshold to surpass.

Chapter 5

Application: Nutritional Recommendation

"You guys know what will make this night even more awesome?... TACOS!"
Marvel Comics Deadpool*,*

5.1 Nutritional recommendations

Recommending a long-term diet is a highly complex yet critically important problem. This complexity primarily stems from the potential impact that a poor recommendation can have on our health, as well as the societal and cultural implications associated with the act of eating.

An ideal long-term food recommendation system must strike a balance between personalized food preferences/interests and nutritional/health requirements. Considering food preferences is crucial since it is unrealistic to expect individuals to drastically alter their diet or tastes based on suggestions. However, factors such as pathologies, allergies, or other medical conditions affecting health (such as intestinal microbiota [184] or genomics [185]) must also be taken into account.

Moreover, it is imperative to correctly understand the intrinsic characteristics of food and how they influence a user's predisposition to choose a dish. For example, a user might opt for a substantially different dish for a midday dessert compared to a main dish for dinner.

Finally, it is essential to comprehend the interplay between physical activity, pathology, and the user and how they impact dietary choices. Whether the objective is transitioning from one physical state to another (e.g., weight loss) or addressing deficiencies caused by a pathology (e.g., anemia), the system must provide diverse recommendations tailored to different user objectives and states.

Earlier works on these topics predominantly focused on addressing user preferences when recommending, as seen in restaurant scenarios. Subsequently, works where health plays a prominent role emerged [186]. Alongside these, a relevant body of work on extracting information and ontologies from recipes, textual and visual data references [187] evolved, becoming a central part of food recommendation systems.

In recent years, more recommender systems have aimed to address both user preferences and nutritional values, recognizing the importance of healthy nutrition in our lives [188]. The focus has shifted towards dealing with multiple constraints and preferences. Works like [189] consider several parameters while classifying ingredients. More recently, [190] concentrates on specific health issues to recommend food for CKD patients, emphasizing fulfilling only particular parameters for them.

Evolutionary algorithms have shown promising results in this context. Works such as [149] and [191] explore the use of evolutionary algorithms to generate recommended bundles or sets (including an additional source of recommendation with physical exercises).

However, our approach specifically focuses on recipes, providing a context to the ingredients. Recent approaches have adopted a multi-objective optimization approach [192], reinforcing the utility of evolutionary algorithms in food recommendation. While these systems learn recipe patterns, most plans and nutritional interventions follow a more standard approach. Our system is tailored for such scenarios, offering full control over the relations obtained when creating the initial population of menus. Additionally, this improves the time performance of the algorithm. Simultaneously, the system allows us to explore sets of parameters that may not produce feasible combinations, providing experts with a starting point for enhancing the recipe database or informing users if their pattern could be problematic (e.g., if a user is physically active but consumes small isolated portions of food).

Therefore, the nutritional application of our recommendation system is an evolutionary algorithms-based application for the creation and refinement of daily menus capable of incorporating constraints and rules at different levels of restriction. These rules can be specified at the precision level of various nutrients and remain robust against user choices.

5.2 Experimentation

5.2.1 Data sources

Personalized nutrition problems usually have a set of different data sources. In our case we are going to follow the notation described in Chapter 4, classifying them into three categories:

- $\mathcal{I}$: The space of items will be created based on the [193] and [194] databases that have data about recipes and their nutritional values. Most of the time, the recommendation processes use ingredients to give nutritional advice[195]. But we do not understand food in our daily lives like this. For that reason we define our primary sub-item space as a set of different recipes. Recipes also have extra information about its seasonality or cooking methods that can be used in the recommendation. Thus, recipes in $\mathcal{I}$ will form the menu that will be recommended to users.

- $\mathcal{U}$: The user space stores all the users of the application with the characteristics that have a direct impact on the system. The user's preferences as well as physical and health status are part of these parameters.

  Other parameters such as the usual quantities of certain recipes consumed by the user, the structure of his menu or their fitness goals must be taken into account during the generation process.

Finally, in this category, we will also find parameters that define the user at a biometric level, such as BMI, weight or height, necessary to calculate the energy expenditure throughout the day.

- $\mathcal{K}$: Expert knowledge will be the most diverse set of parameters in this problem. Within this set, we will have information about the choices the user can make regarding some of its parameters (i.e. the different menu structures supported). These parameters also encode the nutritional values associated with the healthy amount of each nutrient. These amounts will be affected according to the user's condition, physical activity or pathologies. Any additional information from nutrition experts will be stored in this set.

5.2.2 Initial creation of complex items

5.2.2.1 User profiler

This first part is already an element of the recommendation process, since some user characteristics $\mathcal{U}$, as well as the constraints of $\mathcal{K}$, are taken into account in this creation.

From $\mathcal{U}$ we collect all the parameters that build a biomedical profile of our user (with data such as BMI, weight, height or the amount of body fat).

On the other hand, with respect to $\mathcal{K}$ we collect two types of constraints. Those that directly relate the healthy amount of nutrients the user should have, according to their data. And those that intervene in the menu generation: type of cuisine, seasonality, possible menu structures.

It is particularly relevant to point out how the structure of the menu is selected. In this case the final structure of the recommendation depends both on the user and on the parameters of $\mathcal{K}$. In this model we make use of expert knowledge and propose common food patterns. This generates a first set of structures that we can use. However, in order to make the system flexible, it is up to the user to choose which of these structures is the most appropriate for their daily diet.

All the parameters selected are included in the final item space considered by our system. This space is made up of daily menus composed of dishes, which follow a structure and relationships set by the user and the expert knowledge.

5.2.2.2 Structured Item creation

For the specified schema, we now have an already inferred structure that should be filled with different sub-items. These will constitute the final recommendation space, encompassing both these sub-items and the structure. For this initial phase, we have obtained optimal values for different evaluation functions from expert sources.

With this information, we configure a fitness function centered around varying quantities of nutrients, weighting kilocalories as the most important objective, followed by macros as the second, and micronutrients. During this first generation, we consider the convergence of the method and the variability in terms of parameters to choose. This also led to modifications of the weighted values due to variations in the dataset itself (taking into account existing recipes where proteins posed more challenges to attain).

This task works through an evolutionary algorithm whose objective is to obtain a subset of elements from $\hat{\mathcal{I}}$ that satisfies $\{u_{1+x}, \ldots, u_{q+x}\}$ y $\{k_{1+z}, \ldots, k_{l-z}\}$.

- $\mathcal{E}(\mathcal{U}, \mathcal{K})$ (as those parameters that intervene in the generation of the structure) From a questionnaire the structure is designed. It admits data from the different main meals, certain types of food that should not be considered (due to dietary restriction or allergies) and the type of food habits it has: one or two dishes, dessert and any other additional consumption such as wine or bread.

- On the other hand, denoted as $\mathcal{V}(\mathcal{U}, \mathcal{K})$ are those parameters on which the fitness functions of the genetic algorithm act and which admit adjustments and variations during the generation process. In this case that would be the different nutrient levels considered healthy for an average male. We first considered a set of 11 micronutrients where healthy ranges are stated along with dietary consumption pyramid (if the food is mainly composed by meat-based ingredient, pant-based, and others). seasonality.

Now on the genetic algorithm stated for the creation:

5.2.3 Fitness function

We define the fitness function of our genetic algorithm following the equation from last chapter:

$$\begin{cases} \mathcal{F}(menu) = W_{\mathcal{V}_\mathcal{U}} f_\mathcal{U}(menu) + W_{\mathcal{V}_\mathcal{K}} f_\mathcal{K}(menu) \\ \text{for all } menu \in \mathcal{H} \subset \hat{\mathcal{I}} \end{cases} \tag{5.1}$$

Where $W_{\mathcal{V}_\mathcal{U}}$ is the weight associated to user's parameters and $W_{\mathcal{V}_\mathcal{K}}$ is the weight associated to the different nutrient we are considering.

The functions $f_\mathcal{U}$ and $f_\mathcal{K}$ are defined:

$$\begin{cases} f_{Nutrient}(menu) = \sum_{\forall k \in \mathcal{V}_{Nutrient}} w_{Nutrient} g_{Nutrient}(x) \\ \text{for all } menu \in \mathcal{H} \subset \hat{\mathcal{I}} \end{cases} \tag{5.2}$$

Specifically, in the nutritional application we select two different set of weights, one for the relative important of the nutrients (from kcalories, macronutrients and micronutrients) and other for the relevance of different micronutrients between them. For example, giving more weight to fiber often leads to and increase in vitamin intake, but not the contrary, so fiber score is weighted more than other vitamins. Finally for every nutrient we create the function:

$$g_{nutrient}(menu) = (\frac{v(optimal_{nutrient_{goal}})}{nutrient_{goal}})(nutrient_{quantity} - nutrient_{goal})^2 \tag{5.3}$$

in this case $optimal_{k,u}$ represents the optimal nutrient quantity to be reached by that specific menu. And $nutrient_{quantity}$ represent the amount of that nutrient that appears in the menu evaluated. In addition, $v(optimal_{k,u})$ is defined as :

$$v(optimal_{k,u}) = optimal_{k,u} * C/(k, u_{range})^2 \tag{5.4}$$

Where we evaluated the nutrient quantities and select their range of acceptance. For example kilocalories can be automatically accepted if fall between $+/-100$ difference between the expected one. This evaluation as said earlier must be adjusted along with

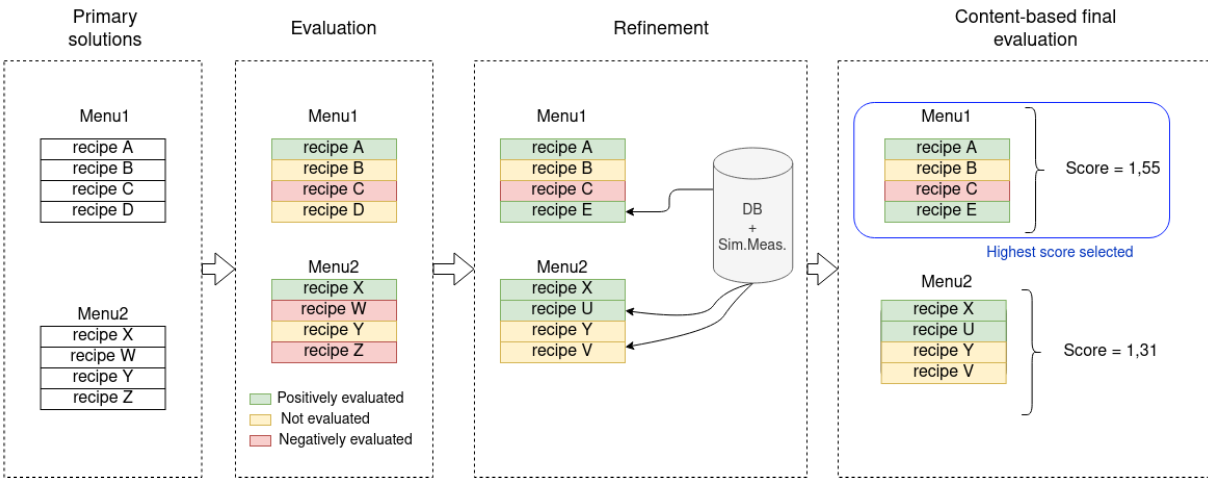

Figure 5.1: Solution refinement phase for the nutritional application.

the weights and the specific dataset we are using.

We proceed with this approach in which the different quantities of elements are weighted according to their importance and the order in which they need to be adjusted, instead of using a multi-evaluated function with an enormous number of parameters. This choice is made for the computational ease of its calculation, given that initially, this system was designed for integration into a mobile app. Furthermore, this summation allows us to interact with weights to address the interpretability of the system. Additionally, by managing weights, we can handle normalization among variables.

5.2.4 Secondary Item adaptation

5.2.4.1 Solution refinement and recommendation

At this point we have a structure filled with sub-items from $\mathcal{I}$ that verify the main constraints selected for the first part of our recommendation process.

From here we move on to the second phase where we refine these solutions according to a more classical approach, based on (mainly) user preferences. A graphic summary can be found in Figure 5.1.

This procedure consists of two parts. On the one hand, the personalization of the generated menus (corresponding to Evaluation and Refinement in Figure 5.1) and on the other hand, the choice of the menus that best fit the user's preferences (corresponding to Content-based final evaluation in Figure 5.1).

Firstly, given the dishes that are associated with our menu (i.e. the elements of $\mathcal{I}$ that appear in $\mathcal{H}$), we can choose those that are negatively valued by the user. These items must be replaced by a similar dish. For them, we will use euclidean similarity, looking for those dishes that are very similar nutritionally speaking, but that are marked as indifferent or liked by the user rather than disliked. It is possible that we are unable to find a realistic improvement in this structure. If this is the case, we still keep the solution for the last phase.

Finally, from the set of secondary recommendations, we will evaluate which is closer to the user's tastes (using the similarities define in section 4.4.1). In this case we recover some information from $\mathcal{I}$ to develop three types of evaluated and weighted scores:

1. Dish category: Those menus that have a higher frequency of labels common to the dishes valued positively by the user: meats, salads, fruit, etc. We will compare the set of tags from a menu with the set of tags from the user and derive a distance from it.

2. Fitness functions: Fitness of the menus evaluated. We will use a cosine similarity metric to compare set of normalize nutrients.

The weighting of these characteristics not only reflects the relative importance of each of the similarities compared to the rest, but can also be related to other types of information. In the nutritional case, similarity could be scored more positively if it is found in a main dish within the menu structure, as there is a correlation between the overall quantity and significance of the main dish compared to smaller side dishes or desserts. Another option could be to assign extra value to the fitness function evaluations.

5.3 Evaluation

5.3.1 Results

For the first evaluation of the model we followed a offline test based on different user profiles. Those where selected in the Stance4Health project. The Stance4Health project (Smart Technologies for personalized Nutrition and Consumer Engagement) (S4H) is a project funded by European Union's Horizon 2020 research and innovation program. Our main focus was to show that this approach can achieve reasonable results for a realistic user in a nutrition intervention. The ability to produce healthy menus has already been validated by nutritionist in [196] for different micronutrients. However In this section we will show that the model can obtain healthy menus from different configuration, showing its robustness to different patterns, sizes or restrictions.

Moreover we will offer novel insights on the secondary recommendation module. For this task we choose a standard healthy male human, with a normal IMC, moderately active which result in a General Metabolic Rate of 2000 kcal a day. For this user we have design the following scenarios, along with and explanation on which situation they can be useful:

- S0:User does not take anything in the morning and single recipes for lunch and dinner. Its portions are standard portions.(Adapt to user behaviour)
- S1:User make 3 meals a day
- S2: User make 4 meals a day
- S3; User make 5 meals a day (Calorie surplus)
- s4: User demand fish as the main ingredient of lunch and dinner (ability to adapt to certain type of dish, this is necessary to provide not daily, but weekly recommendation, where we can incorporate Mediterranean patterns in the diet)

Course	**Dish**	**Name**
Breakfast	Main	Skimmed milk with wholemeal bread
Mid-morning snack	Main	Skimmed yogurt and toast with jam
Lunch	First	Vegetable lasagna
Lunch	Main	Artichokes with Iberian Serrano Ham
Lunch	Dessert	Watermelon
Dinner	Main	Brown rice with leek
Dinner	Dessert	Flat peach

Table 5.1: Example Menu for user.

- s5: User demand meat as the main ingredient of lunch and dinner (ability to adapt to certain type of dish, this is necessary to provide not daily, but weekly recommendation, where we can incorporate Mediterranean patterns in the diet)

- s6: User is allergic to milk

- s7: User demand higher portion of dishes (have a good appetite, vegan diets)

- s8: User demand the smallest quantity possible. (Troubles eating, inability to cook)

An example of the output menu can be seen in figure 5.1.

We divided the tests in different stages, one for the nutritional adjustments of the menu (which is the main focus of the first part) And one for the second part, centered in the content based recommendation system.

For the evaluation of the genetic evolution process we run a set of 30 different menus modifying the algorithmic-centered parameters, those are: number of generations, fitness functions, initial population, type of crossover, probability of crossovers. Based on the first set of tests, we chose the parameters described in table 5.2 for the next part of the evaluation. For the threshold we base our results on the scores of the nutritional validation [196].

GA parameter	Value
Nº of Generations	300
Initial population size	15
treshold	300
Tourtnament type	Elitism 3 random selection
Probability of mutation	0.5
Crossover	Half menu exchange

Table 5.2: Summary of the values used in the Genetic Algorithm part.

With this set of values we then run the primary recommendation system throught 50 different days in each scenario. In this process we start using the most basic fitness function declare in the previous section taking only the data from the kcalories. We re run this experiments using a more advance definition of the fitness function taking into account kcalories and macronutrient levels. Finally, we produce the last batch of experiments using the most advanced fitness function. As we can see in figures 5.2-5.7 along 200 steps in the generation process, all fitness functions stay below their acceptance threshold, but the variability of data force us to reach between 250 and 300 generations to stay below the accepted level in all situations. These evaluations use all the three possible fitness functions built with the nutrient levels. It is worth noting that we are using only the last one for evaluating the population, where all the parameters are encoded as the formula presented in Eq- (5.1). The two upper plots represent how these less complex fitness functions built as Eq. (5.1) but with less nutritional parameters decay as well as the most complex one is used.

On the second part of the recommendation we proceed to change the menus according to the preferences in food types the user give us. Every recipe in the dataset has two set of characteristics: one with a general pyramid-type of food and other ontological related to that one related to the type of dish (over 35 different categories). We use that information and the user preferences to calculate the distance between the user preferences in terms of food types and the user disfavors in term of the menu food types. For that we use a Jaccard similarity type as boolean vector of length 35 (has this food type, do not have it) and evaluate to both types of preferences. Ideally the distance of like categories will

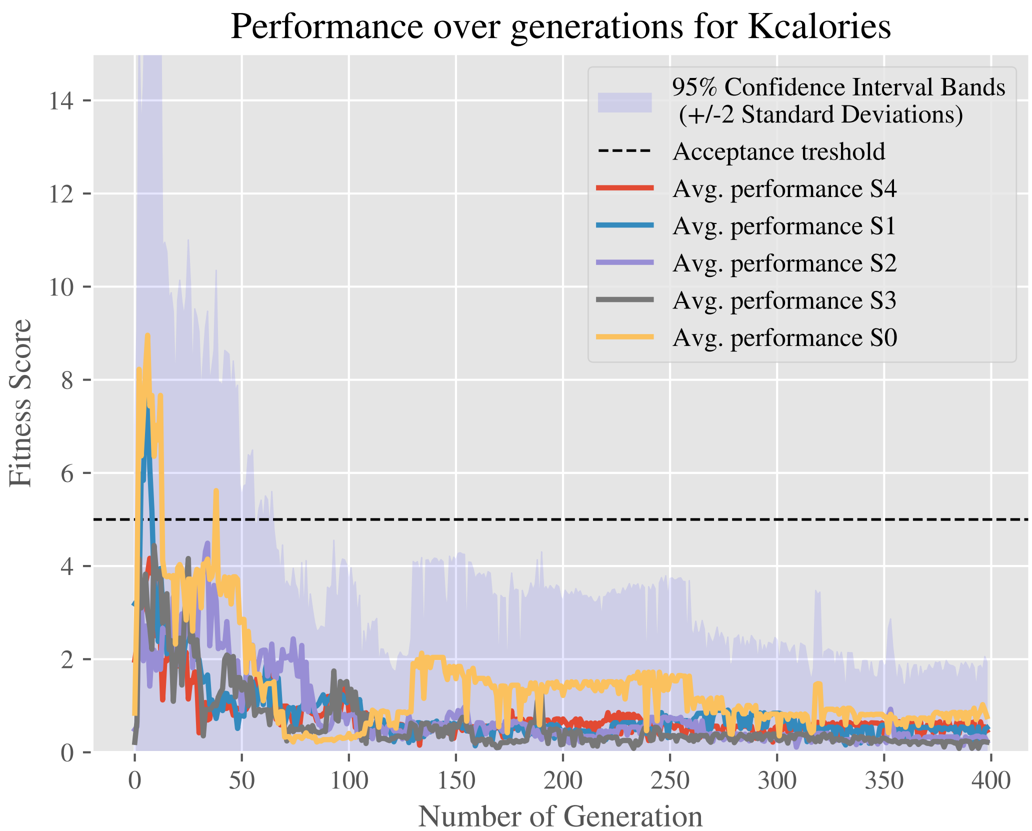

Figure 5.2: GA convergence scoring only the Kilocalories levels in the scenarios S0,S1,S2,S3,S4

decrease while the distance to the disliked types will increase.

However we still have to take into account the nutritional score. Throughout the preference metrics we will add up the fitness score function to penalize those menu changes that diverge too much from the recommendation (due to the absence of similar dishes that match the preferences).

We perform this second part with a random assignation of like and dislike dishes, and two more oriented one: only like meat and no vegetables and the opposite: plant-based diet with little to no-meat. The variability of the distance from the liked-by-user pattern and disliked-by-user pattern is shown in Tables 5.3,5.4 and 5.5 . Kcalories and all nutrient oscillations are also shown.

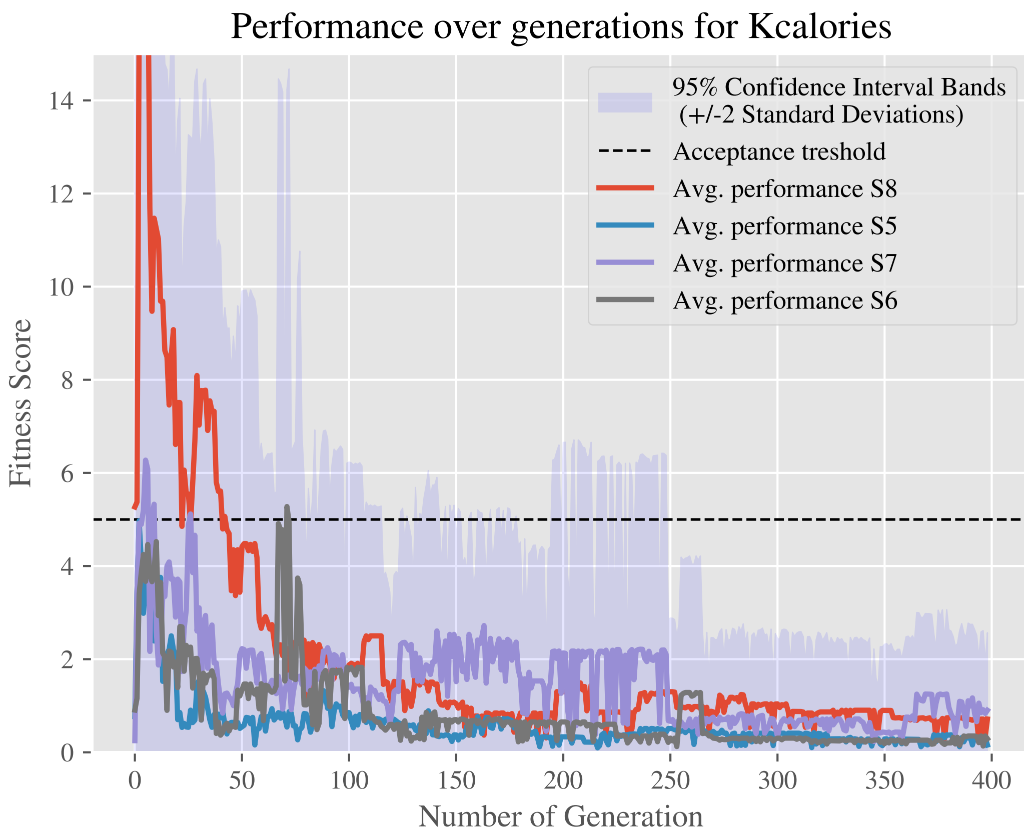

Figure 5.3: GA convergence scoring only the Kilocalories levels in the scenarios S5,S6,S7,S8

For this section we forced at least one change in the menu. However if we cannot obtain a healthy alternative, no changes where accepted. On overall we get an decrease in the like metrics while a increase in the distance form the dislike options. Along this changes, most of the nutritional level stayed close to the original ones and the objective one.

Another visual representation of those changes can be seen in Figures 5.8, 5.9 and 5.10, those are and additional visualization from the ones shown in [197], obtained replicating the experiment using the python package presented in Chapter 7. In those we can visually see who the second module increase the variability of the primary recommendations as can be seen in kilocalories, which are close to the original goal, but have higher variability 5.8.

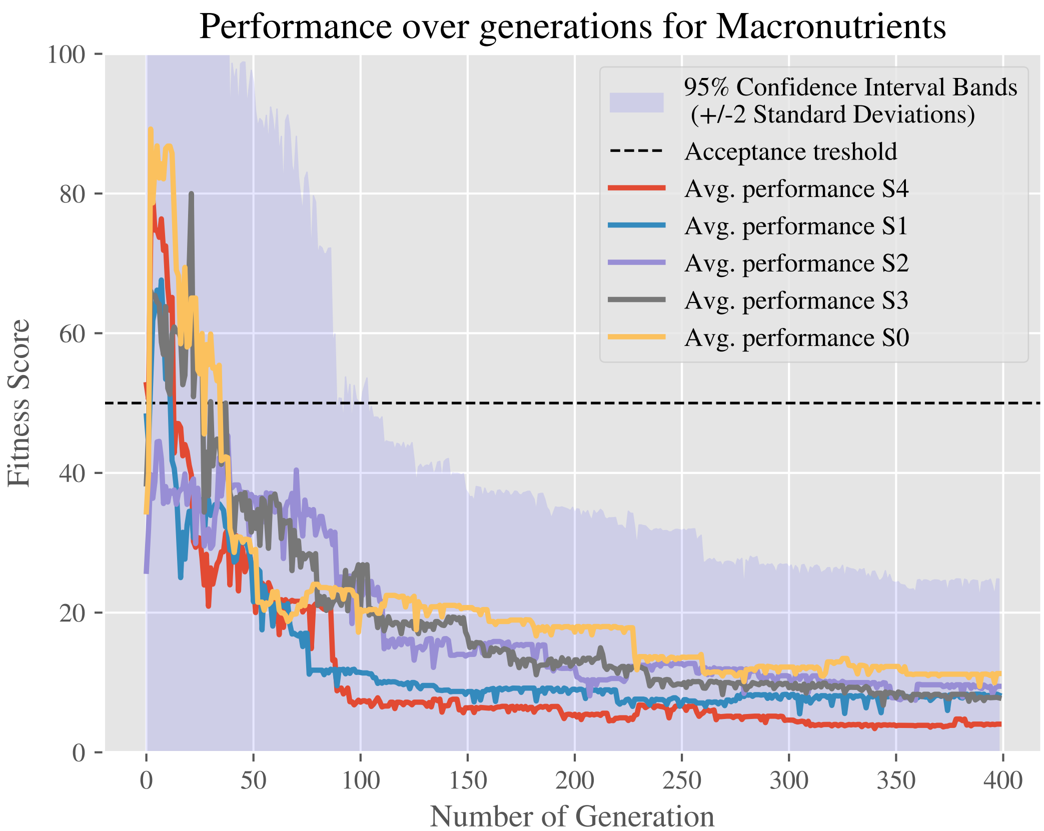

Figure 5.4: GA convergence scoring Kilocalories and macronutrient levels in the scenarios S0,S1,S2,S3,S4

Also, we can see how several menus are refined to obtain a modification that results in an increase of the distance from the dislike pattern, while approaching to a like pattern 5.10. Finally, other quantities as carbohydrates are usually affected by the module, aligned with previous experiments, by lowering their amount in the final menus as seen in 5.9. On average, a conservative approach in the secondary module preserves the objectives marked in the first step of the process.

5.3.2 Continued usage

One of the greatest challenges in the process of creating and recommending menus is the user's engagement with these menus. The reason is straightforward: the effects of a

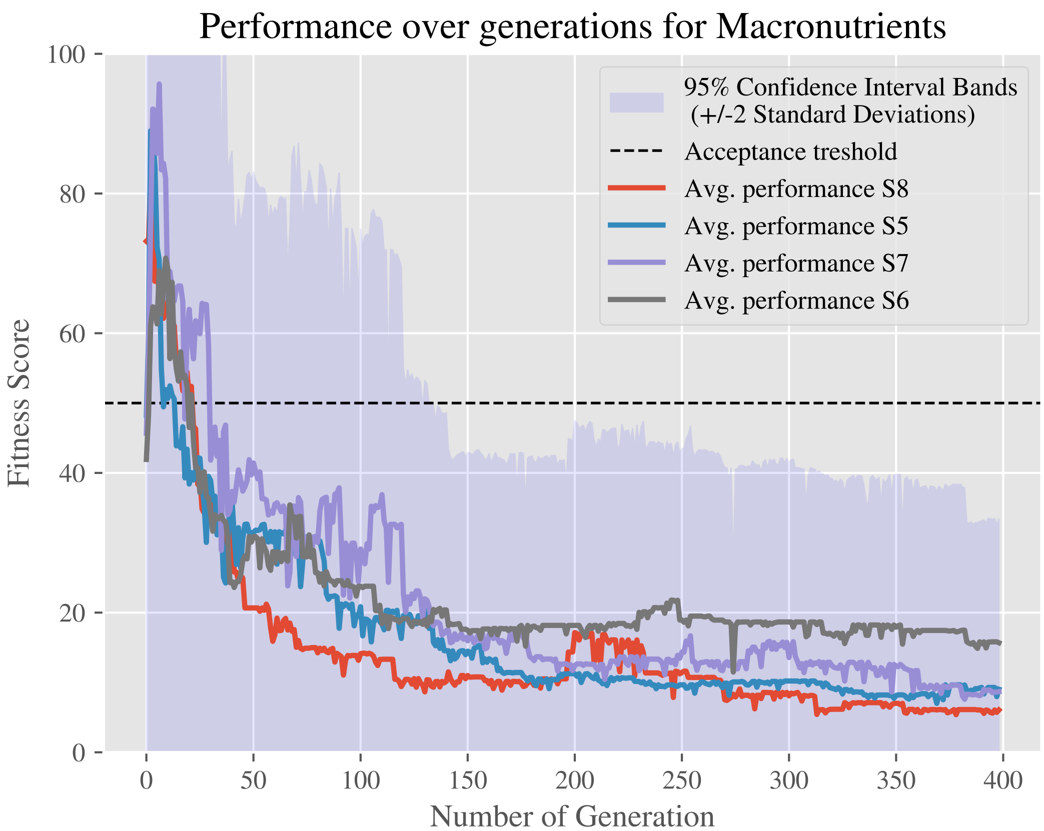

Figure 5.5: GA convergence scoring Kilocalories and macronutrient levels in the scenarios S5,S6,S7,S8

single healthy menu that the user consumes occasionally will be hardly perceptible. And not only perceptible to the user but also to their health. Any improvement in the user's health due to following healthier or more tailored dietary patterns will come with regular adherence to the recommendations. This approach poses an interesting problem; we need the user to feel inclined to follow the proposed recommendations and to interact with the system for an extended period of time as it gains a better understanding of the user. In summary, we require the user to have continuous usage of this recommendation system. This presents three different challenges, one focus on the technological aspects and two related to the psychology of the user.

Regarding psychological objectives:

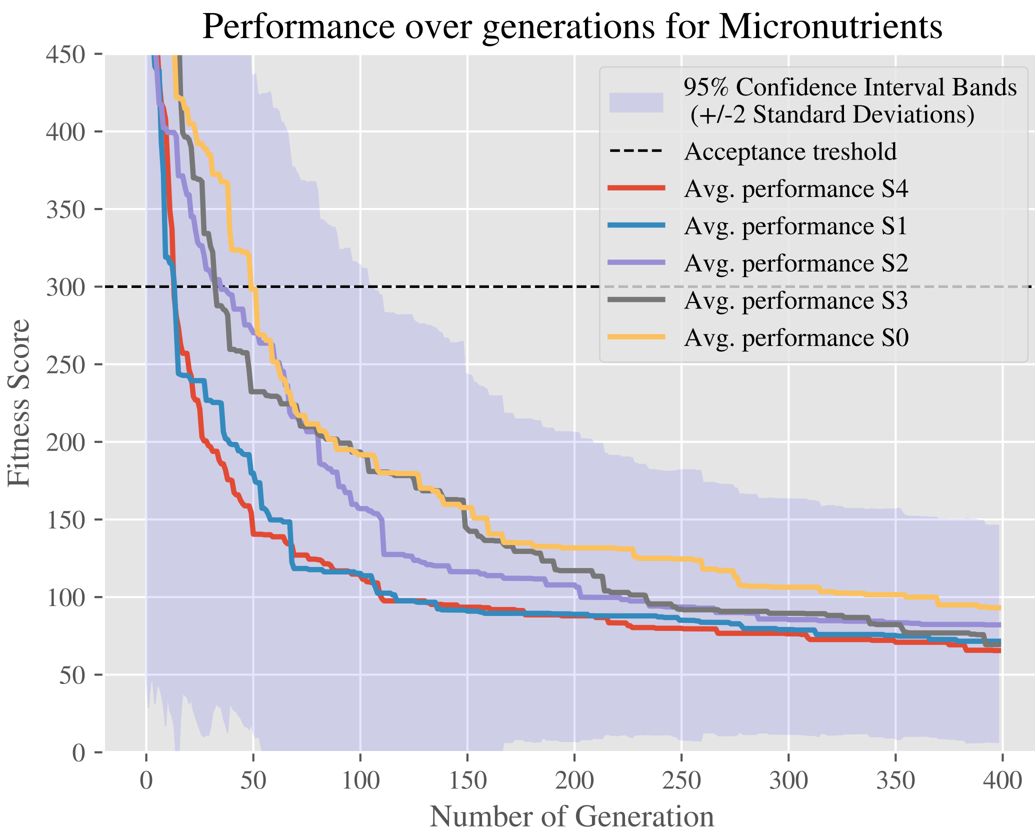

Figure 5.6: GA convergence scoring the most complex function (Kcal, macro nutrients and micronutrients) in the scenarios S0,S1,S2,S3,S4

1. We need the user to feel comfortable using our recommendation system and interacting with it. This is crucial not only for the user to follow the recommendation but also for the user to indicate their preferences, dislikes, and changes made to the proposed recommendations. All these interactions create frictions that become more significant if the user does not understand how to interact with the recommendation system or its associated app.

2. We need the user to be motivated by the app and feel that the recommendations align with their goals. Improving the explainability of the system is essential as a means for the user to remember why they are in this process.

Both aspects are thoroughly studied in Chapter 8 of the thesis, establishing how they

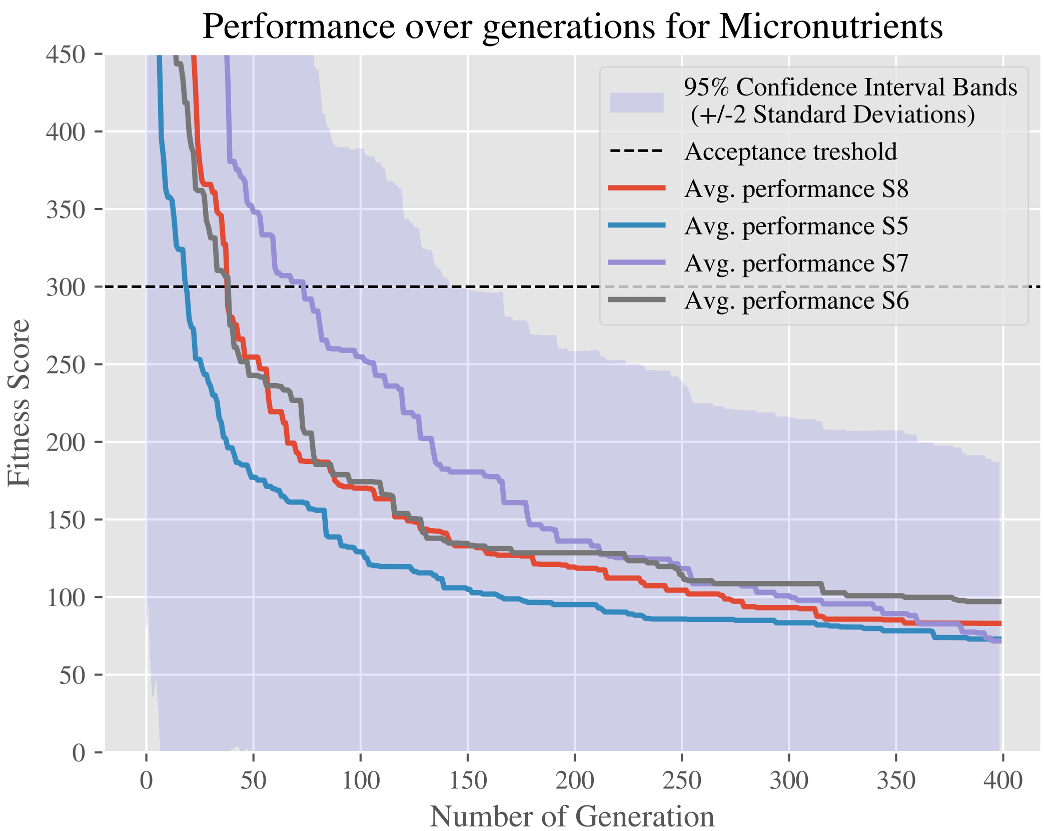

Figure 5.7: GA convergence scoring the most complex function (Kcal, macro nutrients and micronutrients) for the scenarios S5,S6,S7,S8

are researched, how they were evaluated in the European project, and what computer-based proposals we offer for their improvement.

Regarding technical objectives: We need to launch the algorithm several times, at least once weekly, to generate daily menus and possibly to readjust or reevaluate the recommendation if the user deviates from the menu. This scenario was encountered in the European project during the development of an application that integrated a system based on what is described here. In the following section, we elaborate on the technical aspects of this implementation.

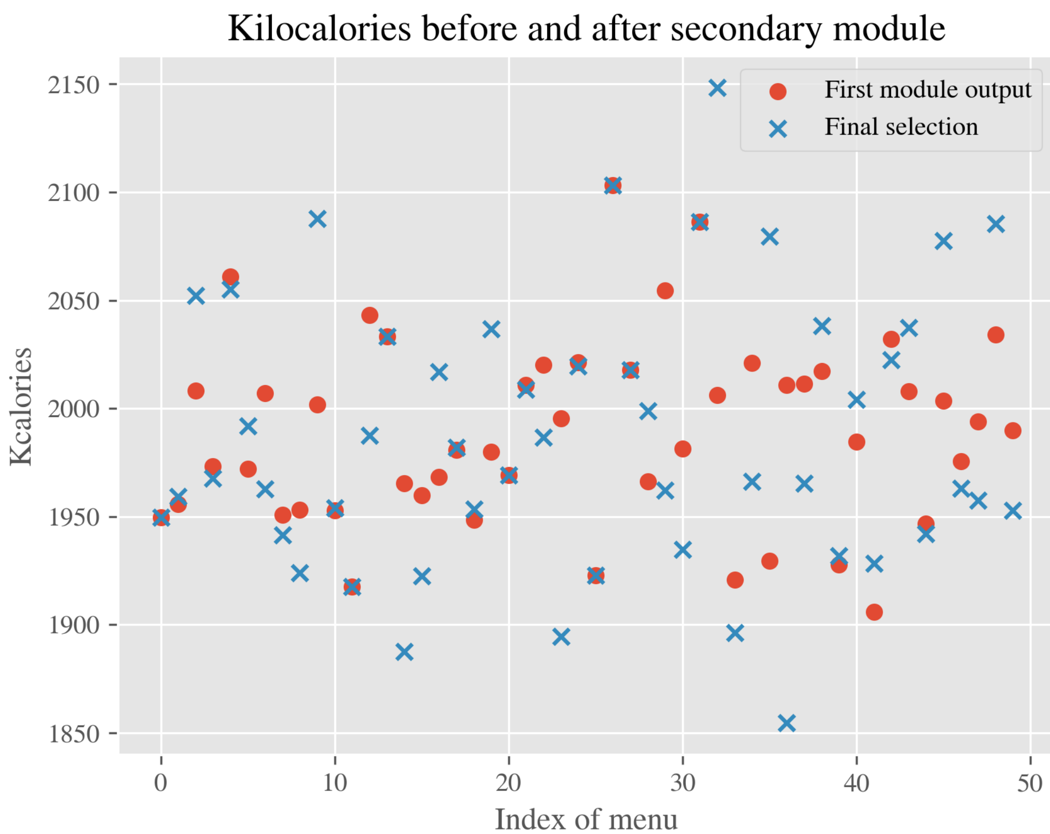

Figure 5.8: Kilocalories variation before and after preference module across 50 menus.

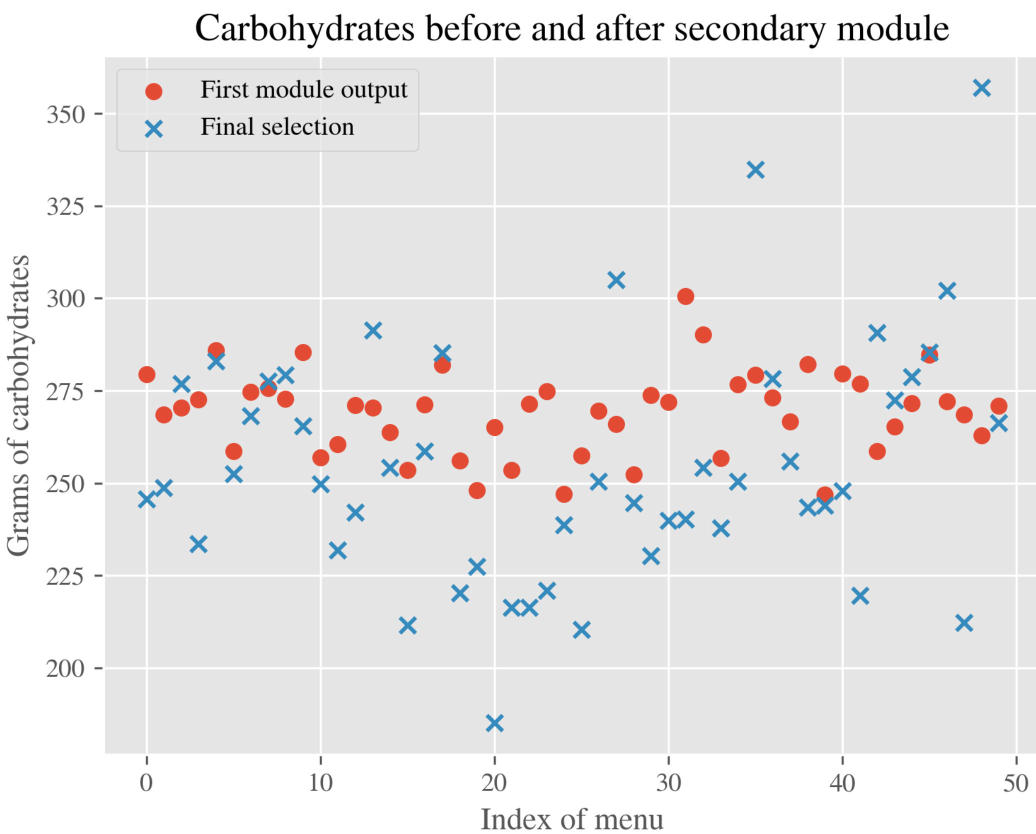

Figure 5.9: Carbohydrates variation before and after preference module across 50 menus.

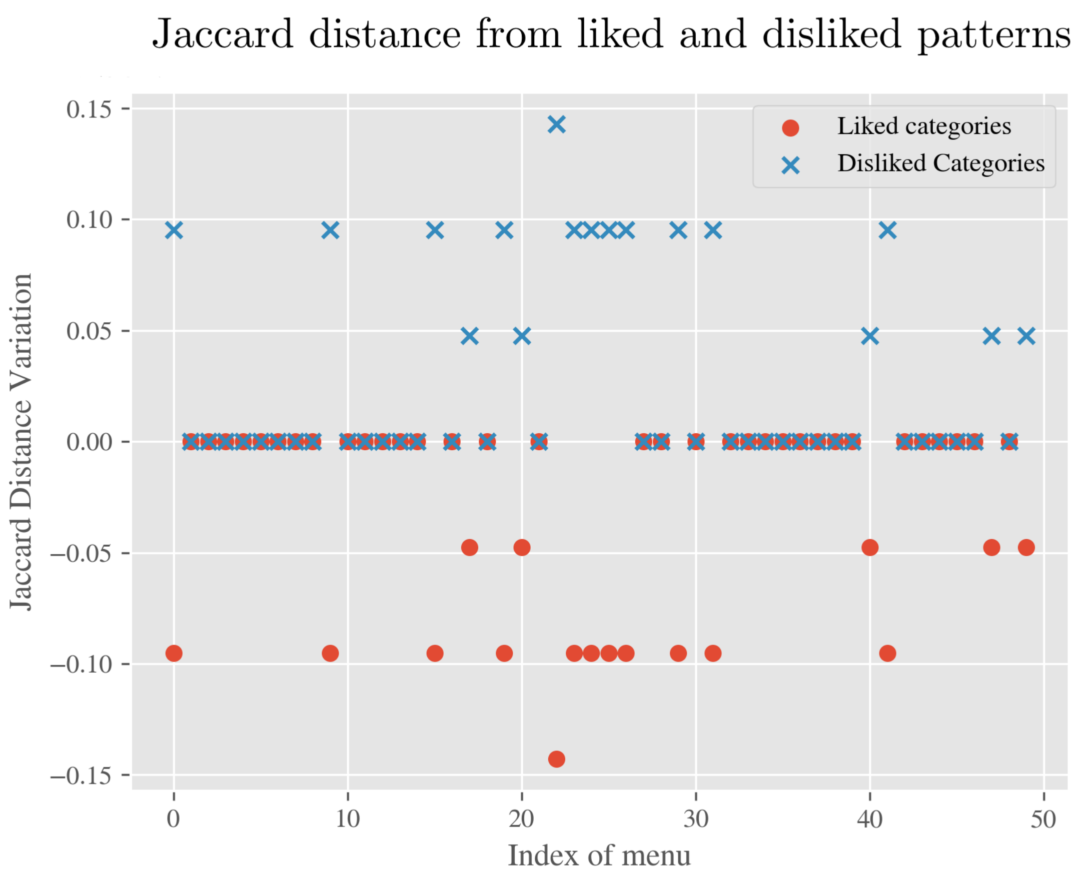

Figure 5.10: Jaccard distance increments from the dislike pattern, and increase to the like pattern across 50 menus.

Randomly recipes based diet		
Nutrient	**Unit**	**Avg. Variation**
Dislike distance	-	+0.04
like distance	-	0.0
kcal	kcal	-13.761230220317449
prot	gr	-0.4787886729870081
carb	gr	-2.3930042006395977
fat	gr	5.5323632772495435
fiber	gr	0.6314800815290088
vitamin c	mgr	23.309959776641275
iron	mgr	1.2869436552125038
calcium	mgr	221.9211939960594
vitamin a	mcg	605.2216795035298
vitamin b6	mg	0.12096667466822812
vitamin d	mg	0.8831782318351888
vitamin e	mg	-0.04191276760525309
vitamin k	mcg	-26.967997170903345
phosphorus	mg	248.23147039009345
iodine	mcg	-13.199232041648225

Table 5.3: Value oscillation after the secondary module. An conservative strategy was adopted and only small variation of the nutrients where allowed.

5.4 Real case study: the Stance4Health European Project

From the development of this application, an additional use case can be highlighted. Simultaneously with the progression of this thesis, the recommendation module associated with the European project Stance4Health has been developed, operating through an app on a smartphone. In this section, we will explore the aspects taken into account for its development, the adaptations from the initial approach, and the situations in which it has been utilized following the work presented in [198].

5.4.1 Technical details

The implementation of this type of system within the Stance4Health project has unfolded through the creation of a smartphone app compatible with Android and iPhone. To

Meat based diet		
Nutrient	**Unit**	**Avg. Variation**
Dislike distance	-	+0.03
like distance	-	-0.02
kcal	kcal	-20.76980993867777
prot	gr	-1.2853054948674079
carb	gr	-11.349871501578805
fat	gr	2.15362956783667
fiber	gr	-0.3597185261571551
vitamin c	mg	4.818489899065986
iron	mg	-0.5324358173383796
calcium	mg	24.34613141594754
vitamin a	mcg	136.43323729369698
vitamin b6	mg	-0.0005443298859073575
vitamin d	mg	1.3463118304824875
vitamin e	mg	0.5697441805741829
vitamin k	mcg	11.294386024089308
phosphorus	mg	-28.59306844771512
iodine	mcg	2.04775629037603

Table 5.4: Value oscillation after the secondary module. An conservative strategy was adopted and only small variation of the nutrients where allowed.

illustrate the app's functionality, we have chosen a high-level schema used in the article on the app [198].

The app was designed following a modular architecture (see figure 5.11 for details), where each of the blocks or functional modules was independent and could be utilized and improved independently of the rest. For a more detailed description, we will differentiate between the frontend (all the services the user either sees or interacts with when they open the app) and the backend (the server-side section built by all the engine services that run as users interact with the surface layer).

Starting with the front end the main parts are the screens. The term "screen" refers to the different graphical displays associated with every module that the users see navigating through the APP. Screens collect the changes and choices of the user and sent them to the backend. They do it in a machine-readable format that it is easily stored in the databases. The author was involve in the creation of the first prototype that used IONIC An open

Plant based diet		
Nutrient	**Unit**	**Avg. Variation**
Dislike distance	-	+0.020
like distance	-	-0.023
kcal	kcal	-27.643320500722336
prot	gr	0.6367802296466921
carb	gr	-13.33946494350169
fat	gr	2.285943875284525
fiber	gr	0.7682935597238089
vitamin c	mg	37.6217961349622
iron	mg	0.580691558837975
calcium	mg	15.47815102283235
vitamin a	mcg	-60.78130001410057
vitamin b6	mg	-0.02462815755091788
vitamin d	mg	0.5734166322332718
vitamin e	mg	2.1383934085201814
vitamin k	mcg	-3.7408984417080737
phosphorus	mg	-0.9323320301525607
iodine	mcg	9.284131809164709

Table 5.5: Value oscillation after the secondary module. An conservative strategy was adopted and only small variation of the nutrients where allowed.

source mobile UI toolkit for building modern, high quality cross-platform mobile apps from a single code base in Angular. After this first prototype we were offering advise during the decision-making process of the design, and finally a development teams built the final frontend as stated in [198], which was built using Angular 8 and Bootstrap 4.

On the backend side the author was directly involve in the creation of all the databases and modules stated in the diagram. The backend is first constituted by the databases. Those are represented in blue in the diagram, and they are the main source of information of the system. The database management system is MariaDB (MySQL), which uses Aria and XtraDB and in turn incorporates two other engines: PBXT and FederatedX. It also incorporates new system-level tables, which help in database optimization tasks thanks to the storage of service statistics.

All these data are submitted to the dietary generator module to produce a dietary

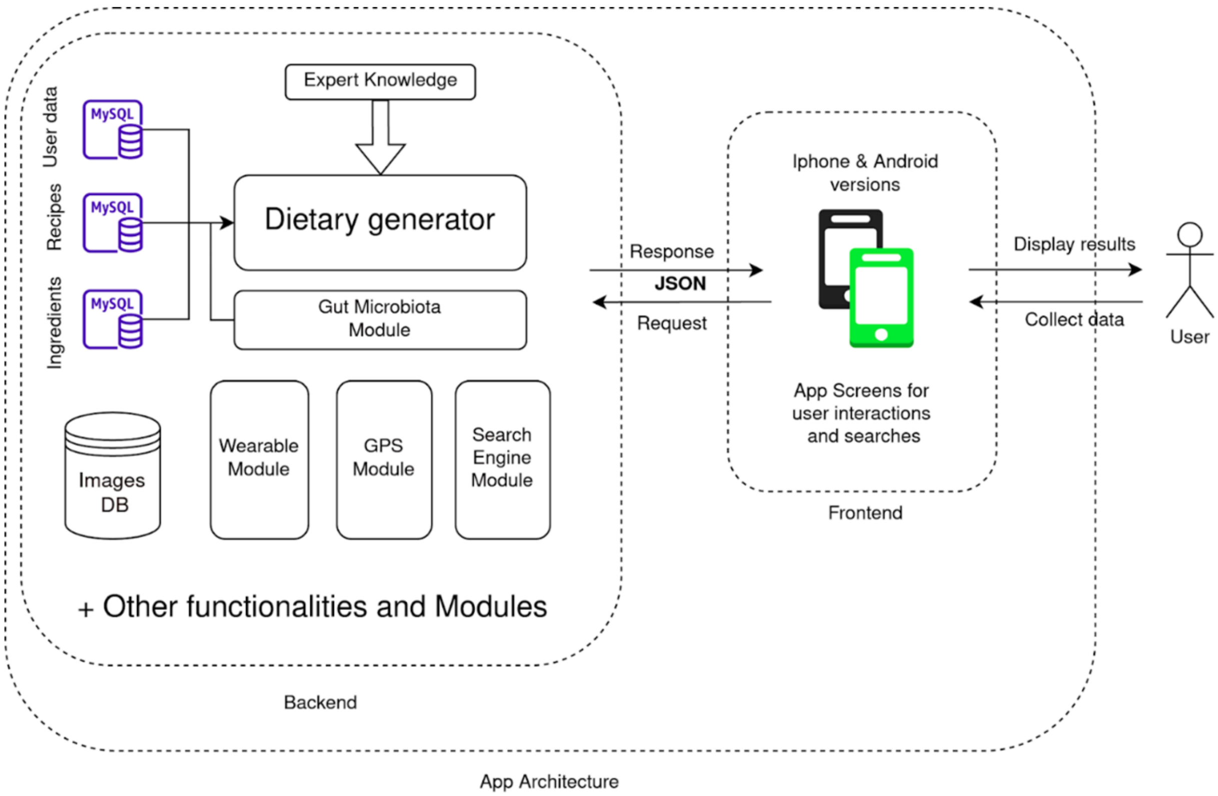

Figure 5.11: High level schema of the S4H App from [198]

recommendation. This led us to the second type of element that appears in the back-end: the modules. Modules are a collection of instructions or functionalities that can be activated while the user uses the app. It is worth noting that some of the modules represent independent functionalities, so they can receive and produce information without affecting the others.

The rest of the backend was built with Java and Spring framework. Furthermore, an application programming interface (API) was enabled to interact with the modules stored on the server, mainly built with Python 3.7 but would be compatible with python 3.10. This API launches the algorithm library that is necessary for certain functionalities as the diet generator and returns the information in a JSON format to be stored and presented.

5.4.2 Data usage

Within the application, three data sources are once again utilized, connecting ingredients, nutrients, recipes, and users. The philosophy and details of these sources differ somewhat from the initial approach outlined in this section; therefore, we proceed to provide a detailed description:

5.4.2.1 User data

We store the main aspects of the user in a MySQL database, including information about biometrics, restrictions and behavior. This information allows us to calculate the nutrient levels we are aiming to recommend. At the same time, it also lets us filter several items in the dietary database that are not suited for the user, either due to age (as coffee or tea recipes in children) or to food allergies (not recommending milk-based products to users allergic to milk proteins). Other than hard restrictions, preference data is also stored, in terms of a boolean (like dislike, 1 and 0) data.

5.4.2.2 Ingredients data

We used the S4H food composition database (FCDB) developed within the framework of the project [199] in which the author also participated. In summary, the S4H FCDB consists of more than 2600 foods with nutritional information on approximately 880 elements, including bioactive compounds. Our dataset contains a Branded Food Products Database consisting of food from supermarkets and hypermarkets of different countries (this is likely to be a significant percentage of the food already purchased and consumed by consumers). We specifically have detailed data from three different countries: Spain (with 89,385 foods products) provided by AECOC (Spanish Association of Manufacturers and Distributors), Germany (with 211,014 foods products) provided by ATRIFY and Greece (with 3312 foods products) provided by researchers [29]. The author also was in charge of the data fussion and merge of 670 different items from fast food restaurants obtained from the publications of the restaurant chains. These fast food items could be a recipe themselves, but as it is rare to solely eat one of them, we stored them as ingredients, to give the user more flexibility when entering the different menus which they could have eaten.

5.4.2.3 Recipe data

: S4H APP, unlike recent approaches in food recommendations, follows recipe-centered meal planning. This means that our system recommends to the user a specific recipe for a specific time of the day. Unlike a single combination of ingredients, recipes give ingredients a context/relationship and a procedure. This allows the user to know "what to eat" and "how to cook it". The recipes were reviewed by inhabitants from each country: it started with the analysis of more than 150,000 recipes from all countries to obtain a set of some 20,000 appropriate recipes (in terms of nutritional value, cultural traditions and diversity in all the possible meal plans). This dataset of recipes was then evaluated in terms of its ingredients' names, weight, retention factors and yield factors according to the cooking technique described for each recipe. Finally, we obtained the nutritional composition of each recipe. Additionally, users can create their own recipes, with ingredients from the ingredients database. Those users' recipes will only be available for the users that have created them.

5.4.2.4 Expert knowledge

In addition to the standard data considered, which encompasses expectations related to diverse nutrients based on biometrics and information from governmental and non-governmental health sources, we also incorporated other aspects requiring nutritional expertise. Among these considerations, daily physical activity emerged as a crucial factor influencing the recommended nutrients. In the S4H APP, we tackled these variations by prompting users to provide a detailed description of their daily physical activity. In order to calculate the metabolic expenditure related with physical activity, we established 5 groups according to the intensity of the activity. Then it is up to the user to provide the time expended along the day (in hours) in each category. Activity values were assigned to each group of activities according to the Food and Agriculture Organization of United Nations (FAO) recommendations on physical activity level [200]. These values are weighted in the number of hours stated by the user, so that the daily physical activity is estimated. Another aspect of the recommendations pertains to specifically designed recipe sizes. Each user selects a meal size based on their preferences, ranging from XXS to XL, depending on their population group. To standardize portion sizes—aligning the weight of the recipe with the assigned size letter—data were collected from 200 volunteers across various countries. This data collection aimed to understand their typical consumption habits and perceptions of portion sizes, utilizing photographic albums and pre-established portion sizes [201], [202].

After that, we created a survey where each participant indicated the usual consumption size of the different food groups and dishes (i.e., a meat-based recipe, a fish-based recipe, a sandwich-based recipe, etc.). This was then compared with the consumption data and an average range was assigned. The results were used to classify the portion sizes in 6 different sizes.

5.4.2.5 Additional Data Sources

In a project of these characteristics, for the normal functioning of the application, we also generate a large amount of images, barcodes, and user interaction data. These data contribute to enhancing user interaction with the system, improving understanding, illustrating recipes, and subsequently allowing the analysis of user behavior after human trials.

5.4.3 Generator

Regarding the generator, its operation follows the aspects developed in Chapter 4. The only noteworthy difference is the inclusion of, in addition to different nutrients, the intestinal microbiota. The introduction of the microbiota is a distinctive aspect in the study, as few systems take it into account. Specifically, this app aims to establish guidelines based on the interaction between the gut microbiota and the diet.

Studies conducted within the S4H project led to the development of an extended reconstruction of dietary metabolism in the human gut microbiota, AGREDA [203], which was subsequently enhanced using an enzyme promiscuity approach [204]. This network establishes crucial metabolic interactions between diet and the gut microbiota. These works enable us to translate this impact into a score, influencing the recommendation process to favor a pattern that promotes microbial metabolite levels closer to those considered healthy.

The information obtained provides an overall gut microbiota score for each food. This score is a real number, which can be positive or negative, depending on the impact of each food on the gut microbiota species. The generator is designed to prioritize recommendations that are most favorable for the microbiota.

Additionally on the generation phase, we produce a weekly menu, evaluating a whole week and generating daily menus for everyday based on healthy eating patterns. If the user state a noteworthy number of changes, we reevaluate our recommendation trying to adjust the deviations from the healthy pattern within the days left in that week (up to a certain health-based limit).

5.4.4 Other modules

For the proper functioning of the application, other modules were necessary in its creation. In this section, we will briefly review these modules, with complete descriptions available in [198].

5.4.4.1 Search engine

Our main goal was in part record the true behaviour of the users, for that reason as stated in the data section we incorporate a huge amount of different data to the user to be ablo to stated what they usually eat despite not following the recommendation. For that task we also have an additional search module based on text that let users look for recipes, ingredientes and commercial food. All the interactions Therefore, we incorporated several ways to accommodate these needs: text-, voice- and camera-based interactions: Camera-based interactions were primarily developed to allow users to have a quick interaction with commercial products as they may be the main source of deviation from the diet. This can be achieved through a comprehensive database linked to the commercial barcodes in the system. Text-based interactions are based on similarity metrics of the text-chains introduced. Finally Voice-based interaction runs on Google Voice recognition API.

It is also worth noting that recipes will be displayed with their information by 100 g if they are searched without context. However, on the menu screen, these recipes will be expressed as g/portion size, which offers a more realistic view of what they are eating.

5.4.4.2 Shopping List

One of the key aspects of every meal plan is the shopping process. This process can be influenced by several factors such as personal preferences or market selection. Moreover, recommending a diet based on a recipe's selection can produce some confusion in the user, as some of the recipe names and photos may not reveal the ingredients needed for its consumption. Therefore, we built another module that helps the user in buying the necessary ingredients. First, for every generated menu, this module compiles all the recipes and produces a unified shopping list that aggregates the necessary ingredients in weekly planning, and gives the total amount that the user will need of each ingredient. The shopping list can also be updated by crossing out those that are already in stock or that have been bought

5.4.5 APP Testing and Validation

Health-related applications are commonly evaluated using objective and reliable scales, such as the Mobile APP Rating Scale (MARS) [205] and the Nutrition APP Quality Evaluation AQEL scale [206]. Among these, MARS, particularly its User Version (uMARS)

[205], is widely utilized. uMARS assesses applications based on four criteria: engagement, functionality, aesthetics, and information, with each criterion rated on a scale from 1 to 5. The average scores for these domains are calculated, resulting in an overall average score indicative of the application's quality.

To nutritionally validate the app, various dietary records were employed, including a food consumption frequency questionnaire (FFQ) and a 24-hour recall, conducted on two non-consecutive weekdays and one weekend day. All questionnaires (uMARS, FFQ, and 24-hour recall) were administered in an online format through Google Forms adapted for research purposes.

Before the nutritional intervention, the app underwent pre-testing for reliability to ensure a seamless user experience. The testing involved load, recalculation, and processing speed assessments. Subsequently, 20 individuals from different countries (Spain, Italy, Germany, and Greece), aged 12 to 60, tested the app and completed the uMARS scale [205]. These participants were recruited from various research centers participating in the European project Stance4Health. Additionally, 20 volunteers (aged 19–25, enrolled in the Nutrition's Bachelor's Degree at the University of Granada, Spain) recorded their food consumption using a 24-hour dietary record and an FFQ to evaluate whether the generated menus aligned with dietary guidelines and actual consumption. All participants volunteered and consented to their data's use for research purposes. Detailed analysis of the study's findings will be presented in Chapter 7.

To date, the app has been utilized by over 500 users across two different countries, with a minimum usage duration of three months. This usage has resulted in the generation of a total of 42,000 daily menus. The outcomes of the nutritional intervention are currently under analysis by an interdisciplinary team comprising nutritionists and marketing researchers. As of the author's knowledge date, the results are actively being evaluated to glean insights into the impact and effectiveness of the nutritional interventions facilitated by the app.

Chapter 6

Application: Podcast content

"Our blessed radio. It gives us eyes and ears into the world. We listen to the German station only for good music. And we listen to the BBC for hope."

The Diary of Anne Frank (Film)*, 1959*

6.1 Podcast recommendations

In 2005, "podcasting" was chosen as "the word of the year" by the New Oxford American Dictionary. By 2015, it was anticipated that this new audio service would challenge the traditional radio industry [207] . Nowadays Podcasting is an increasingly popular pastime in the U.S with 79 percent of respondents knowing of the format, while over 82 million people listened to podcasts in 2021. This number is estimated to rise even further, reaching over 100 million listeners in 2024 [208].

Focusing on their recommendation, there are some interesting works on it. [209] highlights several approaches in recommendation as content modeling and topic retrieval based textual and non-textual information . In the content modeling approach, several works as [210]–[212] have shown that textual and non-textual aspects of podcasts can improve the performance of topic-based podcast popularity prediction. The addition of knowledge graphs has also improve the recommendation of shows [213] and their approach to generate sequential recommendations as [209].Despite this advances there are still plenty work to develop in this recommendation area, beginning with the gathering of podcast information that is still somewhat opaque [214].

On almost every study we found, most recommendations tend to focus on whole podcasts or individual episodes. The reasons behind the inclusion of these cultural productions in a recommendation list depend on the preferences of the listener, and it's evident that there are podcasts of varying lengths, including longer ones. However, the podcast ecosystem is diverse, with several trends favoring the creation of shorter-duration podcasts that can be consumed on a daily basis. The study in [209] already state an interesting point, assuming there are some kind of structure or temporal pattern in the listening. The challenge then arises: How can we effectively recommend more than one podcast? How to produce more structured recommendations?

This is the question that our application seeks to answer. We utilize our algorithm to develop a recommendation system that generates and structures schedules of podcast episodes based on user profiles. The objective is to provide users with content through a structured schedule, tailored to their preferences and either learned or specified by the user.

6.2 Experimentation

6.2.1 Data sources

Access to podcast information is often opaque, and there is no benchmark dataset for comparison.This application started with this problem, that lead us to the inclusion of several data sources in a very similar way of the nutritional case. We used two different data sources based on the main streamers of podcast data today, iTunes and Spotify. This is due to the lack of linked information between different podcast shows (which already have descriptions and categories), the episodes produced within those podcasts, and, finally, the information that connects podcast and episode data with user evaluations of any kind. This lack of standardization has been acknowledged in previous work [214], which extends beyond descriptions, as there are sources of podcast transcriptions that could enhance information extraction by including details such as guests, format, or explicit language. Despite this situation, podcasts were originally and still are associated with an RSS feed, which serves as an interesting source of useful information (the iTunes dataset is based on it) [215], especially textual information.

We chose to gather information in this area as it is already an established field for extracting useful data from the textual information associated with podcasts. Recent research has specifically focused on extracting and modeling topics and additional information from the short text accompanying it [211]. Audio is also an interesting source of data, but extraction often lacks explicability and does not necessarily yield better results [216].

- User dataset: iTunes listener :

  The Podcast Reviews dataset encompasses 2 million reviews across 100,000 podcasts, with monthly updates, providing a dynamic portrayal of user evaluations in the podcasting ecosystem. It contains 2 datasets connected with the following format:

  Reviews dataset:

  - **Podcast ID:** Unique identifier for a podcast within the dataset.
  - **Title:** The title of an individual podcast.
  - **Content:** The textual content of a user's review for a particular podcast.
  - **Rating:** A numerical value representing the user's rating for a podcast, on a scale of 1 to 5.
  - **Author ID:** Unique identifier for the author of a review.
  - **Created At:** Date of the evaluation.

Podcast dataset:

- **Podcast ID:** Unique identifier for a podcast within the dataset.
- **iTunes ID:** Unique identifier for a podcast within the iTunes store.
- **Slug:** slug for the podcast.
- **iTunes URL:** URL for the podcast inside iTunes.
- **Title:** Podcast title (for the show, not episodes).
- **Author:** Author of the podcast.
- **Description:** Podcast show description.
- **Average Rating:** Average rating across all users.
- **Ratings Count:** Total number of reviews in the dataset.
- **Scraped At:** Date of scraping.

The issue with this information is that it is too broad to make a truly granular recommendation. For a broadcast recommendation, we would lack information about the duration of episodes, which is not suitable for our problem. While we could potentially provide a podcast show recommendation, to delve deeper at the episode level and tailor the recommendation we are seeking, we require more details. That is why we used and additional datasource form Spotify:

- Item dataset: Spotify episode dataset The podcast dataset contained about 100k podcasts filtered to contain only documents which the creator tags as being in English or Portuguese, as well as by a language filter applied to the creator-provided title and description. We expect that there will be a small amount of multilingual content that may have slipped through these filters. Episodes were sampled from both professional and amateur podcasts including episodes produced in a studio with dedicated equipment by trained professionals, as well as episodes self-published from a phone app — these vary in quality depending on professionalism and equipment of the creator. The episodes represent eda wide range of:

- Audio quality: we can expect professionally produced podcasts to have high audio quality, but there is significant variability in the amateur podcasts. We have included a basic popularity filter to remove most podcasts that are defective or noisy.
- Topics: the episodes represent a wide range of topics, both coarse- and fine-grained. These include lifestyle and culture, storytelling, sports and recreation, news, health, documentary, and commentary.
- Structural formats: podcasts are structured in a number of different ways. These include scripted and unscripted monologues, interviews, conversations, debate, and included clips of other non-speech audio material

Each of the episodes in the dataset included an audio file, a text transcript, and some associated metadata. The associated metadata of the episodes is the part we are most interested in. The fields relevant for us in it were:

- **show_uri** : Unique identifier for a podcast within the dataset.
- **show_name**: Podcast name
- **show_description**: Podcast description
- **publisher**: Publisher
- **episode_uri**: Unique identifier for a episode within the dataset.
- **episode_name**: Episode name
- **episode_description**: Episode description
- **duration**: Episode duration

6.2.2 Initial creation of complex items

For the initial creation of elements, we conducted a data cross-referencing between episode data, linking them with the data of each podcast. This results in a preliminary dataset where each podcast is associated with all its episodes. Simultaneously, by utilizing the second dataset of user ratings, we can connect each program with the programs in the Spotify dataset, thus linking user opinions with the rest of the data.

6.2.2.1 User Profiler

Each user selected for the trial provides at least 10 evaluations of different programs. Due to the earlier relationship, we can filter these evaluations as positive or negative. This allows us to obtain the set of podcasts positively evaluated by the user, creating a profile of preferred themes where the importance of each theme is weighted by the quantity of positive ratings a podcast with that label has. This results in a final taste profile.

In addition to this profiling, we obtain a set of characteristics to form our final structure: the themes of different blocks and their duration.

6.2.2.2 Item Profiler

The Spotify episodes, in addition to the podcasts, have associated characteristics for evaluating our functions. In this case, we have various parameters: content categories (three per episode), podcast description, episode description, episode name, duration, and language. For this scenario we select the first content category as the primary one and the others two as secondary, to have a starting point for the selection of constraints based on topics in addition to duration of the episode.

This allows us to select episodes of specific themes encoded in the structure suggested by the user's profile and operate the genetic algorithm to find combinations of podcasts that meet the established duration. Furthermore, for the second phase of the algorithm, we will need to evaluate the themes and descriptions of the episodes for two different purposes: avoiding repetitions of themes in different episodes of the same structure and aligning with the user's preferences, either through podcast categories or textual similarity of descriptions.

In this specific problem, after the initial generation through genetic algorithms, a refinement is needed. Due to some of the podcast characteristics, we have information about the episode number and the air date. This enables us to restructure our recommendation so that, in the event of recommending different episodes of the same podcast, we prioritize those in chronological order. This step can also be deactivated if not necessary or deemed irrelevant.

6.2.2.3 Results

For this generation we use our novel python package GenRecs that allow us to represent the structure and data in a simpler way. Apart from that, GenRecs is agnostic to the genetic algorithms so we can experiment with several approaches. Starting from our genetic algorithm defined in Chapter 4, we apply the competition approach, but we also do it using the NSGA-II algorithm as a selection mechanism. Both evaluations are done as a multivariable function of the three defined objectives. Additionally, we compare it with a function initially proposed, which translates into a weighted sum of the three aforementioned functions, i.e., given:

$$g_{length}(Section1) = (\tfrac{v(optimal_{lengthgoal})}{lengthgoal})(\textstyle\sum_{Episodes \in Section3} length(episodes) - length_{goal})^2 \tag{6.1}$$

$$g_{length}(Section2) = (\tfrac{v(optimal_{lengthgoal})}{lengthgoal})(\textstyle\sum_{Episodes \in Section3} length(episodes) - length_{goal})^2 \tag{6.2}$$

$$g_{length}(Section3) = (\tfrac{v(optimal_{lengthgoal})}{lengthgoal})(\textstyle\sum_{Episodes \in Section3} length(episodes) - length_{goal})^2 \tag{6.3}$$

in this case $optimal_{k,u}$ represents the optimal length of the section to be reached by the set of episodes in $Section_{1,2,3}$. In addition, $v(optimal_{length})$ is defined as :

$$v(optimal_{length}) = optimal_{length} * C/(length_{range})^2 \tag{6.4}$$

Three experiments are proposed for the first generation:

- Multi-variable function ($F = (g_1, g_2, g_3)$) with tournament selection (Table 6.1).
- Scalarized function ($F = \sum_{i=1,2,3} w_i g_i$)) with tournament selection (Table 6.2).

GA parameter	Value
Nº of Generations	20
Initial population size	50
threshold	1
Selection	Tourtnament-Elitism 3 random selection
Probability of mutation	0.5
Crossover	Half section exchange

Table 6.1: Multi-variable function with tournament selection.

GA parameter	Value
Nº of Generations	20
Initial population size	50
threshold	1
Selection	Tourtnament-Elitism 3 random selection
Probability of mutation	0.5
Crossover	Half section exchange

Table 6.2: Weighted function with tournament selection

GA parameter	Value
Nº of Generations	20
Initial population size	50
threshold	1
Selection	Tourtnament using NSGAII selection
Probability of mutation	0.5
Crossover	Half section exchange

Table 6.3: Multi-variable function with NSGAII selection

- Multi-variable function ($F = (g_1, g_2, g_3)$) with NSGAII selection (Table 6.3).

For the experiments, we select users that had a number of different podcast rated. Following the guidelines of user profiling, we obtained a set of three categories consumed by the user. With this information, we created a podcast scheduling goal based on three listening stages, one for each favorite category of the user. Our goal was to produce a podcast schedule with episodes across three different sections based on user preferences, with a duration of 60 minutes for each section. The topics were selected based on the two most liked by the users (that were consistently: 'True crime' and 'Comedy') and others less liked to test the limits of the algorithm in terms of preferences (we selected 'Health').

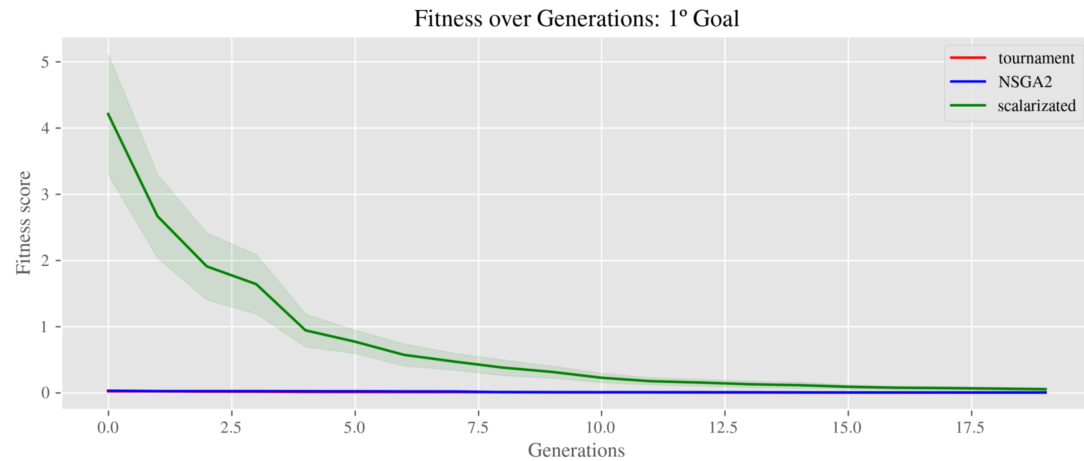

Figure 6.1: Genetic Algorithm Evolution in 20 Generations for Time-Based Structure for the first goal (duration within first section of content).

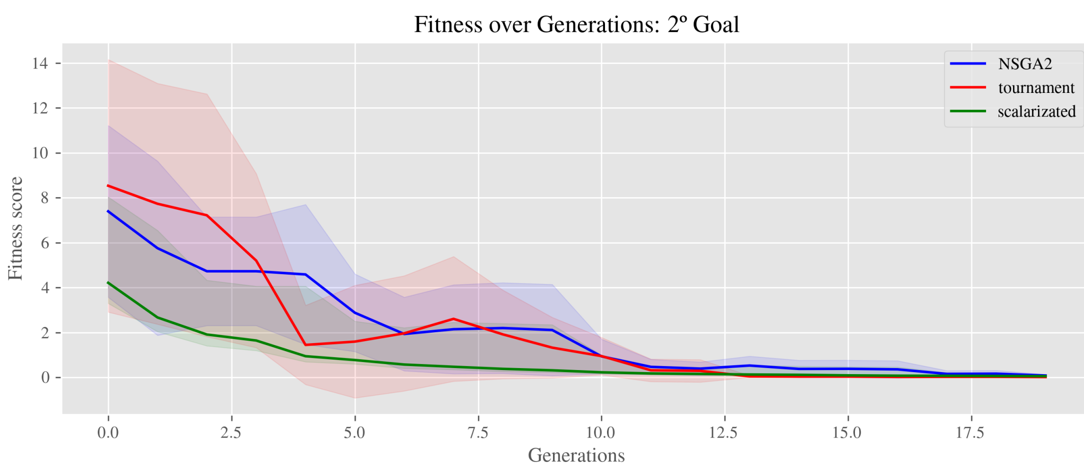

Figure 6.2: Genetic Algorithm Evolution in 20 Generations for Time-Based Structure for the second goal (duration within second section of content).

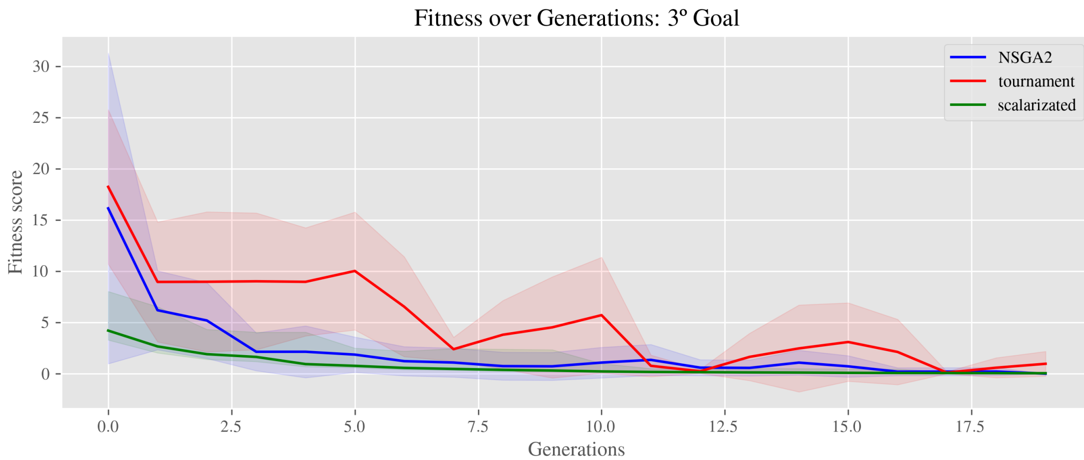

Figure 6.3: Genetic Algorithm Evolution in 20 Generations for Time-Based Structure for the third goal (duration within third section of content).

In Figures 6.1, 6.2 and 6.3, the evolution in the fitness function over the 20 generations for all the 80 experiments is depicted . It can be observed that the objective function reaches satisfactory convergence, indicating that the genetic algorithm has successfully found a solution that aligns with the desired time-based structure. The convergence, however, depends on the different episodes we an find in the dataset. In this scenario, all three lengths are equally important, but the 3º Goal is more unstable as it is more difficult to find elements in the dataset that complies with our expectations.

These results support the effectiveness of the genetic optimization approach for generating recommendation structures based on the duration of podcast episodes. Among them, the scalarized version and NSGA selection approaches where the fastest.

6.2.3 Final item adaptation

6.2.3.1 Content adaptation

For improving the created individuals, adapting them to user preferences, we compare two methods:

- Episode categories.

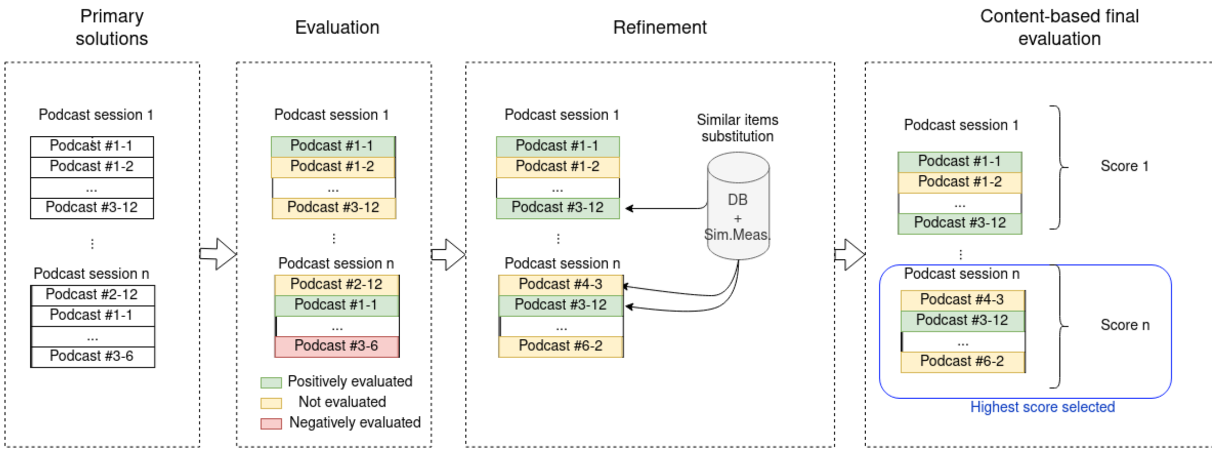

Figure 6.4: Solution refinement phase for the podcast application.

For the categories similarity we chose the weighted Jaccard similarity for every episode of a podcast, we define the similarity metric as follows:

Let C_1 and C_2 be two sets of categories associated with two different podcast episodes, and W_1 and W_2 be their respective weights based on the profile distilled from the user. That is if the user mostly liked a specific podcast category, being close to that category will increase the score.

Selecting both the user profile and the podcast categories, we can calculate the Jaccard similarity as

$$\text{Weighted Jaccard}(C_1, C_2) = \frac{\sum_i \min(CAT_1[i], CAT_2[i]) \cdot \delta(C_1[i], C_2[i])}{\sum_i \max(CAT_1[i], CAT_2[i]) \cdot \delta(C_1[i], C_2[i])}$$

where $\delta(x, y)$ is the Kronecker delta function that will be 0 in those category not involve in the description of a particular episode.

- Transformer-Based Embedding Cosine Similarity

To measure the similarity between the textual descriptions of podcast episodes using BERT-based embeddings using the package SentenceTransformers [217] and the model *all-MiniLM-L6-v2*, we employ cosine similarity:

Let $Pod_1 ... Pod_n$ be the textual descriptions of podcast episodes. We encode these descriptions into high-dimensional vectors using BERT embeddings, denoted as $Pod_{E_1}, ..., Pod_{n_E}$, we perform the same procedure with the podcast rated with 4 or more stars by the user. We then split those embedding between different categories and perform the average of all the vectors in the categories, this creates a semantic representation of which kind of content is liked by the user $User_{E_{Category_1}} ... User_{E_{Category_N}}$.

Having this profile from the user, when evaluating a certain episode or comparing a change proposal we perform the cosine similarity is computed as:

$$\text{Cosine Similarity}(Pod_E, User_E) = \frac{Pod_E \cdot User_E}{\|Pod_E\| \cdot \|User_E\|}$$

where $\cdot$ denotes the dot product, and $\|\|$ represents the Euclidean norm.

This metric assesses the similarity of the semantic content in the textual descriptions, leveraging the power of BERT embeddings to capture nuanced semantic relationships. This semantic relationships help us identify possible duplication in themes and also evaluate the similarity to the user generate embedding, based on the themes they like.

6.3 Evaluation

6.3.1 Results

Results of the experiments can be read in Tables 6.4 and 6.5 for the numerical values of the outcome. However, Genrecs is able to generate plots specifically for each episode during the recommendation so, for the sake of representativity, we also offer the result through a scatter plot in Figures 6.5 to 6.12.

During the improvement process, we proceed to evaluate the items and classify those where we believe there could be an enhancement. In both procedures, this process, as expected, increases the variability with respect to the expected duration, which is much more uniform in the output of the genetic algorithm. However, this variability mostly remains within a range of 10 minutes above and below the expected target.

Regarding the improvement in preferences, the graphs show a considerable enhancement in all episodes. Our approach is quite aggressive, but those episodes that are close to the user would have already completed their improvement process. In the rest, the system can find alternatives to this process, but there are still some cases (minimal) where the algorithm cannot improve the result beyond the planned. Therefore, if the system, while staying below the target, improves it, we stick with that option. If, however, there is no improvement, we retain the initial proposal (as we would understand it to be the closest to the objectives).

In addition to the above, there is a dataset that also deserves to be mentioned. In Figure 6.6, it can be observed that a large set of improvements has the same score. This might seem strange, knowing that the improvements are independent of different chapters. The key to this behavior comes from one of the user's most evaluated podcasts belonging to this category and having a large number of episodes with the same description. This results in the episodes we select in this section, and if 1 or two of them (remembering that the average diversity of the block is 1.67) belong to this podcast, even if they are different, the distance with the description shortens quickly. This leaves us with an interesting starting point about the user's preferences, but also a point of improvement, easily solvable by lowering the episode description if we had that more granular user rating.

In the case of similarity using Jaccard categories, we can observe something interesting. We choose the health category to observe the behavior of our system towards a goal about which we know little from the user. The problem with the Jaccard category rating is that we only have 3 possible evaluations, and due to the obtaining of the user profile, health podcasts with other categories or with a category intertwined with the two main ones will be consistently devalued. The system, still, tries to find elements that share health with some of the other categories that the user enjoys, but it is unable to surpass the limit imposed in the other cases. This fact leads us to think that an additional measure of confidence in the results is necessary. That is, given an improvement, even if we compare similarity with the user, if the user has chosen that category even if they do not listen to it and it deviates from their tastes, we must introduce an additional measure of uncertainty that disperses the results more.

Jaccard Metrics		
Parameter	**Unit**	**Avg. Variation**
Preferences First category similarity	-	0.08
Preferences Second category similarity	-	0.11
Preferences Third category similarity	-	0.13
First block	Time var (fitness score)	0.24
Second block	Time var	-0.19
Third block	Time var	2.59
Diversity	Number of episodes per block	1.64

Table 6.4: Value oscillation after the secondary module, using the Jaccard similarity metrics for the improvements.

Transformer embedding Metrics		
Parameter	**Unit**	**Avg. Variation**
Preferences First category similarity	-	0.15
Preferences Second category similarity	-	0.17
Preferences Third category similarity	-	0.19
First block	Time var	-0.69
Second block	Time var	-1.41
Third block	Time var	1.79
Diversity	Number of episodes per block	1.67

Table 6.5: Value oscillation after the secondary module, using the Transformer-based similarity metrics for the improvements.

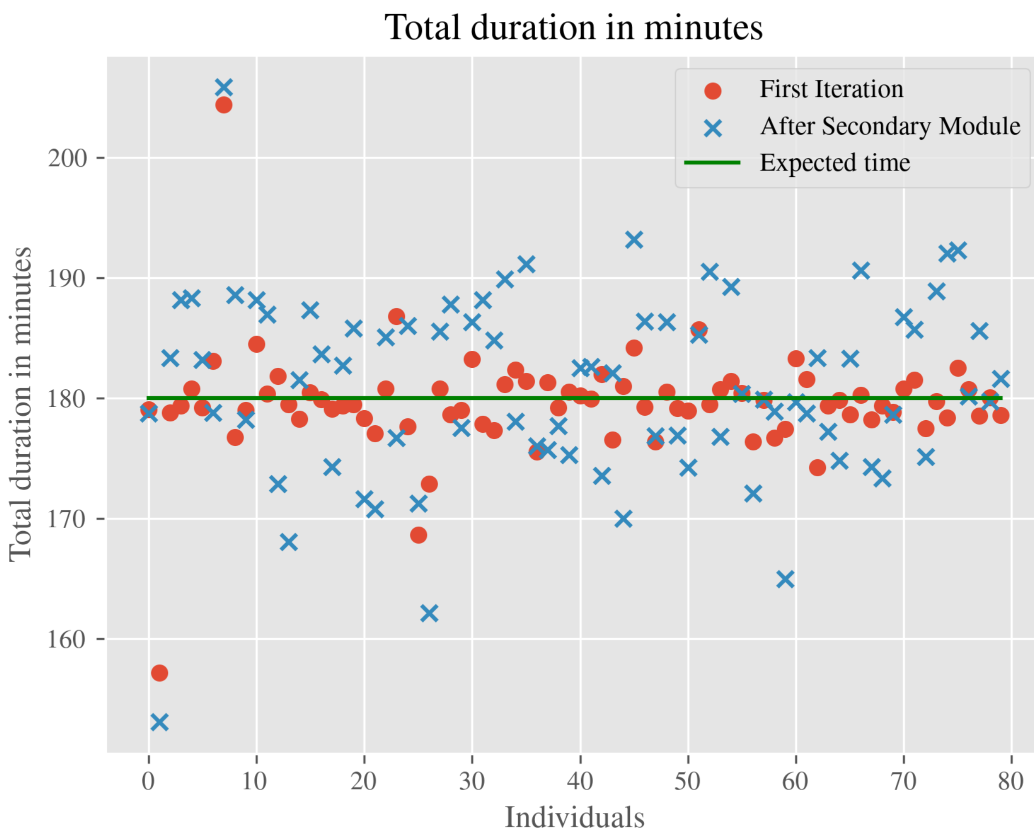

Figure 6.5: Duration oscillation after the secondary module, using the Transformer-based similarity metrics for the improvements.

Section	**Podcast**	**Episode**	**Time**	**Category**
1	Savage Days Podcast	Member Mountain.	60.12	Comedy
2	Serial Killers	The Beast of British Columbia" Pt. 2 - Clifford Robert Olson Jr	60.18	Crime and Murder
3	I Am I Have	I Am I Have The Panel Podfest For Mental Health	42.89	Health
3	Casey Zander Health	How to Reprogram Your Subconscious Mind in 10 Secret Steps To Attract Anything	15.72	Health

Table 6.6: Example of a Recommendation.

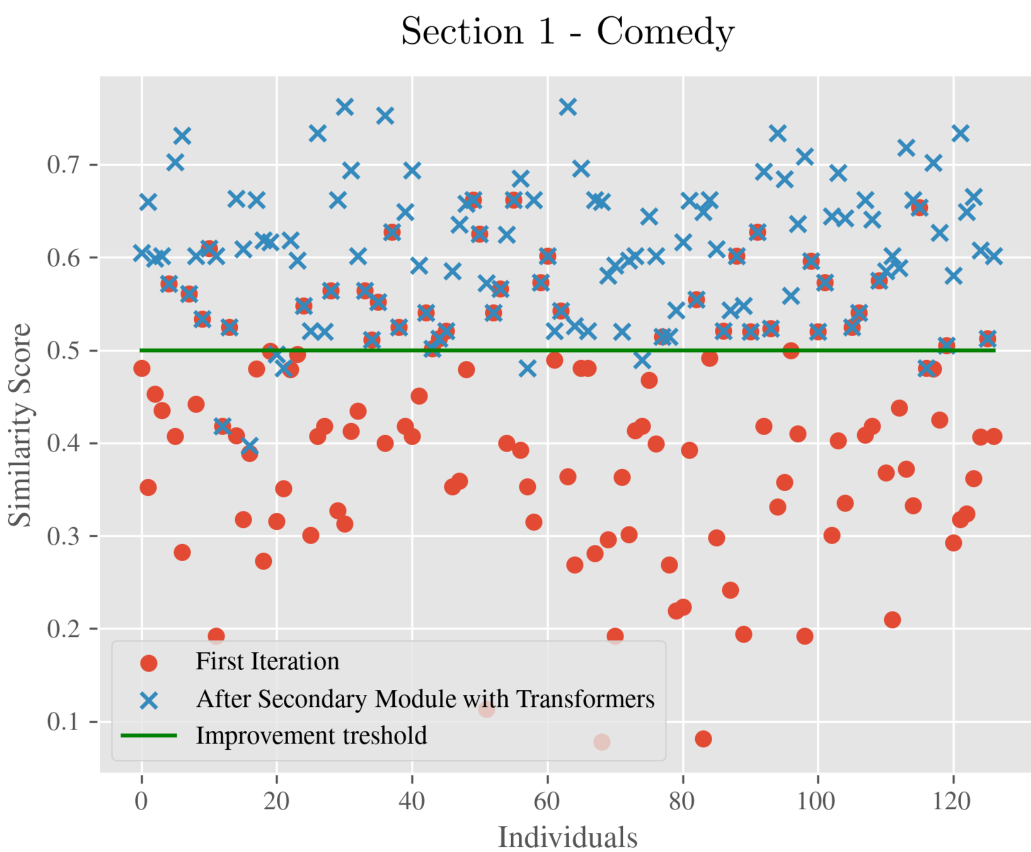

Figure 6.6: Preference score variation in Section 1, using the Transformer-based similarity metrics for the improvements.

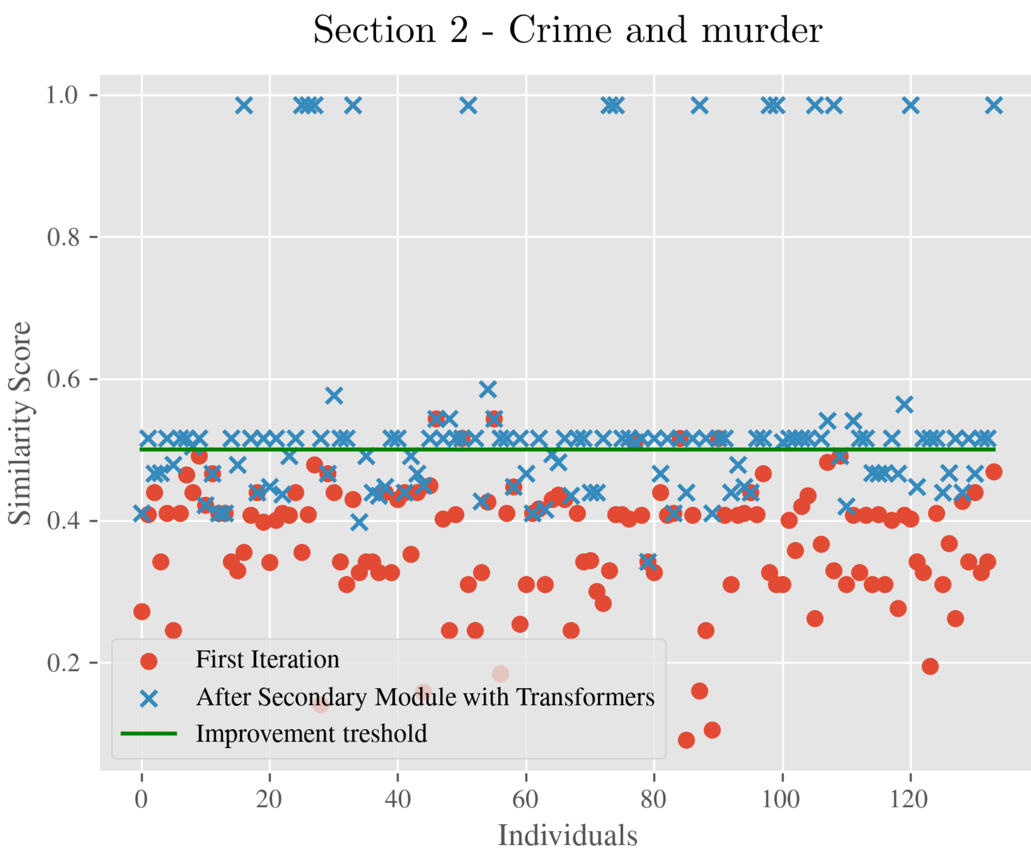

Figure 6.7: Preference score variation in Section 2, using the Transformer-based similarity metrics for the improvements.

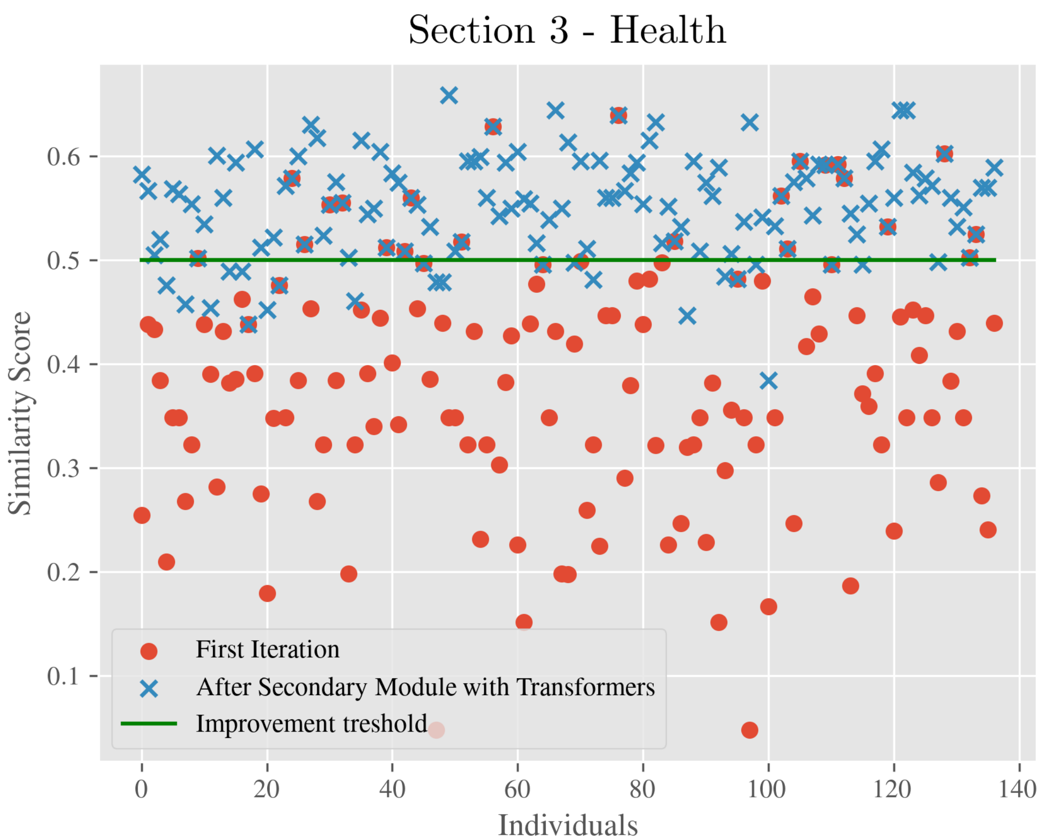

Figure 6.8: Preference score variation in Section 3, using the Transformer-based similarity metrics for the improvements.

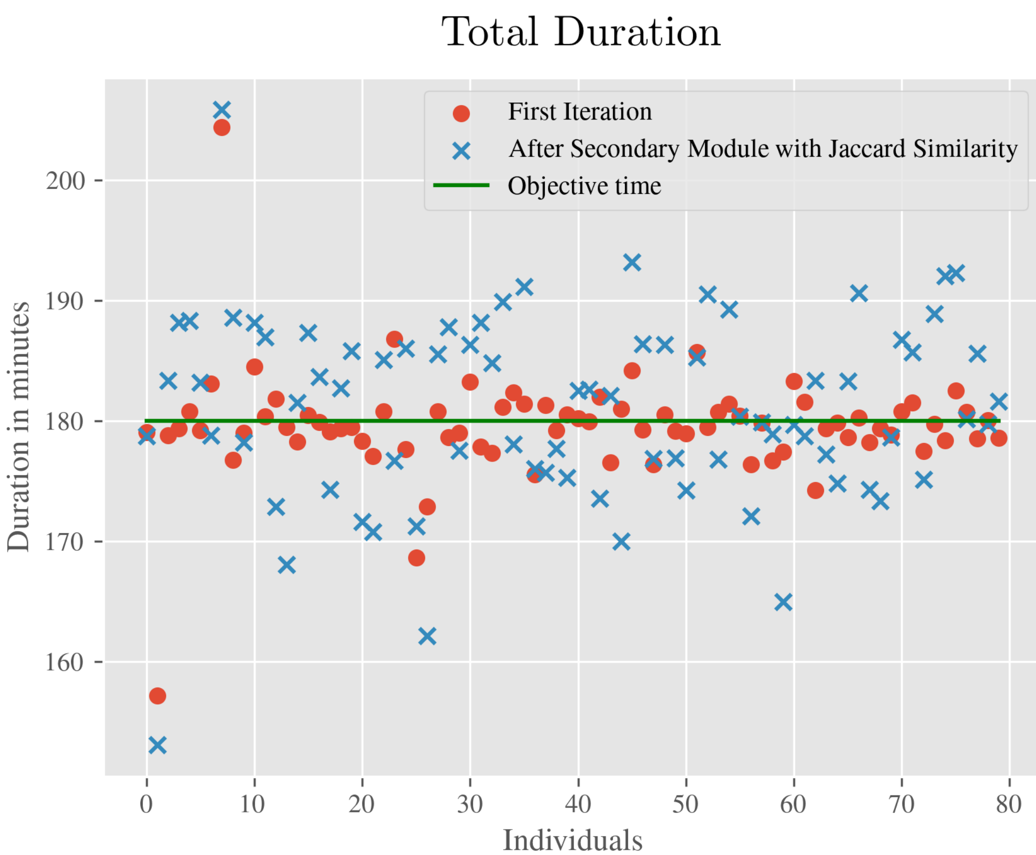

Figure 6.9: Duration oscillation after the secondary module, using the Jaccard similarity metrics for the improvements.

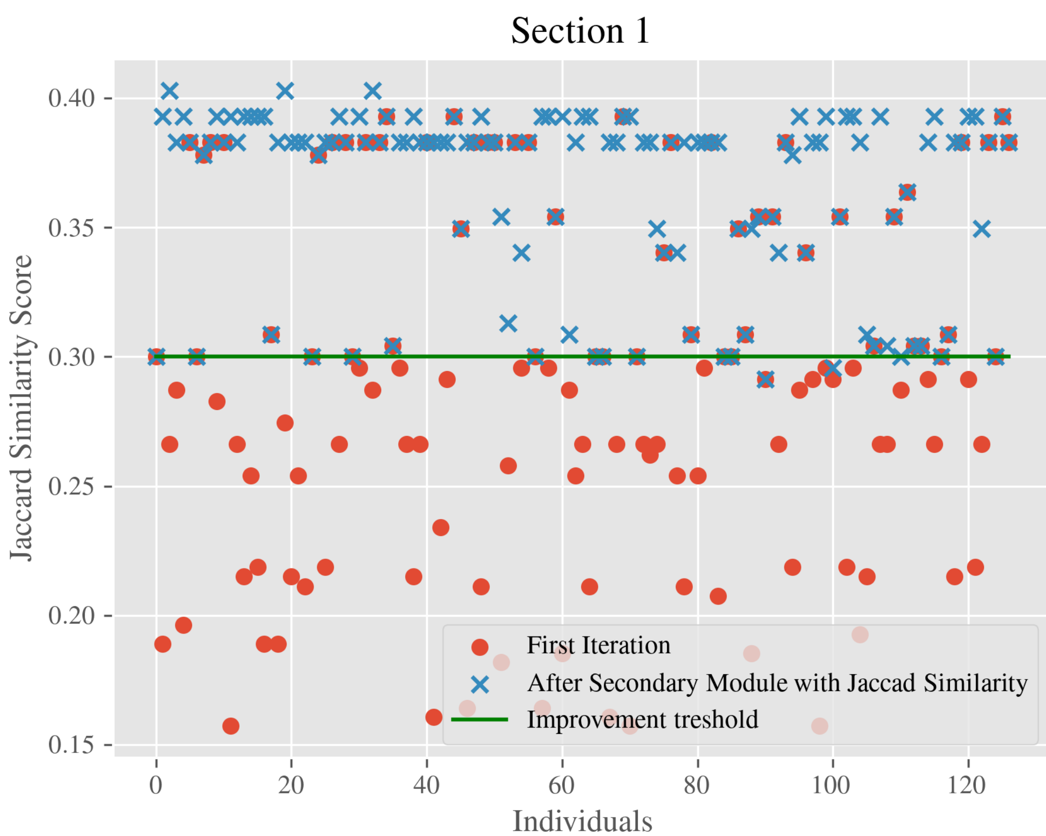

Figure 6.10: Preference score variation in Section 1, using the Jaccard similarity metrics for the improvements.

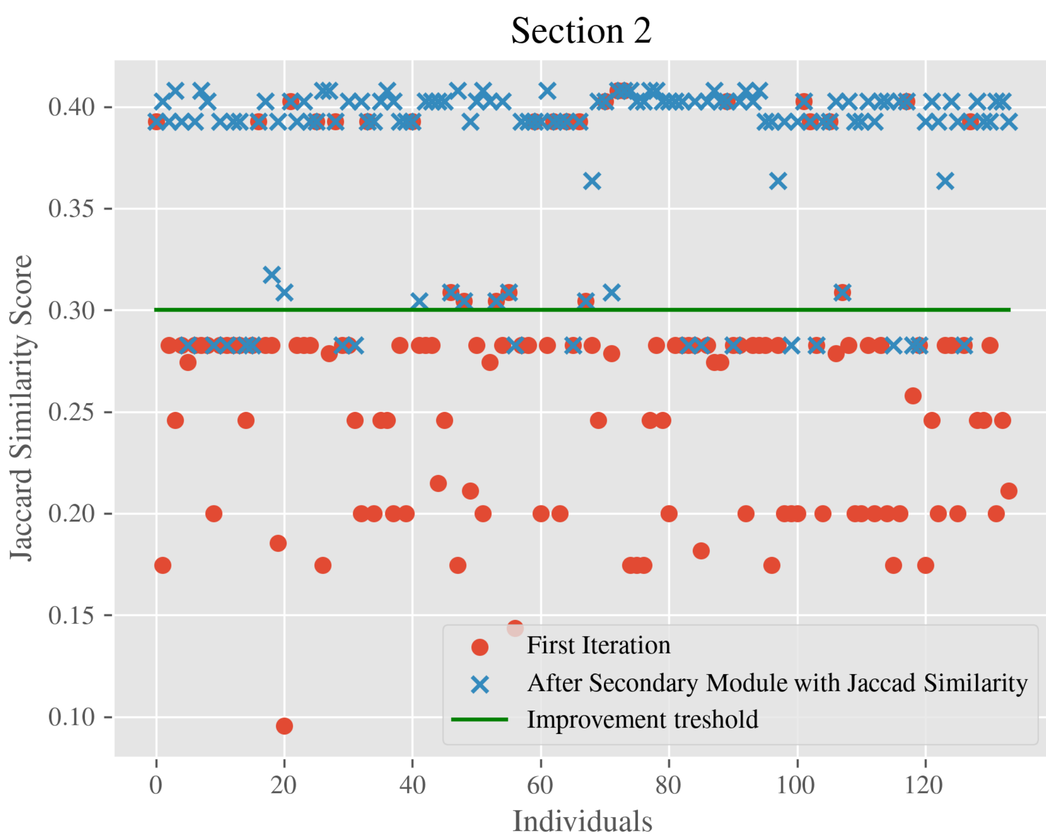

Figure 6.11: Preference score variation in Section 2, using the Jaccard similarity metrics for the improvements.

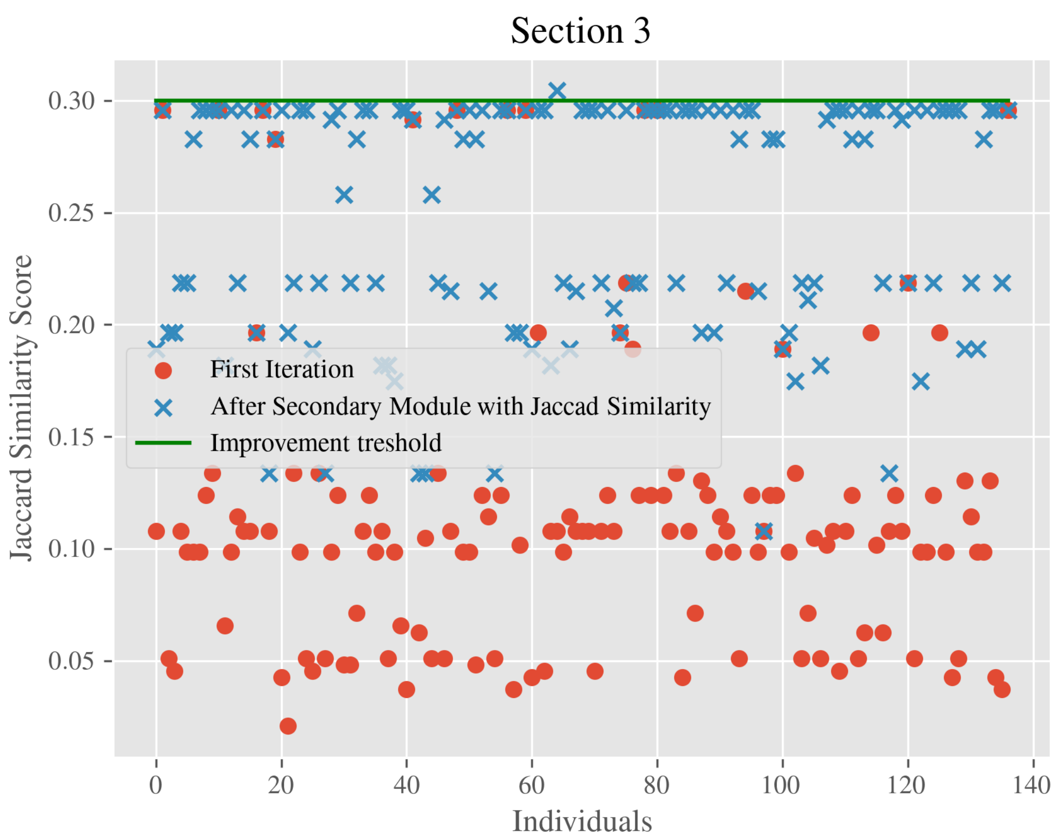

Figure 6.12: Preference score variation in Section 3, using the Jaccard similarity metrics for the improvements.

Chapter 7

Genrecs

"Debugging: what an odd word. As if "bugging" were the job of putting in bugs, and debugging the task of removing them. But no. The job of putting in bugs is called programming."
The Bug Ellen Ullman *2003*

7.1 GenRecs: A recommendation package

In the pursuit of solving complex recommendation problems, the imperative for a reusable and reproducible framework, agnostic to specific domains, became evident. In response, we present GenRecs, a Python package meticulously crafted to offer standardized notation for complex recommendation problems while seamlessly interfacing with pivotal libraries, including Transformers and evolutionary algorithms.

GenRecs serves as an enabler for experiment replicability, the creation of novel applications, and the exploration of innovative algorithms within the recommendation domain. This open-source package is freely accessible on GitHub, extending an invitation to researchers, developers, and enthusiasts to harness its robust functionalities.

Figure 7.1: GenRecs logo package

7.1.1 Key Features

GenRecs encompasses a spectrum of features, including:

- Advanced tools for preprocessing diverse recommendation datasets.
- Implementation of evolutionary algorithms, comprising a genetic algorithm and NSGA-II, meticulously tailored for recommendation systems.
- Utilities tailored for content-based recommendation.
- Functionalities addressing collaborative filtering recommendation.

This library not only facilitates the implementation of existing recommendation algorithms but also serves as a canvas for crafting and experimenting with novel methodologies. The package seamlessly integrates with prevalent Python libraries, ensuring compatibility and user-friendliness.

7.1.2 Development tools

In terms of technology usage and python implementation we follow the most extended approach and tools for the python package creation. Those involve additional secondary packages that make easier the incorporation of necessary utilities. Those are (in a list that may be updated and can be consulted directly on github page).

- **Poetry** for dependency management and packaging
- **Black** for code formatting
- **Pytest** for testing
- **Sphinx** for documentation and **Read the Docs** for hosting documentation

We will use git as the main version control system as it is widely used in the software development community. We will support the current and future development of the project through an issue tracker with associated pull request.

7.1.3 Package structure

The package is structured following the main objects that can be found in the definition of our algorithm. To make our algorithm work, we must have a defined initial dataset that allows us to establish a user profile on which to base recommendations and the objects that are part of the recommendation. A full description of the package and tools can be reviewed in Figure 7.2 along while reading the section.

The idea behind the development is to provide the package with tools to import these datasets, operate easily with them, and insert them as elements from which the final items will be created. We describe below how we achieve this goal through the objects of subitem, item, and dataset.

- **Subitem:** refers to the "i" in our methodology nomenclature. They are the simplest items we want to use to form more complex ones. The structure of these objects is simple, based on Python dictionaries. This decision is made to preserve the semantic relationship of the attributes and to be able to retrieve and display them at all times. It also offers greater ease of serialization (for incorporating our system into an API that exchanges JSONs) and, finally, allows for much flexibility in being added, updated, or removed. This feature is positively valued in these phases of the project to facilitate development.

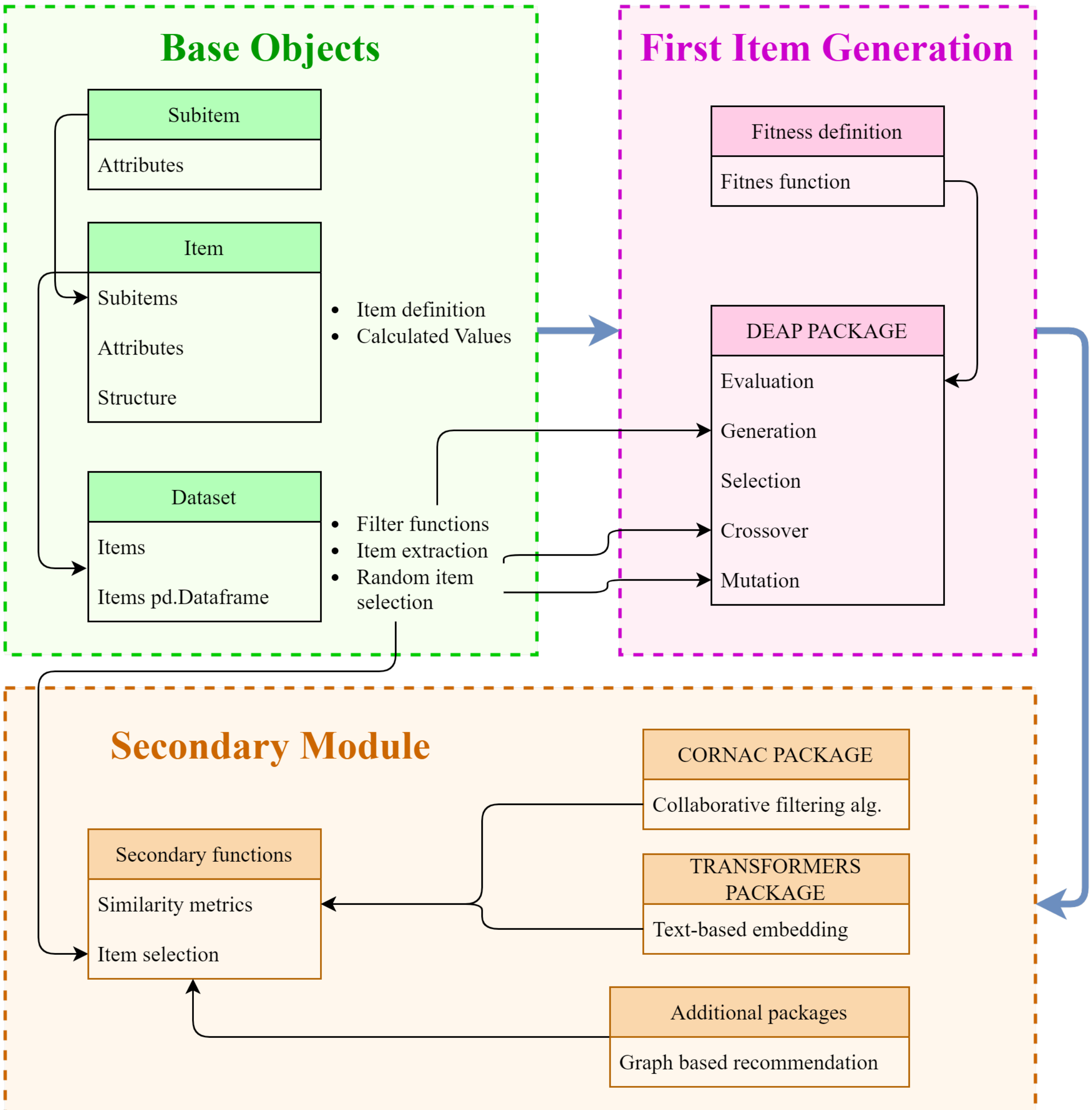

Figure 7.2: Genrecs package and modules structure.

- **Item:** Items are the central element of the package and the foundation of the recommendation. Items have two main attributes in addition to a series of associated functions. First is a dictionary where subitems (based on the aforementioned structure) associated with a key indicating the identifier of the section of the structure they are in. Subsequently, there is the structure attribute, which deserves more independent development.

  The structure is a variable, again based on Python dictionaries for their ease and flexibility. When we select a specific problem, we gather information from experts and users to formulate a structure that will initially be populated with evolutionary algorithms. By disconnecting this structure from the Item, we not only have the possibility to manipulate it but also to establish modifiers and generators based on user histories, a point we will touch upon in the last chapter of the thesis. The structure is, therefore, based on the basic information provided by users, and within its dictionary, the identifiers of each category are placed. Since the structure is complemented with hard constraints, the objective attributes of each category are established through another associated dictionary. For example, if the user indicates a preference for a menu with 3 courses and 4 dishes, three identifiers will be generated (one per course). The structure would then take the form:

  $\{"idBlock_1" : "\#\text{ of dishes}" : 1, "kcal" : estimate, "macronutrients" : estimate\}, "idBlock_2" : \{\#\text{ of dishes} : 2, "kcal" : estimate, "macronutrients" : estimate\}, "idBlock_3" : \{\#\text{ of dishes} : 1, "kcal" : estimate, "macronutrients" : estimate\}\}$

  In the case of a podcast, for instance, if we want a 2-hour program with a 40/60 distribution of comedy and science, the structure might look like:

  $\{"idBlock_1" : \{'category' :' Comedy',' duration' : 40\%total\}, idBlock_2 : \{'category' :' Science',' duration' : 60\%total\}\}$.

  Additionally, the item also receives two more parameters for its creation. The first is attribute information. For each selected attribute in our recommender for adjustment, we introduce a description of how its adjustment will be: approximate to a specific value, within a certain range of values, or approximately binary (has a quality or does not have it). An example of this case could be at the nutritional level:

$\{"Vit_C" : Range, "Kcal" : Accumulative, "tags" : List, "Allergens" : Bool\}$

This description would indicate that the adjustment of *vitamin C* should be considered within an acceptable range and discarded otherwise. The adjustment of *Kcal* is accumulated from all the elements and should approximate a specific value. Additionally, the elements must meet certain categories from the list inside *Tags* (imagine, for example, being marked as vegan recipe in the dataset), and finally, the presence of a specific ingredient in *Allergens* is obligatory not to be present, as bool indicates that every selection should have or not have it. Essentially, all these parameters are the selection of filters for the random selection in the genetic algorithm within the space of all possible subitems. In fact, this filters were coded with their MYSQL translation in the Stance4Health App [198].

Finally, for each identifier, we have the *model's adjusted diversity*. In each block of the structure, models can accommodate one or several subitems. In case the scenario allows a higher number of subitems in each sections of the structure, we can increase the diversity of that part. The most basic case would be a diversity of 1, where a single subitem makes up that section. The most flexible case would be a range of subitems (as it would mean *"this section can contain from 1 to 4 subitems"*).

Items also have a series of basic functions that facilitate simpler evaluations. These functions include the sum of range and cumulative values of the items, the maximum of numerical values, the mean of these values, and the diversity of other categorical variables. In practice, these functions primarily serve as tests and evaluations, as most evaluations functions will be constructed with more complex mathematical formulas.

- **Class Dataset**: The dataset class acts as an additional connection to the external data source. It is designed to be worked through the pandas DataFrame object and constitutes its central axis. One of the pending tasks is to enable it to connect with more databases. However, given the research package's objective rather than being a production library, it is not considered an urgent issue at the moment. What this object does have are fundamental associated functions for the genetic algorithm and the improvement phase:

1. Get Function Indices: In this function, we collect the dataset with variables acting as filters (those with range and bool). This function allows us to obtain a set of ids on which we can make more specific requests in the future. Essentially, these are the elements that can be part of our final item.

2. Extract Subitems: Once we have selected the items, this function uses the previous one to transition from the subitem ids to a set of subitems with all the loaded parameters. This function loads into memory all subitems associated with the provided ids. It is useful if it is necessary to maintain a set of subitems, for example, in the final phase where we can select a subset of items to check if any of them improve user suitability.

3. Extract Random Subitems: In the case of the genetic algorithm, we need a set of subitems that, while complying with the hard rules defined by the problem, are completely random. This function gathers the indices of the elements that can be extracted and uses them to construct subitems. For example, populating an item initially, but also for generating mutations in the genetic algorithm.

7.1.3.1 Primary and Secondary Item construction

The construction of items is associated with the variables defined in the previous objects and forms the method that builds the item from these variables and dataset and subitem objects. This function incorporates all the previous parameters, in addition to the dataset. Through it, we traverse the item structure and progressively fill it with possible items. That is the point were our genetic algorithm schema takes place:

- **Evolutionary Algorithm:**

  There are various evolutionary algorithm packages, so our project has always advocated for code reuse rather than creating another framework that is completely disconnected from the current ecosystem. Instead of that, we have chosen to integrate our objects into DEAP, a well-known library for evolutionary algorithms. This approach allows us, while still maintaining freedom (DEAP allows operating with custom objects), to access all the functionalities of DEAP, including the crossover, selection, or mutation functions already included in its package, and its step-by-step approach for creating specific evolutionary algorithms. Most importantly, this choice facilitates the incorporation of users already familiar with the framework.

  Among all these functionalities, relating them to the selected functions, we initially highlight the generation of individuals, selection, and the selection function:

  - *Generation of individuals:* To generate the initial population and the occasionally needed individuals, we rely on our item construction function, once the structure and objectives are selected. As mentioned earlier, individuals in DEAP can inherit any object structure, so we only need to add ours, and we can work using these dictionary structures.

  - *Fitness function*: we encounter the first difference and begin to understand the package's potential. Being entirely granular, we can define multiple fitness functions and load them into the evolutionary algorithm process based on the experiment we want to conduct. Following the initially created function, we can use functions where quantities are weighted according to their importance to obtain an initial value, while the weights retain the explanatory value of the ranking between objects. However, again, we can use the potential of DEAP, as by modifying only this function, we can transition to a minimization of N-variables, depending on the N objectives we have defined. This style proves useful for obtaining very specific solutions in problems where the structure blocks have lower interdependence (as in the case of podcasts) or lower dimensionality.

It's worth making an additional clarification: this interdependence does not necessarily mean that the structure lacks it. It implies that in the pure process of the genetic algorithm, it is not taken into account. This can happen if it genuinely doesn't need to be considered, or, if it is already considered at the beginning or end of the generation. For example: in the genetic algorithm for nutrition, it may not need to consider intake relationships based on the nutritional pyramid if, prior to generation, these relationships are already considered and are already conditioning the possible indices of recipes that the genetic algorithm can select.

  - *Selection function:* this package allows us to describe the function we want to implement, but at the same time, it also provides several built-in selection functions. This enables us to test different state-of-the-art algorithms and make comparisons, as shown in section 6.3.1, where the comparison between a selection designed from scratch and NSGA-II differs only by commenting or uncommenting a single line of code.

- **Secondary refinement:** Moving on to the second part of the recommendation process, refinement, the package supports various approaches: It allows the creation of a user object with the user's profile. This object is initially calculated considering variables defined as preferences. Following our content-based approach, we select those variables and build a representation of the user through the objects they have indicated. To do this, we divide this representation into categories and variables expressible through numerical vectors.

  With just these two components, we can evaluate the distance between the generated item and the improvement proposals, either using the weighted Jaccard matrix or through cosine similarity. Cosine similarity for textual content is achieved by importing the transformers library, allowing us to define textual fields for comparison through embedding, choosing a Hugging Face model, or importing one stored locally.

Finally, in cases where we have additional information or interactions between users and items, we have chosen to introduce another well-known library, CORNAC. This library allows us not only to reuse the mentioned textual and categorical fields but also the explicit interactions of users to predict ratings. This additional information can be complemented with content-based information or used alongside it. This step is the most recent one, as there are not many datasets that allow us to have all the data sources proposed in this thesis. We believe that the described procedure can help in creating these datasets. The creation of such datasets can, in turn, assist other researchers in formulating new theoretical concepts and validation metrics.

Chapter 8

User validation and trustworthiness

"The real question is not whether machines think but whether men do. The mystery which surrounds a thinking machine already surrounds a thinking man."
Contingencies of Reinforcement; A Theoretical Analysis B.F. Skinner*, 1969*

At this point in the thesis, we already possess tools to address the recommendation of structured items in complex scenarios. However, regardless of the performance of our system, a central question arises: In what way do metrics reflect the performance of our system for a user? We can measure the closeness to the preferences, or if some values are effectively predicted from the already-seen data. But this metrics reflects how algorithms works, not how users think [218]. This question is pertinent because recommendation systems are not intended to be innocuous information services. Recommendation systems filter information and alter our perception of reality. While it is debatable whether their role is to suggest or propose (the distinction here implies a intention in the latter for the suggestion to be accepted), the final decision (accepting a recommendation or not) depends of the user and understanding how this decision is made is important [219], [220].

Understanding the psychological aspects of a user choice is a multidisciplinary problem, but it is crucial in some of the scenarios outlined in the thesis: if my recommendation system is built to offer you healthy dishes, its implicit goal is to facilitate your access to such dishes, and it will work better the more healthy dishes you consume (remember the implications of this in the content-based multi-objective system proposed in this work). Several aspect may help for this task (as we will see during the chapter), but specially for health or other related issues, our focus is centered in improving the trust our user may have on the algorithm. It is therefore legitimate to think that I can enhance the performance of my system by complementing the output with some form of extra algorithm step if it aids or enhance the trust our users have in our recommendations.

In this section, we will delve into this hypothesis, trying to elucidate a method to improve the probabilities of our recommendation to be accepted by the user, from a psychological point of view. For that task we will start describing how we define what a trustworthy recommendation system is, then we will deep in the psychological process behind the user evaluation of this trust, and what factors or strategies can improve it. We will analyze some of this factors from the S4H European Project. Finally once one of these strategies is selected, we will describe how we developed and integrated it into our recommendation system workflow.

8.1 Trustworthy recommendation systems

What should be considered trustworthy when it comes to recommendation systems? Providing an entirely objective answer to this question is challenging, but given the potential impact of recommendation systems, it is crucial to address it before proceeding with the chapter. To do so, we will approach the question from two slightly different perspectives. The first one focuses on the European directive aimed at promoting trustworthy artificial intelligence [221]. This work is analyzed from a legal standpoint in [222], emphasizing four fundamental principles (in addition to seven key aspects) around which trustworthy artificial intelligence should revolve: respect for human autonomy, prevention of harm, fairness, and explainability. These four objectives can be understood in relation to recommendation systems as well.

- **Respect for human autonomy:** Individuals engaging with AI systems should retain complete and effective self-determination. These systems should be crafted to enhance human cognitive, social, and cultural skills. Our system them, should not be used to impose any decision, as their role is to suggest.

- **Prevention of harm:** AI systems must not cause or amplify harm and should refrain from adversely affecting human beings. Harm should understood here as both physical and mentally. Recommendation systems, then, specially in complex situation should have measure to limited the worst-possible scenario.

- **Fairness:** Ensuring the equitable and just distribution of both benefits and costs is vital, along with the commitment to liberate individuals and groups from unfair bias, discrimination, and stigmatization. The prevention of unfair biases within AI systems holds the potential to contribute positively to societal fairness.

- **Explainability:** AI-based systems should run through transparent processes, open communication regarding the capabilities and objectives of AI systems. Without such information, the ability to contest a decision in a meaningful manner is compromised. Therefore our systems should be auditable and , if possible we should be able to explain how they reach certain outputs.

Although this line are design for all AI-based systems, recommendation systems have continued to develop a fundamental theoretical framework for analyzing their risks and their contribution to ethical consumption. Consequently, there is already abundant literature analyzing the risks of these systems and potential countermeasures. We will utilize the six keys proposed by [223] to provide a final theoretical framework from which to understand what is considered trustworthy and in what aspects the trustworthiness enhances our technological solution. This framework is derived from the general guidelines and the relationship between both approaches can be seen in Figure 8.1. According to [223], we can divide the most crucial dimensions in achieving trustworthy recommender systems into six aspects:

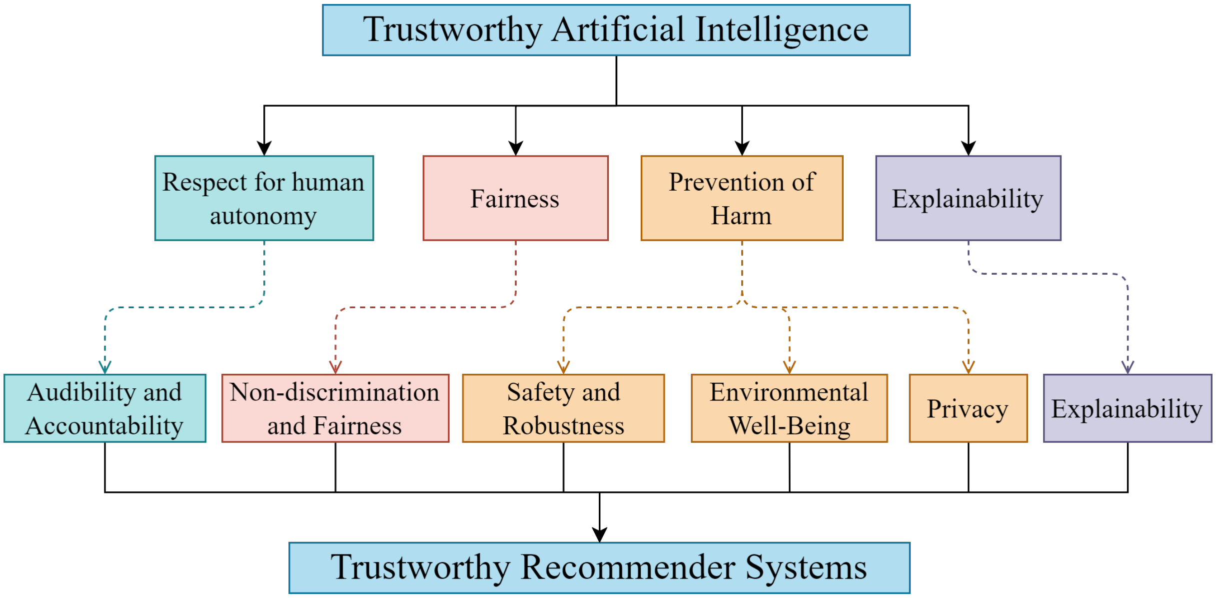

Figure 8.1: Relation between the definition of trustworthy Artificial Intelligence and Trustworthy Recommender Systems

1. **Safety and Robustness:** This focus on creating measures and tools to ensure that the system operates securely, reliably, and without causing harm to users or unintended negative consequences. This aspect directly relates to the principles of autonomy and harm prevention or so-called attacks. It encompasses all attacks that may disrupt the correct functioning of the recommendation system, causing it to offer recommendations that, in their context, are considered harmful (whether specific or not). Additionally, it refers to the system's resilience capacity in the face of such situations. A survey on this area can be found in [224], [225], where several attacks and other ill-data injections are described.

2. **Non-discrimination and Fairness:** In this case, we emphasize the principle of fairness. For a system to be trustworthy, we understand that it must provide recommendations of the same quality for all individuals who have access to it, without discriminating against individuals or protected groups [226]. Not all systems are free from this bias, which is why it is essential to study and delineate it properly. In our nutritional case, for example, it would be unrealistic to assume that our system

can recommend a balanced diet for a person with no access to a variety of different ingredients or several allergies (even with same characteristics even cultural aspect may affect [227]). Increasing the fairness is also a direct result of exploring different scenarios and having analysis and control mechanisms that reassess recommendations (as we explore in Chapter 5). And for that, it is crucial to have the assistance of experts in the field.

3. **Explainability:** In the field of artificial intelligence, the concept of explainability is highly debated and not entirely delimited in terms of what constitutes or contributes to the explainability of a system. It connects with the fourth principle of the guidelines. The explainability of a system refers to its capacity to clarify or present information in a way that is understandable to a human [228]. In the context of recommendation models, a system possessing this explainability is named as explainable recommendations model. These models not only offer users recommendations but also provide explanations for why these recommendations are made. One can understand that if we focus on the end-user of our recommendation systems, evaluating from a subjective standpoint the presence of explainability mechanisms is fundamental for the end-user to understand that they can trust this system.

   Once again, this area has a considerable scientific production focused on the various aspects involved in being explainable. It is in this aspect where our approach shines above other algorithms, coupled with an extensive amount of expert knowledge embedded in the system. We can make our system to be more explainable, therefore in theory, it should be more trustworthy.

4. **Privacy:** It alludes to the principle of preventing harm to individuals. The privacy aspect of a recommender system pertains to the consideration and protection of users' personal information and sensitive data throughout the operation of the recommendation process. Privacy is a critical concern in recommender systems, as they often handle large amounts of user data to provide personalized recommendations. In collaborative systems, user profiles are frequently created based on various

types of data, including browsing history, purchase behavior, and demographic information. Disclosing this information can release sensitive user data, especially when dealing with health-related data. Furthermore, the universal requirement for new data sources with increasingly detailed information can be a critical point and may conflict with this aspect. Hence, ensuring effective anonymization is crucial to prevent the identification of individuals and protect their privacy when we release information about our recommendation algorithms. A more in-depth exploration of the cryptography of recommendation systems is developed in [229].

5. **Environmental Well-Being:** Once again, an aspect related to harm to humans, due to the environmental impact of research in these systems. This is exacerbated by the widespread use of techniques based on neural networks for obtaining embeddings or generating new predictive algorithms. All of this implies a greater demand for resources, offering two points of focus: model compression and model acceleration. Model compression is designed to minimize the size of recommendation models. A path very similar to current development in large language models. Acceleration techniques, on the other hand, focus on reducing the time required for training or inference.

6. **Audibility and Accountability:** This aspect aligns with the first principle, autonomy, and the prevention of harm. Systems, especially those utilizing health data from patients, should adhere to both principles. However, we often overlook that the rest of commercial systems should also comply with these guidelines. There are however noticeable differences between accountability and audibility, so we proceed to define both:

   Accountability: involves the clear assignment of responsibilities for the actions and outcomes of a system. It requires individuals or entities involved in the system's development, deployment, and operation to be answerable for their decisions and the system's behavior. *Audibility*, on the other hand, pertains to the system's capability to be comprehensively monitored, traced, and audited. This involves maintaining a transparent and traceable record of the system's processes, decisions, and data flows.

This thesis itself serves as an example of accountability, as it details the design decisions of the system, its strengths, and areas for improvement. Additionally, the proposed system offers a notable auditability capability where different options for creation and evaluation (through metrics and fitness) can be considered to track whether a machine is making an erroneous judgment.

Given the technical definition and studies conducted on the aspects that make a recommendation system trustworthy, our system has the potential to detect situations where it cannot provide coherent responses and mitigate them. We would achieve this by evaluating various scenarios of the model and reinforcing situations where the performance is low with expert knowledge. In the final chapter, we will provide an analysis of the environmental impact of the model. Lastly, we will delve into explainability.

Our model processes a significant amount of specific information that can help improve explainability. However, there are many possible avenues for action [228]. We may not necessarily lean towards the best measure, as we must not forget that our ultimate goal is to make users more likely to accept the recommendation because they have more confidence in the system. What types of explanations help a user trust the system more? To answer this question, in the next section, a result of the research visit to Aarhus University, we will delve into the psychology of a user and how it can be measured.

8.2 Psychology of end-user

To study the crucial features influencing users' interactions with artificial intelligence applications in general and recommendation systems in particular, technology acceptance models are employed. These models propose an interrelated scheme that emphasizes a set of variables monitored by the application and a set of user-extrinsic variables that modify their behavior. The primary objective of these studies is to comprehend which of these variables impact the user's application usage and investigate the relationships among them to predict their evolution over time.

Before delving into these models, we begin by describing the variables considered and how they relate to recommendation systems:

8.2.1 Personalization, Engagement and continuous usage

Personalization [230], [231] is the process of tailoring information, experiences, or recommendations to individuals based on their unique preferences, characteristics, or behaviours. It involves utilizing user data and insights to deliver content that is specifically relevant and customized for each user.

Personalization has emerged as a pivotal factor in enhancing user experiences and driving user engagement [232]. The ability to produce these extreme personalized recommendations is the key differential factor of these systems and its potential to change user behaviours [233], [234] effectively boosting user engagement and foster repeat app usage.

Furthermore, personalization enables adaptive user experiences, wherein apps dynamically adjust the user interface and features to accommodate individual interactions and preferences. Health recommendation apps [235], [236] are an example, for instance, adapting food recommendations based on user daily routines and preferences. The implementation of these personalized approaches within mobile apps facilitates the delivery of tailored and again, engaging experiences [235], thereby enhancing user satisfaction and encouraging sustained app usage. It is also worth noting that not all studies strictly linked maximum personalization with maximum engagement, specifically in health recommendation apps [237].

At the same time, all these factors introduce new dynamics and challenges, such as concerns over privacy [238], trust in AI algorithms, or even the degree of anthropomorphism [239]. Despite the growing importance of understanding consumer attitudes and behaviours towards AIRecSys, limited studies have specifically focused on them (table 1). Furthermore, there are still low levels of consensus on which kind of strategies work best, and there is no one-fit-all approach [240]. Thus, we need to dig deep into the correct strategies on how to use these systems (and therefore its personalization capabilities) to increase the engagement and continuous usage of these apps.

Continuance usage, as described in [241], [242], is defined the sustained utilization of a product by individual users beyond the initial adoption phase. This indicates that users perceive value in the product and actively choose to incorporate it into their regular activities or routines.

On the other hand, engagement, as outlined in [243], [244], involves ongoing interaction between users and a product. It encompasses various facets of user participation and involvement: personal engagement, such as personal interest, motivation, and attention, as well as interactive engagement, which includes active communication, collaboration, and feedback. Assessing the level of engagement of users in an AI-based app allows us to gauge their degree of interest, satisfaction, and connection to the system they are using [245].

Both concepts are vital for comprehending user behaviour and assessing the long-term success and impact of an app. It is worth noting that continuance usage can be considered an outcome of engagement since users who are engaged with a product are more inclined to continue using it over time.

There are multiple models that could provide a suitable theoretical framework for recommendation systems but, for our purpose, we will introduce two foundation models, along with one specifically designed for recommendation systems. A connection between this factors and it measures es presented in Figure 8.2.

8.2.2 Theoretical models for technology acceptance

The majority of studies focused on exploring consumer attitudes towards mobile apps aside their area of application rely on the Technology Acceptance Model (TAM) [232], [246], [247]. At its core, TAM suggests that users' behavioural intention to use a technology is influenced by their attitude towards the technology, which is in turn influenced by factors such as perceived usefulness (PU) and perceived ease-of-use (PEOU).

Perceived usefulness (PU) refers to the degree to which individuals believe that using a particular technology will enhance their performance or help them achieve their goals. If users perceive a technology to be useful in fulfilling their needs, they are more likely to have a positive attitude towards it. On the other hand, perceived ease-of-use (PEOU) relates to the extent to which users believe that using the technology will be effortless and free from complexity. If a technology is perceived as easy to use, users are more likely to develop a favourable attitude towards it.

Since its inception, the TAM has undergone several revisions and extensions. We specially highlight the Unified Theory of Acceptance and Use of Technology (UTAUT) [248]are major advancements in the model. UTAUT provides a comprehensive framework that includes various determinants of technology adoption (Expectation of performance, Expectation of effort) as well as demographic data and is widely applied on ai-based technologies like voice assistant and e-commerce services.

Another remarkable extension is the UTAUT2 model [249] with extracted factors of the original UTAUT model for the consumer context and extended it by incorporating the following three factors which improved the prediction of behavioural intention and use behaviour, based on new technology trends (Hedonic motivation, Price value where it is applicable and Habits).

Although these models can be used to evaluate our system, there are also specific proposals in the scientific literature for recommendation systems, such as [250]. In this model, two major areas of evaluation are distinguished in recommendation systems. On one hand, there are those that assess the system's accuracy and the effort required to achieve an acceptable accuracy, which connects with the previously described PU (Perceived Usefulness) and PEOU (Perceived Ease of Use) variables. On the other hand, it emphasizes the user's trust in the system's recommendations and their intentions once they have received and evaluated the recommendation.

How do we combine the measure of the effectiveness of our app along with the effectiveness of our algorithm? If we focus on common data points, certain values of the app

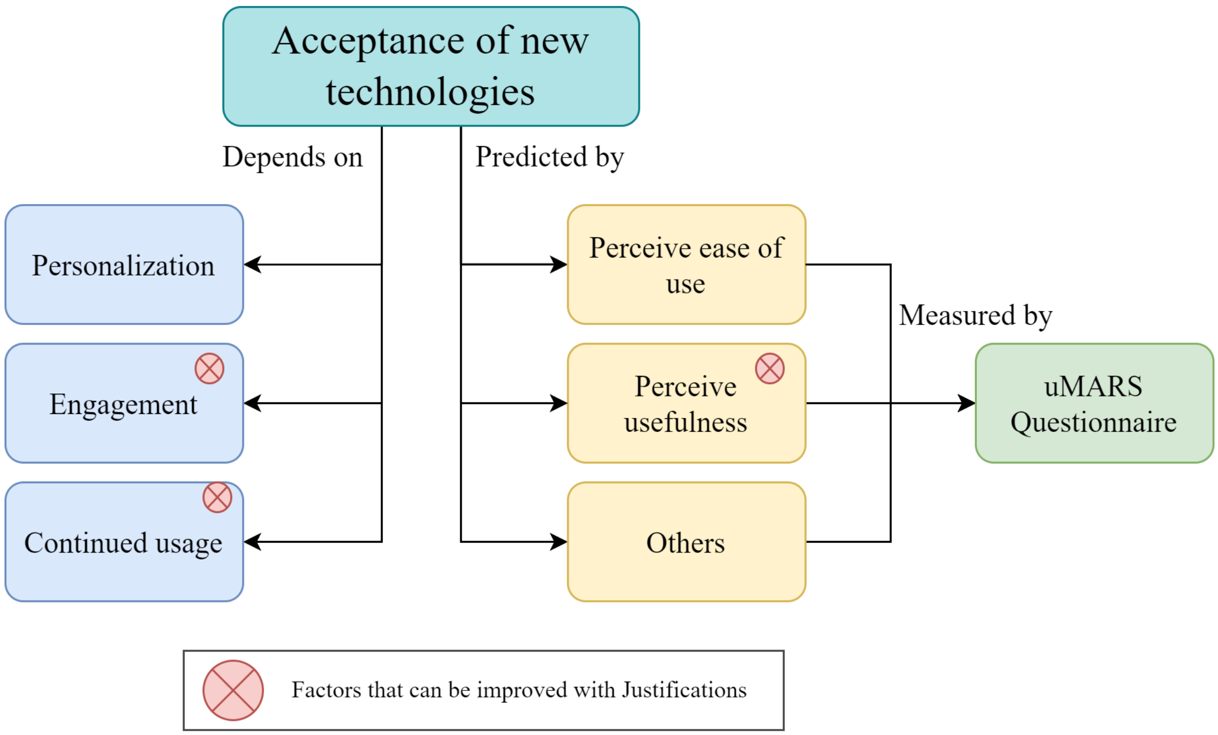

Figure 8.2: Relationship between Technology acceptance models and its factors. The ones that can be improved by creating justifications are marked.

may point out. First we may choose a test that able us to derive interesting insights. For this task we selected survey called uMARS questionnaire [205], [251]. uMARS questionnaire contains a series of questions divided in six different sections. An essential step in this process is aligning the constructs from the Technology Acceptance Models with the relevant uMARS items. This strategic alignment ensures that we are measuring the same or closely related aspects of user perception and satisfaction. For instance, when examining the TAM construct of Perceived Ease of Use (PEOU), we can seamlessly pair it with uMARS items that delve into ease of navigation, user-friendliness, or the application's learning curve (SECTION A, B,C). Similarly, when evaluating Perceived Usefulness (PU), uMARS items pertaining to the value of the app's features, its alignment with users' needs, or its overall utility (SECTIONS E and F).

8.2.3 Results on Stance4Health App

Back to the S4H app we perform the uMARS questionnaire to all users in the first trial.The APP was used by different team members, collaborators and test subjects in user mode. During the continuous stress and performance testing of the server, no problems were found, except for one that was due to server downtime. The APP works in an asynchronous way to reduce the loading times, which means that usually any change is stored in the server and performed independently. In these cases, the APP notifies the user that a change is going to be made, then the modification is performed in the server, uploading the results as soon as they are ready. Taking all this information into account, the spent time for a whole weekly menu calculation on the server-side was $24s + / - 1$. The results of the uMARS questionnaires are depicted in Figure 4. The most valued by users was the information (with a score of 4.25/5), giving importance to the quality and credibility of the information included in the APP. Aesthetic appeal was the second most important and engagement was in last place (with a score of 3.68/5). Overall, the average quality score of the APP from users was 3.97/5. In addition, 88 of users would recommend the APP. Furthermore, 80 also reported that the use of the APP increased their nutritional literacy. After that we also perform the questionnaire to all users in the human trial.

8.3 Algorithmic approach to Improve trustworthiness

After reviewing the aspects discussed in the previous sections regarding our recommendation system, we contemplate how we can enhance user adherence to different diets. If we consider all the aspects highlighted in the psychology of the end user, we find many interesting points where we can increase the score of these systems. The challenge lies in engineering terms, as we must focus on points that allow for the procedural generation of options and can be implemented through a technological process that integrates them into the app's functionality. In other words, improving the front-end will increase the explainability of our system but is not a purely technological process where we can contribute programmatic ideas.

On the other hand, as various studies suggest, improving the explainability of the system is a procedure that links with the pure generation of content and aligns with our system. This procedure, whatever it may be, directly assists and impacts the parameters we have highlighted within the psychological explanations of user adherence in various ways. Firstly, user trust and satisfaction are expected to increase as the system provides personalized justifications for the recommended items, enhancing transparency and auditability. The system's ability to offer explanations fosters user involvement and educates them on the internal logic behind recommendations, thereby improving user understanding and satisfaction.

Additionally, the introduction of an explanation AI system contributes to the effectiveness and persuasiveness of recommendations. Users are more likely to be convinced to purchase recommended items when they understand the reasoning behind the suggestions. The efficiency of the system is also positively impacted, as users are provided with information that helps them make informed decisions, reducing uncertainty and streamlining the decision-making process.

Moreover, the organizational interfaces of explanation-based recommender systems have been shown to be particularly effective in promoting user satisfaction, convincing users to make purchases, and encouraging them to return to the platform. Overall, the incorporation of explanation features enhances various dimensions of user evaluation, creating a more user-friendly and trustworthy recommendation experience.

Increasing the explainability of the system is also aligned with one of the objectives of artificial intelligence systems by major government agencies, as we could see in the first section.

In complex systems, especially those based on user health, it has been demonstrated [252] that generating health-based justifications has a significant impact on user decisions, in addition to enhancing system explainability. How those justifications where generated? Most studies opt for preformed constructions, which are templates that allow for some variation but have been manually generated.

In contrast, we sought to represent a change in this paradigm. We decided to automatically generate justifications within our systems based on health knowledge and the nutritional evaluation of dishes. The latter step is algorithmically obtained effortlessly thanks to our fitness functions, which must evaluate each menu. However, it is necessary to convey this knowledge in a user-understandable manner, something we will achieve through natural language processing systems and retrieval-augmented systems.

8.3.1 Implementation of our proposal

8.3.2 Question generation

The core component of our system for generating explanations involves converting any selected recipe recommendation into a question that mirrors the user's uncertainties. This approach is rooted in the idea that when we receive a recommendation, our underlying query often revolves around the specific attributes that distinguish the recommended item, making it superior or inferior to other choices in our particular context. However, not every user can identify the question that can effectively aid in interpreting the results. This is the fundamental innovation of our system, as it seamlessly translates the user's concerns into natural language questions.

Particularly, let focus on health-centered systems (multi-objective). In this case, the question underlying a user that want to know why they are being recommended something would be similar to: *What characteristics does this recipe have to make it beneficial for my health?* This question also admits another reading, which is: *What characteristics does this recipe have to help me achieve a certain health-oriented goal?*

However, that question is a generic question that does not add really any value and probably would need a very long answer to fulfill its purpose. In addition, most of the knowledge extraction systems we can build would result in a less useful explanation. On the contrary, this question can be further transformed if we process the recommended recipe and obtain its main characteristics and how they relate to the rest of the recipes in the dataset (a small selection). Thus, if a recipe has been recommended in a multi-objective system, it is because some characteristics make it more desirable than others. These characteristics give specificity to the question.

To give a specific example of the procedure if we select a recipe that stands out for its amount of vitamin C (the reason why the system has recommended it over another), the question would result in: *What benefits on my health/objective does vitamin C provide?* This question is more specific and directly refers to quantifiable aspects of the recipe.

8.3.3 Retrieval Augmented Systems

Once the question is generated, we need to produce the answer from expert sources. Two similar approaches have been followed, with the last generation engine being the only difference. However, before proceeding, let's introduce the concept underlying this generative application: retrieval-augmented systems.

These types of systems emerged as early as 2018 when it was observed that non-specific language models generated short responses, and information retrieval models did not produce comprehensible answers [253]. If we start with a text generation problem given a specific context in an input sequence xx to an output sequence yy, the conventional formulation is $y = f(x)$. However, in retrieval-augmented generation, as defined in [254], this formulation is extended to include a set of relevant instances $z_1...z_2$ retrieved from external sources, expressed as $y = f(x, z)$.

The set $z_1...z_2$ typically consists of pairs

$$<x_r, y_r>$$

, where x_r represents instances retrieved from sources such as the training corpus, external datasets, or large-scale unsupervised corpora. The main idea behind this paradigm is that the retrieved instances y_r can provide valuable information during the generation process if they are similar or relevant to the input x.

This approach has already been established in medical contexts as [255] or [256] but its application is recent and there is still an active research field.

In our case, we have large amounts of verified information, and we cannot afford to produce erroneous justifications. Therefore, Retrieval-Augmented Generation (RAG) systems emerge as a great option for designing our automatic justification system. Furthermore, these justifications are based on scientific consensus, and their health-focused approach is also endorsed as one of the primary drivers to 'seduce' users. In the following two sections, we elaborate on how and what technologies we have implemented for this purpose.

8.3.4 Paragraph selection

Once the set of different questions are selected, we can choose which one will be used as our objective. Generally, we could prioritize questions based on the following scale: the nutritional pyramid, macro-nutrients, and micronutrients. We focus on the latter two cases. Before searching for important paragraphs in the text, we break down the relevant text chunks into shorter paragraphs. This is because we need specific information to adequately answer the question but also because we need to locate those specific smaller sections due to the inability of the transformer to support a big number of tokens to be processed.

With the text properly divided, we conducted a semantic search to determine the similarity between the paragraphs related to the nutrient and our question. For this purpose, we used the *multi-qa-mpnet-base-dot-v1* transformer model, which maps sentences and paragraphs to a 768-dimensional dense vector space and was specifically designed for semantic search. We deployed it using the sentence-transformers[217], [257] library and HuggingFace. The selection of this model prioritizes the size of the embedding and the performance of the benchmark models. This model used the pre-trained mpnet-base model with specific training on 215M (question, answer) pairs. The evaluating function was Multiple Negatives Ranking Loss using CLS-pooling, dot-product as a similarity function, and a scale of 1. More details on their model card [258].

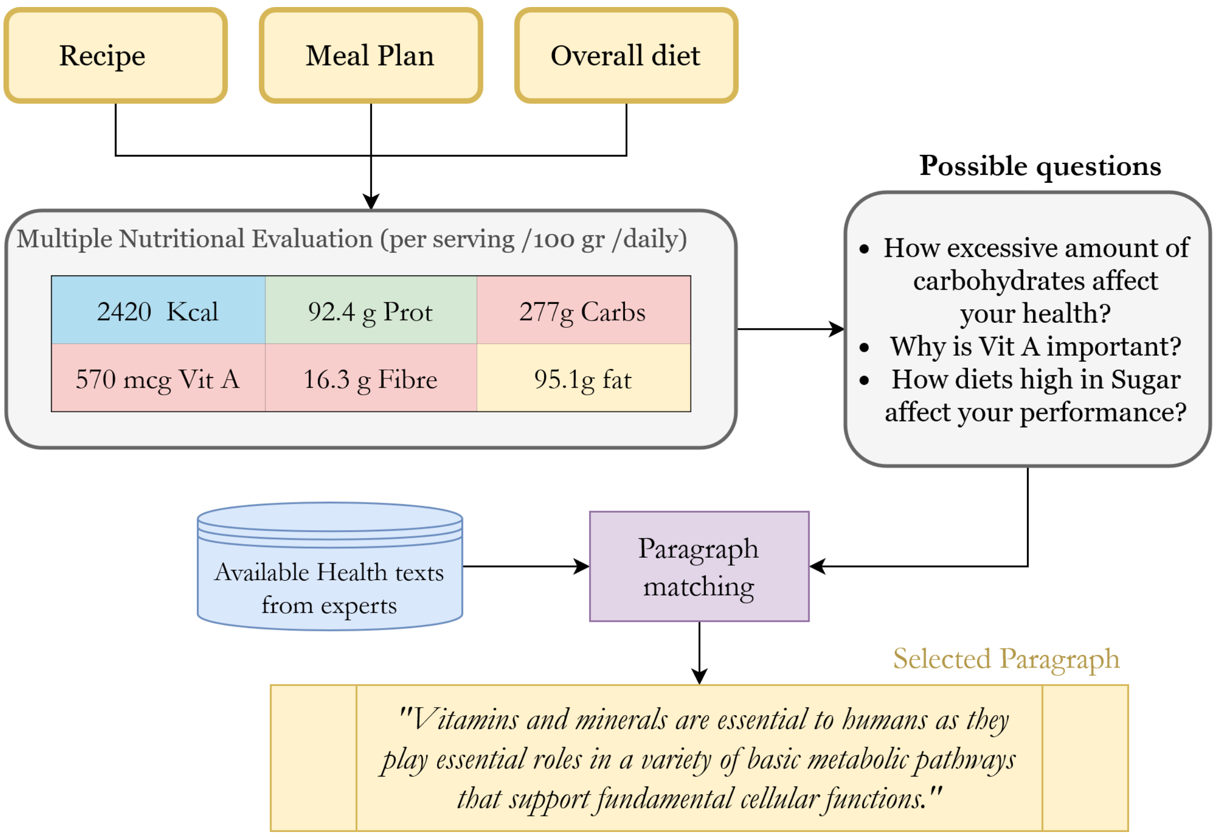

Figure 8.3: Example diagram of the system pipeline based on the nutrient evaluation from USDA.

8.3.5 Question answering

In our methodology, we implement a dual-step process for answering questions. Initially, we collect a set of paragraphs containing relevant information. Subsequently, we employ two distinct models to generate answers. The first model relies on the distilled version of the Roberta-base model[259], fine-tuned using the SQuAD2.0 dataset[260], particularly the *tinyroberta-squad2* variant [261], known for its proficiency in Question Answering. This model has undergone training on diverse question-answer pairs, including unanswerable questions. Evaluation of the selections is based on a confidence score, indicating the level of confidence in the predicted answer.

As final set, if we wanted to use the answer for production ready services, we would extract the whole sentence in which the answer is found. This would let us use correct grammatical sentences while assuring no generative process is taken place.

Simultaneously, we pursue an alternative approach employing a large language model based on Mistral7b [262]. This is a small Large language model, a fine-tuned version of the Mistral-7B-v0.1 generative text model using a variety of publicly available conversation datasets. Notably, we differ in our process by using the paragraph as the contextual backdrop for the prompt sent to the Mistral model. This approach aims to minimize the possibility of hallucination and enhance the readability of the responses, as it grounds the model's understanding in the specific content. Similar to the previous model, evaluation involves a confidence score ranging from 0 to 1, where a higher score reflects increased confidence in the accuracy of the generated answer.

8.3.6 Results

In this section, we present the main results of the proposed system in the article. Initially, we selected a corpus of medical texts described in section 3. We used articles from NHI[263], EFSA[264] and EUFIC as [265], [266] as the text corpora.

We do not focus on a specific set of recipes of recommendations. Instead, we analyzed the whole public MealRec dataset, searching for those nutrients that appear in higher o lower quantities from a healthy standard diet (Using a user based on a young male adult with an expenditure of 2000 kcal per day). Those nutrients will be the ones targeted by our justifications, either because we recommend them and want to encourage their consumption or because the user selected them and we would like to discourage their intake.

For the micronutrients, we made a selection based on three main reasons: the existence of reference material on both pages, its disponibility and their distribution. We follow a general guide by the FDA where below 5 % of the daily recommended quantity is considered "low in a nutrient" and above 20 % is considered "High in a nutrient". Based on these results select specific nutrients, as users need to understand why those recipes are/are not recommended. Following this guide we started evaluating the following nutrients :

Sugars (g), Sodium (mg), Carbohydrates (g), Vitamin B6 (mg), Calories (kcal), Thiamin (mg), Fat (g), Calcium (mg), Dietary Fiber (g), Magnesium (mg), Iron (mg), Protein (g), Vitamin A - IU (IU), Potassium (mg), Saturated Fat (g) and Vitamin C (mg) as a selection that represents different nutrients with different nature (vitamins, minerals, macros, etc). We discard Folate (mcg), Calories from Fat (kcal), Cholesterol (mg) and Niacin Equivalents (mg) because lack of data in the recipes.

The results of the nutritional evaluation can be seen in 8.4. Pairing this nutrients with their nutritional evaluation, we focused on dietary fibre, calcium, potassium and vitamin C, as they are not present in recipes and any health-centered recommendation is likely to recommended higher amounts. We also detect excesses in saturated fat, sodium, Iron and carbohydrates. Moreover FDA and WHO have regulative information on saturated fat and sugar, so we also tried to find a justification for not recommending them for discouraging users that take them. Finally, suppose we selected a nutrient N, we made the following questions: *Role of N in health? Effect of N/N deficit in health?, Why are N beneficial for health?* trying to maximize the different formulations used in the text to describe the effect of nutrients in health.

Nutrient	Question	Selected Text for Answer	Confidence Score	Final Sentence from text
Calcium	Role of calcium in health?	To form and maintain strong bones and teeth	0.4562	One of the key roles of calcium, together with phosphorus, is to form and maintain strong bones and teeth.
Calcium	Effect of calcium deficit in health?	Increases the risk of rickets	0.4356	In children, calcium deficiency increases the risk of rickets, a disease that makes the bones softer and weaker.
Carbohydrates	Why are carbohydrates beneficial for health?	Essential for the proper functioning of the body	0.8055	Carbohydrates are one of the three macronutrients in our diet and are essential for the proper functioning of the body.
Fiber	Effect of fiber in health?	Decreases the risk of cardiovascular and coronary heart disease as well as the risk of obesity	0.2399	Fiber decreases the risk of cardiovascular and coronary heart disease, as well as the risk of obesity.
Iron	Effect of iron in health?	Preventing iron deficiency anemia and related problems	0.4098	Iron's most important contribution to health is preventing iron deficiency anemia and related problems.
Iron	Role of iron in health?	Growth and development	0.3631	Iron is a mineral that the body needs for growth and development.
Potassium	Effect of potassium in health?	Helps in the digestion process	0.2454	Potassium also aids in the digestion of foods by supporting the release of saliva and gastric acids, facilitating the digestion and absorption of proteins and carbohydrates.

Table 8.1: Set of results for the unsupervised search using tinyroberta-squad2

Nutrient	Question	Selected text for answer	Confidence Score	Final sentence from text
potassium	What is the effect of potassium deficit in health?	blood pressure	0,666	Having the right balance of these three minerals in our diets – particularly by making sure we eat enough potassium and keep our salt intake within the recommended values – is key to support a healthy blood pressure.
saturated fat	role of saturated fat in health?	one of the most hotly debated areas in nutrition	0,3183	Saturated fat is one of the most hotly debated areas in nutrition.
sodium	effect of sodium in health?	one of the major minerals	0,5378	Sodium is one of the major minerals, which our bodies need in relatively larger amounts to keep healthy.
sodium	What is the effect of sodium deficit in health?	cause our bodies to remove excessive amounts of this mineral	0,2856	Sodium deficiency is mainly associated with metabolic disorders or specific health conditions (such as severe episodes of diarrhoea or kidney malfunction) that cause our bodies to remove excessive amounts of this mineral.
vitamin c	effect of vitamin c in health?	can help our bodies absorb more iron	0,2358	Vitamin C can help our bodies absorb more iron from plant-based foods (non-haeme iron) which is less absorbed than iron from animal sources (haeme iron).
vitamin c	What is the effect of vitamin c deficit in health?	scurvy	0,5848	People who get little or no vitamin C (below about 10 mg daily) for many weeks can get scurvy.
vitamin c	Why is vitamin c beneficial for health?	helping to protect cells from the damage caused by free radicals	0,1741	In the body, it acts as an antioxidant, helping to protect cells from the damage caused by free radicals.

Table 8.2: Second set of results for the unsupervised search using tinyroberta-squad2

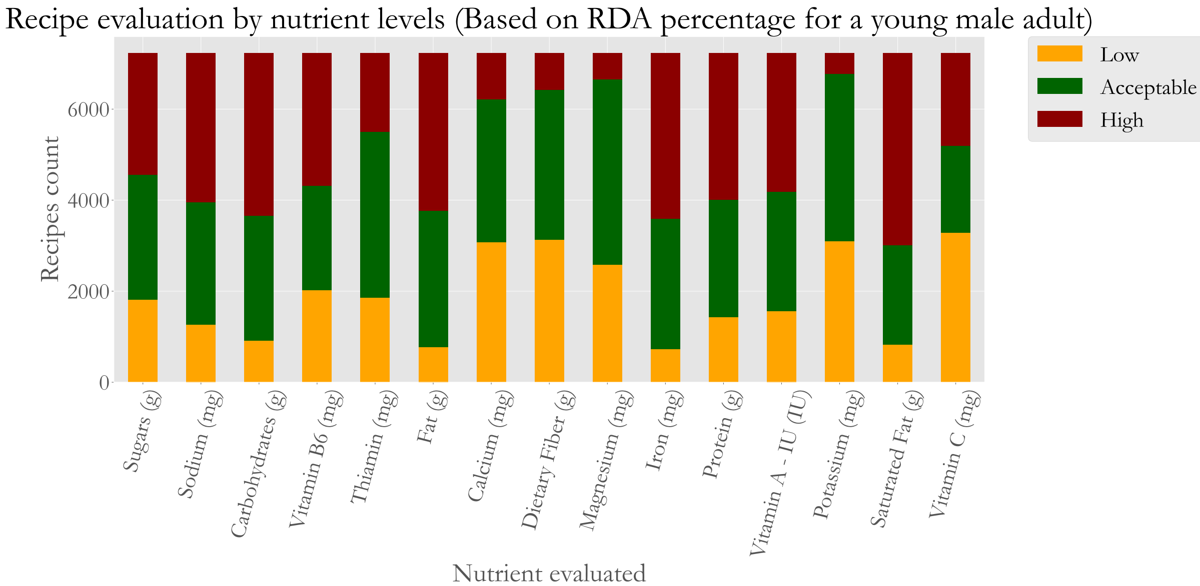

Figure 8.4: A nutritional evaluation of recipes from MealRec based on USDA parameters.

Regarding performance, all consulted texts contained the necessary information to appropriately answer the questions. Although many nutrients could yield satisfactory responses at the level of complete sentences, the system still exhibits considerable variability both across nutrients and questions. On a positive note, we only selected responses with a confidence level greater than 0.2. In Table 8.2, it can be observed that when a response cannot be found, confidence decreases, making it a robust tool for ensuring response coherence but not infallible.

The results also suggest that nutrients with a more direct relationship to health are more easily retrievable. Conversely, others, such as saturated fats, exhibit a clear negative association with health but through more intricate mechanisms (e.g., cholesterol). This implies that the context necessary for an accurate response is larger than our initial encoding capacity, leading to the omission of response details and the selection of erroneous phrases. These issues may be mitigated through further training on the specific corpus (as part of the initial phase) and by increasing the quantity of available texts for retrieval.

Examples for the highlighted micronutrients with the tinyroberta-squad2 approaches can be seen in Tables 8.1 and 8.2.

On the other hand, using the Mistral7b approach, a set of the results can be seen in tables 8.3, 8.4, 8,5 and 8,6. We used the following promtp:

Human: You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question. If you don't know the answer, just say that you don't know. Use two sentences maximum and keep the answer concise. Question: question Context:context Answer:

Where the *question* was replace by one of the chosen one previously and the *context* where the selected paragraphs found through the $multi-qa-mpnet-base-dot-v1$ model.

We were able to introduce more context and the language model transform it, leading to a final generated output that do not need any changes or finding the sentence in the text. The extracted information was able to collect information from several paragraphs and produce an understandable sentence, more profound and interesting than the previous ones. Moreover we introduce a conditional context, where the model is prone to warned us if there were no information in the provided context to be able to answer our question properly. There was still some cases where the information was not accurate due to language errors in the encoding phase, for example in the benefits of Sodium a text was selected omitting the word "reducing" at first, which totally change the sentence meaning ("Sodium helps ..." instead of "Reducing sodium helps..."). This kind of errors can be controlled by crossing multiple sources and differentiating between trends in consumption (reducing / increasing diets).

Nutrient	**Benefits**
Calcium	Calcium is important for bone health and muscle strength. It can be found in dairy products, leafy green vegetables, nuts, fish, and fortified foods.
Dietary Fibre	A diet rich in fibre can decrease the risk of cardiovascular and coronary heart disease, reduce obesity, lower blood pressure and LDL cholesterol, facilitate weight maintenance, improve bowel movements, and impact our gut bacteria in a beneficial way for long-term health benefits.
Dietary Fibre	The benefits of dietary fiber in health include decreased cholesterol and increased numbers of beneficial bacteria.
Iron	Iron has several benefits in health. It is essential for the production of haemoglobin and myoglobin, which help transport oxygen throughout the body. Vegetarians and vegans should pay attention to their iron intake, as they mainly consume non-haem iron from plant-based sources.
Iron	Iron helps prevent anemia and related problems.
Carbohydrates	Carbohydrates provide energy to the body.
Carbohydrates	Carbohydrates provide energy for the body and support brain function. They also play a role in maintaining healthy blood sugar levels. High-quality carbohydrates, such as those found in legumes, fruits, and vegetables, can also improve digestion and overall health.
Vitamin C	The benefits of Vitamin C in health include supporting the immune system, producing collagen, and helping the body make hormones and neurotransmitters. It can be found in citrus fruits, vegetables, meat, fish, kidney, liver, parsley, sorrel, and chives. The recommended daily intake varies based on age, sex, and life stage.
Potassium	Potassium has several benefits in health such as helping muscles and heart contract, nerve cell communication, water balance, substance transportation, pH regulation, blood pressure, digestion, and maintaining the right balance with other minerals like chloride and sodium.

Table 8.3: Set of results for the unsupervised search using Mistral7b

Nutrient	Benefits
Sodium	Sodium helps maintain healthy blood pressure and supports muscle and nerve cell function. However, consuming too much sodium can lead to high blood pressure and negatively impact calcium balance in the body. The recommended daily intake varies based on age, sex, and life stage.
Sodium	Sodium can lower blood pressure and reduce the risk of heart disease, stroke, and some types of cancer. However, it may also increase fluid retention and cause other health problems if consumed in excess.
Saturated Fat	Reducing saturated fat consumption can positively affect health and reduce the prevalence of heart disease.
Saturated Fat	The benefits of Saturated fat in health are not clear cut. Some studies suggest that reducing saturated fat consumption can positively affect health and reduce the prevalence of heart disease. However, other studies suggest that there may be no significant difference between saturated and unsaturated fats in terms of health.

Table 8.4: Set of results for the unsupervised search using Mistral7b

Nutrient	**Drawbacks**
Calcium	There are several drawbacks of calcium in health, including muscular weakness, irregular heart beating, kidney disease, and damage to soft tissues. High-dose calcium supplementation may also increase the risk for osteoporosis.
Calcium	There are no known drawbacks of calcium in health when consumed in adequate amounts. However, a diet low in dairy products may lead to hypocalcaemia or hypercalcaemia.
Dietary Fibre	A deficit in dietary fiber can lead to negative health effects such as increased risk of cardiovascular disease, mortality, malignancy, and impaired colonic health.
Dietary Fibre	Dietary fibre deficit can lead to several negative health effects including increased risk of certain cancers, cardiovascular diseases, and type II diabetes.
Iron	Iron can impair growth and development, particularly cognitive development, especially in children. Vegetarians and vegans may have difficulty absorbing iron from plant-based foods. Too much iron from supplements can be harmful and should be discussed with a doctor or registered dietitian/nutritionist.
Iron	Iron deficiency can impair growth and development, particularly cognitive development, while too much iron can lead to anaemia.
Carbohydrates	The drawbacks of carbohydrates in health include weight gain, increased risk of heart disease, and type 2 diabetes.
Carbohydrates	Carbohydrates deficit can lead to low blood glucose levels which may cause fatigue, weakness, dizziness, and headaches.
Vitamin C	Vitamin C deficiency can cause fatigue, anaemia, joint pain and muscle weakness. Severe deficiency can lead to scurvy, which affects collagen production and causes tooth loss, joint pain, gum inflammation, and poor wound healing. Children may experience bone malformations due
Potassium	The drawbacks of Potassium in health include increased risk of hypertension and stroke.

Table 8.5: Set of results for the unsupervised search using Mistral7b

Nutrient	Drawbacks
Sodium	High sodium intake has several drawbacks in health, including increasing the risk of cardiovascular diseases and kidney disease. It is important to pay attention to your sodium intake when you have metabolic disorders or specific health conditions that cause your body to remove excessive amounts of this mineral.
Salt	Excessive salt consumption may increase the risk of cardiovascular diseases, such as hypertension, in children and adults.
Saturated Fat	The drawbacks of Saturated fat in health include raising LDL-C concentration which can lead to atherosclerotic cardiovascular disease.
Saturated Fat	High consumption of saturated fats can lead to health issues such as increased risk of cardiovascular disease and higher mortality from all causes.

Table 8.6: Set of results for the unsupervised search using Mistral7b

A full description and dataset obtained from the experiments can be accessed in `https://github.com/thebooort/Unsupervised-health-justifications` along with the successive updates planned with bigger text corpora.

Chapter 9

Environmental impact

> *"We need the reduction of our carbon footprint to stem not from an economic incentive or a superficial motive [...] but from the deepest part of our being."*
>
> *2020* Andreu Escrivà*,* -

In this section, we will briefly analyze the environmental impact of the research conducted in this thesis, focusing on the energy consumption of the recommendation systems developed herein.

The reason behind this section is clear for the author: not a single thesis of the one he have read had it. While we as developers and scientists are usually forced into a fast "model-experiment-publication"-system, at some tipping points we should stop and reflect on our contribution to the climate chrisis we are facing. Nowadays the trade off between red and green machine learning[267], [268] (bigger models, vs more efficient, adapted to the problem technologies) is a relevant topic in the field, growing each year. But not only the models and algorithms we are building may affect the climate, or its training, but also, the applications we are developing with them [269]. Thats is the main reason why The analysis will be comprehensive, centering on the energy requirements of the system and subdividing each of its phases. Subsequently, we will also assess the environmental impact of the explainability aspects developed, providing estimations for the algorithm as a whole. These final estimates will be correlated with various environmental impacts and ultimately integrated with other associated impacts.

9.1 CO2 Emissions Related to Google Colab Usage

Experiments were conducted using Google Cloud Platform in region asia-southeast1. This assumption is made based on the expected region for a T4 with high RAM usage when selected in google colab, as stated in (`Google Regions and zones`) with the information gathered by running `!curl ipinfo.io` in the notebook, that pointed out the region of the computation in Singapore. Thus this computation had a carbon efficiency of 0.42 kgCO_2eq/kWh. A cumulative of 100 hours of computation was performed on hardware of type T4 (TDP of 70W). Total emissions are estimated to be 2.94 kgCO_2eq of which 100 percents (at least this is what it is said) were directly offset by the cloud provider.

Estimations were conducted using the MachineLearning Impact calculator presented in [270].

9.2 CO2 Emissions Related to Running the algorithm

Several models have been launched during the process of this thesis. Several tries around trying new weigths, testing functions, improving methodologies and processes. As this project start with no carbon emission technologies is hard to compute how this was, but we offer here a minimum approach. We calculate a rough estimate on how many times the algorithm has been launch across all the application with a surplus for the training and coding activities.

What we offer here is a minimal estimation on how much this thesis has emitted based solely on computational work. We compute this by using the package Codecarbon [271] and running a selected script with the whole python package, a based case (100 of them, for obtaining the evaluation) and a content-based recommendation.

This led us to a consumption of 0.007051 kWh for a whole experiment. Based on the applications and testing a rough estimation of 300 users (launching the app) 7 days a week (for the nutritional case) during at leas 12 weeks produce a 25200 launches of the script in one of its several forms. This is expected to produce 177.6852 kWh for the whole duration of the thesis.

In 2022, Spain's power sector emission intensity stood above 217 grams of carbon dioxide per kilowatt-hour (gCO_2/KWh) of electricity generated (based on the Statista webpage [272]), slightly up from the previous year, which led us to a 385.576884 kg of CO_2 emitted.

9.3 Results and final remark

Round in up the data, we could argue that the process of creating this thesis and the developments in it has produce around 400 kg of CO2. That is whit out account several travel some of them in plane) due to academic activities (conferences and research visits).

While is hard to think about this numbers, I want to put them into perspective through a series of qualitative and quantitative comparisons. Those were calculate with the help of the tool develop by the energy commission of the United States (based on [273]). We start analyzing how 400Kg of CO_2 are the equivalent to CO_2 emissions from different contaminant sources in table 9.1.

This is not the only way to acknowledge our consumption, from a more interesting point of view, it would be the equivalent to greenhouse gas emissions avoided by recycling different amounts of waste, as can be seeing in 9.2.

And finally, in terms of the different countryside elements that can incorporate or sequestrate CO2, 9.3

The calculation stated in this section are not mean to be final or anyway 100% accurate. They are an estimation, but a estimation that should be made to face the true impact of our work, not only to the academic or scientific community, but also to the whole global current situation.

Equivalent	Quantity	Unit
Gasoline Vehicles Usage	0.089	Years
Gasoline Consumed	170.34	liters
Coal Burned	203.21	kilograms
Coal Burned	0.002	Railcars
Energy Usage	0.05	In a Home for a Year
Distance Driven (gasoline vehicle)	1,650	kilometers
Diesel Consumed	148.71	liters
Full Gasoline Deposit	0.005	in Trucks trank
Electricity Usage	0.078	In a Home for a Year
Barrels of Oil Consumed	147.44	liters
Smartphones Charged	48,657	times

Table 9.1: Various Equivalents in the International System

Equivalent	Quantity	Unit
Waste Recycled	0.138	Tons
Waste Recycled	17.3	trash bags
Incandescent Lamps to LEDs	15.2	-
Waste Recycled	0.02	garbage trucks
Wind Turbines running	0.0001	Years

Table 9.2: Greenhouse Gas Emissions Avoided Equivalents

Equivalent	Quantity	Unit
Trees (grown during 10 years)	6.6	-
U.S. Forest Area (one year)	0.477	hectares
Preserved U.S. Forest Area (one year)	0.003	hectares

Table 9.3: Carbon Sequestration Equivalents for the project carbon footprint

Chapter 10

Conclusions

"We will never make a 32-bit operating system."

launch of MSX Bill Gates*, 1989*

10.1 Conclusions and remarks

Throughout this work, we have followed a chronological order, presenting all the advancements that this thesis has entailed. Following the first hypothesis of this study, we explored recommendation systems by providing a detailed description of their components, along with a review of how the problem has been addressed in scientific research to date. After this overview of various recommendation models and techniques, some of which we would later employ, we observed, in line with hypothesis 1, that the majority of works focused on recommendations for specific items in very particular situations.

To delve even further into the approaches used in recommendations that deviate from this simple scenario, we conducted an additional study where we introduced a classification distinct from any developed thus far. In this classification, we defined impact and complexity, using both characteristics as the backbone for addressing different situations and examining how they have been approached to date. This concludes objectives 1.1 and 1.2 of the thesis, contributing to solidifying the hypotheses 1 and 2 put forth in the study.

In the subsequent chapters 4, 5, and 6, we apply the knowledge acquired in the previous chapters to propose a recommendation system that takes into account our initial approach derived from the review, along with the second hypothesis of the study. This leads us to a flexible hybrid recommendation model that employs genetic algorithms to construct complex items based on the characteristics of the problem. Subsequently, this recommendation is enhanced in a second module that utilizes a content-based approach following user preferences. This entire system fulfills objectives O2.1 and O2.2.

In addition to this arrangement, in the following chapters, we present two practical applications of the algorithm introduced in the methodology. In the first one, we focus on the nutritional recommendation process. Within this type of system, there is an extensive body of literature, but we leverage the structure-generating potential of our algorithm to provide a novel and validated recommendation system focused on nutritional interventions. Here, we can adjust our fitness function to accommodate calories, macronutrients, and 11 different micronutrients, an aspect not addressed until now.

Furthermore, this system is part of the European project stance4health, which has been implemented in two countries with numerous users, aiming to identify and correlate dietary patterns with microbiota values. Following this application, we provide an additional recommendation where we delve into a more cultural and subjective realm—podcasts. On this occasion, our system works to create a radio-style program based on user requirements and preferences.

Once again, our approach demonstrates its effectiveness in offering recommendations that do not rely on previous data or consumption datasets. In fact, it can be used to generate new and interesting datasets from existing ones. In this particular case, we harness the full capabilities of content-based recommendation, thanks to new text embedding techniques, addressing the challenge of generating podcast episode schedules. Our approach not only proves effective but also has the capability to produce new information (requirement coverage, constructed datasets, structural space, etc.) that can contribute to future research steps.

This concludes objectives 2.3 and, consequently, all those outlined in section 2. In addition to all the above, Chapter 6 serves to present a Python package that allows replicating a portion of the thesis results. It is available to the international community and open-source, seamlessly connecting with other well-known recommendation system packages.

Finally, hypothesis 3 posits the need to better understand end users to assess their thought processes, opinions about the application, and how we can assist in decision-making in impactful situations. To address this, we present an initial study where we identify how a trustworthy system is defined and how we can measure that perception from the user. Thanks to the European project and the stay in Denmark, we were able to delve deeper into how this interest can be psychologically measured and which variables related to it can predict whether a user will continue using the application.

In conjunction with the results of the European project application, we suggest that one way to increase user trustworthiness in the application is through the justifications of the proposals we make. To achieve this, we transform recommendations into questions based on the fitness or preference evaluations conducted during the recommendation process. When these recommendations, which are crucial according to the state of the art, are based on health and scientific consensus, our method can autonomously search for them in the literature and provide a user-specific response using different language models. This previously unseen advancement in the literature concludes the third objective of the study.

The conclusion of the three main objectives of this work constitutes a unique contribution with several works presented in research journals and research conferences. It represents a recommendation system diverging from the current trend that it is able to assist users in complex environments, creating itself a recommendation item from simpler information. And, what is more it has clear and manifest applications in different fields. However, it has also prove a titanic task in terms of implementing a scientific idea and materializing it in an interdisciplinary European scientific project, benefiting numerous researchers. Therefore, it constitutes a dual achievement, both in research and engineering.

10.2 Reflections and future work

This work contributes to or attempts to address complex situations in an interdisciplinary and multifactorial manner. To achieve this, various branches of both engineering and computer science, as well as psychology and user behavior, have been studied. Recommendation systems, or at least the future the author envisions, should evolve to be systems that accompany and guide rather than impose. Systems that advise and educate. The ultimate goal of a system (for example, in the nutritional context) should be the same as that of a real-life nutritionist—to provide tools to face different situations and educate the individual. The alternative assumes that we can never detach ourselves from the systems we create and place all the responsibility on them. In the case of content consumption, it is similar, as what we consume and how we do it has an impact on the cultural creations that emerge. Therefore, it is necessary for us to understand how recommendation systems work, what they are providing us, how they are deciding, so that we have more autonomy over how to use them and when to stop.

This work presents an approach to this problem and, compared to other similar works, seeks to stand out by advocating for this philosophy. However, many interesting questions remain to be answered.

In the technical section, we would like to delve into the mechanisms and techniques that this system can offer and develop its full potential. Knowledge graphs have proven useful in describing the state of the art, as well as sequential recommendations. Additionally, there is a lack of a theoretical foundation to describe metrics that can be used to compare this algorithm with others. These metrics pose another interesting challenge where many approaches are possible.

In the psychological section, content-based recommendation along with the refinement process opens the system to explainability mechanisms, offering a multitude of tools at our disposal. Moreover, the natural incorporation of this process with language models provides an interesting model to study more interactions and reasoning strategies with users, delving into the psychological aspects of recommendation.

Finally, for present and future research, as well as for the entire community, the gradual release of the datasets we are generating is essential. This will allow us to compare future improvements of the algorithm, but it can also serve as a starting point to use these systems to alleviate cold start issues in recommendations while feeding sequence-based recommendation models.

Bibliography

[1] G. Jawaheer, P. Weller, and P. Kostkova, "Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 4, no. 2, pp. 1–26, 2014.

[2] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukiás, "Multicriteria user modeling in recommender systems," *IEEE Intelligent Systems*, vol. 26, pp. 64–76, 2011. [Online]. Available: `https://api.semanticscholar.org/CorpusID:16752808`.

[3] Z. Yu, J. Lian, A. Mahmoody, G. Liu, and X. Xie, "Adaptive user modeling with long and short-term preferences for personalized recommendation," in *International Joint Conference on Artificial Intelligence*, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:199466436`.

[4] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, pp. 239–249, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:71145948`.

[5] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. Luo, "A review of content-based and context-based recommendation systems," *Int. J. Emerg. Technol. Learn.*, vol. 16, 2021. [Online]. Available: `https://api.semanticscholar.org/CorpusID:233393374`.

[6] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: Challenges and remedies," *Artificial Intelligence Review*, vol. 52, pp. 1–37, 2019.

[7] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," *Recommender Systems Handbook*, pp. 1–35, 2021.

[8] D. W. Oard and J. Kim, "Implicit feedback for recommender systems," 1998.

[9] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, "Personalized itinerary recommendation with queuing time awareness," *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017. [Online]. Available: `https://api.semanticscholar.org/CorpusID:24365448`.

[10] H. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on cnn-lstm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1653–1663, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:208093192`.

[11] R. Abolghasemi, P. E. Engelstad, E. E. Herrera-Viedma, and A. Yazidi, "A personality-aware group recommendation system based on pairwise preferences," *Inf. Sci.*, vol. 595, pp. 1–17, 2022. [Online]. Available: `https://api.semanticscholar.org/CorpusID:247090365`.

[12] G. Manogaran, R. Varatharajan, and M. K. Priyan, "Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system," *Multimedia Tools and Applications*, vol. 77, pp. 4379–4399, 2017. [Online]. Available: `https://api.semanticscholar.org/CorpusID:3339832`.

[13] D. Jannach, "Multi-objective recommender systems: Survey and challenges," *arXiv preprint arXiv:2210.10309*, 2022.

[14] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016, vol. 1.

[15] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *applied sciences*, vol. 10, no. 21, p. 7748, 2020.

[16] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*, Springer, 2007, pp. 325–341.

[17] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[18] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in *International Conference on Trust Management*, Springer, 2006, pp. 93–104.

[19] D. Ben-Shimon, A. Tsikinovsky, L. Rokach, A. Meisles, G. Shani, and L. Naamani, "Recommender system from personal social networks," in *Advances in Intelligent Web Mastering: Proceedings of the 5th Atlantic Web Intelligence Conference–AWIC'2007, Fontainbleau, France, June 25–27, 2007*, Springer, 2007, pp. 47–55.

[20] O. Arazy, N. Kumar, and B. Shapira, "Improving social recommender systems," *IT professional*, vol. 11, no. 4, pp. 38–44, 2009.

[21] S. Bouraga, I. Jureta, S. Faulkner, and C. Herssens, "Knowledge-based recommendation systems: A survey," *International Journal of Intelligent Information Technologies (IJIIT)*, vol. 10, no. 2, pp. 1–19, 2014.

[22] A. Felfernig and R. Burke, "Constraint-based recommender systems: Technologies and research issues," in *Proceedings of the 10th international conference on Electronic commerce*, 2008, pp. 1–10.

[23] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth, "Case-based recommender systems," *The Knowledge Engineering Review*, vol. 20, no. 3, p. 315, 2005.

[24] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "Constraint-based recommender systems," in *Recommender systems handbook*, Springer, 2015, pp. 161–190.

[25] F. O. Isinkaye, Y. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[26] R. Burke, “Hybrid web recommender systems,” *The adaptive web: methods and strategies of web personalization*, pp. 377–408, 2007.

[27] E. Çano and M. Morisio, “Hybrid recommender systems: A systematic literature review,” *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017.

[28] R. Seth and A. Sharaff, “A comparative overview of hybrid recommender systems: Review, challenges, and prospects.,” *Data Mining and Machine Learning Applications*, 2022.

[29] S. Kulkarni and S. F. Rodd, “Context aware recommendation systems: A review of the state of the art techniques,” *Computer Science Review*, vol. 37, p. 100 255, 2020.

[30] Z. Dong, Z. Wang, J. Xu, R. Tang, and J. Wen, “A brief history of recommender systems,” *arXiv preprint arXiv:2209.01860*, 2022.

[31] M. S. Reddy and T. Adilakshmi, “Music recommendation system based on matrix factorization technique -svd,” *2014 International Conference on Computer Communication and Informatics*, pp. 1–6, 2014. [Online]. Available: `https://api.semanticscholar.org/CorpusID:5826606`.

[32] A. Mnih and R. R. Salakhutdinov, “Probabilistic Matrix Factorization,” in *Advances in Neural Information Processing Systems*, vol. 20, Curran Associates, Inc., 2007. [Online]. Available: `https://papers.nips.cc/paper_files/paper/2007/hash/d7322ed717dedf1eb4e6e52a37ea7bcd-Abstract.html` (visited on 01/12/2024).

[33] S. Fu and Q. Ren, “Multi-behavior recommendation with svd graph neural networks,” *ArXiv*, vol. abs/2309.06912, 2023. [Online]. Available: `https://api.semanticscholar.org/CorpusID:261705561`.

[34] S. Rendle, "Factorization machines," *2010 IEEE International Conference on Data Mining*, pp. 995–1000, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:17265929.

[35] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, 57:1–57:22, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:5499886.

[36] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 43–50.

[37] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[38] U. Michelucci, "An introduction to autoencoders," *arXiv preprint arXiv:2201.03898*, 2022.

[39] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[40] G. K. Dziugaite and D. M. Roy, "Neural network matrix factorization," *ArXiv*, vol. abs/1511.06443, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:10560000.

[41] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," *Proceedings of the 26th International Conference on World Wide Web*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:13907106.

[42] S. Rendle, W. Krichene, L. Zhang, and J. R. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," *Proceedings of the 14th ACM Conference on*

Recommender Systems, 2020. [Online]. Available: `https://api.semanticscholar.org/CorpusID:218719424`.

[43] H.-T. Cheng, L. Koc, J. Harmsen, *et al.*, "Wide & deep learning for recommender systems," *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016. [Online]. Available: `https://api.semanticscholar.org/CorpusID:3352400`.

[44] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction," *ArXiv*, vol. abs/1703.04247, 2017. [Online]. Available: `https://api.semanticscholar.org/CorpusID:970388`.

[45] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[46] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, pp. 5929–5955, 2020.

[47] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 495–503.

[48] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[49] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1040–1048.

[50] G. Zheng, F. Zhang, Z. Zheng, *et al.*, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 167–176.

[51] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[52] Q. Guo, F. Zhuang, C. Qin, *et al.*, “A survey on knowledge graph-based recommender systems,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2020.

[53] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[54] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[55] H. Wang, F. Zhang, J. Wang, *et al.*, “Ripplenet: Propagating user preferences on the knowledge graph for recommender systems,” in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 417–426.

[56] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, “Kgat: Knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[57] C. Musto, M. d. Gemmis, P. Lops, F. Narducci, and G. Semeraro, “Semantics and content-based recommendations,” in *Recommender systems handbook*, Springer, 2012, pp. 251–298.

[58] C. Musto, G. Semeraro, M. De Gemmis, and P. Lops, “Learning word embeddings from wikipedia for content-based recommender systems,” in *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, Springer, 2016, pp. 729–734.

[59] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*, 2017.

[60] H. A. M. Hassan, G. Sansonetti, F. Gasparetti, A. Micarelli, and J. Beel, "Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation?" In *RecSys (Late-Breaking Results)*, 2019, pp. 6–10.

[61] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, and M. Fragkoulis, "Leveraging large language models for sequential recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1096–1102.

[62] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[63] F. Sun, J. Liu, J. Wu, *et al.*, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[64] L. Meng, F. Feng, X. He, X. Gao, and T.-S. Chua, "Heterogeneous fusion of semantic and collaborative information for visually-aware food recommendation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3460–3468.

[65] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2545–2553.

[66] W.-T. Chu and Y.-L. Tsai, "A hybrid recommendation system considering visual information for predicting favorite restaurants," *World Wide Web*, vol. 20, pp. 1313–1331, 2017.

[67] R. Yin, K. Li, J. Lu, and G. Zhang, "Enhancing fashion recommendation with visual compatibility relationship," in *The World Wide Web Conference*, 2019, pp. 3434–3440.

[68] Y. Li, H. Wang, H. Liu, and B. Chen, "A study on content-based video recommendation," in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4581–4585.

[69] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," *arXiv preprint arXiv:1808.03912*, 2018.

[70] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[71] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2015.

[72] Y. Zheng and D. X. Wang, "A survey of recommender systems with multi-objective optimization," *Neurocomputing*, vol. 474, pp. 141–153, 2022.

[73] T. Horváth and A. C. de Carvalho, "Evolutionary computing in recommender systems: A review of recent research," *Natural Computing*, vol. 16, pp. 441–462, 2017.

[74] E. Zangerle and C. Bauer, "Evaluating recommender systems: Survey and framework," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[75] L. Zhang, "The definition of novelty in recommendation system.," *Journal of Engineering Science & Technology Review*, vol. 6, no. 3, 2013.

[76] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: Evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257–260.

[77] M. Kunaver and T. Požrl, “Diversity in recommender systems–a survey,” *Knowledge-based systems*, vol. 123, pp. 154–162, 2017.

[78] T. Di Noia, J. Rosati, P. Tomeo, and E. Di Sciascio, “Adaptive multi-attribute diversity for recommender systems,” *Information Sciences*, vol. 382, pp. 234–253, 2017.

[79] S. Patil, D. Banerjee, and S. Sural, “A graph theoretic approach for multi-objective budget constrained capsule wardrobe recommendation,” *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 1, pp. 1–33, 2021.

[80] N. E. I. Karabadji, S. Beldjoudi, H. Seridi, S. Aridhi, and W. Dhifli, “Improving memory-based user collaborative filtering with evolutionary multi-objective optimization,” *Expert Systems with Applications*, vol. 98, pp. 153–165, 2018.

[81] X. Cai, Z. Hu, P. Zhao, W. Zhang, and J. Chen, “A hybrid recommendation system with many-objective evolutionary algorithm,” *Expert Systems with Applications*, vol. 159, p. 113 648, 2020.

[82] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, and Y. Wu, “Nnia-rs: A multi-objective optimization based recommender system,” *Physica A: Statistical Mechanics and its Applications*, vol. 424, pp. 383–397, 2015.

[83] N. Ranjbar Kermany, W. Zhao, J. Yang, J. Wu, and L. Pizzato, “A fairness-aware multi-stakeholder recommender system,” *World Wide Web*, vol. 24, pp. 1995–2018, 2021.

[84] Y. Zheng, N. Ghane, and M. Sabouri, “Personalized educational learning with multi-stakeholder optimizations,” in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 2019, pp. 283–289.

[85] N. R. Kermany, W. Zhao, J. Yang, J. Wu, and L. Pizzato, “An ethical multi-stakeholder recommender system based on evolutionary multi-objective optimization,” in *2020 IEEE international conference on services computing (SCC)*, IEEE, 2020, pp. 478–480.

[86] G. Hadash, O. S. Shalom, and R. Osadchy, "Rank and rate: Multi-task learning for recommender systems," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 451–454.

[87] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 165–174.

[88] Y. Gu, Z. Ding, S. Wang, L. Zou, Y. Liu, and D. Yin, "Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2493–2500.

[89] I. Kamehkhosh, G. Bonnin, and D. Jannach, "Effects of recommendations on the playlist creation behavior of users," *User Modeling and User-Adapted Interaction*, vol. 30, no. 2, pp. 285–322, 2020.

[90] J. A. Delgado, R. Kalluri, K. Gutta, A. B. Krishna, and D. Turner, "Personalized voice search for internet tv.," in *ComplexRec@ RecSys*, 2017, pp. 19–22.

[91] M. R. Hasan, A. K. Jha, and Y. Liu, "Excessive use of online video streaming services: Impact of recommender system use, psychological factors, and motives," *Computers in Human Behavior*, vol. 80, pp. 220–228, 2018.

[92] O. Moskalenko, D. Parra, and D. Saez-Trumper, "Scalable recommendation of wikipedia articles to editors using representation learning," in *Workshop on Recommendation in Complex Scenarios at the ACM RecSys Conference on Recommender Systems (RecSys 2020)*, Virtual Event, Brazil, 2020.

[93] G. Prato, F. Sallemi, P. Cremonesi, M. Scriminaci, S. Gudmundsson, and S. Palumbo, "Outfit completion and clothes recommendation," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–7.

[94] V. S. Vairale and S. Shukla, "Recommendation of food items for thyroid patients using content-based knn method," in *Data Science and Security*, Springer, 2020, pp. 71–77.

[95] A. Mustaqeem, S. M. Anwar, and M. Majid, "A modular cluster based collaborative recommender system for cardiac patients," *Artificial Intelligence in Medicine*, vol. 102, p. 101 761, 2020.

[96] P. Chavan, B. Thoms, and J. Isaacs, "A recommender system for healthy food choices: Building a hybrid model for recipe recommendations using big data sets," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, p. 3774.

[97] D. A. Rohani, A. Quemada Lopategui, N. Tuxen, M. Faurholt-Jepsen, L. V. Kessing, and J. E. Bardram, "Mubs: A personalized recommender system for behavioral activation in mental health," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[98] L. Esmaeili, S. Mardani, S. A. H. Golpayegani, and Z. Z. Madar, "A novel tourism recommender system in the context of social commerce," *Expert Systems with Applications*, vol. 149, p. 113 301, 2020.

[99] C. K. Leung, A. Kajal, Y. Won, and J. M. Choi, "Big data analytics for personalized recommendation systems," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, 2019, pp. 1060–1065.

[100] J. Dalton, V. Ajayi, and R. Main, "Vote goat: Conversational movie recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1285–1288.

[101] S. Dara, C. R. Chowdary, and C. Kumar, “A survey on group recommender systems,” *Journal of Intelligent Information Systems*, vol. 54, no. 2, pp. 271–295, 2020.

[102] S. Norouzi, A. K. Ghalibaf, S. Sistani, *et al.*, “A mobile application for managing diabetic patients’ nutrition: A food recommender system,” *Archives of Iranian medicine*, vol. 21, no. 10, pp. 466–472, 2018.

[103] A. Yan, S. Cheng, W.-C. Kang, M. Wan, and J. McAuley, “Cosrec: 2d convolutional neural networks for sequential recommendation,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2173–2176.

[104] Q. Wang, J. Ma, X. Liao, and W. Du, “A context-aware researcher recommendation system for university-industry collaboration on r&d projects,” *Decision Support Systems*, vol. 103, pp. 46–57, 2017.

[105] D. Chen, L. Xie, A. K. Menon, and C. S. Ong, “Structured recommendation,” *arXiv preprint arXiv:1706.09067*, 2017.

[106] J. Tang and K. Wang, “Personalized top-n sequential recommendation via convolutional sequence embedding,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.

[107] M. Li, X. Bao, L. Chang, Z. Xu, and L. Li, “A survey of researches on personalized bundle recommendation techniques,” in *International Conference on Machine Learning for Cyber Security*, Springer, 2020, pp. 290–304.

[108] D. Shi, T. Wang, H. Xing, and H. Xu, “A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning,” *Knowledge-Based Systems*, vol. 195, p. 105 618, 2020.

[109] T. M. Phuong, T. C. Thanh, and N. X. Bach, “Neural session-aware recommendation,” *IEEE Access*, vol. 7, pp. 86 884–86 896, 2019.

[110] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 43–56, 2016.

[111] R. Conforti, M. de Leoni, M. La Rosa, W. M. van der Aalst, and A. H. ter Hofstede, "A recommendation system for predicting risks across multiple business process instances," *Decision Support Systems*, vol. 69, pp. 1–19, 2015.

[112] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," *Expert Systems with Applications*, vol. 167, p. 114 324, 2021.

[113] D. Wang, X. Zhang, D. Yu, G. Xu, and S. Deng, "Came: Content-and context-aware music embedding for recommendation," *IEEE transactions on neural networks and learning systems*, 2020.

[114] T. Wei, J.-X. Shi, and Y.-F. Li, "Probabilistic label tree for streaming multi-label learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1801–1811.

[115] J. Wang, N. Gao, J. Peng, and J. Mo, "Dcar: Deep collaborative autoencoder for recommendation with implicit feedback," in *International Conference on Artificial Neural Networks*, Springer, 2019, pp. 172–184.

[116] R. Xie, R. Wang, S. Zhang, Z. Yang, F. Xia, and L. Lin, "Real-time relevant recommendation suggestion," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 112–120.

[117] H. Chen, Y. Chen, X. Wang, *et al.*, "Curriculum disentangled recommendation with noisy multi-feedback," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[118] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose, "Relational collaborative filtering: Modeling multiple item relations for recommendation," in *Proceedings of*

the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 125–134.

[119] S. Gupta and V. Kant, "An aggregation approach to multi-criteria recommender system using genetic programming," *Evolving Systems*, vol. 11, no. 1, pp. 29–44, 2020.

[120] E. Munoz, J. Meza, L. Recalde, and L. Teran, "Finding the appropriate housing: A fuzzy-model-based recommender system," *2021 8th International Conference on eDemocracy and eGovernment, ICEDEG 2021*, pp. 139–145, 2021.

[121] F. Luo, G. Ranzi, W. Kong, G. Liang, and Z. Dong, "Personalized residential energy usage recommendation system based on load monitoring and collaborative filtering," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1253–1262, 2021.

[122] F. Liu and W.-W. Guo, "Research on house recommendation model based on cosine similarity in deep learning mode in grid environment," *International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2019*, pp. 121–124, 2019.

[123] A. Rashed, S. Jawed, L. Schmidt-Thieme, and A. Hintsches, "Multirec: A multi-relational approach for unique item recommendation in auction systems," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 230–239.

[124] A. Rashed, J. Grabocka, and L. Schmidt-Thieme, "Attribute-aware non-linear co-embeddings of graph features," in *Proceedings of the 13th ACM conference on recommender systems*, 2019, pp. 314–321.

[125] M. Cartagena, P. Cerda, P. Messina, F. D. Río, and D. Parra, "Curatornet: Visually-aware recommendation of art image," in *Proceedings of the Fourth Workshop on Recommendation in Complex Environments*, Virtual Event, Brazil, 2020. [Online]. Available: `https://arxiv.org/abs/2009.04426`.

[126] B. Yilma, N. Aghenda, M. Romero, Y. Naudet, and H. Panettoy, "Personalised visual art recommendation by learning latent semantic representations," *SMAP 2020 - 15th International Workshop on Semantic and Social Media Adaptation and Personalization*, 2020.

[127] D. Goncalves, L. Liu, J. Sá, T. Otto, A. Magalhães, and P. Brochado, "The importance of brand affinity in luxury fashion recommendations," in *Recommender Systems in Fashion and Retail*, Springer, 2021, pp. 3–19.

[128] E. M. Daly, A. Botea, A. Kishimoto, and R. Marinescu, "Multi-criteria journey aware housing recommender system," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 325–328.

[129] M. Uta, A. Felfernig, and D. Helic, "Constraint-aware recommendation of complex items," in *CEUR Workshop Proceedings*, RWTH Aachen, vol. 2960, 2021.

[130] S. P. Erdeniz, R. Samer, and M. Stettinger, "Towards similarity-aware constraint-based recommendation," in *Advances and Trends in Artificial Intelligence. From Theory to Practice: 32nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2019, Graz, Austria, July 9–11, 2019, Proceedings*, Springer, vol. 11606, 2019, p. 287.

[131] R. Zhang, T. Yu, Y. Shen, H. Jin, and C. Chen, "Text-based interactive recommendation via constraint-augmented reinforcement learning," *Advances in neural information processing systems*, vol. 32, 2019.

[132] Ö. Sürer, R. Burke, and E. C. Malthouse, "Multistakeholder recommendation with provider constraints," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 54–62.

[133] S. Haussmann, O. Seneviratne, Y. Chen, *et al.*, "Foodkg: A semantics-driven knowledge graph for food recommendation," in *International Semantic Web Conference*, Springer, 2019, pp. 146–162.

[134] Y. Chen, A. Subburathinam, C.-H. Chen, and M. J. Zaki, "Personalized food recommendation as constrained question answering over a large-scale food knowledge graph," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 544–552.

[135] Y. Tian, Y. Yang, X. Ren, *et al.*, "Joint knowledge pruning and recurrent graph convolution for news recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 51–60.

[136] Y. Chung, H.-W. Jung, J. Kim, and J.-H. Lee, "Personalized expert-based recommender system: Training c-svm for personalized expert identification," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Springer, 2013, pp. 434–441.

[137] C. D. Casuat, A. S. M. Isira, E. D. Festijo, A. S. Alon, J. N. Mindoro, and J. A. B. Susa, "A development of fuzzy logic expert-based recommender system for improving students' employability," in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, IEEE, 2020, pp. 59–62.

[138] S.-i. Song, S. Lee, S. Park, and S.-g. Lee, "Determining user expertise for improving recommendation performance," in *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, 2012, pp. 1–5.

[139] Z. Duan, W. Xu, Y. Chen, and L. Ding, "Etbrec: A novel recommendation algorithm combining the double influence of trust relationship and expert users," *Applied Intelligence*, vol. 52, no. 1, pp. 282–294, 2022.

[140] M. M. Sridevi and R. R. Rao, "Personalized recommender by exploiting domain based expert for enhancing collaborative filtering algorithm: Prec," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 3, 2019.

[141] T. Qi, F. Wu, C. Wu, and Y. Huang, "Personalized news recommendation with knowledge-aware interactive matching," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 61–70.

[142] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, and X. He, "Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1791–1800.

[143] Z. Zhu, Y. He, X. Zhao, and J. Caverlee, "Popularity bias in dynamic recommendation," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2439–2449.

[144] A. T. Wibowo, A. Siddharthan, C. Lin, and J. Masthoff, "Matrix factorization for package recommendations," in *Proceedings of the RecSys 2017 Workshop on Recommendation in Complex Scenarios (ComplexRec 2017)*, CEUR-WS, 2017.

[145] Z. Zheng, C. Wang, T. Xu, *et al.*, "Drug package recommendation via interaction-aware graph induction," in *Proceedings of the Web Conference 2021*, 2021, pp. 1284–1295.

[146] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, "Bundle recommendation and generation with graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[147] L. Chen, Y. Liu, X. He, L. Gao, and Z. Zheng, "Matching user with item set: Collaborative bundle recommendation with deep attention network.," in *IJCAI*, 2019, pp. 2095–2101.

[148] A. Pathak, K. Gupta, and J. McAuley, "Generating and personalizing bundle recommendations on steam," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1073–1076.

[149] H. Alcaraz-Herrera and I. Palomares, “Evolutionary approach for’healthy bundle’wellbeing recommendations.,” in *HealthRecSys RecSys*, 2019, pp. 18–23.

[150] L. Chen, L. Wu, K. Zhang, R. Hong, and M. Wang, “Set2setrank: Collaborative set to set ranking for implicit feedback based recommendation,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 585–594.

[151] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu, “A group recommendation system for online communities,” *International journal of information management*, vol. 30, no. 3, pp. 212–219, 2010.

[152] J. Shi, B. Wu, and X. Lin, “A latent group model for group recommendation,” in *2015 IEEE International conference on mobile services*, IEEE, 2015, pp. 233–238.

[153] J. Zhang, C. Gao, D. Jin, and Y. Li, “Group-buying recommendation for social e-commerce,” in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, IEEE, 2021, pp. 1536–1547.

[154] B. Hao, H. Yin, J. Zhang, C. Li, and H. Chen, “Self-supervised graph learning for occasional group recommendation,” *arXiv preprint arXiv:2112.02274*, 2021.

[155] S. Zan, Y. Zhang, X. Meng, P. Lv, and Y. Du, “Uda: A user-difference attention for group recommendation,” *Information Sciences*, vol. 571, pp. 401–417, 2021.

[156] Z. Huang, Y. Liu, C. Zhan, C. Lin, W. Cai, and Y. Chen, “A novel group recommendation model with two-stage deep learning,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[157] T. Mavridis, S. Hausl, A. Mende, and R. Pagano, “Beyond algorithms: Ranking at scale at booking. com,” in *Proceedings of the Fourth Workshop on Recommendation in Complex Scenarios. CEUR-WS*, 2020.

[158] Y. Sun and Y. Zhang, “Conversational recommender system,” in *The 41st international acm sigir conference on research & development in information retrieval*, 2018, pp. 235–244.

[159] R. Cai, J. Wu, A. San, C. Wang, and H. Wang, "Category-aware collaborative sequential recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 388–397.

[160] S. Zhang, D. Yao, Z. Zhao, T.-S. Chua, and F. Wu, "Causerec: Counterfactual user sequence synthesis for sequential recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 367–377.

[161] Y. Cui, H. Sun, Y. Zhao, H. Yin, and K. Zheng, "Sequential-knowledge-aware next poi recommendation: A meta-learning approach," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 2, pp. 1–22, 2021.

[162] M. Donciu, M. Ionita, M. Dascalu, and S. Trausan-Matu, "The runner–recommender system of workout and nutrition for runners," in *2011 13th international symposium on symbolic and numeric algorithms for scientific computing*, IEEE, 2011, pp. 230–238.

[163] V. A. Nguyen, H.-H. Nguyen, D.-L. Nguyen, and M.-D. Le, "A course recommendation model for students based on learning outcome," *Education and Information Technologies*, vol. 26, no. 5, pp. 5389–5415, 2021.

[164] C. Li, Y. Lu, W. Wang, *et al.*, "Package recommendation with intra-and inter-package attention networks," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 595–604.

[165] G. Cai, K. Lee, and I. Lee, "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos," *Expert Systems with Applications*, vol. 94, pp. 32–40, 2018.

[166] B. Ortiz Viso, “Evolutionary approach in recommendation systems for complex structured objects,” in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 776–781.

[167] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, “Visually-aware fashion recommendation and design with generative image models,” in *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2017, pp. 207–216.

[168] C. C. T. Loon, J. Kavikumar, D. Nagarajan, and V. Yuvaraj, “A comprehensive study of personalized garment design using fuzzy logic,” in *AIP Conference Proceedings*, AIP Publishing LLC, vol. 2282, 2020, p. 020 002.

[169] P. Usip, F. Osang, and S. Konyeha, “An ontology-driven fashion recommender system for occasion-specific apparels,” *Journal, Advances in Mathematical & Computational Sciences*, vol. 8, no. 1, 2020.

[170] M. S. Hegde, G. Krishna, and R. Srinath, “An ensemble stock predictor and recommender system,” in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2018, pp. 1981–1985.

[171] T. De Pessemier, B. De Deyn, K. Vanhecke, and L. Martens, “Recommendations for sports games to bet on,” in *Second Workshop on Recommendation in Complex Scenarios (ComplexRec 2018), in conjunction with the 12th ACM Conference on Recommender Systems (RecSys 2018)*, 2018, pp. 8–12.

[172] E. Ringger, A. Chang, D. Fagnan, *et al.*, “Finding your home: Large-scale recommendation in a vibrant marketplace,” *ComplexRec 2018*, p. 13, 2018.

[173] H. J. Jun, J. H. Kim, D. Y. Rhee, and S. W. Chang, ““seoulhouse2vec”: An embedding-based collaborative filtering housing recommender system for analyzing housing preference,” *Sustainability*, vol. 12, no. 17, p. 6964, 2020.

[174] R. Logesh, V. Subramaniyaswamy, and V. Vijayakumar, “A personalised travel recommender system utilising social network profile and accurate gps data,” *Electronic Government, an International Journal*, vol. 14, no. 1, pp. 90–113, 2018.

[175] Q. Shen, W. Tao, J. Zhang, H. Wen, Z. Chen, and Q. Lu, "Sar-net: A scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4094–4103.

[176] C. Obeid, I. Lahoud, H. El Khoury, and P.-A. Champin, "Ontology-based recommender system in higher education," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 1031–1034.

[177] S. Pandey, A. Lan, G. Karypis, and J. Srivastava, "Learning student interest trajectory for mooc thread recommendation," in *2020 International Conference on Data Mining Workshops (ICDMW)*, 2020, pp. 400–407.

[178] Z. A. Pardos and W. Jiang, "Designing for serendipity in a university course recommendation system.," in *Proceedings of the Tenth International Conference on Learning Analytics and Knowledge*, 2020, pp. 350–359.

[179] A. Eetemadi, N. Rai, B. M. P. Pereira, M. Kim, H. Schmitz, and I. Tagkopoulos, "The computational diet: A review of computational methods across diet, microbiome, and health," *Frontiers in microbiology*, vol. 11, p. 393, 2020.

[180] L. R. Ferretto, E. A. Bellei, D. Biduski, *et al.*, "A physical activity recommender system for patients with arterial hypertension," *IEEE Access*, vol. 8, pp. 61 656–61 664, 2020.

[181] C. Trattner and D. Elsweiler, "Food Recommender Systems: Important Contributions, Challenges and Future Research Directions," *arXiv:1711.02760 [cs]*, Nov. 2017, arXiv: 1711.02760. [Online]. Available: `http://arxiv.org/abs/1711.02760` (visited on 06/11/2019).

[182] K. Yanai, T. Maruyama, and Y. Kawano, "A cooking recipe recommendation system with visual recognition of food ingredients.," *International Journal of Interactive Mobile Technologies*, vol. 8, no. 2, 2014.

[183] W.-Y. Chao and Z. Hass, “Choice-based user interface design of a smart healthy food recommender system for nudging eating behavior of older adult patients with newly diagnosed type ii diabetes,” in *International Conference on Human-Computer Interaction*, Springer, 2020, pp. 221–234.

[184] M. Dello Russo, P. Russo, J. Á. Rufián-Henares, *et al.*, “The stance4health project: Evaluating a smart personalised nutrition service for gut microbiota modulation in normal-and overweight adults and children with obesity, gluten-related disorders or allergy/intolerance to cow’s milk,” *Foods*, vol. 11, no. 10, p. 1480, 2022.

[185] V. A. Mullins, W. Bresette, L. Johnstone, B. Hallmark, and F. H. Chilton, “Genomics in personalized nutrition: Can you “eat for your genes”?” *Nutrients*, vol. 12, no. 10, p. 3118, 2020.

[186] M. Ge, F. Ricci, and D. Massimo, “Health-aware food recommender system,” in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys ’15, event-place: Vienna, Austria, New York, NY, USA: ACM, 2015, pp. 333–334, ISBN: 978-1-4503-3692-5. (visited on 06/11/2019).

[187] S. Fang, Z. Shao, R. Mao, *et al.*, “Single-view food portion estimation: Learning image-to-energy mappings using generative adversarial networks,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 251–255.

[188] C. Iwendi, S. Khan, J. H. Anajemba, A. K. Bashir, and F. Noor, “Realizing an efficient iomt-assisted patient diet recommendation system through machine learning model,” *IEEE Access*, vol. 8, pp. 28 462–28 474, 2020.

[189] R. Y. Toledo, A. A. Alzahrani, and L. Martinez, “A food recommender system considering nutritional information and user preferences,” *IEEE Access*, vol. 7, pp. 96 695–96 711, 2019.

[190] R. Shandilya, S. Sharma, and J. Wong, “Mature-food: Food recommender system for mandatory feature choices a system for enabling digital health,” *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100 090, 2022.

[191] H. Alcaraz-Herrera, J. Cartlidge, Z. Toumpakari, M. Western, and I. Palomares, “Evorecsys: Evolutionary framework for health and well-being recommender systems,” *User Modeling and User-Adapted Interaction*, pp. 1–39, 2022.

[192] J. Zhang, M. Li, W. Liu, S. Lauria, and X. Liu, “Many-objective optimization meets recommendation systems: A food recommendation scenario,” *Neurocomputing*, vol. 503, pp. 109–117, 2022.

[193] S. G. d. S. y Nutrición and I. SL, “I-diet food composition database, updated from original version of g,” *Martín Peña FCD*, vol. 1, 2019.

[194] D. Hinojosa-Nogueira, S. Pérez-Burillo, B. Navajas-Porras, *et al.*, “Development of an unified food composition database for the european project “stance4health”,” *Nutrients*, vol. 13, no. 12, p. 4206, 2021.

[195] T. Theodoridis, V. Solachidis, K. Dimitropoulos, L. Gymnopoulos, and P. Daras, “A survey on AI nutrition recommender systems,” en, in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA ’19*, Rhodes, Greece: ACM Press, 2019, pp. 540–546, ISBN: 978-1-4503-6232-0. (visited on 07/23/2019).

[196] D. Hinojosa-Nogueira, B. Ortiz-Viso, B. Navajas-Porras, *et al.*, “Stance4health nutritional app: A path to personalized smart nutrition,” *Nutrients*, vol. 15, no. 2, p. 276, 2023.

[197] B. Ortiz-Viso, A. Morales-Garzón, M. J. Martin-Bautista, and M.-A. Vila, “Evolutionary Approach for Building, Exploring and Recommending Complex Items With Application in Nutritional Interventions,” *IEEE Access*, vol. 11, pp. 65 891–65 905, 2023, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3290918.

[198] D. Hinojosa-Nogueira, B. Ortiz-Viso, B. Navajas-Porras, *et al.*, “Stance4Health Nutritional APP: A Path to Personalized Smart Nutrition,” en, *Nutrients*, vol. 15, no. 2, p. 276, Jan. 2023, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2072-6643. DOI: `10.3390/nu15020276`. [Online]. Available: `https://www.mdpi.com/2072-6643/15/2/276` (visited on 01/24/2023).

[199] D. Hinojosa-Nogueira, S. Pérez-Burillo, B. Navajas-Porras, *et al.*, “Development of an Unified Food Composition Database for the European Project “Stance4Health”,” en, *Nutrients*, vol. 13, no. 12, p. 4206, Dec. 2021, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2072-6643. DOI: `10.3390/nu13124206`. [Online]. Available: `https://www.mdpi.com/2072-6643/13/12/4206` (visited on 02/27/2023).

[200] “Human energy requirements: Report of a joint FAO/WHO/UNU expert consultation: Rome, 17-24 october 2001,” *World Health Organization. Human Energy Requirements: Report of a Joint FAO/WHO/UNU Expert Consultation: Rome*, pp. 92–95, 2001.

[201] A. Marzooqi, H. M. Burke, A. Ghazali, and A. Yousuf, “The development of a food atlas of portion sizes for the united arab emirates,” *J Food Compos Anal Elsevier*, vol. 43, pp. 140–148, 2015.

[202] S. V. Marcos, M. J. Rubio, F. R. Sanchidrián, and D. de Robledo, “Spanish national dietary survey in adults, elderly and pregnant women,” *EFSA Support. Publ.*, vol. 13, no. 6, Jun. 2016.

[203] T. Blasco, S. Pérez-Burillo, F. Balzerani, *et al.*, “An extended reconstruction of human gut microbiota metabolism of dietary compounds,” en, *Nat. Commun.*, vol. 12, no. 1, p. 4728, Aug. 2021.

[204] F. Balzerani, D. Hinojosa-Nogueira, X. Cendoya, *et al.*, “Prediction of degradation pathways of phenolic compounds in the human gut microbiota through enzyme promiscuity methods,” en, *NPJ Syst. Biol. Appl.*, vol. 8, no. 1, p. 24, Jul. 2022.

[205] S. R. Stoyanov, L. Hides, D. J. Kavanagh, and H. Wilson, "Development and validation of the user version of the mobile application rating scale (uMARS)," en, *JMIR MHealth UHealth*, vol. 4, no. 2, e72, Jun. 2016.

[206] K. N. Difilippo, W. Huang, and K. M. Chapman-Novakofski, "A new tool for nutrition app quality evaluation (AQEL): Development, validation, and reliability testing. JMIR MHealth UHealth," vol. 5, 2017.

[207] S. Chan-Olmsted and R. Wang, "Understanding podcast users: Consumption motives and behaviors," en, *New Media & Society*, vol. 24, no. 3, pp. 684–704, Mar. 2022, Publisher: SAGE Publications, ISSN: 1461-4448. DOI: 10.1177/1461444820963776. [Online]. Available: https://doi.org/10.1177/1461444820963776 (visited on 01/08/2024).

[208] T. t. p. g. i. S. a. n. l. f. t. i. g. b. c. o. c. D. t. v. u. cycles and S. C. D. M. D. D. T. R. i. t. Text, *Topic: Podcasting Industry*, en. [Online]. Available: https://www.statista.com/topics/3170/podcasting/ (visited on 01/08/2024).

[209] G. W. Benton, G. Fazelnia, A. Wang, and B. Carterette, "Trajectory Based Podcast Recommendation," *ArXiv*, Sep. 2020. [Online]. Available: https://www.semanticscholar.org/paper/Trajectory-Based-Podcast-Recommendation-Benton-Fazelnia/35106f6f4f882c79130a3485b78aa171a5d580ac (visited on 07/27/2023).

[210] L. Yang, Y. Wang, D. Dunne, M. Sobolev, M. Naaman, and D. Estrin, "More than just words: Modeling non-textual characteristics of podcasts," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 276–284.

[211] F. B. Valero, M. Baranes, and E. V. Epure, "Topic modeling on podcast short-text metadata," in *European Conference on Information Retrieval*, Springer, 2022, pp. 472–486.

[212] K. Martikainen, "Audio-based stylistic characteristics of podcasts for search and recommendation: A user and computational analysis," M.S. thesis, University of Twente, 2020.

[213] M. Aziz, A. Wang, A. Pappu, *et al.*, "Leveraging semantic information to facilitate the discovery of underserved podcasts," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3707–3716.

[214] R. Jones, H. Zamani, M. Schedl, *et al.*, "Current challenges and future directions in podcast information access," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1554–1565.

[215] M. Sharpe, "A review of metadata fields associated with podcast rss feeds," *arXiv preprint arXiv:2009.12298*, 2020.

[216] N. Shah, V. Srivastava, M. Bhardwaj, *et al.*, "It's what you say and how you say it: Exploring audio and textual features for podcast data," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2023, pp. 1972–1977.

[217] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: `http://arxiv.org/abs/1908.10084`.

[218] P. Pu, L. Chen, and R. Hu, "Evaluating recommender systems from the user's perspective: Survey of the state of the art," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 317–355, 2012.

[219] L. Chen, M. De Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro, "Human decision making and recommender systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 3, no. 3, pp. 1–7, 2013.

[220] B. P. Knijnenburg and M. C. Willemsen, "Evaluating recommender systems with user experiments," in *Recommender systems handbook*, Springer, 2015, pp. 309–352.

[221] *Ethics guidelines for trustworthy AI | Shaping Europe's digital future*, en, Apr. 2019. [Online]. Available: `https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai` (visited on 12/26/2023).

[222] N. A. Smuha, "The eu approach to ethics guidelines for trustworthy artificial intelligence," *Computer Law Review International*, vol. 20, no. 4, pp. 97–106, 2019.

[223] W. Fan, X. Zhao, X. Chen, *et al.*, "A comprehensive survey on trustworthy recommender systems," *arXiv preprint arXiv:2209.10117*, 2022.

[224] K. Zhang, Q. Cao, F. Sun, *et al.*, "Robust recommender system: A survey and future directions," *ArXiv*, vol. abs/2309.02057, 2023. [Online]. Available: `https://api.semanticscholar.org/CorpusID:261556958`.

[225] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Trans. Internet Techn.*, vol. 7, p. 23, 2007. [Online]. Available: `https://api.semanticscholar.org/CorpusID:16802889`.

[226] Y. Deldjoo, D. Jannach, A. Bellogín, A. Difonzo, and D. Zanzonelli, "Fairness in recommender systems: Research landscape and future directions," *User Modeling and User-Adapted Interaction*, pp. 1–50, 2022. [Online]. Available: `https://api.semanticscholar.org/CorpusID:254927017`.

[227] C. Walker, E. R. Gibney, and S. Hellweg, "Comparison of environmental impact and nutritional quality among a european sample population – findings from the food4me study," *Scientific Reports*, vol. 8, 2018. [Online]. Available: `https://api.semanticscholar.org/CorpusID:5483597`.

[228] S. Robbins, "A misdirected principle with a catch: Explicability for ai," *Minds and Machines*, vol. 29, pp. 495–514, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:204707186`.

[229] E. Aghasian, S. K. Garg, and J. Montgomery, "User's privacy in recommendation systems applying online social network data, a survey and taxonomy," *ArXiv*, vol. abs/1806.07629, 2018. [Online]. Available: `https://api.semanticscholar.org/CorpusID:49318466`.

[230] S. Chandra, S. Verma, W. M. Lim, S. Kumar, and N. Donthu, "Personalization in personalized marketing: Trends and ways forward," en, *Psychol. Mark.*, vol. 39, no. 8, pp. 1529–1562, Aug. 2022.

[231] A.-S. Riegger, J. F. Klein, K. Merfeld, and S. Henkel, "Technology-enabled personalization in retail stores: Understanding drivers and barriers," en, *J. Bus. Res.*, vol. 123, pp. 140–155, Feb. 2021.

[232] G. McLean, K. Osei-Frimpong, K. Al-Nabhani, and H. Marriott, "Examining consumer attitudes towards retailers' m-commerce mobile applications – an initial adoption vs. continuous use perspective," en, *J. Bus. Res.*, vol. 106, pp. 139–157, Jan. 2020.

[233] E. C.-X. Aw, G. W.-H. Tan, T.-H. Cham, R. Raman, and K.-B. Ooi, "Alexa, what's on my shopping list? transforming customer experience with digital voice assistants," en, *Technol. Forecast. Soc. Change*, vol. 180, no. 121711, p. 121 711, Jul. 2022.

[234] N. Dhiman and M. Jamwal, "Tourists' post-adoption continuance intentions of chatbots: Integrating task–technology fit model and expectation–confirmation theory," en, *Foresight*, Sep. 2022.

[235] E. M. Grua, M. De Sanctis, I. Malavolta, M. Hoogendoorn, and P. Lago, "An evaluation of the effectiveness of personalization and self-adaptation for e-health apps," en, *Inf. Softw. Technol.*, vol. 146, no. 106841, p. 106 841, Jun. 2022.

[236] Y. Cheng, S. Sharma, P. Sharma, and K. Kulathunga, "Role of personalization in continuous use intention of mobile news apps in india: Extending the UTAUT2 model," en, *Information (Basel)*, vol. 11, no. 1, p. 33, Jan. 2020.

[237] D. Pérez-Troncoso, D. M. Epstein, and J. A. Castañeda-García, "Consumers' preferences and willingness to pay for personalised nutrition," en, *Appl. Health Econ. Health Policy*, vol. 19, no. 5, pp. 757–767, Sep. 2021.

[238] P.-F. Hsu, T. K. Nguyen, and J.-Y. Huang, "Value co-creation and co-destruction in self-service technology: A customer's perspective," en, *Electron. Commer. Res. Appl.*, vol. 46, no. 101029, p. 101 029, Mar. 2021.

[239] E. Moriuchi, "An empirical study on anthropomorphism and engagement with disembodied AIs and consumers' re-use behavior," en, *Psychol. Mark.*, vol. 38, no. 1, pp. 21–42, Jan. 2021.

[240] A. V. Bogoviz, V. A. Elykomov, V. S. Osipov, K. G. Kelina, and L. A. Kripakova, "Barriers and perspectives of formation of the e-healthcare system in modern russia," in *Ubiquitous Computing and the Internet of Things: Prerequisites for the Development of ICT*, ser. Studies in computational intelligence, Cham: Springer International Publishing, 2019, pp. 917–923.

[241] H. Cho, C. Chi, and W. Chiu, "Understanding sustained usage of health and fitness apps: Incorporating the technology acceptance model with the investment model," en, *Technol. Soc.*, vol. 63, no. 101429, p. 101 429, Nov. 2020.

[242] C. Yoon and E. Rolland, "Understanding continuance use in social networking services," en, *J. Comput. Inf. Syst.*, vol. 55, no. 2, pp. 1–8, Jan. 2015.

[243] R. J. Brodie, D. Linda, B. Hollebeek, and A. Jurić, "Customer engagement: Conceptual domain, fundamental propositions, and implications for research," *Journal of Service Research*, vol. 14, no. 3, pp. 252–271, 2011.

[244] J. Fang, Z. Zhao, C. Wen, and R. Wang, "Design and performance attributes driving mobile travel application engagement," en, *Int. J. Inf. Manage.*, vol. 37, no. 4, pp. 269–283, Aug. 2017.

[245] A. Dovaliene, A. Masiulyte, and Z. Piligrimiene, "The relations between customer engagement, perceived value and satisfaction: The case of mobile applications," en, *Procedia Soc. Behav. Sci.*, vol. 213, pp. 659–664, Dec. 2015.

[246] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q*, vol. 13, no. 3, p. 319, Sep. 1989.

[247] S. Koul and A. Eydgahi, "A systematic review of technology adoption frameworks and their applications," *Journal of Technology Management & Innovation*, 2017.

[248] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, 2003, Publisher: Management Information Systems Research Center, University of Minnesota, ISSN: 0276-7783. DOI: `10.2307/30036540`. [Online]. Available: `https://www.jstor.org/stable/30036540` (visited on 06/19/2023).

[249] V. Venkatesh, J. Y. L. Thong, and X. Xu, "Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology," *MIS Quarterly*, vol. 36, no. 1, pp. 157–178, 2012, Publisher: Management Information Systems Research Center, University of Minnesota, ISSN: 0276-7783. DOI: `10.2307/41410412`. [Online]. Available: `https://www.jstor.org/stable/41410412` (visited on 06/19/2023).

[250] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 157–164.

[251] S. R. Stoyanov, L. Hides, D. J. Kavanagh, O. Zelenko, D. Tjondronegoro, and M. Mani, "Mobile app rating scale: A new tool for assessing the quality of health mobile apps," en, *JMIR MHealth UHealth*, vol. 3, no. 1, e27, Mar. 2015.

[252] A. D. Starke, C. Musto, A. Rapp, G. Semeraro, and C. Trattner, ““tell me why”: Using natural language justifications in a recipe recommender system to support healthier food choices,” *User Modeling and User-Adapted Interaction*, pp. 1–34, 2023.

[253] J. Weston, E. Dinan, and A. Miller, “Retrieve and Refine: Improved Sequence Generation Models For Dialogue,” in *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, A. Chuklin, J. Dalton, J. Kiseleva, A. Borisov, and M. Burtsev, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 87–92. DOI: `10.18653/v1/W18-5713`. [Online]. Available: `https://aclanthology.org/W18-5713` (visited on 12/28/2023).

[254] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, *A Survey on Retrieval-Augmented Text Generation*, arXiv:2202.01110 [cs], Feb. 2022. DOI: `10.48550/arXiv.2202.01110`. [Online]. Available: `http://arxiv.org/abs/2202.01110` (visited on 12/28/2023).

[255] A. Lozano, S. L. Fleming, C.-C. Chiang, and N. Shah, *Clinfo.ai: An Open-Source Retrieval-Augmented Large Language Model System for Answering Medical Questions using Scientific Literature*, arXiv:2310.16146 [cs], Oct. 2023. DOI: `10.48550/arXiv.2310.16146`. [Online]. Available: `http://arxiv.org/abs/2310.16146` (visited on 12/28/2023).

[256] P. Lewis, E. Perez, A. Piktus, *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: `https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html` (visited on 12/28/2023).

[257] N. Reimers and I. Gurevych, “Making monolingual sentence embeddings multilingual using knowledge distillation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2020. [Online]. Available: `https://arxiv.org/abs/2004.09813`.

[258] *Sentence-transformers/multi-qa-mpnet-base-dot-v1*, https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1, Accessed: 2023-03-23.

[259] A. Warstadt, Y. Zhang, H.-S. Li, H. Liu, and S. R. Bowman, "Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually)," *arXiv preprint arXiv:2010.05358*, 2020.

[260] E. Hulburd, "Exploring bert parameter efficiency on the stanford question answering dataset v2. 0," *arXiv preprint arXiv:2002.10670*, 2020.

[261] *Tinyroberta-squad2 model card*, https://huggingface.co/deepset/tinyroberta-squad2, Accessed: 2023-03-23.

[262] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7B*, arXiv:2310.06825 [cs], Oct. 2023. DOI: 10.48550/arXiv.2310.06825. [Online]. Available: http://arxiv.org/abs/2310.06825 (visited on 01/23/2024).

[263] *Dietary Supplement Fact Sheets*, en. [Online]. Available: https://ods.od.nih.gov/factsheets/list-all/ (visited on 03/17/2023).

[264] EFSA Panel on Dietetic Products, Nutrition and Allergies (NDA), "Scientific opinion on dietary reference values for iron," *EFSA Journal*, vol. 13, no. 10, p. 4254, 2015. DOI: https://doi.org/10.2903/j.efsa.2015.4254. eprint: https://efsa.onlinelibrary.wiley.com/doi/pdf/10.2903/j.efsa.2015.4254. [Online]. Available: https://efsa.onlinelibrary.wiley.com/doi/abs/10.2903/j.efsa.2015.4254.

[265] *Eufic calcium factsheet*, https://www.eufic.org/en/vitamins-and-minerals/article/calcium-foods-functions-how-much-do-you-need-more, Accessed: 2023-03-23.

[266] *Eufic fibre factsheet*, https://www.eufic.org/en/whats-in-food/article/what-is-dietary-fibre-and-is-it-beneficial, Accessed: 2023-03-23.

[267] P. Dhar, "The carbon impact of artificial intelligence," en, *Nature Machine Intelligence*, vol. 2, no. 8, pp. 423–425, Aug. 2020, Number: 8 Publisher: Nature Publishing Group, ISSN: 2522-5839. DOI: 10.1038/s42256-020-0219-9. [Online]. Available: https://www.nature.com/articles/s42256-020-0219-9 (visited on 01/11/2024).

[268] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, *Green AI*, arXiv:1907.10597 [cs, stat], Aug. 2019. DOI: 10.48550/arXiv.1907.10597. [Online]. Available: http://arxiv.org/abs/1907.10597 (visited on 01/11/2024).

[269] J. Coleman, *AI's Climate Impact Goes beyond Its Emissions*, en. [Online]. Available: https://www.scientificamerican.com/article/ais-climate-impact-goes-beyond-its-emissions/ (visited on 01/11/2024).

[270] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.

[271] *Mlco2/codecarbon: Track emissions from Compute and recommend ways to reduce their impact on the environment.* [Online]. Available: https://github.com/mlco2/codecarbon (visited on 01/11/2024).

[272] *Spain: Power sector carbon intensity 2022*, en. [Online]. Available: https://www.statista.com/statistics/1290486/carbon-intensity-power-sector-spain/ (visited on 01/11/2024).

[273] O. US EPA, *Greenhouse Gases Equivalencies Calculator - Calculations and References*, en, Data and Tools, Aug. 2015. [Online]. Available: https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references (visited on 01/11/2024).