Development of recommender systems using social media data in order to palliate the cold start problem

Programa de Doctorado en Tecnologías de la Información y la Comunicación

Universidad de Granada

Julio Herce Zelaya
julioherce@gmail.com

Editor: Universidad de Granada. Tesis Doctorales
Autor: Jorge Chamarro Padial
ISBN: 978-84-1117-997-3
URI: https://hdl.handle.net/10481/84449

UNIVERSIDAD
DE GRANADA

Programa de Doctorado en Tecnologías de la Información y la Comunicación

Development of recommender systems using social media data in order to palliate the cold start problem

Desarrollo de sistemas de recomendación usando información de redes sociales para paliar el problema del arranque en frío

Memoria de tesis doctoral presentada por:

Julio Herce Zelaya

Para optar al título de Doctor por la Universidad de Granada
dentro del Programa de Doctorado en Tecnologías de la Información y la Comunicación
Granada, Mayo de 2023.

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingenierías Informática y de Telecomunicación
Universidad de Granada

UNIVERSIDAD
DE GRANADA

Programa de Doctorado en Tecnologías de la Información y la Comunicación

Development of recommender systems using social media data in order to palliate the cold start problem

Memoria de tesis doctoral presentada por:

Julio Herce Zelaya

Para optar al título de Doctor por la Universidad de Granada
dentro del Programa de Doctorado en Tecnologías de la Información y la Comunicación
Granada, Mayo de 2023.

DIRECTORES:
Carlos Gustavo Porcel Gallego
Enrique Herrera Viedma

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingenierías Informática y de Telecomunicación
Universidad de Granada

El Doctorando D. Julio Herce Zelaya y los directores de la tesis D. Carlos Gustavo Porcel Gallego y D. Enrique Herrera Viedma:

Garantizamos, al firmar esta Tesis Doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores al ser citados, cuando se han utilizado sus resultados o publicaciones.

D. Julio Herce Zelaya
Doctorando

Dr. D. Carlos Gustavo Porcel Gallego
Director

Dr. D. Enrique Herrera Viedma
Director

Como directores de la tesis titulada “Desarrollo de sistemas de recomendación usando información de redes sociales para paliar el problema del arranque en frío”, D. Carlos Gustavo Porcel Gallego y D. Enrique Herrera Viedma consideramos que la modalidad más apropiada para su presentación es por agrupación de publicaciones y, por tanto, AUTORIZAMOS la presentación de la tesis bajo dicha modalidad.

Dr. D. Carlos Gustavo Porcel Gallego
Director

Dr. D. Enrique Herrera Viedma
Director

Research project and funding

El desarrollo de esta tesis ha sido posible gracias a las subvenciones obtenidas con los siguientes proyectos de investigación:

- TIN2016-75850-R, titulado "Sistemas inteligentes de toma de decisión y consenso en ambiente difuso: Aplicaciones en e-salud y e-comercio" y financiado por el Ministerio de Economía y Competitividad.
- PID2019-103880RB-I00, titulado "Sistemas de toma de decisiones en grupo disruptivos en ambiente difuso: aplicaciones en gestión inteligente de energía y empleados" y financiado por el Ministerio de Ciencia e Innovación
- P20_00673, titulado "Nuevos sistemas difusos para la toma de decisiones: Aplicaciones en entornos digitales" y financiado por la Conserjería de Transformación Económica, Industria, Conocimiento y Universidades. Junta de Andalucía.

Acknowledgements

Quiero empezar agradeciendo a mis dos directores de tesis: Enrique y Carlos. Enrique por haber facilitado tantas tareas y haber ayudado con su experiencia de investigador de altísimo prestigio. Carlos por haber tenido la paciencia para estar siempre ahí para ayudarme, para enseñarme y para motivarme cuando lo necesitaba.

Otros de los grandes responsables de que esté hoy aquí y a los que quiero agradecer son Álvaro y Juan, compañeros y amigos que siempre me han ofrecido ayuda y consejo y que antes incluso de empezar la tesis me guiaron y me ayudaron a encauzar mis ideas por el camino de la investigación. De un valor incalculable son también las charlas que manteníamos durante las tardes y las noches en Múnich hablando de temas relacionados con la tesis y otros miles más.

También me gustaría agradecer a mi familia. A mis padres porque me dieron la oportunidad de llegar donde estoy y me enseñaron que todo esfuerzo tiene su recompensa.

Gracias a ti también, Abuela. Tu luz permanece.

Por último, pero no por ello menos importante, me gustaría agradecer a Carmen, mi mujer, por aguantar mis ausencias y a mi hijo Julio porque, sin saberlo, me dio la motivación para siempre seguir adelante.

Table of contents

List of Figures

List of Tables

Abstract

Nowadays we live in a period where there are plenty of options for consuming content online, either books, films or music. New material is released every day and users can consume this content with just a couple of clicks. Despite of this vast amount of options, or maybe due to that, it is more difficult than ever for users to find content that they would enjoy consuming. Sometimes this process can feel like looking for a needle in a haystack. The role of recommender systems is to filter all this content and to provide only the interesting items to the users. These systems are normally based on historical data from the users with other items. For example previous ratings of items can be used to recommend similar items to the ones were ranked highest. One of the most common pitfalls from these systems is the cold start problem. This problem occurs when either a new user or new item is introduced in the system and, therefore, there are no previous data that could be leverage by the recommender systems in order to create recommendations. This problem has an ever-growing importance due to the huge offer of online services for consuming content. These systems need to be prepared to engage users that recently join their platforms by offering them contents that the users would enjoy. Otherwise there is a high risk that these users would leave and find another platform. This topic is widely studied in the literature but due to its importance and its peculiarities, like different domain behaviour or lack of appropriate datasets to study this problem, there is still much to study and research and the current state-of-the-art algorithms have room for improvement. In this proposal, this issue is addressed from different perspectives and applied for different domains and scenarios. The goal of this work is to alleviate the cold start problem and for that we develop models using artificial intelligence algorithms that make use of users' contextual data from their social media profiles. These models outperform the state-of-the-art models for cold start problem. In addition to that, in this proposal a dedicated dataset has been created. This dataset is optimised for the study of the cold start problem and has data about movie rating, movie description and user description. This will ease future researches since such datasets are rather scarce in the literature. Other areas of this proposal is hedge fund management or reinforcement for recommendation of education resources in the field of oral surgery and implantology and how to address these topics when there is not much previous data available.

This work obtained optimal results, matching, and even improving results from state-of-the-art algorithms for recommendation systems with cold-start problem. This is obtained through the leverage of implicit data, extracted automatically from different sources not having to require the user to provide any manual data.

Resumen

A día de hoy vivimos en la edad de la información y estamos expuestos a una sobrecarga de contenido. En los últimos años la cantidad de opciones disponibles para consumir contenido ha crecido exponencialmente. A veces, esta gran cantidad de contenido dificulta la elección del usuario cuando tiene que tomar alguna decisión, por lo que es de gran importancia contar con herramientas automáticas que ayuden en esa toma de decisiones. Es aquí donde cobran una gran importancia los sistemas de recomendación, que usando nuestros patrones de búsqueda, visionado y valoraciones son capaces de recomendarnos productos que sean de nuestro interés. Sin embargo uno de los problemas más comunes de los sistemas de recomendación es el problema del arranque en frío, esto es, la situación de cuando un usuario acaba de unirse a una plataforma y la plataforma no tiene ningún dato sobre él que pueda ser usado para crear recomendaciones. Este problema tiene una importancia creciente a día de hoy ya que cada vez hay más servicios distintos de retransmisión de películas en streaming (Netflix, Prime Video, HBO, Filmin, Apple TV, Disney+), de reproducción de música en streaming (Spotify, Apple Music, Amazon Music) y no es raro que usuarios cambien de una plataforma a otra frecuentemente. Es por eso que se convierte en un problema de gran relevancia y se vuelve vital que las plataformas dispongan de sistemas de recomendación que sean resilientes al problema de arranque en frío.

En esta propuesta, el problema del arranque en frío se palia mediante el uso de información implícita sobre el usuario usando datos de distintas fuentes como redes sociales, portales de noticias o información acerca de la confianza en otros usuarios similares. Este problema ha sido ampliamente estudiado en la literatura, pero por su importancia y peculiaridades, como comportamiento variables en distintos entornos o la dificultad de encontrar datos en los que basar este tipo de sistemas, aún hay mucho trabajo por hacer. En este trabajo se abordan los sistemas de recomendación desde enfoques y dominios tan diversos como la recomendación de películas a usuarios de portales de películas basándose en datos extraídos de redes sociales; la elección de acciones para optimizar inversiones usando históricos del mercado y datos de noticias relacionadas con esas acciones; y recomendación de recursos académicos usando un sistema de confianza entre usuarios.

Los resultados obtenidos en las diferentes propuestas de este trabajo han sido prometedores porque igualan e incluso sobrepasan resultados obtenidos en la literatura con respecto a recomendación bajo la influencia del problema de arranque en frío. Estos resultados se obtienen gracias al uso de información implícita acerca del usuario extraída de distintas fuentes de manera automática, sin necesitar que el usuario proporcione, activamente, datos de forma manual.

Chapter 1

Introduction

In this chapter, the Doctoral Thesis will be introduced. The chapter consists of a general introduction, with three subsections; one for recommender systems in general, one focused on the cold start problem and the last one for the justification of this Doctoral Thesis. Afterwards, there is one section for each study included in this Thesis where the particular studies will be introduced.

1.1 General Introduction

1.1.1 Recommender Systems

Recommender systems are algorithms or techniques used to provide personalized recommendations to users based on their preferences, interests, and past behavior. These systems aim to assist users in discovering items or content they are likely to find interesting or useful, thereby enhancing user experience and increasing user engagement.

Recommender systems typically work by analyzing large amounts of data, including user data and item data, to identify patterns and make predictions about user preferences. The following list describes a step-by-step breakdown of how recommender systems normally work:

1. Data Collection: The system collects data about users and items. User data can include explicit feedback such as ratings or reviews, as well as implicit feedback like browsing history or purchase records. Item data consists of attributes and features that describe the items.

2. Data Preprocessing: The collected data is processed and transformed into a suitable format for analysis. This may involve cleaning the data, handling missing values, normalizing scales, or extracting relevant features.

3. User and Item Representation: The system represents users and items in a way that captures their characteristics. This representation could be based on demographic information, historical behavior, or content-based features such as item descriptions or genres.

4. Similarity or Preference Estimation: The system calculates the similarity between users or items based on their representations. Similarity measures can include cosine similarity, Euclidean distance, or correlation coefficients. Alternatively, the system estimates the preference of a user for an item using collaborative filtering techniques or matrix factorization methods.

5. Recommendation Generation: Based on the calculated similarities or estimated preferences, the system generates recommendations for each user. These recommendations can be ranked and presented to the user in various ways, such as a list of top-N items, personalized rankings, or even in the form of targeted advertisements.

Recommender systems are often classified in three classical types: collaborative filtering, content-based filtering and hybrid approach [9, 11].

1. Collaborative Filtering: This approach recommends items to users based on the preferences and behaviors of similar users. It identifies users with similar tastes and recommends items that those similar users have liked or rated highly [8].

2. Content-Based Filtering: Content-based recommender systems recommend items to users based on the characteristics and attributes of the items themselves. It analyzes the content or features of items that users have liked in the past and suggests similar items [29].

3. Hybrid Recommender Systems: These systems combine multiple approaches, such as collaborative filtering and content-based filtering, to provide more accurate and diverse recommendations. By leveraging the strengths of different techniques, hybrid systems can overcome limitations and improve recommendation quality [10].

Other new classifications can include the following recommender system types.

1. Knowledge-Based Filtering: Knowledge-based filtering relies on explicit knowledge about users' preferences and item characteristics. It uses domain-specific knowledge or expert-defined rules to make recommendations. This approach is useful when there is limited or no user interaction data available.

2. Context-Aware Recommender Systems: Context-aware recommender systems take into account contextual information, such as time, location, or user context, to provide personalized recommendations. By considering the situational factors surrounding the user, these systems can offer more relevant and timely recommendations.

3. Demographic-Based Filtering: Demographic-based filtering recommends items based on demographic information, such as age, gender, or location. It assumes that individuals with similar demographic characteristics may have similar preferences.

Recommender systems play a crucial role in various domains and industries for several reasons:

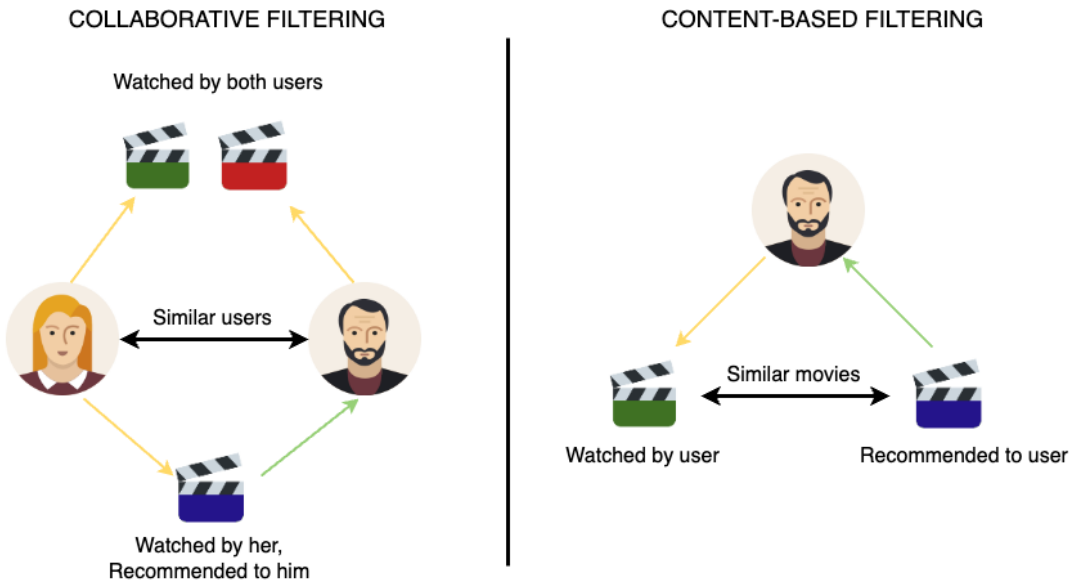


Figure 1.1: Classic Recommender systems approaches

1. Personalization: Recommender systems enable personalized experiences by suggesting items tailored to individual preferences. This helps users find relevant and interesting content in an overwhelming sea of options.

2. Discovery and Exploration: Recommender systems can introduce users to new items or content they might not have discovered otherwise. By suggesting items outside their usual preferences, users can explore and broaden their interests.

3. Increased Engagement and Sales: By recommending items users are likely to be interested in, recommender systems increase user engagement, leading to longer sessions and increased interactions. This can result in higher conversion rates, sales, and customer satisfaction.

4. Improved User Experience: Recommender systems enhance the overall user experience by reducing information overload and making the user interface more intuitive. By tailoring recommendations to individual users, they make the interaction with the system more efficient and enjoyable.

5. Business Insights: The data collected and analyzed by recommender systems provide valuable insights into user behavior, preferences, and market trends. This information can be used for business intelligence, marketing strategies, and decision-making processes.

In summary, recommender systems utilize data analysis and machine learning techniques to provide personalized recommendations to users, improving user experience, engagement, and business outcomes.

1.1.2 Cold start problem in recommender systems

The cold start problem in recommender systems refers to the challenge of providing accurate and effective recommendations for new users or items that have limited or no historical data available. It arises when there is insufficient information to understand the preferences and characteristics of these new entities. The cold start problem can occur in two forms [23, 32]: the user cold start and the item cold start as indicated in figure 1.2.

Figure 1.2: Cold start problem types

1. User Cold Start: This occurs when a new user joins the system, and there is limited or no information about their preferences, ratings, or behavior. Without sufficient data, it becomes challenging to generate personalized recommendations for the user [12, 19, 38].

2. Item Cold Start: This occurs when a new item is added to the system, and there is limited or no historical data about its usage, ratings, or characteristics. Without prior user feedback, it is difficult to understand the item's properties and make accurate recommendations [45].

Addressing the cold start problem is crucial for recommender systems because it allows for better user engagement, improved recommendation quality, and increased system usefulness. Failing to address this problem can result in poor user experiences, missed opportunities for user satisfaction, and potential loss of revenue for businesses.

Various approaches have been proposed in the scientific literature to tackle the cold start problem in recommender systems:

1. Content-Based Methods: Content-based approaches rely on item features or attributes to generate recommendations. In the case of the item cold start, these methods can utilize item content, such as textual descriptions or metadata, to infer the item's characteristics

and make recommendations. For user cold start, content-based methods can rely on demographic information or user profiles to generate initial recommendations.

2. Knowledge-based Approaches: Knowledge-based techniques utilize domain-specific knowledge or expert systems to provide recommendations. These methods leverage domain knowledge, such as item-item relationships, item characteristics, or user preferences inferred from explicit user input, to generate recommendations even in the absence of historical data.

3. Hybrid Methods: Hybrid approaches combine multiple recommendation techniques to address the cold start problem. By combining collaborative filtering, content-based filtering, or knowledge-based methods, these approaches can leverage the strengths of different techniques to handle new users and items more effectively.

4. Active Learning: Active learning techniques involve actively collecting feedback from users to gather initial data and reduce the cold start problem. These methods incorporate mechanisms to encourage user participation, such as explicit rating requests or surveys, to quickly gather user preferences and bootstrap the recommender system.

5. Context-aware Methods: Context-aware recommender systems consider contextual information, such as time, location, or user context, to provide more personalized recommendations. By leveraging contextual information, these systems can make better inferences about new users or items and provide relevant recommendations.

Recent advancements in addressing the cold start problem have explored the application of deep learning techniques, such as neural networks and deep neural networks, to improve recommendation accuracy [42]. These approaches can handle sparse data [31] more effectively and capture complex patterns and relationships between users and items, even with limited information.

Additionally, researchers have explored the use of transfer learning, where knowledge from a source domain with sufficient data is transferred to the target domain with limited data. This enables the system to leverage knowledge gained from existing users or items to make accurate recommendations for new users or items. [18]

In conclusion, the cold start problem poses a significant challenge for recommender systems. However, through various techniques such as content-based methods, knowledge-based approaches, hybrid methods, active learning, and context-aware methods, researchers have made advancements in addressing this problem. Recent scientific literature has explored the application of deep learning and transfer learning techniques to improve recommendation accuracy for new users and items. By addressing the cold start problem, recommender systems can provide more effective and personalized recommendations, leading to enhanced user experiences and improved system performance.

1.1.3 Justification of doctoral thesis

Recommender systems have become essential tools for personalized content delivery and enhancing user experiences in different domains. However, the cold start problem poses a significant challenge, limiting the effectiveness and accuracy of these systems for new users and items. As the demand for recommender systems continues to grow, addressing the cold start problem becomes imperative to ensure their widespread adoption and utility. Recommender systems from real world are facing more than ever the cold-start problem, since there are more and more new online content services everyday which may lack from data from new users, and therefore have to learn to deal with this problem and build techniques to palliate it. This doctoral thesis aims to provide a comprehensive study of the cold start problem in recommender systems, investigate its underlying causes, and propose novel solutions to mitigate its impact.

Even though, the literature has extensively treated the cold start problem, there is still room for improvement in the accuracy of systems, in the efficiency of using contextual data and in the dataset foundation level, since there is a lack of specialized datasets optimized for the study of the cold start problem.

In the following sections from this chapter, the different studies from this doctoral thesis will be explained.

1.2 Study 1

Recommender systems are more relevant than ever, since they help users to find relevant content within huge amount of content. In order to achieve that, these systems leverage previous behaviours and analogies between users to predict new demands or preferences [11, 14, 25].

Many recommendation proposals leverage data provided by the user in order to be able to recommend items or products to them. These data could be acquired either in a explicit (e.g. by rating a product) or implicit way (e.g. by establishing a connection with another user). However, when a user is new in the system, there is a lack of ratings history. Due to these circumstances it is difficult to infer user's preferences. This problem is referred to as the cold-start problem [8, 18, 44]. Online services providing content are aware of these issues and in order to palliate them, they often offer to the users the opportunity to make a voting tour in where they are asked to rate a series of items before the site can recommend others to them. Other sites ask users to give some info about them [36].

One drawback from these methods is that they require some explicit action and effort from the user and they tend to be reluctant to make these movie rate tours or give more info about them.

The main goal from this study is to leverage implicit data from the user's social media stream [5, 43] creating a profile which would be determined by a collection of features depicting their behaviours, tastes and character.

Since the amount of information we had to deal was huge, we leverage Big Data tools and machine learning techniques to create the recommendations [28, 31, 33, 35]. Our approach

is based on the use of decision trees and random decision forests that will assist us for the classification of the users according to their profiles, allowing us to obtain significantly better recommendations.

Then, the starting point from our study is a model to process the social stream for every single user. This stream will be used to create the so called Twitter Profile. Then this Twitter profile is utilized in order to classify the users in relation with the actual rating of every item, with the help of decision trees (either single decision tree or random forest). After this classification process the items predicted to be recommended for the user with the highest probability will be selected. The results achieved after the validation and experimentation phase points a favourable behaviour of the proposed model, being therefore very suitable for its application in a real environment.

1.3 Study 2

In the recent years, cognitive computing has experienced a substantial growth [20, 21]. The main cloud providers provide APIs for the developers' community to run these services and create a wider range of applications [22]. One of the areas covered by these services is sentiment analysis, which includes a combination of natural language processing, text analysis, computational linguistics, and biometrics to identify, extract, and quantify in a systematic way affective states and subjective information inherent in the human communication [1, 2, 4, 24].

Sentiment analysis has undergone a remarkable development in the last years too, becoming one of the most prolific research areas in the Natural Language Processing field [3, 4]. The computation of sentiments relies heavily on the existence of polarity dictionaries, where lemmas are given a score (usually between -1 and 1) representing the contribution of words containing this lemma to the overall sentiment of the particular sentence. We face the so called thresholding and scaling problem.

Relying too heavily on polarity scores presents a number of challenges, particularly in relation to contextual bias. Polarity dictionaries assign an immutable score to each lemma, without accounting for contextual nuances. Creating context-aware polarity scores is difficult due to the lack of guiding principles or systematic methods for obtaining scores. In previous work [6, 7], methods for polarity bias modelling within a specific context were defined, along with a volatility score to assess the reliability of the bias modelling.

This article seeks to move beyond quantifying polarity bias and explore automated methods of inferring polarity scores for the finance markets domain. Sentiment analysis has been widely used in predicting stock market movements, identifying change points, assessing market sentiment, and quantifying bearish or bullish phases. The finance markets domain is particularly suitable for studying emotions and sentiments due to the large number of finance-related news articles produced each day, as well as the availability of almost real-time pricing and trading volume information. By assuming that choice of words, tonality, emotional load, and sentiment are correlated with stock market changes, historical price and news data can be used to correct existing polarity scores or develop new ones for words in the news.

The main contribution of this paper is an approach to automatically extracting a polarity dictionary from the stock market domain without human intervention and addressing the scaling and thresholding problem. This is achieved through a new technique for extracting and labeling news with price change magnitude, creating a weighted news collection for further processing. A binned corpus is introduced to facilitate standard information retrieval techniques, such as term frequency-inverse document frequency, to extract guiding polarity values for each term. An embeddings-based approach is used to compute term neighborhoods and disseminate guiding polarity values to other terms. Finally, crisp polarity scores are transformed into fuzzy linguistic sets to provide more generalizable results and reduce the impact of imposed thresholds. The volatility of the polarity score is also computed based on support from the domain content.

Even though in this work, we do not present the topic as such, this proposal can be extrapolated and incorporated to recommender systems to solve the cold start problem by using sentiment analysis for analysing existing texts from the user (i.e. from social stream) and to enable the implicit profile data extraction reducing the manual provided data to the minimum and establishing in this way more efficient user profile creation.

Overall, this approach solves the scale and thresholding problem by providing fuzzy linguistic sets instead of crisp polarity values, addresses human bias by inferring polarity values without human intervention, and contextualizes polarity values to the specific domain.

1.4 Study 3

Web technologies are driving the development of innovative pedagogical models that complement traditional education [29]. These new technologies enhance teaching and learning processes by providing efficient and easy information broadcasting and global communication tools that encourage collaborative learning [13, 39]. Personalized education [37] can be highly beneficial in helping students to reinforce areas where they need help and maximizing their potential in areas where they excel. It is important for education to be dynamic and adaptable to meet the changing needs of students.

Recommender systems are personalized services that seek to identify valuable information items for users. These systems require information about each user, such as their ratings of analyzed items, to generate a list of personalized recommendations [11, 17, 40]. Collaborative approaches are commonly used to generate recommendations, based on the ratings of users with similar profiles. However, the need for many ratings to obtain a good performance can pose a challenge, as users typically provide only a few ratings, making it difficult to compute user similarity [34]. Trust is a crucial element in online social networks, and incorporating trust models into recommendation systems can help overcome this challenge [26, 41].

This article introduces a new fuzzy linguistic recommender system that incorporates the concept of trust in the recommendations generation engine. The system is designed for students of Dentistry from the Dentistry School of the University of Granada in Spain. Its key features include an approach to estimating trust scores between users, personalized recommendations based on trusted users rather than users with similar ratings history or pedagogical needs, a

user-friendly interface using multi-granular fuzzy linguistic modeling [27, 30], the ability to use it anywhere and anytime, and reliable information and exercises endorsed by a team of experts in oral surgery from the Dentistry School of the University of Granada. In this work we aim to palliate the cold start problem by leveraging the propagated trust among users. Trust is supporting the palliation of the cold start problem since, in this work, the knowledge is extracted from users' social environment

1.5 Study 4

Recommender systems have become the go-to experts for helping users find items that match their preferences [11, 25]. These systems rely on previous user data to generate recommendations, either by using content-based methods or collaborative filtering. However, the cold start problem arises when there is no previous data from the user, making it impossible to provide tailored recommendations [8, 18, 44]. The typical solutions, such as asking users about their interests or asking them to rate items, require significant effort and time from the user.

This work proposes a new approach that takes implicit data from the user's social media stream instead of explicitly asking for data [15, 16, 31]. These data are used to generate a user profile that can be used to classify users and create predictions for the recommender system. To overcome the lack of a suitable dataset for the cold start problem, we created a new dataset optimized for this purpose. The synthetic dataset includes movies, user profiles, and ratings, and its data is extracted from Filmaffinity and Twitter. This dataset has a duality nature, it contains user domain data (ratings about movies), and it also contains user behavior data. Also it provides item related data (movie description). This duality nature can serve, together with recommender techniques, to palliate cold start problem since cross-domain correlation can be leveraged for the creation of algorithms.

Compared to other public datasets available for recommender systems, this new dataset is unique in the sense that it includes behavioral user data, making it ideal for creating models that leverage implicit data from users. The authors also provide an example of using this dataset to design and evaluate a recommender system model that uses a mixed approach between collaborative filtering and content-based methods. The evaluation results indicate that the model provide highly accurate recommendations.

Bibliography

[1] Appel, Orestes; Chiclana, Francisco; Carter, Jenny y Fujita, Hamido: «A hybrid approach to the sentiment analysis problem at the sentence level». *Knowledge-based Systems*, 2016, **108**, pp. 110–124.

[2] ——: «A Consensus Approach to the Sentiment Analysis Problem Driven by Support-Based IOWA Majority». *International Journal of Intelligent Systems*, 2017, **32(9)**, pp. 947–965.

[3] ——: «Cross-ratio uninorms as an effective aggregation mechanism in Sentiment Analysis». *Knowledge-Based Systems*, 2017, **124**, pp. 16–22.

[4] ——: «Successes and challenges in developing a hybrid approach to sentiment analysis». *Applied Intelligence*, 2018, **48(5)**, pp. 1176–1188.

[5] Bernabé-Moreno, J.; Tejeda-Lorente, A.; Porcel, C. y Herrera-Viedma, E: «A new model to quantify the impact of a topic in a location over time with Social Media». *Expert Systems with Applications*, 2015, **42**, pp. 3381–3395.

[6] Bernabé-Moreno, Juan; Tejeda-Lorente, Alvaro; Porcel, Carlos y Herrera-Viedma, Enrique: «A Fuzzy Linguistics Supported Model to Measure the Contextual Bias in Sentiment Polarity». En: *Advances in Fuzzy Logic and Technology 2017*, pp. 199–210. Springer, 2017.

[7] ——: «An Embeddings Based Fuzzy Linguistics Supported Model to Measure the Contextual Bias in Sentiment Polarity». En: *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain, 26-28 September 2018*, pp. 735–748, 2018. doi: 10.3233/978-1-61499-900-3-735. https://doi.org/10.3233/978-1-61499-900-3-735

[8] Bobadilla, J.; Ortega, F.; Hernando, A. y Bernal, J.: «A collaborative filtering approach to mitigate the new user cold start problem». *Knowledge-Based Systems*, 2012, **26**, pp. 225–238.

[9] Bobadilla, J.; Ortega, F.; Hernando, A. y Gutiérrez, A.: «Recommender systems survey». *Knowledge-Based Systems*, 2013, **46**, pp. 109–132.

[10] Burke, R.: «Hybrid Web Recommender Systems». *P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS*, 2007, **4321**, pp. 377–408.

[11] Burke, R.; Felfernig, A. y Göker, M.H.: «Recommender systems: An overview». *AI Magazine*, 2011, **32**, pp. 13–18.

[12] Chien, C.; Yu-Hao, W.; Meng-Chieh, C. y Yu-Chun, S.: «An effective recommendation method for cold start new users using trust and distrust networks». *Information Sciences*, 2013, **224**, pp. 19–36.

[13] Dascalu, M.I.; Bodea, C.N.; Moldoveanu, A.; Mohora, A.; Lytras, M. y Ordoñez de Pablos, P.: «A recommender agent based on learning styles for better virtual collaborative learning experiences». *Computers in Human Behavior*, 2015, **45**, pp. 243–253.

[14] Edmunds, A. y Morris, A.: «The problem of information overload in business organizations: a review of the literature». *International Journal of Information Management*, 2000, **20**, pp. 17–28.

[15] Esmaeili, L.; Mardani, S.; Golpayegani, S.A.H. y Madar, Z.Z.: «A novel tourism recommender system in the context of social commerce». *Expert Systems With Applications*, 2020, **149**, p. 113301. doi: https://doi.org/10.1016/j.eswa.2020.113301.

[16] García-Sánchez, F.; Colomo-Palacios, R. y Valencia-García, R.: «A social-semantic recommender system for advertisements». *Information Processing and Management*, 2020, **57**, p. 102153. doi: https://doi.org/10.1016/j.ipm.2019.102153.

[17] Goga, M.; Kuyoro, S. y Goga, N.: «A recommender for improving the student academic performance». *Procedia - Social and Behavioral Sciences*, 2015, **180**, pp. 1481–1488.

[18] Gonzalez Camacho, L.A. y Nice Alves-Souza, S.: «Social network data to alleviate cold-start in recommender system: A systematic review». *Information Processing and Management*, 2018, **54**, pp. 529–544.

[19] Hernando, A.; J., Bobadilla.; Ortega, F. y Gutiérrez, A.: «A probabilistic model for recommending to new cold-start non-registered users». *Information Sciences*, 2017, **376**, pp. 216–232.

[20] High, Rob: «The era of cognitive systems: An inside look at IBM Watson and how it works». *IBM Corporation, Redbooks*, 2012.

[21] Hwang, Kai y Chen, Min: *Big-data analytics for cloud, IoT and cognitive computing*. John Wiley & Sons, 2017.

[22] Koneru, Anupriya; Bhavani, Nerella Bala Naga Sai Rajani; Rao, K Purushottama; Prakash, Garikipati Sai; Kumar, Immadisetty Pavan y Kumar, Velimala Venkat: «Sentiment Analysis on Top Five Cloud Service Providers in the Market». En: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 293–297. IEEE, 2018.

[23] Lika, B.; Kolomvatsos, K. y Hadjiefthymiades, S.: «Facing the cold start problem in recommender systems». *Expert Systems with Applications*, 2014, **41**, pp. 2065–2073.

[24] Liu, Bing: «Sentiment analysis and opinion mining». *Synthesis lectures on human language technologies*, 2012, **5(1)**, pp. 1–167.

[25] Martínez-Cruz, C.; Porcel, C.; Bernabé-Moreno, J. y Herrera-Viedma, E.: «A Model to Represent Users Trust in Recommender Systems using Ontologies and Fuzzy Linguistic Modeling». *Information Sciences*, 2015, **311**, pp. 102–118. doi: doi:10.1016/j.ins.2015.03.013.

[26] Massa, P. y Avesani, P.: *Computing with Social Trust*. capítulo Trust metrics in recommender systems, pp. 259–285. Springler, 2009.

[27] Mata, F.; Martínez, L. y Herrera-Viedma, E.: «An Adaptive Consensus Support Model for Group Decision Making Problems in a Multi-Granular Fuzzy Linguistic Context». *IEEE Transactions on Fuzzy Systems*, 2009, **17(2)**, pp. 279–290.

[28] Meng-Yen, H.; Tien-Hsiung, W. y Kuan-Ching, L.: «A keyword-aware recommender system using implicit feedback on Hadoop». *J. Parallel Distrib. Comput.*, In press.

[29] Money, W.H. y Dean, B.P.: «Incorporating student population differences for effective online education: A content-based review and integrative model». *Computers & Education*, 2019, **138**, pp. 57–82.

[30] Morente-Molinera, J.A.; Pérez, I.J.; Ureña, R. y Herrera-Viedma, E.: «On multi-granular fuzzy linguistic modelling in group decision making problems: a systematic review and future trends». *Knowledge Based Systems*, 2015, **74**, pp. 49–60.

[31] Natarajan, S.; Vairavasundaram, S.; Natarajan, S. y Gandomi, A.H.: «Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data». *Expert Systems With Applications*, 2020, **149**, p. 113248. doi: https://doi.org/10.1016/j.eswa.2020.113248.

[32] Panda, D.K. y Ray, S.: «Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review». *Journal of Intelligent Information Systems*, 2022, **59**, pp. 341–366.

[33] Pliakos, K.; Joo, S.H.; Park, J.Y.; Cornillie, F.; Vens, C. y Noortgat, W.V.: «Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems». *Computers & Education*, 2019, **137**, pp. 91–103. doi: https://doi.org/10.1016/j.compedu.2019.04.009.

[34] Porcel, C.; Ching-López, A.; Lefranc, G.; Loia, V. y Herrera-Viedma, E.: «Sharing notes: An academic social network based on a personalized fuzzy linguistic recommender system». *Engineering Applications of Artificial Intelligence*, 2018, **75**, pp. 1–10.

[35] Portugal, I.; Alencar, P. y Cowan, D.: «The use of machine learning algorithms in recommender systems: A systematic review». *Expert Systems With Applications*, 2018, **97**, pp. 205–227.

[36] Reza Zafarani, Huan Liu, Mohammad Ali Abbasi: *Social Media Mining*. Cambridge University Press, 2014.

[37] Segal, D.; Gal, K.; Shani, G. y Shapira, B.: «Sharing notes: An academic social network based on a personalized fuzzy linguistic recommender system». *A difficulty ranking approach to personalization in E-learning*, 2019, **130**, pp. 261–272.

[38] Son, L.H.: «Dealing with the new user cold-start problem in recommender systems: A comparative review». *Information Systems*, 2016, **58**, p. 87–104.

[39] Tan, H.C.: «Using a structured collaborative learning approach in a case-based management accounting course». *Journal of Accounting Education*, 2019, **49**, p. 100638.

[40] Tejeda-Lorente, A.; Porcel, C.; Peis, E.; Sanz, R. y Herrera-Viedma, E.: «A quality based recommender system to disseminate information in a University Digital Library». *Information Science*, 2014, **261**, pp. 52–69.

[41] Victor, P.; Cornelis, C.; DeCock, M. y Pinheiro da Silva, P.: «Gradual trust and distrust in recommender systems». *Fuzzy Sets and Systems*, 2009, **160(10)**, pp. 1367–1382.

[42] Wei, J.; He, J.; Chen, k.; Zhou, Y y Tang, Z.: «Collaborative filtering and deep learning based recommendation system for cold start items». *Expert Systems With Applications*, 2017, **69**, pp. 29–39.

[43] Wu, H.; Yue, K.; Pei, Y.; Li, B.; Zhao, Y. y Dong, F.: «Collaborative Topic Regression with social trust ensemble for recommendation in social media systems». *Knowledge-Based Systems*, 2016, **97**, pp. 111–122.

[44] Zhang, Y.; Shi, Z.; Zuo, W.; Yue, L. y Li, X.: «oint Personalized Markov Chains with social network embedding for cold-start recommendation». *Neurocomputing*, 2019, **Available online December 2019**. doi: https://doi.org/10.1016/j.neucom.2019.12.046.

[45] Zhu, Yu; Lin, Jinhao; He, Shibi; Wang, Beidou; Guan, Ziyu; Liu, Haifeng y Cai, Deng: «Addressing the Item Cold-start Problem by Attribute-driven Active Learning», 2018.

Chapter 2

Objectives

The overall objective of the whole Doctoral Thesis is to create recommender systems that are resilient to the cold start problem and that are able to create accurate recommendations in all kind of situations with low to none human input. In order to do that, the approaches taken are based on AI-based tools and models that do not require much human interaction. For mitigating the cold start problem, this Doctoral Thesis introduces novel techniques for addressing the cold start problem in recommender systems, leveraging advancements in machine learning, data mining, and artificial intelligence. The proposed solutions will be evaluated through extensive experimentation, comparing their performance against state-of-the-art approaches and demonstrating their effectiveness in mitigating the cold start problem.

The present Doctoral Thesis is composed of a total of 4 studies. They are classified into three different sections: Section I focuses on the cold-start problem for movie recommendation domain; Section II focuses on trust-based recommendation of resources for academic environments; Section III is focused on predicting stock price by creating polarity dictionary from financial news. In the following sections, the general and specific objectives from the different studies will be explained. General objectives are composed by a list of specific objectives.

2.1 Section I

This section includes the Study 1 and Study 4 from this Doctoral Thesis.

General objective 1: to provide meaningful and accurate recommendations even under the cold-start problem situation, that is, when no there is a lack of previous data about the user-item rating.

- Specific objective 1.1: Extract user-related data from different sources that can be leveraged for the creation of the recommendations
- Specific objective 1.2: Create models that leverage contextual user data extracted from social media stream in order to extrapolate it to movie domain.
- Specific objective 1.3: Curate extracted data and elaborate a dataset that can be leveraged

for future research works within the area of recommender systems in general and cold-start problem in particular

2.2 Section II

This section includes the Study 2 from this Doctoral Thesis.

General objective 2: create predictions on stock price change based on data obtained from financial news.

- Specific objective 2.1: create a polarity dictionary fetching data from news feed
- Specific objective 2.2: to extract sentiment polarity automatically producing fuzzy polarity dictionary extracted from financial news feed in order to predict stock price changes.

2.3 Section III

This section includes the Study 3 from this Doctoral Thesis.

General objective 2: to create a trust based recommender system leveraging fuzzy linguistic modelling in order to provide recommended activities to academics in the field of oral surgery and implantology.

- Specific objective 3.1: create a system that calculates the trust between different users
- Specific objective 3.2: provide students the most relevant resources for their further development using fuzzy linguistic models that builds a trust based recommender system without having to compare users' previous rating

Chapter 3

Methodological overview

For the development of the present Doctoral Thesis and its corresponding studies, we have used data obtained from different sources as public APIs, web scrapping or surveys conducted by sci2s [1], our research group in Spain. The systems created leverage different machine learning techniques to support users in the decision making process. In the following sections, the specific methodology for every study will be described.

3.1 Study 1

With the aim of palliating the cold-start problem, in this study a new technique was developed. This technique was based on the extraction of data from the social stream of the user and in the leverage of classifier trees in order to predict the target value (recommendation flag indicating whether a movie is suitable to be recommended to a user or not).

3.1.1 Data gathering

In this technique data is obtained from social media stream for determine the user profile and from movie rating platform for getting the ratings for the movies.

3.1.2 Data transformation

After the data is being gathered the N most-rated movies are selected. For every of these movies a model will be created.

3.1.3 Model creation

These models will leverage random forest and will take as input features the user profiles created in previous step and the target would be a flag indicating if the movie should be recommended for the users. The training test data distribution would be around 80% - 20%. The separation would be based on the users, that is, users from training dataset will not show up in test dataset and vice versa. In this way, we are simulating a cold-start scenario. After the models have been

[1] The research group Soft Computing and Intelligent Information Systems (`https://sci2s.ugr.es/`)

executed, the results will be grouped by user, selected the ones with recommended flag and sort them by probability which represents the certainty of the obtained recommendations.

3.1.4 Extracting top rated movies per user

As final step, the top M movies would be provided to the user as recommended items. The process is summarized in the Figure 3.1

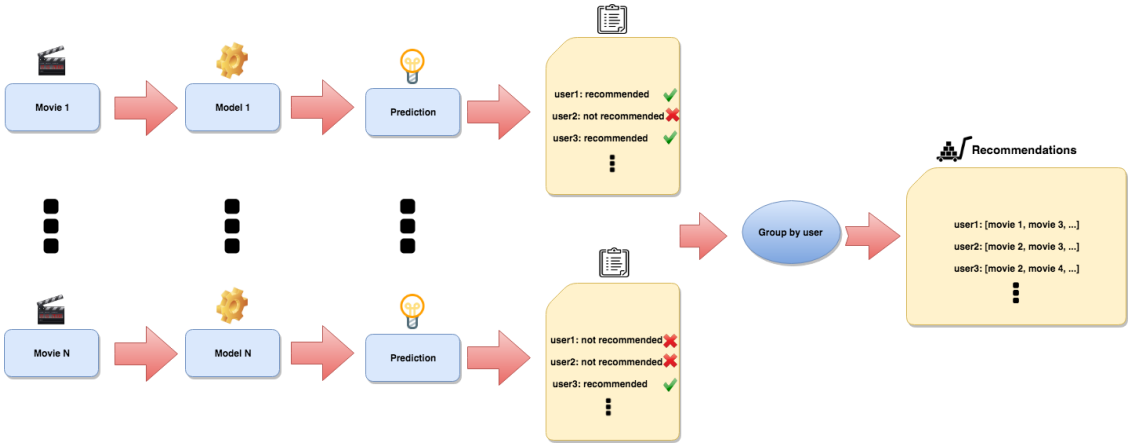


Figure 3.1: Recommendation prediction diagram

3.2 Study 2

The system for automatic sentiment polarity extraction from financial news. The system uses a weighted news collection, where weights are assigned to news entries based on the in-percentage daily price changes of a specific stock. The system gathers data from finance portals and market data portals, tags days with price changes over a particular threshold, selects news that match these days, and creates a collection of news per stock symbol labeled with a threshold value and a sign. The system then defines Weighted Bins and Signed Weighted Bins for the news, and the weighted news collection is the set of all news referred to the selected stocks and their corresponding Signed Weighted Bin. The system is designed to find stocks with substantial media presence and large trading volumes, using a well-known index, such as Euro Stoxx 50, as an input. The overall process is shown in the Figure 3.2

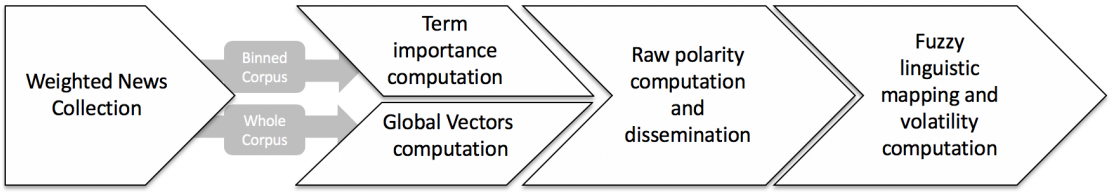


Figure 3.2: Overview of the Fuzzy Sentiment Polarity Dictionary creation process

3.2.1 Weighted news collection

We first choose the stocks we will user for the system. We picked the stocks under Euro Stoxx 50. After that, for every stock market data as well as data from news websites will be collected and matched creating a weighted news collection.

3.2.2 Term importance computation in binned Corpus

We obtain in this step the positive and negative guiding polarities for the most representative terms. We use the TF-IDF algorithm to estimate how important that term is with respect to the corpus.

3.2.3 GloVal Vectors Computation in broader Corpus

In the broader corpus, we will apply the GloVe algorithm to compute the vectorial representation of our terms. By the end of this step, we will obtain the positive and negative guiding polarities for the terms that are most representative, which we will use in combination with the GloVal vectors.

3.2.4 Guiding Polarity dissemination

This step consists of passing the guiding polarity of all terms identified in the TF-IDF algorithm onto their own Embeddings Neighbourhood. The raw polarity dictionary is the union of the disseminated Polarities to the signed Guiding Polarities.

3.2.5 Fuzzy Linguistic Mapping and Volatility computation

We lastly map the polarity values in the raw polarity dictionary fetched in the step before to linguistic labels. In order to estimate how much evidence is behind the polarity definition of a specific term, we define a measure for the stability, based on both of the term in the corpus and the number of occurrences . Therefore, the user of the polarity dictionary can have the choice of disregard volatile polarities. The Polarity Domain Value in combination with the Polarity Supporting indicator forms our context-aware fuzzy sentiment polarity dictionary.

3.3 Study 3

The different concepts assessed in the system are the following: degree of trust of a student relative to another, the predicted degree of relevance of a resource for a student, the degree of satisfaction with a recommended resource expressed by a student and the membership degree of a resource scope or student needs with respect to each of the defined reinforcing subgroup.

The algorithm is composed of three main steps that are described below.

3.3.1 Evaluate trust relations between users

The first step of the algorithm is to identify a set of trusted users for every user. The estimation consists on flagging users as trusted users if the trust degree is above the mid linguistic label. This means the trust score between every pair of users will be estimated.

3.3.2 Fetch data from users

The second step is to fetch assessments provided by trusted users for items.

3.3.3 Assign relevance to items for users

The third and last step consists of providing a relevance degree to every item using linguistic weighted average operator and providing those with higher relevance to the users and therefore create recommendation of academics activities based on the users with higher trust score.

3.4 Study 4

With the aim of establishing a foundation and a reference for future studies on the palliation of cold-start problem in recommender systems, we have published a dataset which is optimised for this purpose.

3.4.1 Data extraction

In order to do that, we have extracted data from Filmaffinity via an ethic web-scraper and from Twitter API. From Twitter, we have obtained user-related data and from the rating movie portal Filmaffinity we have obtained movie-related data and also movie ratings from user. The process of connect both sources was first to harvest all user profiles from Filmaffinity and choose only those, where in their profile page a link to their twitter page was provided. After that all ratings for this users was extracted. From this subset of users, their twitter page link was used in order to fetch all their tweets and their twitter profile details. After the collect of the raw data, the user-related data (tweets and twitter profile details) was used in order to create a user profile with different behavioural features. These data is persisted and published in Github for its further reference for future research: https://github.com/lynchblue/movie-rating-dataset

3.4.2 Model creation

Taking this dataset as input, a mixed prediction model was created in order to create a recommender system that recommends movies based on the recommendations from users with similar behaviours (similar user profiles). This model is using a mixed approach between collaborative filtering and content-based approach. The process is shown in the Figure 3.3.

The code for the algorithm is published in the following Github repository: https://github.com/lynchblue/csp-dataset-in-action

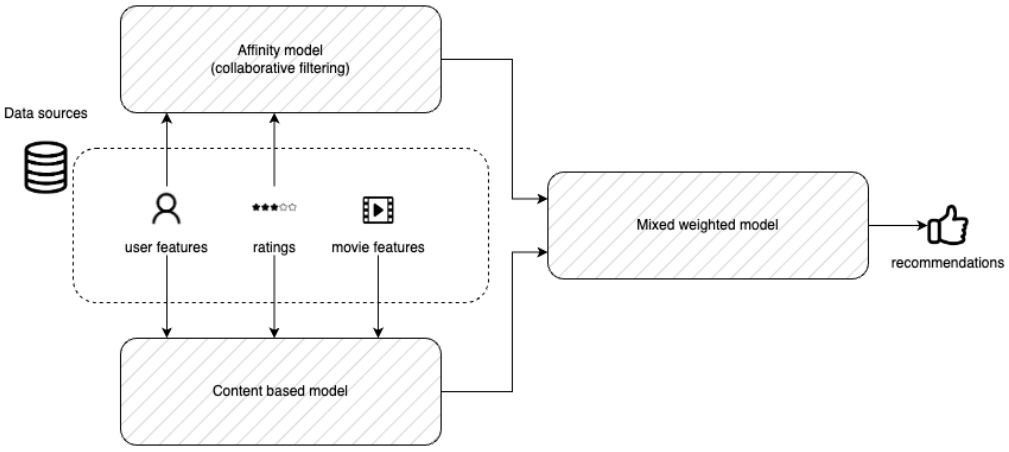


Figure 3.3: Overview data flow and model creation for CSP-Dataset

Chapter 4

Results

The present Doctoral Thesis obtains positive results by using explicit data from different sources and using AI-based algorithms. In the following sections, the result from every study is described in detail.

4.1 Study 1

In this work we focus on creating recommendations for users under cold-start problem influence. The data used is extracted from Twitter and filmaffinity and data domain is movie recommendation. The algorithm used is based on classification trees. We have trained the models with our training data and made two parallel calculations: one with a single classification tree and another one with a random forest. We have used several metrics (average rating, accuracy error, RMSE error, f1 error) to calculate the error values.

Average rating is the average of the ratings from the user for the items recommended by the system. Accuracy error, also known as classification error, is a metric used to measure the performance of a classification model. It represents the proportion of incorrect predictions made by the model on a given dataset. It is calculated by dividing the total number of incorrect predictions by the total number of predictions made. RMSE (Root Mean Square Error) is a widely used metric to evaluate the performance of regression models. It measures the average deviation between the predicted values and the actual values in the dataset. It calculates the square root of the average of the squared differences between the predicted and actual values. F1 error, also known as F1 score, is a metric commonly used to evaluate the performance of binary classification models. It combines precision and recall into a single value and provides a balanced measure of the model's accuracy.

For every execution, we calculate these four metrics for the model built with Classification Trees (CT) as well as the model built with Random Forests (RF). In this way we can make a comparison between both models. All these metrics are computed for each different models (one per movie) and afterwards, the average of all of them will be calculated. We have observed that generally the accuracy error is better for the random forest model as the one from classification tree. Nevertheless, in the average rating for the predictions, the classification tree takes slightly

Table 4.1: Accuracy error after 20 executions

	Mean	Standard Deviation	Variance
Decision Trees	0.338	0.01325	0.0001757
Random Forest	0.298	0.01093	0.0001195

the lead. In Figure 4.1 is shown the resulting classifier tree for the movie with movie id 971380.

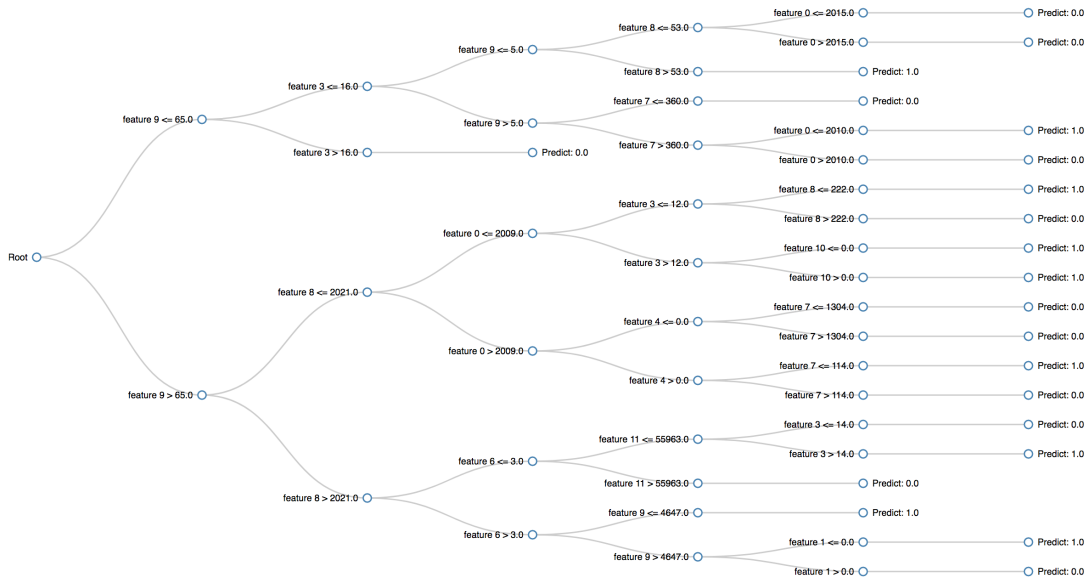


Figure 4.1: Tree classifier for features for movieId 971380.

Thus we can say that random forest performs better than a single classification tree. As we can observe in Table 4.1 from the 20 executions we obtained an average accuracy error of 0.298 for the random forests over the 0.338 for the single decision tree. Based on the standard deviation and variance we can also confirm that the obtained results are highly stable, especially in the case of the random forest.

4.2 Study 2

In this other work, we focus on the prediction of stock prices, using data extracted from news portal. The news associated to the different price variations bins have undergone the pre-processing routing (tokenizing, stop words removal, lemmatizing and PoS Tagging). For evaluation we only selected nouns, verbs and adjectives, since those are typically the highest contributors to the sentiment of a sentence. This results in a fully normalized corpus with one document per financial news gathered. Then we created the binned corpus and applied TF-IDF to obtain the signed guiding polarities.

After the aggregation of both signed guiding polarities and disseminated polarities, we apply the fuzzy linguistic mapping assigning a linguistic label to each polarity value. For our

implementation, we opted for a level 5 label set with the labels Almost non-existent, Slight, Medium, Strong, Very Strong for both positive and negative polarities, to obtain the Polarity Domain Values

In order to complete our fuzzy polarity dictionary, the Polarity Supporting Indicator for each term is computed. For this, we use the following level 5 label set: Very weak, Weak, Medium, Strong, Very Strong

In the table 4.2 we show the terms with the highest fuzzy polarity.

	term	maxpolarity	maxpolarityfuzzy	support	supportfuzzy
1	peapod	0.07	Very Strong positive	0.00	Very weak
2	directly	0.07	Very Strong positive	0.00	Very weak
3	meal	0.06	Very Strong positive	0.01	Very weak
4	northeast	0.06	Very Strong positive	0.01	Very weak
5	bol.com	0.06	Very Strong positive	0.01	Very weak
6	bol	0.06	Very Strong positive	0.01	Very weak
7	nationality	0.06	Very Strong positive	0.01	Very weak
8	fresh	0.05	Very Strong positive	0.01	Medium
9	globe	0.05	Very Strong positive	0.02	Medium
10	sportswear	0.05	Very Strong positive	0.00	Very weak
11	jewelry	0.05	Very Strong positive	0.02	Medium
12	house	0.05	Very Strong positive	0.03	Strong
13	mall	0.05	Very Strong positive	0.01	Weak
14	creativity	0.05	Very Strong positive	0.01	Weak
15	owned	0.04	Very Strong positive	0.01	Medium
16	shelf	0.04	Very Strong positive	0.01	Weak
17	relationship	0.04	Very Strong positive	0.17	Very Strong
18	optical	0.04	Very Strong positive	0.02	Medium
19	router	0.04	Very Strong positive	0.01	Weak
20	compensation	0.04	Very Strong positive	0.13	Very Strong

Table 4.2: Top positive terms in the fuzzy polarity dictionary

As we can observe, the Polarity Supporting Indicator helps understanding the reliability of the inferred polarities. In table 4.3, we provide the distribution of Polarity Supporting Indicator labels by Polarity Domain Value label. As we can see, they are quite balanced.

	Medium	Strong	Very Strong	Very weak	Weak
Almost non-existent negative	52	35	45	48	42
Almost non-existent positive	66	38	46	93	58
Medium negative	55	71	66	29	22
Medium positive	48	59	71	27	28
Slight negative	50	46	64	82	65
Slight positive	33	44	69	25	28
Strong negative	69	52	24	38	49
Strong positive	60	50	36	26	23
Very Strong negative	37	28	10	38	24
Very Strong positive	50	38	24	43	40

Table 4.3: Distribution of Supporting labels by Polarity labels

The whole dictionary can be downloaded from https://bit.ly/2XdkyqQ

4.3 Study 3

In this work, we are creating the recommendations by using info extracted from trust networks. In order to perform the experiments we have implemented the approach considering different parameters, for example the values for the horizon. To compare the results, we also have implemented some collaborative approaches. We have implemented both item-based and user-based approaches with different configurations.

We also analysed the coverage achieved with each approach, classifying users and items into different types. The users are classified in these types: cold start users who provided from 1 to 4 ratings, heavy raters who provided more than 10 ratings, opinionated users who provided

more than 4 ratings and whose standard deviation is greater than 1.5, black sheep users who provided more than 4 ratings and for which the average distance of their rating on corresponding item with respect to mean rating of the item is greater than 1. The items are classified in: niche items which received less than 5 ratings and controversial items which received ratings whose standard deviation is greater than 1.5.

Figure 4.2 displays respectively the MAE and coverage for cosine/Pearson similarity measure for item-based and users-based collaborative approaches for the different user groups. The MAE obtained is generally similar for all combinations, but the coverage is better with the user-based collaborative approach. In both cases, the higher the number of neighbours, the better the results, especially in coverage.

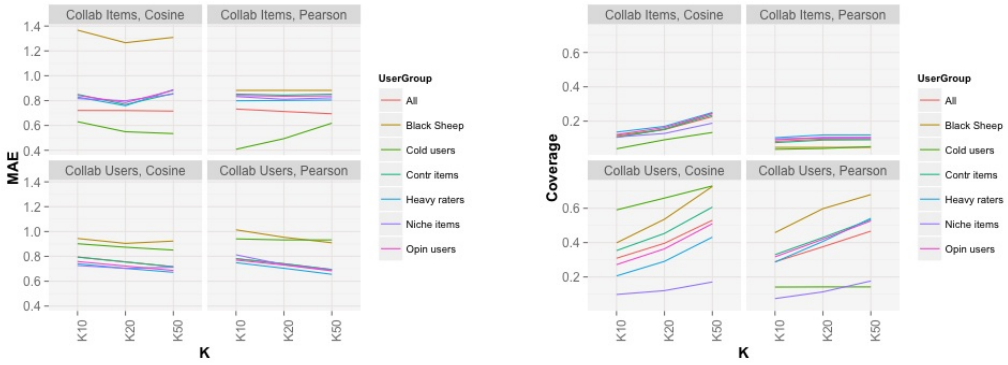


Figure 4.2: MAE and Coverage obtained with differents configurations of collaborative approach.

After that we analysed the results obtained with the approach based on trust. Figure 4.3 shows respectively the MAE and coverage obtained with the new proposal for different horizons. These figures show that a higher horizon value penalizes the time to results as it means much higher execution time.

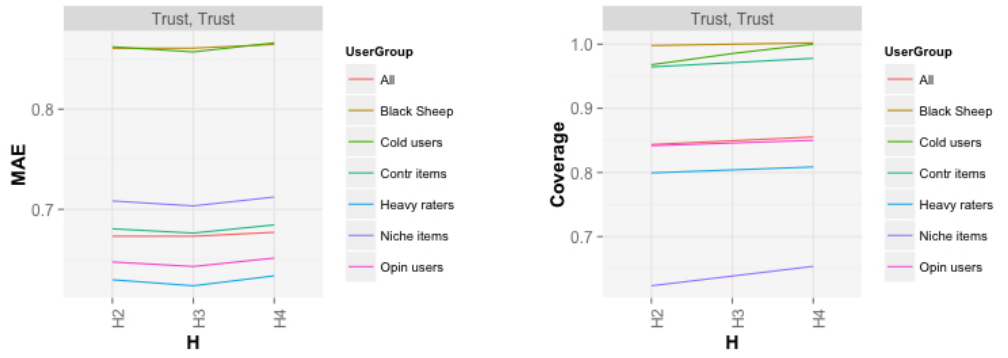


Figure 4.3: MAE and Coverage obtained with our suggested approach for differents horizons.

Figure 4.4 show the results of the comparison. We can observe that the better MAE is obtained with item-based collaborative implementation for cold users, but only for very specific situations. However, in general terms, we see that the new proposal based on trust clearly

outperforms the other approaches; specifically, we have achieved an improvement of 2.71%. Moreover the best results of our proposal manifest in terms of coverage because it outperforms the other methods.

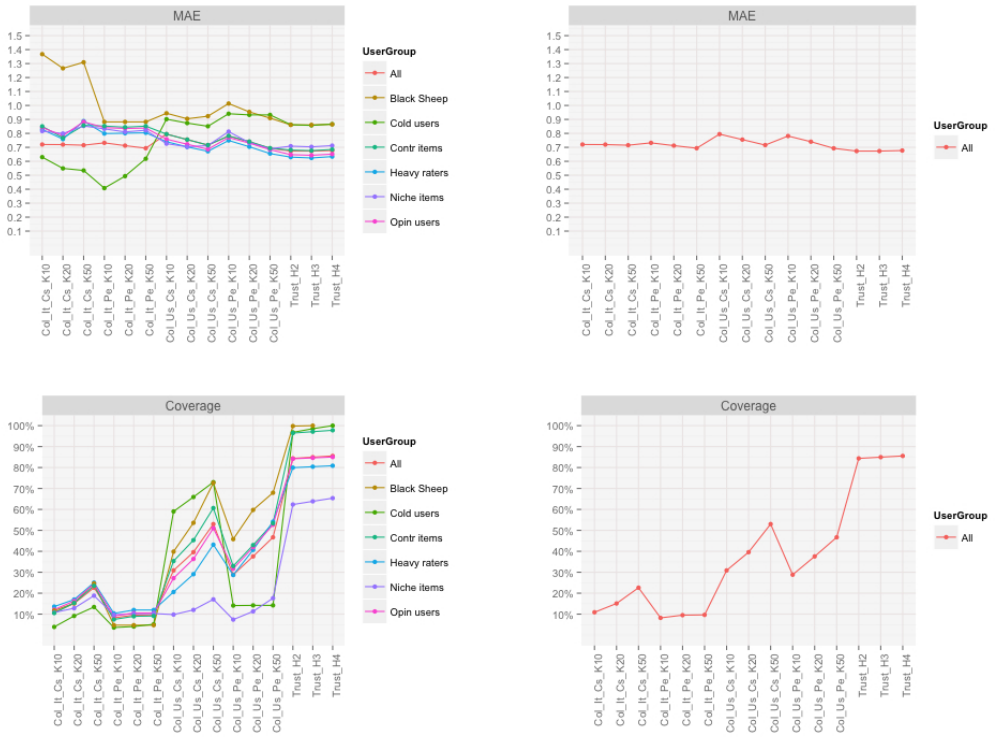


Figure 4.4: Comparison of MAE and coverage between new and the previous schemes.

4.4 Study 4

In this last work, we focused on creating recommendations for movie domain, using a dataset created by us, by fetching data from different sources about movie info, user info and ratings. We have also implemented some hybrid recommender system algorithms using these data, obtaining the results described below.

The average results from the utilized metrics applied to the whole dataset are an MRR of 0.457 and an accuracy of 60%. Lastly, recommended item's average metric is obtained by calculating the average of the real rating of the top N recommended items. The results are shown in Table 4.4.

Table 4.4: Results of the recommended item average.

Average Rating of Recommended Items	Average Rating from User	Improvement over Average Rating
8.6 (out of 10)	7.38 (out of 10)	16.53%

Figure 4.5 shows the recommended items average (red line) and the average rating of the user (blue line) together with the rating distribution to show the quality of the recommendations (green bars).

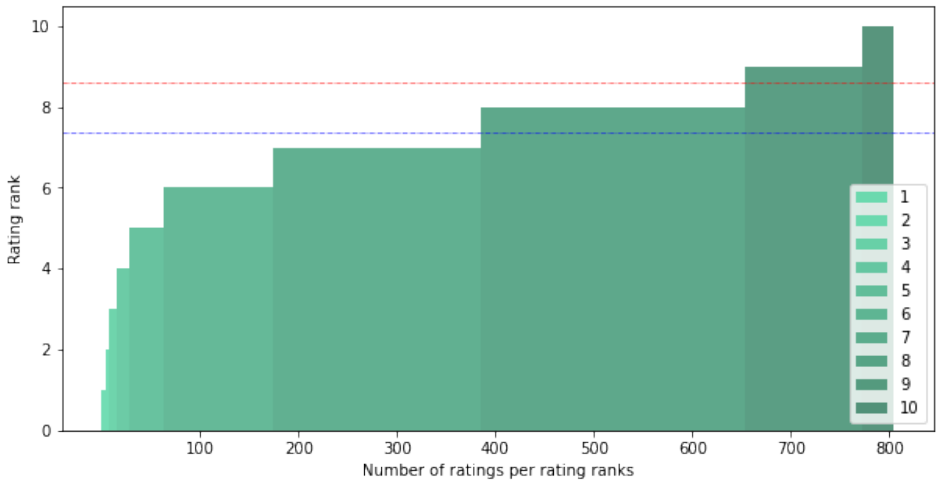


Figure 4.5: User rating distribution in comparison with the average of the recommended items (red) and the item rating average from the user (blue).

Chapter 5

New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests

In this chapter we include the following paper:

- New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests.
  - Authors: J. Herce-Zelaya, C. Porcel, A. Tejeda-Lorente, J. Bernabé-Moreno, E. Herrera-Viedma.
  - Journal: Information Sciences, 536, 156-170, 2020. ISSN: 0020-0255.
  - DOI: `https://doi.org/10.1016/j.ins.2020.05.071`
  - Impact factor source: Web Of Science - Journal Ctitation Report.
  - Impact factor: 6.795 (year 2020)
  - Category: Computer Science, Information System.
  - Quartile: Q1.
  - Ranking: 18 of 161.

Abstract

The aim of recommender systems is to provide users with items that could be of their interest. However one of the biggest drawbacks from recommender systems is the so called cold start problem, which occurs when new users or products are added to the system and therefore there is no previous information about them. There are many proposals in the literature that aim to deal with this issue. In some cases the user is required to provide some explicit information

about them, which demands some effort on their part. Because of that and due to the great boom of social networks, we will focus on extracting implicit information from user's social stream. In this paper we will present an approach on which social media data will be used to create a behavioural profile to classify the users and based on this classification will create predictions making use of machine learning techniques such as classification trees and random forests. Thus the user will not have to provide actively any kind of data explicitly but their social media source, alleviating in this way the cold start problem since the system would use this data in order to create user profiles, which will be the input for the engine of the recommender systems. We have carried out numerous experiments, as well as a comparison with some other state-of-the-art new user cold-start algorithms, obtaining very satisfactory results.

Keywords: Recommender systems, cold start problem, social media, decision tree classifier, random forest.

5.1 Introduction

As years go by the amount of content that we can find in the Internet is exponentially growing. This enables users to find any type of content. Nevertheless it also adds another implicit problem: the difficulty to find relevant items for a determined user [14]. The more content we have, the more arduous is to spot the relevant items for the users out of the total amount of content. Recommender systems palliate this problem by helping us making decisions or finding what we are looking for [5]. Recommender systems automate the process of recommendation that we follow in our daily life, by asking for opinion to other users [10, 26].

In order to accomplish that, these systems utilize previous behaviours and analogies between users to predict new demands or preferences. Recommender systems have been implemented in numerous fields, mainly e-commerce [9, 12, 36], but also in many others, such as university digital libraries [37, 40, 41], or in educational field [17]. In fact, these systems have a crucial role in highly rated Web sites, such as Amazon [1], YouTube [2], Netflix [3], Tripadvisor[4], Last.fm [5] or IMDb [6] [12].

Many recommendation proposals utilize data provided by the user in order to be able to recommend items or products to them. This data could be acquired either in a explicit (e.g. by directly rating a product) or implicit way (e.g. by establishing a connection with another user). However, when a user first joins a site, they have not yet expressed their interest about any product: they do not have any ratings history. Due to these circumstances it is hard to infer what the users are going to like when they start on a site. The problem is referred to as the cold-start problem [4, 18, 48]. Another case of cold-start problem would be when a new

[1] http://www.amazon.es/
[2] www.youtube.com/
[3] www.netflix.com/
[4] www.tripadvisor.es/
[5] www.lastfm.es/
[6] www.imdb.com/

product is released and therefore there are no data for this product since no one could yet rate it. These problems are known as new user cold-start problem and new item cold-start problem. As an example of the new item cold-start problem, we could imagine a web site which provides streaming media and video-on-demand. This site lacks of any data from the user that just joined and they do not have any hint on what such users may like. Therefore it can not be established whether certain movie is suited to be recommended for the user or not. These kind of sites are aware of these issues and in order to address them, they often offer to the users the opportunity to make a voting tour in where they are asked to rate a couple of items before the site can recommend others to them. Other sites tend to ask users to give some info about them, such as interests or hobbies [33].

One drawback from these methods is that they require some explicit action from the user and they tend to be reluctant to make these movie rate tours or give more info about them. Nowadays there are a vast amount of different movie streaming services such as Netflix, Movistar, HBO, Filmin, Sky, or Watchever, and it would be a very laborious task for the users to make these rate tours or filling up the extra info about them every time they are trying a new service. These services often offer a free trial first month of service and therefore the time the user arrived is the most crucial time since user will not stay if the do not like the service offering. Thus it is even more decisive for them to show to the user relevant content that the user would like to see as soon as possible convincing them to stay.

Despite of the reticence from users in general to fill up data forms and to make these rating tours in that kind of situation, there are other contexts where these users can actively provide more information about their likes, tastes and behaviours. We refer here indeed to the social media systems [2, 46]. Nowadays users leave in microblogs and social media systems a huge amount of information about their interest expressing their tastes and opinions [3, 42, 46]. Social media systems are in general an environment where users tend to express themselves in a broad way leaving an enormous digital footprint that could be converted in valuable information [1]. Due to the ever-growing usage of social media and microblogs, used for keeping in touch with people as well as for expressing their opinion about very different topics, a vast amount of information could be extracted and converted into useful knowledge in order to integrate them into a recommender system and generate better estimates [18, 26, 42, 49]. Therefore, this user's social content could be used and processed with the purpose of elaborating a user's profile that could help the decision making process [1, 3]. This procedure reduces significantly the users' interaction since we do not require much of actively action from them which eases the required flow previous to the recommendation process.

In this paper, our main target is to create a behavioural profile leveraging these implicit data extracted from the user's social stream [18, 27, 28] which would be determined by a collection of features depicting their preferences, tastes and character. We have corroborated that in previous proposals those approaches have worked satisfactorily [35, 39]. However our proposal provides the novelty of being able to bring those topics to the practical area since we provide a data set merging two data sources and matching them together in order to be able to bind the social stream data with the rating data. After having a behavioural mapping for the users, the different

attributes from their profiles will be used to establish a classification matching the behavioural profiles with the target attribute, which is the user's rating for a determined item. A prediction model will be created for every item from the catalogue in order to predict whether such item should be recommended to a determined user or not. Concretely, the concept is to extract information provided on the most well-known and used microblog system, i.e., Twitter [7] and use this knowledge in the recommendation scheme, an integration that is giving increasingly better results [11, 24, 42]. Then this knowledge is linked with some rating data in order to create the models.

However, the amount of information we have to deal with in this type of processes is so massive that it is necessary to resort to more advanced additional techniques that could help us to obtain useful knowledge. Therein we propose to integrate in this process some machine learning techniques, since they are the techniques that recently have been utilized with most success in recommendation systems [28, 30, 31]. To assist the processing of massive data, we will adopt Big Data tools in order to give better recommendations [27]. Specifically we propose to use Apache Spark[8] which is a fast and general engine for large-scale data processing that offers a broad diversity of machine learning algorithms. Our approach will be based on the use of decision trees and random decision forests that will assist us for the classification of the users according to their profiles, allowing us to obtain significantly better recommendations [22].

Then, the starting point of our proposal is a model to process the social stream for every single user [15, 16, 28]. From this stream we will extract some features creating the so called *Twitter Profile* which defines the behaviour of the users of the social media site. Then this Twitter profile is utilized to classify the users in relation with the actual rating of every item (which would be the target), with the assist of decision trees (either single decision tree or random forest). After this classification process we will select the items which are predicted to be recommended for the user with the highest probability. We have developed the proposed model and, to check the functionality of the system, we have applied it in the movies recommendation field [11]. We have selected the field of movies because the data is more accessible and it is a topic in which an extensive amount of people is interested and thus will be easier to find people for whom the movies are among their interests. Furthermore, movies are a very common and active topic of discussion in the social networks. The results achieved after the validation and experimentation phase indicate a favourable behaviour of the proposed model, being therefore very suitable for its application in a real environment.

The rest of the paper is structured as follows. In Section 5.2, the necessary preliminaries are exposed. Next in Section 5.3, we describe our proposal. Thereafter, in Section 5.4 we describe the experiments and evaluation of the system. Finally, some closing remarks and future works are pointed out in Section 5.5.

[7] `https://twitter.com/`
[8] `https://spark.apache.org/`

5.2 Preliminaries

5.2.1 Basis of recommender systems

Recommender systems are information filter tools that aids users in their information access processes, through prediction and recommendations of information items that could be from the user's concern [5, 10]. This process is offered as an alternative to ordinary social recommendation process. Namely, the traditional process in which we all follow by requesting opinions from experts or acquaintances when we have to make a choice for acquiring a new product without possessing any information about the product itself.

Recommender systems are not just a search method, they go beyond the search. They do not respond to punctual information needs, on the contrary they bring us deep into the discovery web. The users do not search for something specific anymore, but they expect to discover things that they did not even know that exist, or things they did not know how to execute the search to find them. The problem that these systems intent to solve is to expose items which are unknown by the user. In order to achieve that, the system will have to execute unknown ratings estimation methods, using known ratings that are persisted in a rating matrix. We could classify this estimation methods in two types [26]:

- Model based: Initially they develop a model through machine learning techniques and afterwards the model is queried in order to provide some useful recommendations about items. For instance, some techniques that could be used are clustering, neural networks or decision trees.
- Heuristic or memory based: Systems that provide recommendations making use of heuristic formulas of similarity and correlation between items and user. Example of these similarity measures are: Pearson or cosine similarity.

In this proposal we focus on the first type, namely we propose a model-based approach.

5.2.2 Cold start problem

As previously mentioned, ratings are estimated making use of a rating matrix. The complication here is that users tend to be reluctant to provide personal information. Thus in real applications these rating matrices tend to be disperse, what complicates the process of estimating recommendations since we can not access the past historical from user nor from items. This situation is what is defined as a cold start problem [18, 23].

This problem appears inevitable in two situations:

- New users cold start problem [4]: It appears when a new users is joined to the system, because they have not supplied any information yet and therefore, it can not be recommended with any item nor be compared with any similar user to them.
- New items cold start problem [35, 45]: It appears when a new item is added to the system. Since there are no ratings about this item, it will not be chosen in any matching process and thus it will not be recommended to any user.

5.2.3 Decision tree classifier

Decision tree classifier learning is a machine learning algorithm that is often utilized in decision making by generating predictions for labelled data using as input a series of observations. The goal is to create a model that predicts the value of a target variable, i.e., label, based on several input variables. They are able to discover complex interactions between variables and create accurate predictions on new data.

There are two main kind of decision trees according to the nature of their outcome, which are the following:

- Classification tree analysis is when the predicted outcome can take a discrete set of values. Commonly the response variable has two categories, e.g. yes or no. If there are more than two categories then the algorithm C4.5 will be used.

- Regression tree analysis is when the predicted outcome is continuous or numeric, e.g. the price of a car, the amount of rain to fall.

Since we want to figure out whether a movie is suitable to be recommended to a user or not, our outcome will be categorical with only two categories, recommended and not recommended. Therefore in our proposal we use a classification tree.

The term Classification And Regression Tree (CART) analysis is an term which alludes to both of the previously described techniques. First introduced by Breiman et al. [8], trees used for regression and trees used for classification have some analogies but also some differences, such as the procedure used to determine where to split.

Decision tree builds classification or regression models in the form of a tree structure. They break down a dataset into smaller and smaller subsets and simultaneously a decision tree is being developed. At the end we have as result a tree with decision nodes and leaf nodes. A decision node (e.g. outlook) has two or more branches (e.g. sunny and rainy). Leaf node (e.g., Play tennis) represents a classification or decision. The best predictor will be positioned at the top node and it is called root node. Decision trees can deal with both categorical and numerical data.

In Table 5.1 is displayed an example of observations for building a classifier tree. The observations' target is *play tennis* and the features are outlook whose categories are rainy or sunny; humidity with categories of high and low; and windy with categories of false or true.

If we calculate the decision tree for the previous values we will obtain a classified tree like the one in Figure 5.1. As described before every node represents a decision. The algorithm utilized to create the tree is known as ID3 and was proposed by J.R. Quinlan in [32].

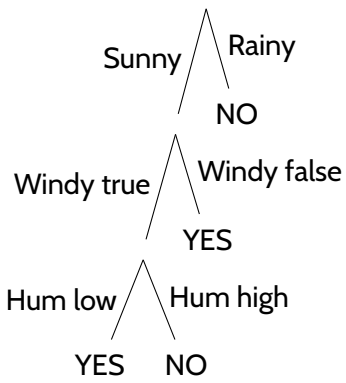
5.2.4 Random Forest

Single decision trees tend to suffer from high variance or high bias. Random forests come as an attempt to mitigate the problems that these high variance or high bias might cause. Random Forests are an ensemble learning method for classification and regression whose mechanism is based on building multiple decision trees at training time and processing their results in order

Table 5.1: Observations

Outlook	Humidity	Windy	Play tennis
Rainy	High	True	No
Sunny	Low	True	Yes
Sunny	High	False	No
Rainy	Low	False	No
Sunny	High	False	Yes
Rainy	High	False	No
Sunny	Low	True	No

Figure 5.1: Classifier tree



to get a more stable prediction out of them. For classification, it selects the class that is the mode of the classes output by the individual trees, while for regression, the mean of different regression trees will be calculated [7].

The previously described process defines the original bagging algorithm for trees [6]. Random forests have one difference with this general scheme: they make use of a slightly modified tree learning algorithm that selects, for every decision tree in the learning process, a random subset of the features. This process is called *feature bagging*. This is done because of the correlation of trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the other trees, causing them to become correlated. With that it is intend to add a random aspect to the algorithm in order to prevent correlation between the different classification trees [21].

Then, random decision forests are a combination of tree predictors: the basic principle from random forest is to combine some *weak learners* in order to obtain a *strong learners*. For this reason, they are a useful tool for making predictions taking into account that they do not overfit because of the law of large numbers. Embracing the right amount and kind of randomness to the algorithm makes them accurate classifiers and regressors.

5.2.5 Related works

We can find in the literature diverse proposals that aim to solve the cold start problem [18, 28]. A possible workaround for solving that problem is to demand users to provide some personal info and to rate a determined number of items in order to be able to establish a profile in the system [19]. If we focus on the new users cold start problem, in [23] a comparative study from different proposals is proposed, some of them will be cited here. In particular, in [4] the authors introduce a new optimized similarity measure through machine learning techniques based neural networks. In [44] is proposed an approach that incorporates association rules, probability based metrics and own users' context in order to alleviate the cold start problem. In [20] a rules based system is also utilized, in this case a probabilistic model. Nevertheless, in [25] a proposal is presented in which classification algorithm C.4.5 and Naive Bayes are combined with diverse similarities and prediction techniques in order to solve the problem. Another interesting proposal is introduced in [13], where they present an approach that utilizes trust and distrust network to find trustworthy users and utilize the suggestion of these users to generate the recommendations. In this sense of fetching social network data to palliate this problem, more recently, in [18] is presented a revision about how it is been working precisely with information extracted from social network, studying some published articles between 2011 and 2017.

On the other hand, in order to deal with the new items cold start problem there are not so many articles to be found, although we could mention two interesting ones. In [45] the authors present a system in which they obtain items features using deep learning architecture SDAE and those features are exploited by integrating them in collaborative schema timeSVD++. Furthermore, in [35] items features and user's generated tags are used applying a further matrix factorization steps is used in order to generate the knowledge.

There are many other proposals that allow to alleviate the problem with their approaches,

although they are not specific for that. In general, we can see that one of the most used and effective techniques is to utilize additional info about users, such as age, gender, education, zip code or any other information that could help to classify the users, which is what is known as demographic information [29].

Other works focus their approach on how the recommender systems can be empowered and developed through machine learning techniques and how challenging is to find the most suitable technique for every use case, reviewing the trends of machine learning and artificial intelligence techniques and identify open questions in the use or research of machine learning algorithms [31]. Other machine learning techniques like deep learning are presented in a broad study [43]. There is also in the literature some works that study the use of based social recommender systems making use of the data from social networks to improve the accuracy of the algorithms [47]. In other works like [34] they also use external behavioural data to fill the gaps between the human and software decision making process. We can find as well some works that aims to solve the cold-start problem from different approaches, as in [38] where we can find a comprehensive review of different studies and a extensive comparative between them.

If we focus on the idea of incorporating information extracted from social media and social networks into the recommendation schemes, in [48] an approach is presented where social relations and temporal informations are integrated to palliate the cold-start problem making use of Markov Chains. In [16] the authors propose an ontology-based advertisement recommendation system that leverages the data produced by users in social networking sites; a shared ontology model is used to represent both users' profiles and the content of advertisements. In [15] the social context is also taken into account to propose a recommendation approach that presents a personalized list of tourist attractions for each tourist, based on the similarity of users' desires and interests, trust, reputation, relationships, and social communities.

5.3 Description of the proposal

We propose a recommender system in which the solely input is the social stream from the user. In our specific case, we will use data extracted from Twitter[9] to generate the user profiles. Then we will classify these profiles and with those data we will establish a prediction model for every item from a determined catalogue in order to predict whether such item is suitable to be recommended to that user or not. In order to do this, we propose to use decision trees that will help us to classify the users according to their profiles, allowing us to obtain far better recommendations [22]. We propose to use single decision trees as also random forest, where several decision trees are ensemble in order to provide a better result [7]. In this section we explain the proposed model that we have applied to a specific environment, which is movie recommendation, and therefore we will explain it considering that specific area.

The input for our system will be the social stream from the users. To be more specific, we will use data from the user's Twitter account in order to create a Twitter profile that represents somehow the personality of the users. Afterwards we will create the prediction models with

[9] `https://www.twitter.com`

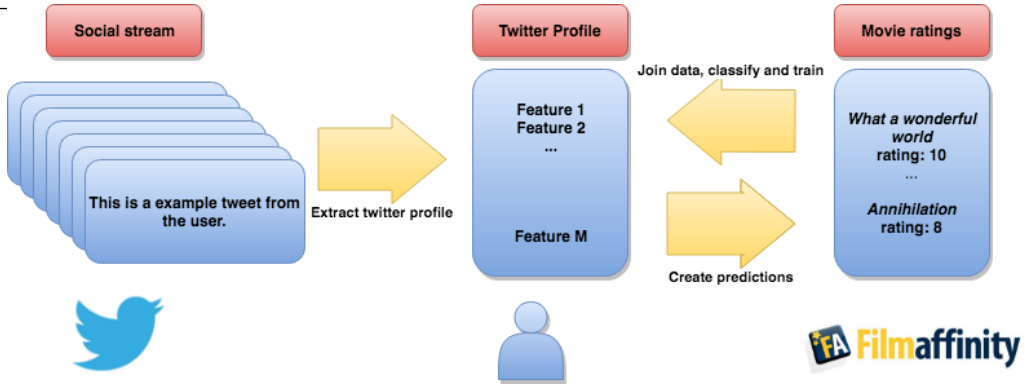


Figure 5.2: Data source diagram.

which we want to classify users according to their Twitter profile and whose target will be whether or not a particular movie is suitable or not to be recommended to certain user. In Figure 5.2 we can see a diagram of the system developed and implemented, consisting of the following phases:

- Data gathering: we will first have to collect the data with which we will generate our recommendations.
- Classification: we will classify the users according to their tastes, character and behaviour (Twitter profile) making use of classification trees and random forest, whose target will be a flag indicating whether a movie is recommended or not. That will be the training process, in which we will use part of the data set to train our model. In this process the connections between the different features from Twitter profile and the target label will be created.
- Prediction: Once the models are trained, we will use the test data in order to create the predictions. Since we have also the real data from the users (movie ratings) we will be able to compare the predictions and real data evaluating so the accuracy from our predictions. At the end from this stage we will have a list of movies (the ones that we are used in our movie catalogue) and a corresponding flag for every movie indicating whether our system marked the movie as recommendable for the user or not.
- Selection: We will select all the movies that are marked as recommendable for the user and then select the ones whose probability higher is. This means, we select the movies that are marked as recommended with more certainty from our prediction model.

Then, the first challenge is to find suitable data that allows us to develop and implement our proposal. Since we need to find users who have movie rating data as well as social stream data, we decided to use Filmaffinity[10] in order to have relevant data. We will keep the focus in our proposal on movie recommendations since we find a very popular topic in social networks and

[10] `https://www.filmaffinity.com/es/main.html`

also because we found a very attractive and useful tuple of source of data (Twitter for defining the users and Filmaffinity for getting the ratings that the users are giving to the movies).

Filmaffinity is an online web site that serves as movie database and where users can rate movies. The users have the possibility of provide in their Twitter url on their profile, which enables a link between both data sources. That will however diminish considerably the amount of users that we can utilize for this experiments since only roughly 10% of the users provide on their profiles their Twitter url.

We will try to predict which products are more suitable to be recommended to a user. In order to accomplish that goal, we will have a product catalogue (movie catalogue in our case), a list of users for which we have both Twitter and Filmaffinity profile. From these users we have a list of all tweets available in Twitter and a list of movie rating.

In order to classify the users depending on their Twitter profile and in relation with every movie from our movie catalogue, we will use Apache Spark. The decision of choosing is based in its ability to process large amounts of data (as is the case that concerns us), its execution speed and its machine learning library which offers a wide variety of algorithms and utilities.

To be more specific we will use the Decision Tree Classifier[11] algorithm from Spark to classify the users according to their Twitter stream and we will create prediction models for every movie in our movie catalogue. Moreover, we will also use the Random Forest Classifier[12] in order to implement the classifier with random forest. Once we have collected all the predictions, we will choose for every user those products which are predicted to be recommended and from this set we will choose those for which their probability is higher.

5.3.1 User data gathering

We have used an ethical web scraping in order to get the data of the users (ratings and Twitter url) from Filmaffinity. We have implemented our web scrap utilizing python as programming language. To be precise, we have used the libraries requests[13] in order to fetch the content of every page and lxml[14] in order to parse the data from every page. For persisting our data we have used a MongoDB[15] database. The reasons for choosing it are the ease to use, the fact that it has no schema and its very powerful aggregate pipeline which can effortlessly process our data records and return computed results.

We are solely interested in the subset of users that have given a Twitter url in their Filmaffinity profiles, we will discard the rest of them (since we can not match the two sources of data for them). Once we have this subset of users, we will use another ethical web scraping for getting the ratings from every of these users.

[11] `https://spark.apache.org/docs/2.0.0/api/java/org/apache/spark/ml/classification/DecisionTreeClassifier.html`

[12] `https://spark.apache.org/docs/latest/api/java/index.html?org/apache/spark/ml/classification/RandomForestClassifier.html`

[13] `http://docs.python-requests.org/en/master/`

[14] `http://docs.python-guide.org/en/latest/scenarios/scrape/`

[15] `https://www.mongodb.com/what-is-mongodb`

5.3.2 Social media data gathering

Making use of the Twitter API [16] we have collected all the tweets (actually only 3200 last tweets from every user are possible to fetch because of Twitter API's restrictions) from the users for which we have data in Filmaffinity. This data are persisted in our MongoDB database as well so that is more convenient for us to aggregate it afterwards.

5.3.3 Elaborating a Twitter profile

We have processed the tweets previously collected extracting features in order to build a Twitter profile for every user. This profile is defined by a series from *features* that we use later on in order to build our model.

Some of the features are directly accessible from Twitter profile, e.g. number of followers, or account creation year. However there are other features for which we need to process all the tweets in order to be able to extract them, e.g. preferred day for writing tweets, early tweeter.

The features extracted for this first iteration are the following:

- Account creation year (values: numeric)
- Early bird: Usually tweets in the morning (values: false or true)
- Night owl: Usually tweets in the night (values: false or true)
- Preferred hour: The hour on which the user more often tweets (Values from 0-23)
- Weekend tweeter: Usually tweets on weekends (values: false or true)
- Week tweeter: Usually tweets on week days (values: false or true)
- Preferred weekday: The day of the week on which the user more often tweets (Values from 0-6)
- Friends count: The number of people the user follows (values: numeric)
- Followers count: The number of followers from user (values: numeric)
- Favourites count: The number of favorited tweets by user (values: numeric)
- Geolocation enabled: If user has geolocation enabled (values: false or true)
- Number of tweets: The number of tweets from user (values: numeric)

On the other side we will have for every user a list of movie ratings. The data are those described below:

- movieId: The id of the movie in Filmaffinity.
- rating: The rating that a given user has given to the movie with id movieId.

[16] `https://developer.twitter.com/en/docs/api-reference-index`

5.3.4 Joining Twitter profile and movie data

As we have mentioned, we utilize the features extracted from Twitter profile in order create predictions for the rating of movies. We take the user's ratings for the movies that we have in our catalogue. The ratings are interval-based ratings drawn in a 10-point scale, that is, the ratings are integer numbers from 1 to 10. Since for us is not so important the exact rating from a user but the fact that the user liked the movie, we will convert this interval-based to a binary rating by labelling this data in two possible labels: 0 for not suitable for the user and 1 suitable for the user. We label ratings between 0 and 6 as not suitable and ratings between 7 and 10 as suitable.

For that we use a determined set of N movies, simulating a streaming service catalogue. In order to have more rating data we chose the N movies most rated from our user group (the users for which we have Twitter profile).

Then we utilize the decision tree learning algorithm [17] in order to classify the users according to their Twitter profile and as label we use a recommended flag whose values could be 0 or 1. Values with 0 means that a movie is not suitable to be recommended to a determined user and on the contrary values with 1 means that movie is suitable to be recommended to the user.

Since we want to know the predictions for several movies and not just one, we will have to use a model per movie. Thus we create N models being N the number of movies in the catalogue. The difference between those models will be only the labels (the recommended flag for a determined movie). Everything else (Twitter profile) will remain identical in every model. As described in the example from Table 5.4, we have a list of observations (one per user who voted the movie 1) with some features (user's features extracted from Twitter profile that were previously described) and at the end we have the label which is the value that we want to predict and have two possible values: 0 would mean that the movie A is not suitable to be recommended for corresponding user; and the value 1, in the other hand, would mean that we could recommend movie A to that user.

From the rating matrix perspective, we iterate over the items and create as many models as items in the catalogue. We simplify the interval-based ratings in 10-point scale by replacing them with binary ratings. Then we incorporate the user profile for every user. After that, we train the models with the user profiles and the ratings being the target. And lastly, we test our system with the respectively trained models and evaluate the expected values with the actual ones.

We have the rating matrix:

$$R_{\text{mn}} \tag{5.1}$$

where:

m = number of users

n = number of items

And on the other side we have the user profile:

[17] http://en.wikipedia.org/wiki/Decision_tree_learning

Table 5.2: For every item = j

r_{0j}
r_{1j}
...
r_{mj}

Table 5.3: For every item = j

p_{00}	p_{01}	...	p_{0t}	r_{0j}
p_{10}	p_{11}	...	p_{1t}	r_{1j}
...	...	...	...	...
p_{m0}	p_{m1}	...	p_{mt}	r_{mj}

$$P_{mt} \tag{5.2}$$

where:

m = number of users

t = number of features of user's profile

We combine the ratings matrix R_{mn} with the user profile P_{mt} where m is the number of users, n the number of items and t the number of features of user's profile.

We iterate over the different items (movies) creating a different model for each starting from the rating matrix as indicated in table 5.2. Afterwards we combine the ratings matrix R_{mn} with the user profile P_{mt} like indicated in table 5.3.

$$R_{ij} \tag{5.3}$$

Our proposed model takes our training data and then creates a decision tree and a random

Table 5.4: Model for movie A

UserId	Feature-1	Feature-2	...	Feature-M	Label
1	6	0	-	22	0
2	7	1	-	20	1
3	1	1	-	18	0
4	2	0	-	18	0
5	0	1	-	17	1

forest per model (one model per movie as previously indicated) and from these models, it will create the corresponding predictions for the values for the test data. This data pipeline is displayed in Figure 5.3.

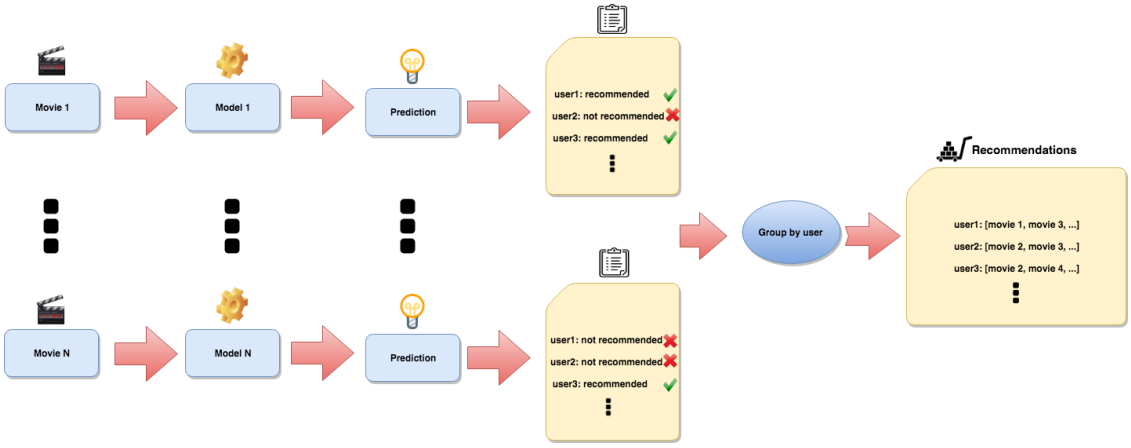


Figure 5.3: Recommendation prediction diagram

5.4 Evaluation and experiments

5.4.1 Predictions evaluation

Once that our classification trees and random forests have created the predictions, we will continue by evaluating the quality of these predictions, exposing the results from the two different variations (single classification tree and random forest) in order to compare them. The target from our prediction model is a flag that indicates whether a movie is recommended or not. We will then select the movies that are marked as recommended and from them we select those which have higher probability to be recommended.

We will quantify the accuracy from the predictions with the following indicators. We have chosen these indicators since they are the most common metrics for evaluation Classification Trees or Random Forest, especially the precision, RMSE and F-measure. The average rating from recommended items gives a quick and clear vision of how good the system was with their predictions.

- Average rating from recommended movies. From all the movies that are predicted to be recommended for the user we calculate the average of ratings provided about each movie. It is an indicator of, how good are in general the movies that are recommended to the user.

- Precision (positive predictive value). Resembles the percentage of success from our recommender over the failures. In the table is recorded the opposite value, i.e. accuracy error (ACC):

  ACC = $\frac{TP}{TP+FP}$

Table 5.5: Twitter profile sample

id	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12
305054	2010	False	False	23	False	False	0	759	123	3	False	8462
990342	2013	True	False	9	False	False	2	1518	1169	27829	True	50207
982478	2010	False	True	23	False	False	2	481	477	829	True	8116
469948	2010	False	False	21	False	False	3	64	122	88	True	3516
832140	2013	True	False	11	False	False	4	805	531	2932	True	10216
547430	2013	True	False	9	False	False	2	1777	6195	19288	False	30364
106161	2011	False	True	21	False	False	0	37	48	107	True	1321
707180	2011	True	False	12	False	False	3	337	425	1623	True	9772
525484	2010	False	True	23	False	False	2	247	248	162	False	2883
541976	2011	False	True	22	False	False	2	654	585	2589	False	3832
715072	2011	False	True	12	False	False	1	467	581	830	False	16469

- Root Mean Squared Error (RMSE). Represents the sample standard deviation of the differences between predicted values and observed values:

  RMSE = $\sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$

- F-measure. The F measure is a measure of the accuracy of a test. It is defined as the weighted harmonic mean of the precision and recall of the test (F1):

  F1 = $\frac{2 \cdot precision \cdot recall}{precision + recall}$

5.4.2 Developed experiments

With the use of web scraping techniques, we extracted from Filmaffinity a total of 8503 users from Filmaffinity web site. From all these users, we filter out those that do not provide their Twitter url in their Filmaffinity profile. That gives a total of 604 users which have a Twitter profile. After filtering out those with false Twitter url and those that doesn't have tweets in their profile we have 482 users left.

From these users we have a total of 781782 ratings to make our experiments. That makes an average of roughly 1622 ratings per user. The total number of movies rated is 55769. On the other side, we have a total of 1142720 tweets to process, an average of 2370 tweets per user.

Once we process the Twitter stream from every user, we generated a Twitter profile for every user which would look like this sample for the movie with movie id 160882 in Table 5.5.

The columns from Table 5.5 are described as follow:

- id: User id
- f1: account year of creation
- f2: early bird
- f3: night owl
- f4: preferred hour
- f5: weekend tweeter
- f6: week tweeter
- f7: preferred weekday

Table 5.6: Twitter profile sample with ratings

id	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	rate
305054	2010	False	False	23	False	False	0	759	123	3	False	8462	5
990342	2013	True	False	9	False	False	2	1518	1169	27829	True	50207	4
982478	2010	False	True	23	False	False	2	481	477	829	True	8116	7
469948	2010	False	False	21	False	False	3	64	122	88	True	3516	5
832140	2013	True	False	11	False	False	4	805	531	2932	True	10216	9
547430	2013	True	False	9	False	False	2	1777	6195	19288	False	30364	6
106161	2011	False	True	21	False	False	0	37	48	107	True	1321	8
707180	2011	True	False	12	False	False	3	337	425	1623	True	9772	4
525484	2010	False	True	23	False	False	2	247	248	162	False	2883	7
541976	2011	False	True	22	False	False	2	654	585	2589	False	3832	5
715072	2011	False	True	12	False	False	1	467	581	830	False	16469	8

- f8: friends count
- f9: followers count
- f10: favourites count
- f11: geo enabled
- f12: number of tweets

Then we interpolate the Twitter profile data with the movie rating data obtaining something like the sample from Table 5.6.

The meaning of the columns from f1 to f12 is the same as in the previous case; in this case we see a new column with the rating from the user to the movie.

After that, we group the observations by movie. The observation will consist of the features from the Twitter profile of every user and the target (or label) will be a flag indicating if the movie is recommended for this user. As previously stated, we consider that a movie is suitable to be recommended to a user if the user has rated it with a rating of 7 (out of 10) or more. And then our entries would look like following sample in Table 5.7. We can see that we do not have the rating column anymore, but we have now the recommended column that is directly calculated from rating and it would have the value 1 when the movie is suitable to be recommended to the user; and otherwise would have value of 0. The description of the columns from f1 to f12 is the same as in the previous two.

Now we will create a classifier tree and random forest per every movie. To do it we use a 70% of the users to train the different models and the 30% remaining to evaluate the predictions.

Based on our trained models and according to our features a classification model would be created. In Figure 5.4 we can see the resulting classifier tree for the movie with movieId 971380.

We will train the models with our training data and we will do two parallel calculations: one with a single classification tree and another one with a random forest. After several execution of our predictions, we have used several metrics and the obtained results are shown in Tables 5.8, 5.9 and 5.10.

In these tables are displayed the error values for 20 executions. For every execution, we calculate the following four metrics for the model built with *Classification Trees* (CT) as well as

Table 5.7: Twitter profile sample with recommended label

id	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	rec
305054	2010	False	False	23	False	False	0	759	123	3	False	8462	0
990342	2013	True	False	9	False	False	2	1518	1169	27829	True	50207	0
982478	2010	False	True	23	False	False	2	481	477	829	True	8116	1
469948	2010	False	False	21	False	False	3	64	122	88	True	3516	0
832140	2013	True	False	11	False	False	4	805	531	2932	True	10216	1
547430	2013	True	False	9	False	False	2	1777	6195	19288	False	30364	0
106161	2011	False	True	21	False	False	0	37	48	107	True	1321	1
707180	2011	True	False	12	False	False	3	337	425	1623	True	9772	0
525484	2010	False	True	23	False	False	2	247	248	162	False	2883	1
541976	2011	False	True	22	False	False	2	654	585	2589	False	3832	0
715072	2011	False	True	12	False	False	1	467	581	830	False	16469	1

Table 5.8: Validation of predictions 1

Iteration	1	2	3	4	5	6	7
avg rating (DT)	7.714	7.826	7.632	7.682	7.772	7.800	7.709
avg rating (RF)	7.279	7.295	7.534	7.475	7.695	7.525	7.244
avg accuracy error (DT)	0.356	0.348	0.335	0.327	0.336	0.324	0.327
avg accuracy error (RF)	0.318	0.306	0.301	0.296	0.295	0.283	0.298
avg RMSE error (DT)	0.588	0.583	0.573	0.564	0.570	0.561	0.564
avg RMSE error (RF)	0.555	0.544	0.542	0.535	0.531	0.520	0.538
avg f1 error (DT)	0.598	0.610	0.626	0.635	0.633	0.648	0.640
avg f1 error (RF)	0.601	0.613	0.624	0.623	0.637	0.650	0.631

Table 5.9: Validation of predictions 2

Iteration	8	9	10	11	12	13	14
avg rating (DT)	7.805	7.894	7.768	7.674	7.758	7.694	7.697
avg rating (RF)	7.610	7.497	7.378	7.426	7.366	7.586	7.600
avg accuracy error (DT)	0.328	0.352	0.345	0.350	0.326	0.341	0.338
avg accuracy error (RF)	0.290	0.292	0.305	0.305	0.296	0.314	0.301
avg RMSE error (DT)	0.562	0.586	0.579	0.583	0.563	0.575	0.574
avg RMSE error (RF)	0.528	0.530	0.542	0.543	0.536	0.549	0.539
avg f1 error (DT)	0.628	0.615	0.613	0.605	0.625	0.617	0.628
avg f1 error (RF)	0.636	0.635	0.613	0.615	0.623	0.610	0.628

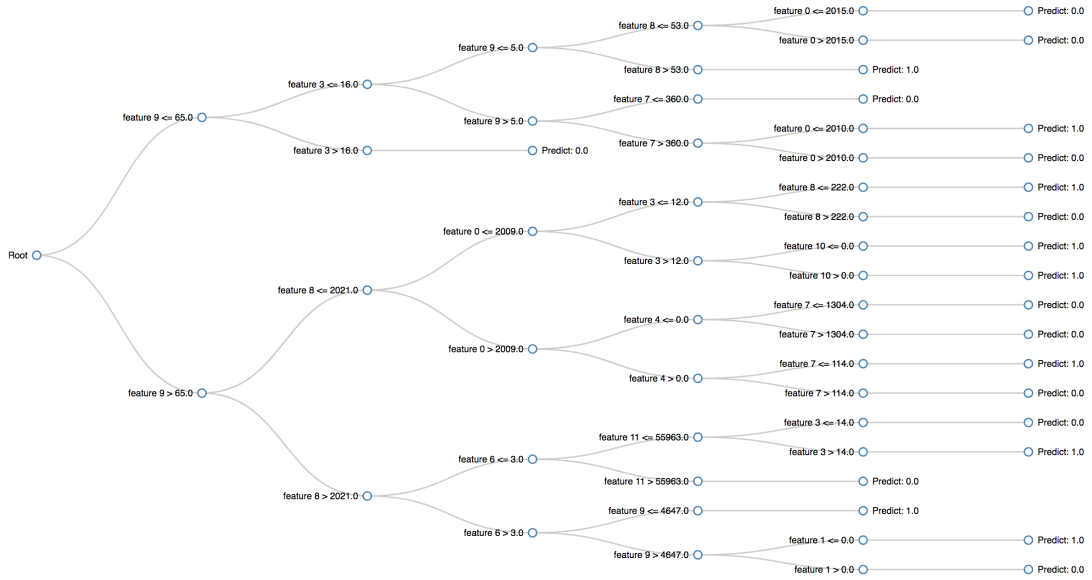


Figure 5.4: Tree classifier for features for movieId 971380.

Table 5.10: Validation of predictions 3

Iteration	15	16	17	18	19	20
avg rating (DT)	8.044	7.754	7.828	7.761	7.759	7.797
avg rating (RF)	7.651	7.408	7.396	7.667	7.507	7.475
avg accuracy error (DT)	0.313	0.354	0.337	0.318	0.361	0.334
avg accuracy error (RF)	0.274	0.315	0.292	0.287	0.304	0.292
avg RMSE error (DT)	0.550	0.587	0.573	0.556	0.595	0.569
avg RMSE error (RF)	0.512	0.552	0.532	0.525	0.544	0.532
avg f1 error (DT)	0.655	0.607	0.634	0.649	0.614	0.634
avg f1 error (RF)	0.663	0.606	0.645	0.653	0.632	0.634

the model built with *Random Forests* (RF). In this way we can make a comparison between both models.

All these metrics are computed for every one of the different models (one per movie) and afterwards, the average of all of them will be calculated. That is the value that we show in every cell of the table.

We can observe that in general the accuracy error is slightly better for the random forest model as the one from classification tree. However in the average rating for the predictions, the classification tree takes the lead. That does not necessarily mean that the predictor is better than the one from the random forest, we have to bare in mind that we consider a movie suitable to be recommended if their rating is great or equal as 7; and average rating, for both models, is in general closer to 8 than to 7.

Thus we can say that random forest performs better than a single classification tree, which

Table 5.11: Accuracy error after 20 executions

	Mean	Standard Deviation	Variance
Decision Trees	0.338	0.01325	0.0001757
Random Forest	0.298	0.01093	0.0001195

Table 5.12: Mean absolute Error comparison of models

	MIPFGWC-CS	NHSM	FARAMS	HU-FCF	MJD	BSSB-RS
MAE	0.672	0.591	0.603	0.608	0.8	0.298

was something expected. As we can observe in Table 5.11 from the 20 executions we obtained an average accuracy error of 0.298 for the random forests over the 0.338 for the single decision tree. Based on the standard deviation and variance we can also confirm that the obtained results are highly stable, especially in the case of the random forest.

Seeing these results we can assert that these results are very positive moreover taking into account the handicap that we do not use the previous rating data from the user to search for similar items. On the contrary the only data that we use from the user is not directly related with the items that we are going to create the recommendations for (movies in our case) but data from their social stream.

Furthermore, we compare the results of our Behavioural Social Stream Based Recommender System (BSSB-RS) with some other state-of-the-art works of new user cold-start problem [4, 38].

It is important to remark that almost all the algorithms are not using any additional data for the decision making. They use instead some rating data from the users (they are not a purely zero ratings algorithm like ours). In addition to that the algorithms MIPFGWC-CS and HU-FCF are also using demographical data for their systems.

We can appreciate in the table 5.12 and in figure 5.5 that our approach outperforms all new user cold-start proposed algorithm even though we are using an absolute zero ratings cold-start users.

It is important to remark that we can not compare these results with another non cold-start state of the art recommender systems since our model is only taking contextual data (Twitter stream data) as input. We only use the rating data of new users for building the model and for validation purposes. Therefore it would not be fair to compare our approach for a cold-start context with another context where a full rating history for all users is provided.

5.5 Conclusions and future work

We have proposed a recommendation approach based on a prediction model, using behavioural information extracted from social media to classify the users according to their behavioural

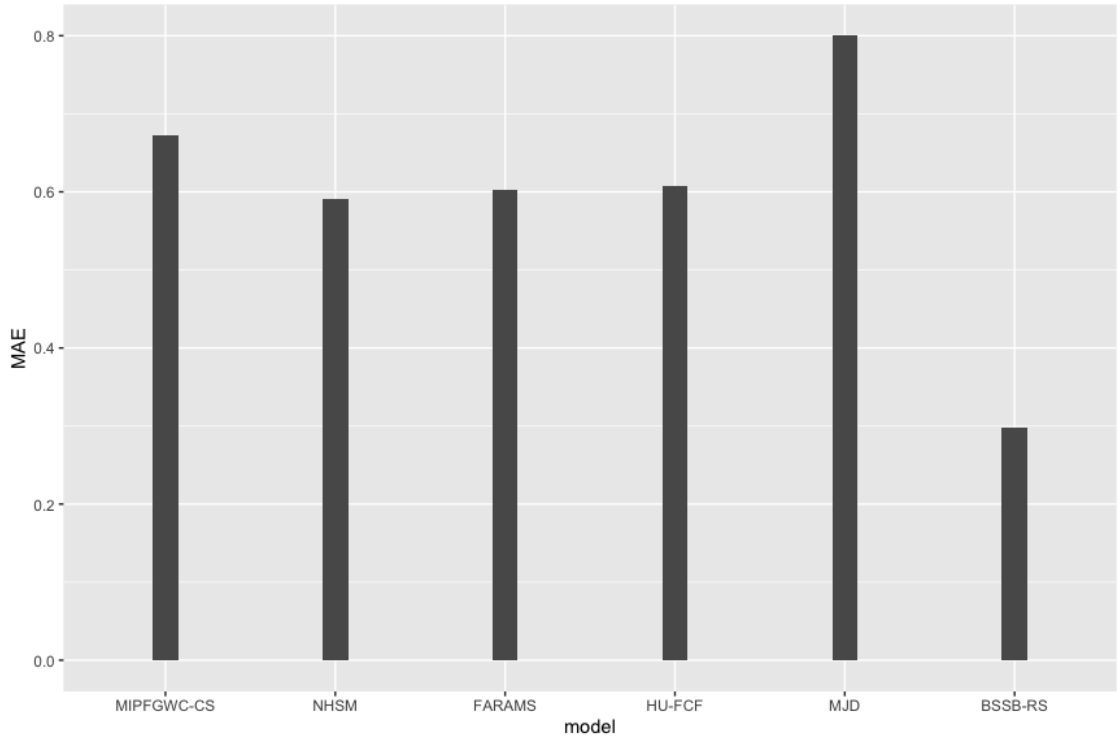


Figure 5.5: RS Cold Start Model Comparison

profiles. Then, the users will not need to explicitly provide any personal information other than the source of their social media, helping in this way to alleviate the cold start problem. One of the main novelties of our system is that we obtained a rich and comprehensive data set that comprises two different data sources, for the rating data and for the social data, that are linked enabling the fulfilment of our experiments. With the help of this implicit data obtained from the social media we can palliate the information gap that we have for new users of the system. Our algorithm is assisted with machine learning techniques, i.e., classification trees and random forest, which help us classify users assigning them a flag for every item indicating if it is suitable to be recommended or not. Although we have used classification trees and random forest, the most important idea of the approach is not the determined machine learning technique that is powering the algorithm but the integration of the behavioural data, obtained from social stream, and the rating data for creating recommendations.

The proposal has been validated in the movie recommendation environment, and the obtained results of the suggested predictions are truly satisfactory and therefore, the generated recommendations are in average very good since the results are outperforming other new user cold-start algorithms. Therefore we could assess that our algorithm (BSSB-RS) is an optimal asset in cold-start situations because it leverages the information we have in social media turning it into a very valuable data source enhancing the quality and precision in the decision making process and providing a much more accurate recommendation of items.

In this work we create our predictions establishing a direct relation between the user profile and the rating for a determined item. An eventual improvement to this process we could ap-

proach in future works would be establishing the relation between user profile and item's features (for example between Twitter profile and comedian genre or determined actor) increasing in this way the granularity of recommendations but also the complexity and execution time from algorithms. This could help us to obtain more granularity in the relations and therefore obtain better recommendations. We would have to deal with a significant more extensive amount of models that we would have to combine to obtain the final prediction. Furthermore, we have used some features for creating our model, however if we go deeper by doing some feature engineer and we extract more and more complex and relevant features from our social stream, i.e. performing sentiment analysis from the tweets the user have published, it could improve significantly the accuracy from the recommendations.

Acknowledgments

This paper has been developed with the FEDER financing of Project TIN2016-75850-R.

Bibliography

[1] Bernabé-Moreno, J.; Tejeda-Lorente, A.; Porcel, C.; Fujita, H. y Herrera-Viedma, E: «Quantifying the emotional impact of events on locations with social media». *Knowledge-Based Systems*, 2018, **146**, pp. 44–57.

[2] Bernabé-Moreno, J.; Tejeda-Lorente, A.; Porcel, C. y Herrera-Viedma, E: «A new model to quantify the impact of a topic in a location over time with Social Media». *Expert Systems with Applications*, 2015, **42**, pp. 3381–3395.

[3] ——: «Leveraging localized social media insigths for industry early warning systems». *International Journal of Information Technology & Decision Making*, 2018, **17**, pp. 357–385.

[4] Bobadilla, J.; Ortega, F.; Hernando, A. y Bernal, J.: «A collaborative filtering approach to mitigate the new user cold start problem». *Knowledge-Based Systems*, 2012, **26**, pp. 225–238.

[5] Bobadilla, J.; Ortega, F.; Hernando, A. y Gutiérrez, A.: «Recommender systems survey». *Knowledge-Based Systems*, 2013, **46**, pp. 109–132.

[6] Breiman, L.: «Bagging predictors». *Machine Learning*, 1996, **24**, pp. 123–140.

[7] ——: «Random Forests». *Machine Learning*, 2001, **45**, pp. 5–32.

[8] Breiman, L.; Friedman, J.; Stone, C.J. y Olshen, R.A.: *Classification and Regression Trees*. Taylor & Francis, 1984.

[9] Burke, R.: «Hybrid Web Recommender Systems». *P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS*, 2007, **4321**, pp. 377–408.

[10] Burke, R.; Felfernig, A. y Göker, M.H.: «Recommender systems: An overview». *AI Magazine*, 2011, **32**, pp. 13–18.

[11] Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R. y García-Sánchez, F.: «Social knowledge-based recommender system. Application to the movies domain». *Expert Systems with Applications*, 2012, **39**, pp. 10990–11000.

[12] Charlotte-Ahrens, S.: *Recommender Systems: Relevance in the Consumer Purchasing Process*. epubli, 2011.

[13] Chien, C.; Yu-Hao, W.; Meng-Chieh, C. y Yu-Chun, S.: «An effective recommendation method for cold start new users using trust and distrust networks». *Information Sciences*, 2013, **224**, pp. 19–36.

[14] Edmunds, A. y Morris, A.: «The problem of information overload in business organizations: a review of the literature». *International Journal of Information Management*, 2000, **20**, pp. 17–28.

[15] Esmaeili, L.; Mardani, S.; Golpayegani, S.A.H. y Madar, Z.Z.: «A novel tourism recommender system in the context of social commerce». *Expert Systems With Applications*, 2020, **149**, p. 113301. doi: https://doi.org/10.1016/j.eswa.2020.113301.

[16] García-Sánchez, F.; Colomo-Palacios, R. y Valencia-García, R.: «A social-semantic recommender system for advertisements». *Information Processing and Management*, 2020, **57**, p. 102153. doi: https://doi.org/10.1016/j.ipm.2019.102153.

[17] Goga, M.; Kuyoro, S. y Goga, N.: «A recommender for improving the student academic performance». *Procedia - Social and Behavioral Sciences*, 2015, **180**, pp. 1481–1488.

[18] Gonzalez Camacho, L.A. y Nice Alves-Souza, S.: «Social network data to alleviate cold-start in recommender system: A systematic review». *Information Processing and Management*, 2018, **54**, pp. 529–544.

[19] Grouplens:. «Movielens - movie recommendations». `http://movielens.umn.edu/login`

[20] Hernando, A.; J., Bobadilla.; Ortega, F. y Gutiérrez, A.: «A probabilistic model for recommending to new cold-start non-registered users». *Information Sciences*, 2017, **376**, pp. 216–232.

[21] Ho, T. K.: «Random Decision Forests». *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, 1995, **1**, p. 278–282.

[22] Ho-Cho, Y.; Kyeong-Kim, J. y Hie-Kim, S.: «A personalized recommender system based on web usage mining and decision tree induction». *Expert Systems with Applications*, 2002, **23**, pp. 329–342.

[23] Hoang-Son, L: «Dealing with the new user cold-start problem in recommender systems: A comparative review». *Information Systems*, 2016, **58**, pp. 87–104.

[24] Kim, Y. y Shim, k.: «TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation». *Information Systems*, 2014, **42**, pp. 59–77.

[25] Lika, B.; Kolomvatsos, K. y Hadjiefthymiades, S.: «Facing the cold start problem in recommender systems». *Expert Systems with Applications*, 2014, **41**, pp. 2065–2073.

[26] Martínez-Cruz, C.; Porcel, C.; Bernabé-Moreno, J. y Herrera-Viedma, E.: «A Model to Represent Users Trust in Recommender Systems using Ontologies and Fuzzy Linguistic Modeling». *Information Sciences*, 2015, **311**, pp. 102–118. doi: doi:10.1016/j.ins.2015.03.013.

[27] Meng-Yen, H.; Tien-Hsiung, W. y Kuan-Ching, L.: «A keyword-aware recommender system using implicit feedback on Hadoop». *J. Parallel Distrib. Comput.*, In press.

[28] Natarajan, S.; Vairavasundaram, S.; Natarajan, S. y Gandomi, A.H.: «Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data». *Expert Systems With Applications*, 2020, **149**, p. 113248. doi: https://doi.org/10.1016/j.eswa.2020.113248.

[29] Pazzani, M.: «A Framework for Collaborative, Content-Based and Demographic Filtering». *Artificial Intelligence Review*, 1999, **13(5-6)**, pp. 393–408.

[30] Pliakos, K.; Joo, S.H.; Park, J.Y.; Cornillie, F.; Vens, C. y Noortgat, W.V.: «Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems». *Computers & Education*, 2019, **137**, pp. 91–103. doi: https://doi.org/10.1016/j.compedu.2019.04.009.

[31] Portugal, I.; Alencar, P. y Cowan, D.: «The use of machine learning algorithms in recommender systems: A systematic review». *Expert Systems With Applications*, 2018, **97**, pp. 205–227.

[32] Quinlan, J. R.: «Induction of Decision Trees». *Machine Learning*, 1986, **1**, pp. 81–106.

[33] Reza Zafarani, Huan Liu, Mohammad Ali Abbasi: *Social Media Mining*. Cambridge University Press, 2014.

[34] Rodger, J.A.: «An expert system gap analysis and empirical triangulation of individual differences, interventions, and information technology applications in alertness of railroad workers». *Expert Systems with Applications*, 2020, **144**, p. 113081. doi: https://doi.org/10.1016/j.eswa.2019.113081.

[35] Sahu, A.K.; Dwivedia, P. y Kant, V.: «Tags and Item Features as a Bridge for Cross-Domain Recommender Systems». *Procedia Computer Science*, 2018, **125**, pp. 624–631.

[36] Sarwar, B.; Karypis, G.; Konstan, J. y Riedl, J.: «Analysis of recommendation algorithms for e-comerce». *Proceedings of ACM E-Commerce 2000 conference*, 2000, pp. 158–167.

[37] Serrano-Guerrero, J.; Herrera-Viedma, E.; Olivas, J.A.; Cerezo, A. y Romero, F.P.: «A Google Wave-based Fuzzy Recommender System to disseminate Information in University Digital Libraries 2.0». *Information Sciences*, 2011, **181**, pp. 1503–1516.

[38] Son, L.H.: «Dealing with the new user cold-start problem in recommender systems: A comparative review». *Information Systems*, 2016, **58**, p. 87–104.

[39] Tao, L.; Cao, J. y Liu, F.: «Dynamic feature weighting based on user preference sensitivity for recommender systems». *Knowledge-Based Systems*, 2018, **149**, pp. 61–75.

[40] Tejeda-Lorente, A.; Porcel, C.; Bernabé-Moreno, J. y Herrera-Viedma, E.: «REFORE: A recommender system for researchers based on bibliometrics». *Applied Soft Computing*, 2015, **30**, pp. 778–791.

[41] Tejeda-Lorente, A.; Porcel, C.; Peis, E.; Sanz, R. y Herrera-Viedma, E.: «A quality based recommender system to disseminate information in a University Digital Library». *Information Science*, 2014, **261**, pp. 52–69.

[42] Terán, L.; Oti-Mensah, A. y Estorelli, A.: «A literature review for recommender systems techniques used in microblogs». *Expert Systems with Applications*, 2018, **103**, pp. 63–73.

[43] Thomas, J. J.; Karagoz, P.; B., Ahamed B. y Vasant, P.: *Deep Learning Techniques and Optimization Strategies in Big Data Analytics*. IGI Global, 2019.

[44] Viktoratos, I.; Tsadiras, A. y Bassiliades, N.: «Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems». *Expert Systems With Applications*, 2018, **101**, pp. 78–90.

[45] Wei, J.; He, J.; Chen, k.; Zhou, Y y Tang, Z.: «Collaborative filtering and deep learning based recommendation system for cold start items». *Expert Systems With Applications*, 2017, **69**, pp. 29–39.

[46] Wu, H.; Yue, K.; Pei, Y.; Li, B.; Zhao, Y. y Dong, F.: «Collaborative Topic Regression with social trust ensemble for recommendation in social media systems». *Knowledge-Based Systems*, 2016, **97**, pp. 111–122.

[47] Yang, X.; Guo, Y.; Liu, Y. y Steck, H.: «A survey of collaborative filtering based social recommender systems». *Computer Communications*, 2014, **41**, pp. 1–10.

[48] Zhang, Y.; Shi, Z.; Zuo, W.; Yue, L. y Li, X.: «oint Personalized Markov Chains with social network embedding for cold-start recommendation». *Neurocomputing*, 2019, **Available online December 2019**. doi: https://doi.org/10.1016/j.neucom.2019.12.046.

[49] Zhoubao, S.; Lixin, H.; Wenliang, H.; Xueting, W.; Xiaoqin, Z.; Min, W. y Hong, Y.: «Recommender systems based on social networks». *The Journal of Systems and Software*, 2015, **99**, pp. 109–119.

Chapter 6

A Context-Aware Embeddings Supported Method to Extract a Fuzzy Sentiment Polarity Dictionary

In this chapter we include the following paper:

- A Context-Aware Embeddings Supported Method to Extract a Fuzzy Sentiment Polarity Dictionary.
  - Authors: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya, C. Porcel, E. Herrera-Viedma.
  - Journal: Knowledge-Based Systems, 190, 105236, 2020. Special issue on "New Innovations in Machine Learning and Software Science (NEMLSS)". ISSN 0950-7051.
  - DOI: `https://doi.org/10.1016/j.knosys.2019.105236`
  - Impact factor source: Web Of Science - Journal Ctitation Report
  - Impact factor: 8.038 (year 2020)
  - Category: Computer Science, Artificial Intelligence.
  - Quartile: Q1.
  - Ranking: 16 of 139.

Abstract

The latest development in cognitive technologies are helping us understand emotions and sentiments with unprecedented precision. Polarity detection is the key enabler to sentiment analysis and typically relies on experimental dictionaries, where terms are assigned polarity scores, yet lacking contextual information and based on human inputs and conventions.

In this article, we present a novel approach to automatically extract a polarity dictionary from a particular domain, the stock market, without human intervention and addressing the scaling

and thresholding problem. Our approach tracks the price changes of particular stocks over time, using it as a guiding polarity value. The magnitude of the price variation for a particular stock is then attributed to the financial news about this stock in corresponding period of time and that's what we use as our working corpus. On top of that, we derive the so-called binned corpus and apply the well-known TF-IDF information retrieval techniques to compute the TF-IDF value for each term. These values are then disseminated within the neighbourhood of each term based on the embeddings-enabled cosine distance. After introducing the problem and providing the background information, we thoroughly describe our method and all the components required to implement the system. Last but not least, we assign the terms to fuzzy linguistic labels and provide a volatility metric indicating how reliable our scores are depending on their distribution of occurrences in the corpus. To show how our approach works, we implement it for the Euro Stoxx 50 from January 2018 to March 2019 and discuss the results compared with typical approaches, pointing out potential improvements for further research work.

Keywords: sentiment analysis, polarity extraction, word embeddings, information retrieval, contextual bias, fuzzy polarity.

6.1 Introduction

In the recent years, cognitive computing -defined as the set of software and hardware techniques mimicing the functioning of the human brain-, has experienced a substantial development [27, 29]. The major cloud providers, such as Google, AWS, Microsoft and IBM, offer ready-to-use APIs for the developers' community to run these services on own data[35] and create a wider range of applications.

One of the areas covered by these services is sentiment analysis, which encompasses a combination of natural language processing, text analysis, computational linguistics, and biometrics to identify, extract, and quantify in a systematic way affective states and subjective information inherent in the human communication [2, 3, 5, 39].

Sentiment analysis has undergone a remarkable development in the last years too, becoming one of the most prolific research areas in the Natural Language Processing field [4, 5]. The computation of sentiments relies heavily on the existence of polarity dictionaries, where lemmas are given a score (usually between -1 and 1) representing the contribution of words containing this lemma to the overall sentiment of the particular sentence (for example, the polarity for the word "death" according with the popular *Syuzhet* dictionary [31] is -0,75). This simplistic conception of polarity does not account for the context of the term. Continuing with the example, the word "death" in a historical context (e.g. to count the fatalities of a battle) is certainly less *negatively-loaded* than "death" in the context of journalism, when press reports breaking news about a terrorist attack in a emotional heart-breaking context.

In addition, we face the so called thresholding and scaling problem. In [20] the authors show the difficulties comparing polarities given as crisp values (e.g. using the *Syuzhet* dictionary, we

obtain -1 for "addict" and "abuse", but also for "unfit" and "sleepless"... so we've lost the possibility of comparing the terms... is "sleepless" better or worse than "addict" or "abuse"? If we ask a human, probably she or he would consider "sleepless" to be "less worse" than "abuse"... but where is the threshold?

The over-reliance on these polarity scores certainly present therefore some challenges. Certainly one of the most critical ones is the fact that contextual information (leading to a contextual bias) is not captured in the polarity dictionaries (a polarity score for a lemma is immutable and not modifiable by the context). Yet, defining context-aware polarity scores for lemmas is challenging, as there is no guiding principle or systematic way of obtaining scores. In previous work [9][10], we defined methods for polarity bias modelling within a particular context, providing also a volatility score to assess how reliable our bias modelling is.

In this article, we want to go beyond polarity bias quantification and explore automatic ways of inferring polarity scores for a particular context: *finance markets*. Sentiment analysis has been extensively used to predict movements in the stock markets, to find change points, to assess the market appetite to buy or to sell and to quantify the duration of a bearish or bullish phase. The Finance markets domain, is quite appropriate to study sentiments and emotions. On one hand, a massive amount of finance related news are written everyday. News tickers provide near real time information about companies' financial health, potential events that might affect the stock course, press releases, analysts reports, product launches, etc. Specialized investment portals usually provide a news feed aggregating and tagging (e.g.: by stock symbol or company name, by index, by commodity, etc) all potential finance news. On the other hand, we have almost real time pricing and traded volume information available in all sorts of granularities. If we assume that the choice of words, tonality, emotional load and ultimately, the sentiment is correlated to (quite important) course changes, we can also use the historical price development and the historical collection of news about a particular market entity (a symbol, a fond, etc) to correct existing polarity scores or simply to learn new ones for the words present in the news.

The main contribution of this paper is *our approach to automatically extract a polarity dictionary from a particular domain, the stock market, without human intervention and addressing the scaling and thresholding problem*. Concretely, our contribution can be broken down into following items:

- We have created a new technique to extract news and label them with a price change magnitude, creating a weighted news collection for further processing.
- We have introduced the concept of *binned corpus*, as we are going to explain in the section 6.3.2
- We have re-purposed standard information retrieval techniques, such as *term frequency - inverse document frequency* to extract the guiding polarity value for each term from the binned corpus.
- We have leveraged an embeddings-based approach to compute the neighbourhood of a term and as a mechanism to disseminate guiding polarity values to the rest of the terms.

- We have transformed crisp polarity scores into fuzzy linguistic sets to make the result more generalizable and less subject to imposed thresholds and also computed the volatility related to the polarity score given the support from the domain content.

But first and foremost, with our approach we solve the three traditional issues inherent to classic polarity dictionaries based approaches:

- The scale and thresholding problem, as we provide fuzzy linguistic sets instead of crisp polarity values
- The human bias problem, as the polarity values are fully inferred without human intervention, providing on top an indicator on how reliable each particular polarity value is.
- The contextualization problem, as the polarity values are specific to our domain.

Our work is structured as follows: after presenting the rational of our attempt in the introduction, we provide the background information supporting our research. Then, we introduce important definitions we are going to use in our approach and explain thoroughly how new/corrected polarity scores are computed and mapped to linguistic fuzzy sets. Subsequently, we discuss the results obtained after applying our method in a practical case with the 50 Euro Stoxx stocks. To finish the paper, we provide the main concluding remarks and point to further research lines.

6.2 Background

In this section we provide the background information required to sustain our work. First we introduce the topic of polarity detection and revise the approaches to automatic polarity extraction. Then we go through the related work exploring the connection between stock prices and financial news, social media, etc. (which is one of the key assumptions for our approach to work). Then we introduce the fuzzy linguistic modelling we use to extract the fuzzy version from our intermediate crisp polarity dictionary. Finally, we review the fundamentals of word embeddings and their usage to establish relationships between terms within a corpus, which we also exploit in our method.

6.2.1 On Polarity Detection

The standard approach to *polarity detection* or *semantic orientation* uses either a pre-trained or a manually labelled polarity lexicon or dictionary. Thus, these dictionaries are at the core of any sentiment analysis related activity. One of the first examples is the dictionary created by Hu et al. [28], consisting of 6779 terms (4776 assigned to -1 and 2003 to +1), extensively used on customer reviews for opinion mining-.

A commercial version issued by Daku and his co-authors, the Lexicoder [19] had a similar aim. The more recent *Syuzhet* dictionary [31] provides over 10K entries, with scoring ranging

from -1 to 1. The Positive Affect Negative Affect Scale technique (PANAS) [58] expands into the psychology domain offering a psychometric scale for detecting mood fluctuations.

The well-known SentiWordNet [7] provides a dictionary-based approach to extract sentiments. This dictionary relies on Part of Speech tagging to apply a lexical dictionary to *synsets* or synonym set groups (adjectives, nouns, verbs, and other grammatical classes). The polarity computation of a given text is an aggregation operation [4] across all the existing synsets, each one contributing with their own positive or negative affect score.

We find a lot of researchers focusing in addition on modeling happiness based on sentiments. In [21], Dodd proposed a dictionary based Happiness Index derived from the Affective Norms for English Words (ANEW). Araujo et al. in [6] suggested a method to map the happiness index to positive or negative polarity values. ANEW has been used for many other applications, such as extraction of emotional profiles for locations [8].

Some authors have approached the polarity problem from furhter angles. Thelwall et al. explored approaches to compute the sentiment strength. Their SentiStrength [56] relies on the existing *Linguistic Inquiry* and *Word Count* dictionary [47] to implement supervised and unsupervised classification methods and extract the strength of the sentiments, including polarity. Similarly, *SenticNet* [14] applies classification techniques to Natural Language Processing structures to infer the polarity for nearly 14K concepts. The new version, *SenticNet 5* [15], implements further improvements based on deep learning techniques.

6.2.2 On Automatic Polarity Extraction

As aforementioned, one of the weaknesses of classical sentiment analysis is the dependency on the quality of a polarity dictionary. As we've seen before, polarity dictionaries are typically biased, inaccurate and not context aware. Thus, many researcher have focused their work on improving the quality of the polarity dictionaries in three different ways: adapting them to the context of particular domains, defining correcting functions to the polarity value and implementing a controlled high-quality automatic way of creating polarity dictionaries.

Back in 2006, Kanayama et al. introduced in [34] the notion of polar atoms and presented an coherency based approach (assuming that similar polarities tend to appear successively in context). The approach implements a redistribution of polarity values based on density and precision of coherency in a corpus. Agathangelou et al. in [1] proposed an approach for domain-specific dictionary building based on the software called NiosTo, which rather that infer polarity values from scratch, relies on existing dictionaries.

Peng and his co-authors presented in [46] an automatic sentiment dictionary generation method, called Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) algorithm, to assign polarity scores to each word in the dictionary, and benchmarked the results with human-labeled dictionaries from AMT and the General Inquirer lexicon.

Back in 2006, Kanayama introduced the notion of polar atoms and presented an coherency based approach (assuming that similar polarities tend to appear successively in context). The approach implements a redistribution of polarity values based on density and precision of coherency in a corpus.

An interesting approach can be found in [23], where the authors proposed a method to readjust polarities based on the presence of *emoticons* on micro-blogs-. Basically, the approach uses the presence of *emoticons* to compute a polarity added value (extension) to the existing scores and later uses SVM to classify the sentiment word to build up the dictionary. In the same research line, Cambria et al. [13] created *Affective 2*, a language visualization and analysis system that allows for reasoning by analogy on natural language concepts. The proposal is then enhanced and generalized in [12]. In [17] the authors propose a richer deep learning powered approach to overcome the language specificity limitation inherent to Affective space vectors. Their approach builds upon the so called *Convolutional Fuzzy Sentiment Classifier* to predict the degree of a particular emotion in the Affective Space, performing in a 4 dimensional emotional space to speed up the classification performance. The recent advances in deep learning technologies have been extensively applied to the sentiment analysis. For example, Ma et al. in [40] obtained promising results augmenting the long short-term memory (LSTM) network with a hierarchical attention mechanism consisting of a target level attention and a sentence-level attention [2], extending the seminal work of [16]. Further deep learning methods, such as capsule networks, allowed for increased performance tackling sentiment classification problems. The capsule network is a structured model that solves many of the problems inherent to deep learning based text analytics. Capsules are locally invariant groups that learn to recognize the existence of visual entities and encode their properties into vectors. Capsule networks utilize a nonlinear function called squashing because capsules (groups of neurons) are represented as a vector. Capsules consider the spatial relationships between entities and learn these relationships via dynamic routing [49]. In [61] a capsule approach based on Recurrent Neural Network (RNN) has been proposed. For a given problem, one capsule is built for each sentiment category e.g.,'positive' and 'negative'. Each capsule has an attribute, a state and three modules: representation module, probability module, and reconstruction module. Based on capsule representation, the probability module computes the capsule's state probability.

The contextual bias problem in polarity detection has been addressed in the literature. In [9] the authors suggested a method to quantify and amend the contextual polarity bias using fuzzy linguistic modelling to define both the correction factor and the volatility of the inferred factor. The same method has been improved one year later introducing embeddings as a tool to capture situational and contextual interdependences ([10]).

6.2.3 On Using Stock Markets and sentiments

Sentiment analysis has been extensively used in the context of stock markets. Our work relies on the correlation between polarity of the financial news and the stock price variation, which has been thoroughly explored to create stock price prediction models.

Bollen et all. [11] analysed how collective mood states derived from large-scale Twitter feeds show some degree to correlation with the value of the Dow Jones Industrial Average over time. For that, they leverage 2 mood tracking tools on daily Tweets, the Opinion Finder and Google Profile of Mood States, establishing the correlation between 6 mood states (Calm, Alert, Sure,

Vital, Kind, and Happy) and potential price variations.

Nguyen et al.[45] developed a model that captures topics and sentiment from the social media feed simultaneously and proposed a new topic model adapting LDA (TSLDA). With their model, they proved that sentiment analysis of social media can help improve the stock prediction

The impact of financial news on stock prices is thoroughly studied by Li and his co-authors in [38, 37]. In their work, they describe the creation of a sentiment space combining different polarity dictionaries (Loughran–McDonald, Harvard psychological dictionary) to enhanced a generic stock price prediction framework, showing a superior performance compared to the models just using bag of words.

Seng et al. [53] suggested an approach to develop a dictionary with grammar and multiword structure, based on sentiment orientation and score of data with added information, which in conjunction with sentiment analysis, allows to investigate the relationship between financial news and stock market volatility. The results prove a strong correlation.

6.2.4 Fuzzy linguistic modelling

The fuzzy linguistic approach is a tool based on the concept of linguistic variable proposed by Zadeh [59]. This theory has given very good results to model qualitative information and it has been proven to be useful in many problems.

The 2-Tuple Fuzzy Linguistic Approach

The 2-Tuple Fuzzy Linguistic Approach [25] is a continuous model of information representation that allows reduction in the loss of information that typically arises when using other fuzzy linguistic approaches, both classical and ordinal [24]. To define it both the 2-tuple representation model and the 2-tuple computational model to represent and aggregate the linguistic information have to be established.

Let $\mathcal{S} = \{s_0, ..., s_g\}$ be a linguistic term set with odd cardinality. We assume that the semantics of labels is given by means of triangular membership functions and consider all terms distributed on a scale on which a total order is defined. In this fuzzy linguistic context, if a symbolic method aggregating linguistic information obtains a value $\beta \in [0, g]$, and $\beta \notin \{0, ..., g\}$, we can represent β as a 2-tuple (s_i, α_i), where s_i represents the linguistic label, and α_i is a numerical value expressing the value of the translation between numerical values and 2-tuple: $\Delta(\beta) = (s_i, \alpha)$ y $\Delta^{-1}(s_i, \alpha) = \beta \in [0, g]$ [25].

In order to establish the computational model negation, comparison and aggregation operators are defined. Using functions Δ and Δ^{-1}, any of the existing aggregation operators can be easily be extended for dealing with linguistic 2-tuples without loss of information [25]. All details can be found in our previous paper [9].

Multi-Granular Linguistic Information Approach

To accommodate the requirements of the different sentiment analysis methods, it's important to support different "granularity levels". Certain methods could for example only deal with yes/no values and direction only (e.g.: *"Negative Bias", "No Bias", "Positive Bias"*). Other methods

might be able to incorporate higher granularity values in the aggregation operation for the sentiment computation (e.g.: *"Lowest", "Low", "Normal", "High", "Highest"*).

To enable the compatibility of sentiment analysis methods, we need to support the different granularities and provide tools to manage the multi-granular linguistic information. In [26] a multi-granular 2-tuple fuzzy linguistic modelling based on the concept of linguistic hierarchy is proposed.

A *Linguistic Hierarchy*, LH, is a set of levels $l(t, n(t))$, where each level t is a linguistic term set with different granularity $n(t)$. The levels are ordered according to their granularity, so that we can distinguish a level from the previous one, i.e., a level $t+1$ provides a linguistic refinement of the previous level t. We can define a level from its predecessor level as: $l(t, n(t)) \rightarrow l(t + 1, 2 \cdot n(t) - 1)$. In [26] a family of transformation functions between labels from different levels was introduced. To establish the computational model we select a level that we use to make the information uniform and thereby we can use the defined operator in the 2-tuple model. This result guarantees that the transformations between levels of a linguistic hierarchy are carried out without loss of information.

6.2.5 On Machine Learning Methods and Word Embeddings

Machine Learning has revolutionized the approach to Natural Language Processing tasks. Sentiment analysis has been one of the areas that has profited the most [60]. Socher et al.[54] implemented the so called RTNT model, exploiting the structure of the sentence to compose the single terms' sentiments in order to get the overall sentiment of the sentence. It represents the words by vectors and takes a class of tensor-multiplication-based mathematical functions to describe compositionality. The big advantage of this model is that it is very interpretable. We can visualize which words it detects to be positive or negative, and how it understands the compositions. However, we need to build an extremely large training set for every specific application. In [30], the authors explore for the use of deep convolution neural networks applied to short messages, in concrete, tweets, with astonishing results.

In the recent years, the usage of a deep learning technologies enable the representation of words as vectors and the emerging of the Words Embeddings Technologies.

The ground principle of Word embeddings (also known distributional vectors) is the continuous vectorial representations of words that follow the distributional hypothesis [50], according to which words with similar meanings tend to occur in similar context. Distributional vectors, as such, are designed to capture the characteristics of the neighbours of a term.

Distribution vectors enable arithmetic operations between words. For example, we can compute how similar 2 words in a corpus are, by using standard similarities functions, such as the cosine distance. Word embeddings are often used as the first data processing layer in a deep learning model. Embeddings are typically trained by optimizing an auxiliary objective in a large unlabelled corpus and can be used in various scenarios, such as predicting a term given its context, where the resulting distributional vectors can capture general syntactical and semantic information.

We can consider Mikolov et al. as the fathers of the distribution vectors. In 2016, these

authors released two seminal papers, [36] and [43], presenting the well-known *word2vec* approach, which guarantees the scalability in the generation of word embeddings (some models available that have been trained with more than 100 billion words). Mikolov et al. presented the vectors algebra as a way to perform operations between words, as the vectors preserved the semantic consistency, for example, vec(King) $-$ vec(woman) is close to vec(Queen). One of the most exploited features, which we extensively use in our proposal, is the support for measuring similarity between vectors, for example using measures such as *cosine similarity* or just the typical euclidean distance.

Mikolov revolutionized the word embedding with his two models: Continuous Bags Of Words, which computes the conditional probability of a target term given the context words surrounding it across a window of size *k* and skip-gram model, which works the other way around: predicting the surrounding context words given the central target word, being the context words assumed to be located symmetrically to the target words within a distance equal to the window size in both directions).

Following the success of word2vec, further ground-breaking algorithms approached the embeddings generation in slightly different ways. *FastText* (presented in [33]) for example learns vectors for the n-grams that are found within each word, as well as each complete word (the mean of the target word vector and its component n-gram vectors are used for training at each training step). The adjustment that is calculated from the error is then used uniformly to update each of the vectors that were combined to form the target, adding additional complexity but showing better performance in some scenarios.

GloVe (Global Vectors for words representation) [48] works similarly as word2vec with a caveat. Instead of predicting the context given word, GloVe learns by constructing a co-occurrence matrix (words x context) that basically counts how frequently a word appears in a context, applying different degrees of factorization to achieve a lower-dimension representation. In this work we are going to apply GloVe to create our embeddings in our 2 different corpora.

Word embeddings present some limitations, for example the inability to represent phrases, where the combination of two or more words (e.g., idioms like “smoke and mirrors” or named entities such as “Real Madrid’) does not represent the combination of meanings of individual words. Some solutions have been researched to overcome this particular problem, such as identifying such phrases based on word co-occurrence and training embeddings for them separately [44], or directly learning n-gram embeddings from unlabelled data [32].

An additional limitation is inherent to the definition of the window for the surrounding words, which is problematic if used in tasks such as sentiment analysis [57] -semantic similarity with colliding polarities might be clustered together. The work performed by Teng and his co-authors [55] suggested a sentiment aware word embedding model based on supervised polarity incorporated into the loss function in the embeddings training phase.

The GloVe algorithm is implemented following these steps:

1. Word co-occurrence statistics gathering in a form of word co-occurrence matrix X, where each element X_{ij} represents how often word i appears in context of word j. Usually

we scan our corpus in the following manner: for each term we look for context terms within some area defined by a $window\_size$ before the term and a $window\_size$ after the term. Also we give less weight for more distant words, usually using this formula: $decay = \frac{1}{offset}$

2. Define a set of soft constraints for each word pair: $w_i^T w_i + b_i + b_j = log(X_{ij})$ where w_i is the vector for the main word, w_j is the vector for the context word, b_i and b_j are scalar biases for the main and context words.

3. Define a cost function: $J = \sum_{i=1}^{V} \sum_{j=1} \varphi(X_{ij})(w_i^T w_i + b_i + b_j - log(X_{ij}))^2$ Where φ is a weighting function which help us to prevent learning only from extremely common word pairs: $\varphi(X_{ij}) = \begin{cases} (\frac{X_{ij}}{X_{max}})^{\alpha} & \text{if } X_{ij} \leq XMAX \\ 1 & otherwise \end{cases}$

6.3 Automatic Sentiment Polarity Extraction

In this section we will describe how our system works to produce a fuzzy polarity dictionary extracted from the financial context. The Fig. 6.1 shows the process steps required to implement our approach. In the subsequent sections, we will introduce the necessary definitions and describe each module, from the data gathering until the final output.

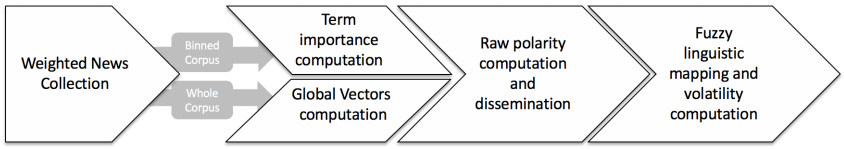


Figure 6.1: Overview of the Fuzzy Sentiment Polarity Dictionary creation process

6.3.1 Creation of Weighted News collection

The purpose of this step is to create a collection of news with a weight assigned to proxy the overall polarity of any particular news entry. As we discussed in the introduction, our idea is to gather all news related to a specific stock and use the in-percentage daily price changes as weights, as we will see below.

There are plenty of exchanges with a large number of stocks. In order to obtain robust polarity values, we need to find stocks that have both substantial media presence and large trading volumes. For this purpose, we opted in this paper for the stocks from a well-known index, such as Euro Stoxx 50 [1] (made up of fifty of the largest and most liquid stocks in the EURO zone).

Fig 6.2 shows how the system for data gathering and preparation works: the stocks are used as an input for our 2 harvesting modules: the *News Harvester* pulls news related to each identified stock symbol from different finance portals (typically using RSS protocol). Likewise,

[1] See `https://www.stoxx.com/index-details?symbol=SX5E`

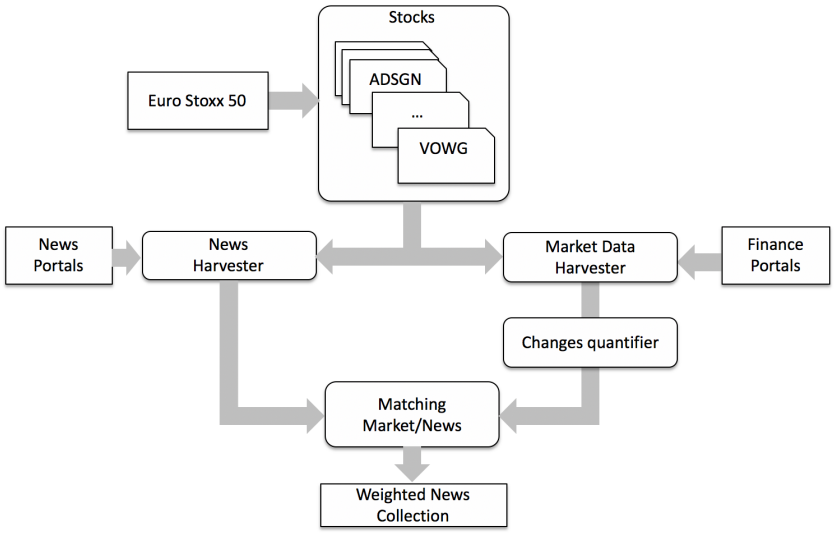


Figure 6.2: Data gathering and Weighted News Collection creation overview

the *Market Data Harvester* connects to specialized finance online portals (such as Yahoo Finance [2] to obtain the current and historical courses of the selected stocks.

The *Change Quantifier* tags those days with price changes over a particular threshold (where the price of a particular stock in absolute terms went over/under a given percentage within a particular time window compared to the price just before entering the time window). As we are going to use the magnitude of the positive or negative price change, we are going to work with different thresholds (according the frequency and magnitude of the changes, we suggest a range of thresholds from -10% or less to 10% or more in steps of 2%).

The *Matching Market/News* module selects the news that match the days labelled with any threshold value, discarding the other ones. The result is a collection of news per stock symbol, where each new is labelled with a threshold value and a sign. (e.g.: ADSGN corresponding to Adidas will have the news from the 12th of March 2018 labelled with a 10% or more, as we can see in the Fig. 6.7, moving from 104 to 122 in 2 days, which is 14%).

If $N(K,T)$ is the whole collection of news gathered for all considered stocks K during a period of time T, we represent $n(k,t) \in N(K,T), k \in K, t \in T$ as a single news referred to a particular stock k in a particular time unit t (usually days)

Let $p(k,t)$ be the close price of the stock k in the time unit t. Let w be a time window of w units (e.g.: 2 days). A more specific selection of the w is thoroughly explained and formalized in [42]. For our purposes it is important to keep a w consistent within the same stock market and long enough to capture the impact of the news but short enough to discern important price movements. Different stock markets might work better with different values of w.

Let TH be a discrete evenly distributed finite vector of thresholds (e.g.: $TH = [2,4,6,8,10]$)

[2] `https://finance.yahoo.com/`

Definition 6.3.1 *Weighted Bin* for a particular news $n(k,t)$ issued in the time t about the stock k, is defined as the maximum value of the threshold th so that the price change of k during the next w time units is equal to or greater than th

$$WB(n(k,t),p(k,t),w,TH) = \text{argmax}_{th \in TH}(|(p(k,t+w) - p(k,t))/100| >= th)$$

Definition 6.3.2 *Signed Weighted Bin* is the *Weighted Bin* with a positive sign indicating a stock price increase or negative indicating a decrease:

$$sWB(n(k,t),p(k,t),w,TH) = \begin{cases} WB(n(k,t),p(k,t),w,TH), \\ \text{if } p(k,t+w) > p(k,t) \\ (-1) * WB(n(k,t),p(k,t),w,TH), \\ \text{otherwise} \end{cases} \tag{6.1}$$

Representing the example above in the newly introduced notations:
$n(ADSGN,'2018/03/12')$ with a window of $2days$ would have a positive weighted bin of 10% $WB(n(ADSGN,'2018/03/12'),p(ADSGN,'2018/03/12'),'2days',TH) = 10\%$

Thus, the weighted news collection is the set of all news referred to the selected stocks and their corresponding *Signed Weighted Bin*. As we are using the positive or negative price change as a proxy for the news polarity, we are interested in significant variations of the price. It can be controlled by the lower end of the TH vector (e.g.: defining a minimum price change of 4% instead of 2%). In [42], Merello et al. formalized the financial news impact problem in a timely dependent manner referred to the selection of w(in time units)

6.3.2 Term importance computation in binned Corpus

Once the weighted collection is ready, we can proceed with the pipeline presented in Fig. 6.3. Each news text goes through a pre-processing step, where tokenization [22], removal of stop words [52], lematization and Part of Speech tagging (implemented with [51]) and selection of particular PoS tags (nouns, verbs, adjectives) and filtering by a minimum of occurrences (to avoid sparsity and noise)

The result is a normalized corpus, containing as many documents as relevant news identified in the subsection above. Taking it as an input, we create a new corpus with as many documents as thresholds employed in the definition of *Signed Weighted Bins*(see Def.6.3.2). Each document is the aggregation of all the news, no matter from which stock, within the same *Signed Weighted Bin*, as explained in Fig. 6.4 and expressed below:

$$[n(k,t), sWB(n(k,t),p(k,t),w,TH)] \rightarrow [sWB(n(k,t),p(k,t),w,TH), \Xi(n(k,t), sWB(n(k,t),p(k,t),w,TH))] \tag{6.2}$$

where $n(k,t)$ represents a particular news about stock k in the time t, sWB has been defined in Def. 6.3.2 and $\Xi(n,th)$ represents a function that aggregates all news belonging to a particular

sWB bin.

The binned corpus allows for applying the well-known algorithm TF-IDF [18], which we use to compute for each and every term, how much that term is important to that document with respect to the corpus.

Due to the nature of the stock market, smaller prices changes are more likely to happen. Thus, we can expect much higher number of news in the lower weighted bands ($\pm$ 2%, $\pm$ 4%) than in the ones reflecting higher prices changes ($\pm$ 8%, $\pm$ 10%). In order to have a proper significance when we extract the polarity, we need to establish a minimum occurrences threshold per term, which shall be proportional to the size of the weighted bin. In addition we introduce following definition to force a minimum of occurrences of a term in both corpus (binned and standard)

Definition 6.3.3 *Polarity Computing Threshold*. This is the minimum number of documents with occurrences of any term t_i in a standard corpus, so that the polarity computation makes sense. It is established for a particular Domain Corpus C and is a constant value $PCT(C) = K$.

As the binned corpus is derived from the standard corpus, the minimum occurrence condition will be only validated in the standard one.

The closer to 1 the TF-IDF value for a particular lemma in a particular signed weighted bin, the more representative is this particular lemma for this signed weighted bin. Using this relationship, we introduce the concept of guiding polarity, which combines the value signed weight bin itself and theTF-IDF of a lemma belonging to this bin:

Definition 6.3.4 *Guiding Polarity* $GP(t, BC)$ is the maximum absolute value obtained after multiplying the tf-idf value for a lemma l in a bin by the signed weighted value of this bin: $GP(l, BC) = |\operatorname{argmax}_{t \in TH}(tf - idf(l, t, BC)) * sWB(t)|$

Definition 6.3.5 *Signed Guiding Polarity* $sGP(t, BC)$ is the *Guiding Polarity* with the signed carried by the sWB that fulfilled the condition for Guiding Polarity

By the end of this step, we will have the positive and negative guiding polarities for the terms that are most representative, which we will use in combination with the GloVal vectors, as explained in the next section, to disseminate the polarity to other terms.

6.3.3 GloVal Vectors Computation in broader Corpus

In the broader corpus, we will apply the GloVe algorithm (as explained in the Background subsection 6.2.5) to compute the vectorial representation of our terms. As mentioned before, we stick to the pipeline of Fig. 6.3, applying all pre-processing steps (tokenization [22], removal of stop words [52], lemmatization, Part of Speech tagging, PoS tags selection (to avoid volatility we suggest to keep the most meaning carrying words, such as nouns, verbs and adjectives, but obviously our approach can be extended to any kind of Part of Speech label) and filtering by a minimum of occurrences.

Although different embeddings technologies can be applied, we have opted for Global

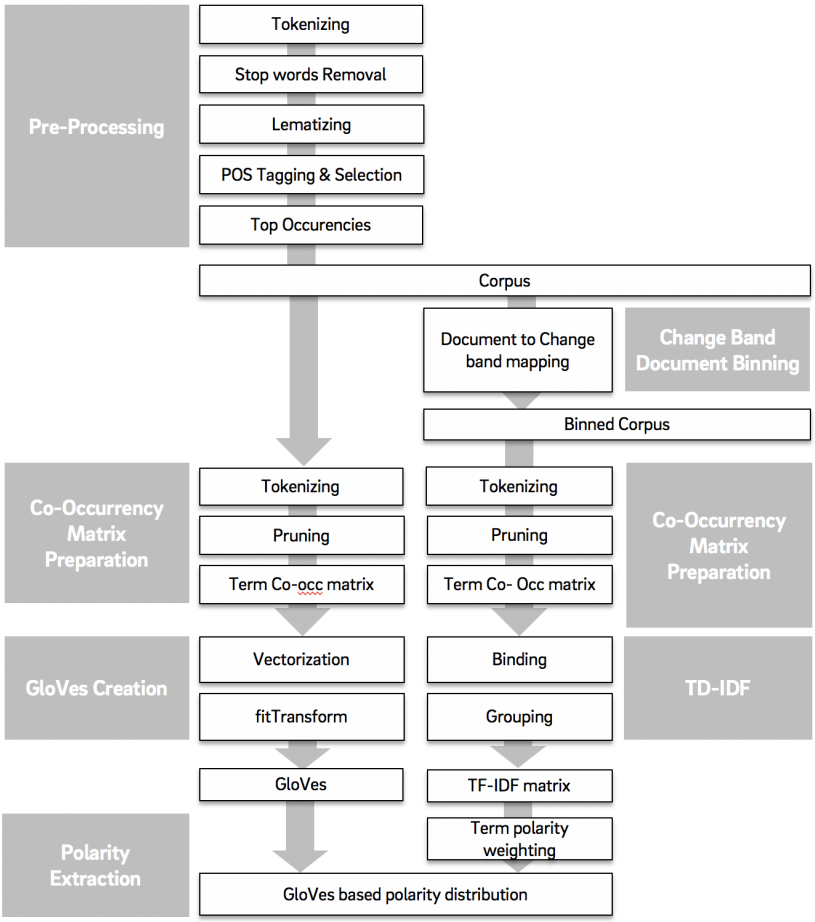


Figure 6.3: Overview of the system modules to perform the context-aware polarity extraction

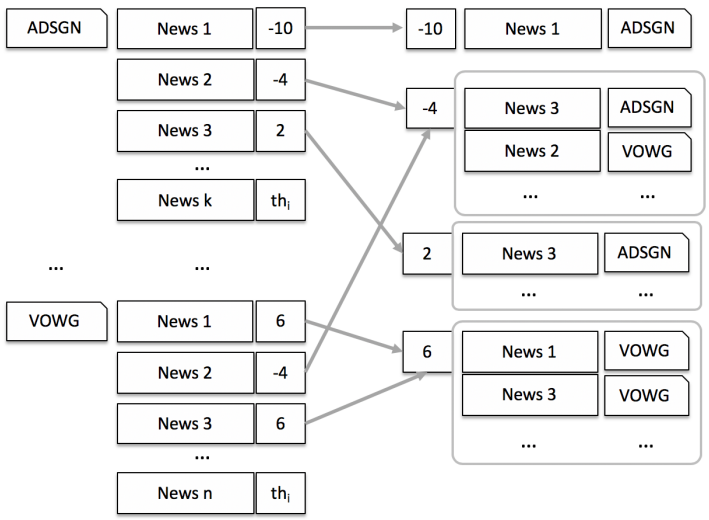


Figure 6.4: Process of creation of binned corpus

Vectors because a) it is very straightforward (i.e., to enforce the word vectors to capture sub-linear relationships in the vector space and therefore shows a higher performance), b) it adds

additional practical meaning into word vectors by considering the relationships between word-pair to word-pair rather than word -word and c) it gives lower weight for highly frequent word pairs so as to prevent the dominance of meaningless stop words-like terms.

For each term, we compute the *(*Embeddings Neighbourhood), defined in [10] as follows:

Definition 6.3.6 *Embedding Neighbourhood*. We define the embeddings neighbourhood of a term t_i given a window length w, $EN(t_i, w)$, as the set of all terms T containing the top w terms to t_i that maximizes the cosine similarity measure, $cos(t_i, t_j) = \frac{t_i \cdot t_j}{||t_i|| \cdot ||t_j||}$

We will use the *(*Embeddings Neighbourhood) as the scope at term level to disseminate the *signed Guiding Polarity*

6.3.4 Guiding Polarity dissemination

The step now consists of passing the guiding polarity of all terms identified in the TF-IDF procedure onto their own Embeddings Neighbourhood. Let's say k is a guiding polarity term.

Let's call K the set of all terms having a signed guiding polarity. The disseminated polarity for a term l is computed as follows:

$$disPolarity(l) = \sum_{k} sGP(k, BC) * cossim(GloVe(l), GloVe(k)) \tag{6.3}$$

$l \in EN(k, w), k \in K, l \notin K$, Where k represents the set of all signed guiding polarity terms in whose neighbourhood the term l is present, $GloVe(l)$ and $GloVe(k)$ the vectorial representation of l and k respectively, and w the size of the window to define the scope of the neighbourhood (constant)

Our raw polarity dictionary is the union of signed Guiding Polarities to the disseminated Polarities.

6.3.5 Fuzzy Linguistic Mapping and Volatility computation

In the previous subsection 6.2.4, we introduced the fundamentals of fuzzy linguistic modelling and defined the 2-tupla based supporting arithmetic operations to enable the computing of sentiment analysis tasks. We now need to map the polarity values in the raw polarity dictionary obtained in the step before to linguistic labels.

In order to provide a sense of how much evidence is behind the polarity definition of a particular term, we define a measure for the stability (as opposed to volatility), based on both number of occurrences of the term in the corpus. Thus, the user of the polarity dictionary can have the choice of disregard volatile polarities.

Definition 6.3.7 *Polarity Stability* This is an indicator for how stable the polarity computation for a particular term is. The minimum value can be the imposed as $PCT(C)$ (as explained in Def. 6.3.7 and the maximum of $\#C$. To standardize this value, we define a normalizing

function ϵ, defined as $\epsilon : [PCT(C), \#C] \longrightarrow [0, 1]$, which makes the Polarity Stability value range between 0 and 1:

$$PS(t_i, C) = \epsilon(\frac{\#M}{\#C}) \tag{6.4}$$

where M represents the set of documents in the standard corpus, where the term t_i is present and C the set of all documents in the Corpus.

For both cases, we are going to use different label sets (S_1, S_2) selected from a *LH* [26]:

- **Polarity Domain Value** of a term in a our context $PDV(t)$, which is assessed in S_1.
- **Polarity Supporting Indicator** applied to the previous indicator $PSV(t)$, which is assessed in S_2.

Although this framework guarantees the flexibility in the choice of the *LH*, we suggest using a 2 level *LH* with 3 and 5 labels each one for the *Bias Model stability indicator* and a 2 level *LH* with 5 and 9 labels for the *Polarity Domain Value* itself. Our suggestion is motivated by the intent of making it more tangible for the reader, but the choice of (S_1, S_2) remains generic and shall be taken depending on the nature of the problem or convenience for further operations.

The *Polarity Domain Value* in combination with the *Polarity Supporting indicator* constitutes our context-aware fuzzy sentiment polarity dictionary D:

$D \equiv [t, PDV(t), PSI(t)]$

6.4 Experimentation

To implement our approach, we chose the stocks listed in the EuroStoxx 50 index [3] (composed by fifty of the largest and most liquid stocks in the EURO zone), because of the trading volumes, financial news richness and variety of industries. The table 6.1 shows the concrete stocks and the number of financial news we have gathered between Jan 2018 and March 2019.

In Fig. 6.7 we show, taken Adidas as example, how the different price changes defining the weighted bins manifest. We have defined a time window of 2 days to register the price change, following the recommendations of [41] about news lags and delays. The period of time we have chosen presents enough price variations to support the analysis. The higher we set the threshold, the less occurrences we observe (for example, in the table 6.2 we just see one occurrence for a positive 10% price variation, no one for a -10%, but as we go down to $\pm$ 8% ,$\pm$6% up to $\pm$2% , we start having almost 2K ocurrences in both positive and negative bins).

The news assigned to the different price variations bins have undergone the pre-processing routing explained in Fig. 6.3 (tokenizing, stop words removal, lemmatizing and PoS Tagging). For our evaluation we just selected *nouns*, *verbs* and *adjectives*, as those are typically the highest contributors to the sentiment of a sentence. The result is a fully normalized corpus with one document per financial news gathered. Applying the formula 6.3.2, we created the binned corpus and applied TF-IDF to obtain the *signed guiding polarities*. In the Fig. 6.5 we can see the top 25 terms per bin visualized.

[3] See `https://www.stoxx.com/index-details?symbol=SX5E`

	stock	#gathered_news	stock	#gathered_news
1	ADIDAS	210	INDITEX	20
2	AIR LIQUIDE	40	ING	210
3	AIRBUS	800	INTESA SANPAOLO	40
4	ALLIANZ	110	KERING	145
5	AMADEUS	50	KONINKLIJKE AHOLD DELHAIZE	40
6	ANHEUSER-BUSCH	300	KONINKLIJKE PHILIPS	50
7	ASML HOLDING	60	LINDE	170
8	AXA	110	LOREAL	60
9	BANCO SANTANDER	130	LOUIS VUITTON	50
10	BASF	160	MUENCHENER RUECKVERSICHERUNG	8
11	BAYER	370	NOKIA	320
12	BBVA	60	ORANGE	220
13	BMW	260	SAFRAN	70
14	BNP PARIBAS	100	SANOFI	370
15	CRH PLC	30	SAP	140
16	DAIMLER	260	SCHNEIDER ELECTRIC	50
17	DANONE	90	SIEMENS	270
18	DEUTSCHE POST	80	SOCIETE GENERALE	80
19	DEUTSCHE TELEKOM	120	TELEFONICA	200
20	ENEL	40	TOTAL	1470
21	ENGIE	300	UNIBAIL-RODAMCO-WESTFIELD	40
22	ENI	270	UNILEVER	310
23	ESSILORLUXOTTICA	5	VINCI	50
24	FRESENIUS	180	VIVENDI	180
25	IBERDROLA	50	VOLKSWAGEN	590

Table 6.1: Euro Stoxx 50 stocks used to extract our corpus and number of financial news gathered in the period of study

Weighted Bin	+	-
0.02	1874	1893
0.04	240	324
0.06	33	43
0.08	10	7
0.1	4	0

Table 6.2: Number of news per weighted bin. E.g. positive 0,02 bin has a total of 1874 news, while the negative 0,08 bin only 7 news

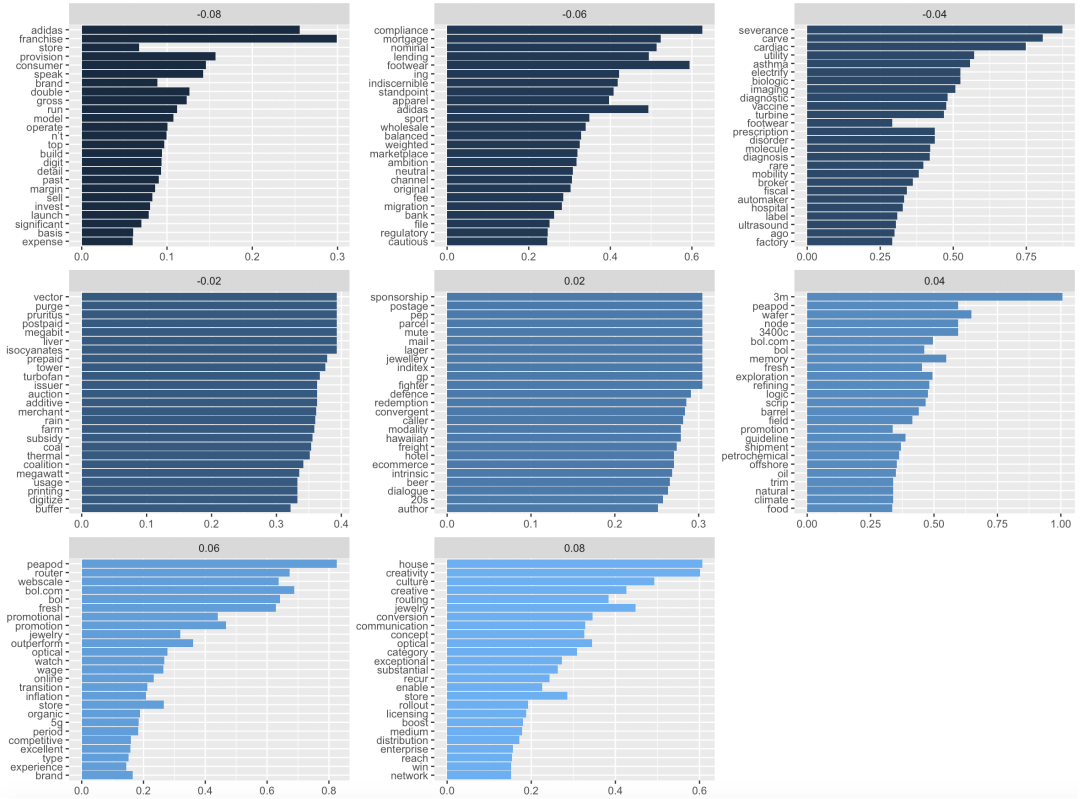


Figure 6.5: Top terms in the TF-IDF step

To proceed with the polarity dissemination, we applied the GloVe algorithm to create the global vectors and compute the Embeddings Neighbourhood (as explained in 6.3.3. In Fig. 6.6 we can see for example, the extended neighbourhood for the terms *retail* and *compliance*.

Figure 6.6: Embeddings visualization for "compliance" and "retail"

After aggregating both *signed guiding polarities* and *disseminated polarities*, we apply the fuzzy linguistic mapping assigning a linguistic label to each polarity value. For our implementation, we opted for a level 5 label set, as explained in 6.3.5 with the labels *Almost non-existent, Slight, Medium, Strong, Very Strong* for both positive and negative polarities, to obtain the *Polarity Domain Values*

To complete our fuzzy polarity dictionary, the *Polarity Supporting Indicator* for each term is computed (as explained in the subsection 6.3.5). For this purpose, we use a different level 5 label set *Very weak, Weak, Medium, Strong, Very Strong*

In the tables 3 and 4 we show the terms with the highest and the lowest fuzzy polarity. As we can also see, the *Polarity Supporting Indicator* helps understanding the reliability of the inferred polarities.

In table 6.4 we provide the distribution of *Polarity Supporting Indicator* labels by *Polarity Domain Value* label. As we can observed, the are quite balanced.

The entire dictionary can be downloaded from `https://bit.ly/2XdkyqQ`

	term	maxpolarity	maxpolarityfuzzy	support	supportfuzzy
1	peapod	0.07	Very Strong positive	0.00	Very weak
2	directly	0.07	Very Strong positive	0.00	Very weak
3	meal	0.06	Very Strong positive	0.01	Very weak
4	northeast	0.06	Very Strong positive	0.01	Very weak
5	bol.com	0.06	Very Strong positive	0.01	Very weak
6	bol	0.06	Very Strong positive	0.01	Very weak
7	nationality	0.06	Very Strong positive	0.01	Very weak
8	fresh	0.05	Very Strong positive	0.01	Medium
9	globe	0.05	Very Strong positive	0.02	Medium
10	sportswear	0.05	Very Strong positive	0.00	Very weak
11	jewelry	0.05	Very Strong positive	0.02	Medium
12	house	0.05	Very Strong positive	0.03	Strong
13	mall	0.05	Very Strong positive	0.01	Weak
14	creativity	0.05	Very Strong positive	0.01	Weak
15	owned	0.04	Very Strong positive	0.01	Medium
16	shelf	0.04	Very Strong positive	0.01	Weak
17	relationship	0.04	Very Strong positive	0.17	Very Strong
18	optical	0.04	Very Strong positive	0.02	Medium
19	router	0.04	Very Strong positive	0.01	Weak
20	compensation	0.04	Very Strong positive	0.13	Very Strong

Table 6.3: Top positive terms in the fuzzy polarity dictionary

	term	maxpolarity	maxpolarityfuzzy	support	supportfuzzy
1	jersey	-0.08	Very Strong negative	0.01	Very weak
2	school	-0.07	Very Strong negative	0.01	Very weak
3	newness	-0.07	Very Strong negative	0.01	Very weak
4	footwear	-0.06	Very Strong negative	0.01	Weak
5	harm	-0.06	Very Strong negative	0.01	Weak
6	sport	-0.06	Very Strong negative	0.02	Medium
7	scalability	-0.06	Very Strong negative	0.01	Weak
8	overhead	-0.06	Very Strong negative	0.02	Medium
9	adidas	-0.05	Very Strong negative	0.02	Medium
10	football	-0.05	Very Strong negative	0.01	Weak
11	nominal	-0.05	Very Strong negative	0.02	Medium
12	franchise	-0.04	Very Strong negative	0.04	Strong
13	den	-0.04	Very Strong negative	0.01	Very weak
14	compliance	-0.04	Very Strong negative	0.01	Medium
15	rolling	-0.04	Very Strong negative	0.02	Medium
16	headcount	-0.04	Very Strong negative	0.02	Medium
17	apparel	-0.04	Very Strong negative	0.02	Medium
18	community	-0.04	Very Strong negative	0.05	Strong
19	replicate	-0.04	Very Strong negative	0.01	Medium
20	mortgage	-0.04	Very Strong negative	0.01	Medium

Table 6.4: Top negative terms in the fuzzy polarity dictionary

	Medium	Strong	Very Strong	Very weak	Weak
Almost non-existent negative	52	35	45	48	42
Almost non-existent positive	66	38	46	93	58
Medium negative	55	71	66	29	22
Medium positive	48	59	71	27	28
Slight negative	50	46	64	82	65
Slight positive	33	44	69	25	28
Strong negative	69	52	24	38	49
Strong positive	60	50	36	26	23
Very Strong negative	37	28	10	38	24
Very Strong positive	50	38	24	43	40

Table 6.5: Distribution of Supporting labels by Polarity labels

6.5 Concluding Remarks

In this article, we introduced a novel approach to automatically extract a polarity dictionary using the the stock market as the reference domain in a fully automated way (no human intervention to define polarities required).

Our system identifies price changes of particular stocks over time, using them as a guiding polarity value. The magnitude of the price variation for a particular stock is then attributed to the financial news about this stock in corresponding period of time and that's what we use as our working corpus. Using this domain corpus as reference, we build the so called *binned corpus* and apply the TF-IDF algorithm to compute the TF-IDF value for each term obtaining the signed guiding polarities. We then disseminate these values within the neighbourhood of each term based on the embeddings-enabled cosine distance. Last but not least, we map the terms to fuzzy linguistic labels and provide a supporting indicator to indicate how reliable our scores are depending on its distribution of occurrences in the corpus.

To show how our approach works, we implement it for the Euro Stoxx 50 from January 2018 to March 2019, discuss the results and made the fuzzy polarity dictionary available.

Our approach solves 3 typical issues inherent to the classic approaches to building polarity dictionaries:

- The scale and thresholding problem, as we provide fuzzy linguistic sets instead of crisp polarity values.

- The human bias problem, as the polarity values are fully inferred without human intervention, providing on top an indicator on how reliable each particular polarity value is.

- The contextualization problem, as the polarity values are specific to our domain.

Further research work could focus on the impact of using of n-grams instead of mono-grams as well as the extension to further Part of Speech label (adverbs, etc). In addition, techniques to transfer the polarity dictionary to a different domain might also pave the way towards a multi-domain generic approach. Last but not least, we'd like to point to all the operationalization of the polarity dictionary to compute sentiment using fuzzy linguistic arithmetic operations.

Acknowledgments

This paper has been developed with the FEDER financing under Project TIN2016-75850-R

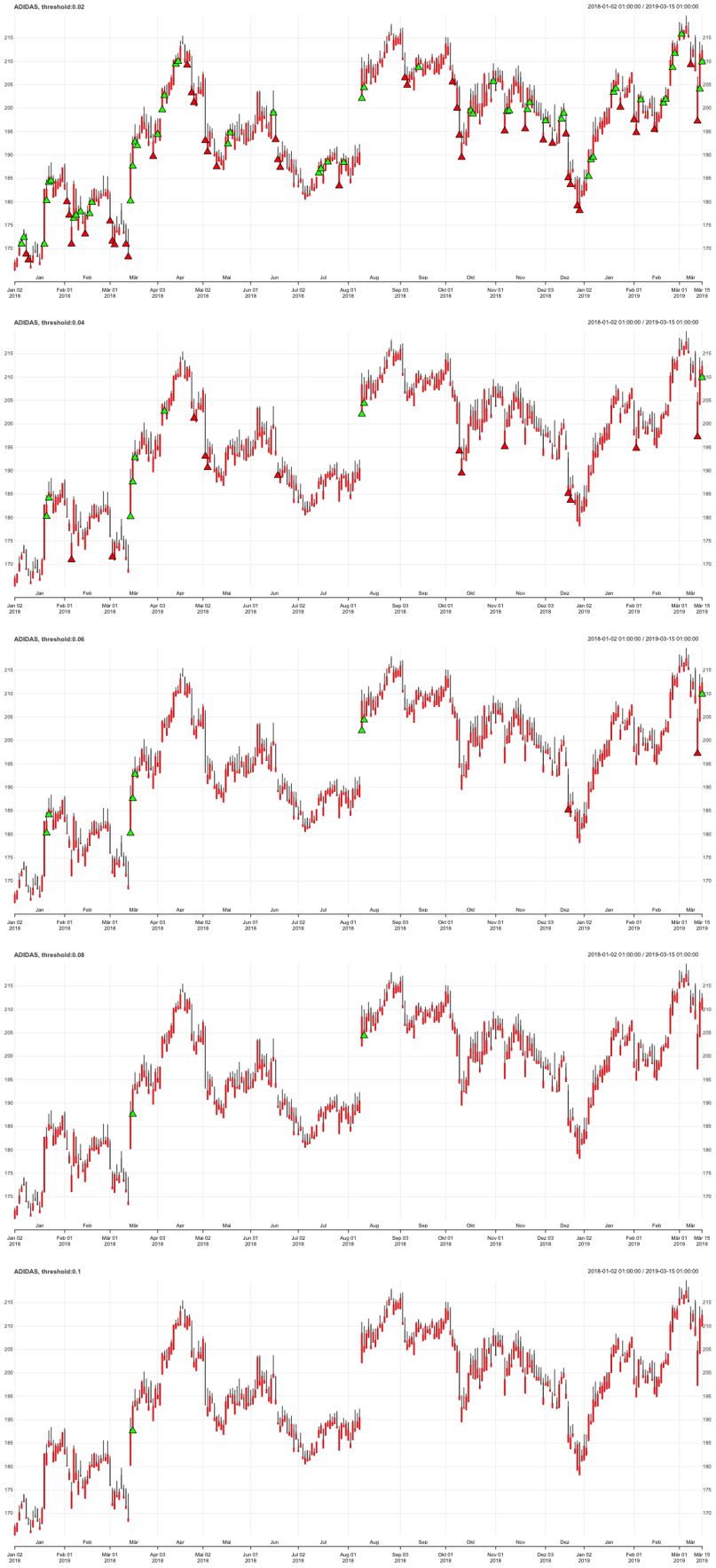


Figure 6.7: Adidas course from 2018 with different change thresholds (0.02, 0.03,0.04,0,08,0.1) indicating positive changes in green and negative ones in red

Bibliography

[1] Agathangelou, Pantelis; Katakis, Ioannis; Kokkoras, Fotios y Ntonas, Konstantinos: «Mining domain-specific dictionaries of opinion words». En: *International conference on web information systems engineering*, pp. 47–62. Springer, 2014.

[2] Appel, Orestes; Chiclana, Francisco; Carter, Jenny y Fujita, Hamido: «A hybrid approach to the sentiment analysis problem at the sentence level». *Knowledge-based Systems*, 2016, **108**, pp. 110–124.

[3] ——: «A Consensus Approach to the Sentiment Analysis Problem Driven by Support-Based IOWA Majority». *International Journal of Intelligent Systems*, 2017, **32(9)**, pp. 947–965.

[4] ——: «Cross-ratio uninorms as an effective aggregation mechanism in Sentiment Analysis». *Knowledge-Based Systems*, 2017, **124**, pp. 16–22.

[5] ——: «Successes and challenges in developing a hybrid approach to sentiment analysis». *Applied Intelligence*, 2018, **48(5)**, pp. 1176–1188.

[6] Araújo, Matheus; Gonçalves, Pollyanna; Cha, Meeyoung y Benevenuto, Fabrício: «iFeel: a system that compares and combines sentiment analysis methods». En: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 75–78. ACM, 2014.

[7] Baccianella, Stefano; Esuli, Andrea y Sebastiani, Fabrizio: «SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.» En: *LREC*, volumen 10, pp. 2200–2204, 2010.

[8] Bernabé-Moreno, Juan; Tejeda-Lorente, A; Porcel, Carlos; Fujita, Hamido y Herrera-Viedma, Enrique: «Emotional profiling of locations based on social media». *Procedia Computer Science*, 2015, **55**, pp. 960–969.

[9] Bernabé-Moreno, Juan; Tejeda-Lorente, Alvaro; Porcel, Carlos y Herrera-Viedma, Enrique: «A Fuzzy Linguistics Supported Model to Measure the Contextual Bias in Sentiment Polarity». En: *Advances in Fuzzy Logic and Technology 2017*, pp. 199–210. Springer, 2017.

[10] ——: «An Embeddings Based Fuzzy Linguistics Supported Model to Measure the Contextual Bias in Sentiment Polarity». En: *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain,*

26-28 September 2018, pp. 735–748, 2018. doi: 10.3233/978-1-61499-900-3-735. https://doi.org/10.3233/978-1-61499-900-3-735

[11] Bollen, Johan; Mao, Huina y Zeng, Xiaojun: «Twitter mood predicts the stock market». *Journal of computational science*, 2011, **2(1)**, pp. 1–8.

[12] Cambria, Erik: «Affective computing and sentiment analysis». *IEEE Intelligent Systems*, 2016, **31(2)**, pp. 102–107.

[13] Cambria, Erik; Fu, Jie; Bisio, Federica y Poria, Soujanya: «AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis». En: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, , 2015.

[14] Cambria, Erik; Poria, Soujanya; Bajpai, Rajiv y Schuller, Björn: «SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives». En: *the 26th International Conference on Computational Linguistics (COLING), Osaka*, , 2016.

[15] Cambria, Erik; Poria, Soujanya; Hazarika, Devamanyu y Kwok, Kenneth: «SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings». En: *Thirty-Second AAAI Conference on Artificial Intelligence*, , 2018.

[16] Cao, Di y Yu, Kai: «Deep Attentive Structured Language Model Based on LSTM». En: *International Conference on Intelligent Science and Big Data Engineering*, pp. 169–180. Springer, 2017.

[17] Chaturvedi, Iti; Satapathy, Ranjan; Cavallari, Sandro y Cambria, Erik: «Fuzzy commonsense reasoning for multimodal sentiment analysis». *Pattern Recognition Letters*, 2019, **125**, pp. 264–270.

[18] Chen, Kewen; Zhang, Zuping; Long, Jun y Zhang, Hao: «Turning from TF-IDF to TF-IGM for term weighting in text classification». *Expert Systems with Applications*, 2016, **66**, pp. 245–260.

[19] Daku, Mark; Soroka, Stuart y Young, Lori: «Lexicoder, version 2.0 (software)». *McGill University, Montreal, Canada*, 2011.

[20] Devitt, Ann y Ahmad, Khurshid: «Sentiment polarity identification in financial news: A cohesion-based approach». En: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 984–991, 2007.

[21] Dodds, Peter Sheridan y Danforth, Christopher M: «Measuring the happiness of large-scale written expression: Songs, blogs, and presidents». *Journal of happiness studies*, 2010, **11(4)**, pp. 441–456.

[22] Dridan, Rebecca y Oepen, Stephan: «Tokenization: Returning to a Long Solved Problem—A Survey, Contrastive Experiment, Recommendations, and Toolkit—». En: *Proceedings of the*

50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volumen 2, pp. 378–382, 2012.

[23] Hao, Xiaohong; Jia, Yifan y Gu, Qun: «An Automatic Construction Approach for Sentiment Dictionary Based on Weibo Emoticons». En: *2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018)*, Atlantis Press, 2018.

[24] Herrera, F. y Herrera-Viedma, E.: «Aggregation operators for linguistic weighted information». *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems*, 1997, **27**, pp. 646–656.

[25] Herrera, F. y Martínez, L.: «A 2-tuple fuzzy linguistic representation model for computing with words». *IEEE Transactions on Fuzzy Systems*, 2000, **8(6)**, pp. 746–752.

[26] ——: «A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making». *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 2001, **31(2)**, pp. 227–234.

[27] High, Rob: «The era of cognitive systems: An inside look at IBM Watson and how it works». *IBM Corporation, Redbooks*, 2012.

[28] Hu, Minqing y Liu, Bing: «Mining opinion features in customer reviews». En: *AAAI*, volumen 4, pp. 755–760, 2004.

[29] Hwang, Kai y Chen, Min: *Big-data analytics for cloud, IoT and cognitive computing*. John Wiley & Sons, 2017.

[30] Jianqiang, Zhao; Xiaolin, Gui y Xuejun, Zhang: «Deep convolution neural networks for twitter sentiment analysis». *IEEE Access*, 2018, **6**, pp. 23253–23260.

[31] Jockers, Matthew L.: *Revealing Sentiment and Plot Arcs with the Syuzhet Package*, 2015. `http://www.matthewjockers.net/2015/02/02/syuzhet/`

[32] Johnson, Rie y Zhang, Tong: «Semi-supervised convolutional neural networks for text categorization via region embedding». En: *Advances in neural information processing systems*, pp. 919–927, 2015.

[33] Joulin, Armand; Grave, Edouard; Bojanowski, Piotr; Douze, Matthijs; Jégou, Hérve y Mikolov, Tomas: «FastText.zip: Compressing text classification models». *arXiv preprint arXiv:1612.03651*, 2016.

[34] Kanayama, Hiroshi y Nasukawa, Tetsuya: «Fully automatic lexicon expansion for domain-oriented sentiment analysis». En: *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 355–363. Association for Computational Linguistics, 2006.

[35] Koneru, Anupriya; Bhavani, Nerella Bala Naga Sai Rajani; Rao, K Purushottama; Prakash, Garikipati Sai; Kumar, Immadisetty Pavan y Kumar, Velimala Venkat: «Sentiment Analysis on Top Five Cloud Service Providers in the Market». En: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 293–297. IEEE, 2018.

[36] Le, Quoc y Mikolov, Tomas: «Distributed representations of sentences and documents». En: *International Conference on Machine Learning*, pp. 1188–1196, 2014.

[37] Li, Xiaodong; Huang, Xiaodi; Deng, Xiaotie y Zhu, Shanfeng: «Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information». *Neurocomputing*, 2014, **142**, pp. 228–238.

[38] Li, Xiaodong; Xie, Haoran; Chen, Li; Wang, Jianping y Deng, Xiaotie: «News impact on stock price return via sentiment analysis». *Knowledge-Based Systems*, 2014, **69**, pp. 14–23.

[39] Liu, Bing: «Sentiment analysis and opinion mining». *Synthesis lectures on human language technologies*, 2012, **5(1)**, pp. 1–167.

[40] Ma, Yukun; Peng, Haiyun y Cambria, Erik: «Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM». En: *Thirty-Second AAAI Conference on Artificial Intelligence*, , 2018.

[41] Masulis, Ronald W y Shivakumar, Lakshmanan: «Does market structure affect the immediacy of stock price responses to news?» *Journal of Financial and Quantitative Analysis*, 2002, **37(4)**, pp. 617–648.

[42] Merello, Simone; Ratto, Andrea Picasso; Ma, Yukun; Oneto, Luca y Cambria, Erik: «Investigating timing and impact of news on the stock market». En: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1348–1354. IEEE, 2018.

[43] Mikolov, T; Chen, K; Corrado, G y Dean, J: «Efficient Estimation of Word Representations in Vector Space. Cornell University Library. 2013». *arXiv preprint arXiv:1301.3781*, 2016.

[44] Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg S y Dean, Jeff: «Distributed representations of words and phrases and their compositionality». En: *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[45] Nguyen, Thien Hai y Shirai, Kiyoaki: «Topic modeling based sentiment analysis on social media for stock market prediction». En: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volumen 1, pp. 1354–1364, 2015.

[46] Peng, Wei y Park, Dae Hoon: «Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization». En: *Fifth International AAAI Conference on Weblogs and Social Media*, , 2011.

[47] Pennebaker, James W; Francis, Martha E y Booth, Roger J: «Linguistic inquiry and word count: LIWC 2001». *Mahway: Lawrence Erlbaum Associates*, 2001, **71(2001)**, p. 2001.

[48] Pennington, Jeffrey; Socher, Richard y Manning, Christopher: «Glove: Global vectors for word representation». En: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[49] Sabour, Sara; Frosst, Nicholas y Hinton, Geoffrey E: «Dynamic routing between capsules». En: *Advances in neural information processing systems*, pp. 3856–3866, 2017.

[50] Sahlgren, Magnus: «The distributional hypothesis». *Italian Journal of Disability Studies*, 2008, **20**, pp. 33–53.

[51] Schmid, Helmut: «Improvements in part-of-speech tagging with an application to German». En: *In proceedings of the acl sigdat-workshop*, Citeseer, 1995.

[52] Schofield, Alexandra; Magnusson, Måns y Mimno, David: «Pulling out the stops: Rethinking stopword removal for topic models». En: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 432–436, 2017.

[53] Seng, Jia-Lang y Yang, Hsiao-Fang: «The association between stock price volatility and financial news–a sentiment analysis approach». *Kybernetes*, 2017, **46(8)**, pp. 1341–1365.

[54] Socher, Richard; Perelygin, Alex; Wu, Jean; Chuang, Jason; Manning, Christopher D; Ng, Andrew y Potts, Christopher: «Recursive deep models for semantic compositionality over a sentiment treebank». En: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[55] Tang, Duyu; Wei, Furu; Yang, Nan; Zhou, Ming; Liu, Ting y Qin, Bing: «Learning sentiment-specific word embedding for twitter sentiment classification». En: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volumen 1, pp. 1555–1565, 2014.

[56] Thelwall, Mike: «Heart and soul: Sentiment strength detection in the social web with sentistrength». *Proceedings of the CyberEmotions*, 2013, pp. 1–14.

[57] Wang, Xin; Liu, Yuanchao; Chengjie, SUN; Wang, Baoxun y Wang, Xiaolong: «Predicting polarities of tweets by composing word embeddings with long short-term memory». En: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volumen 1, pp. 1343–1353, 2015.

[58] Watson, David; Clark, Lee A y Tellegen, Auke: «Development and validation of brief measures of positive and negative affect: the PANAS scales.» *Journal of personality and social psychology*, 1988, **54(6)**, p. 1063.

[59] Zadeh, L.A.: «The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Part I, Information Sciences 8 (1975) 199-249, Part II, Information Sciences 8 (1975) 301-357, Part III, Information Sciences 9 (1975) 43-80», 1975.

[60] Zhang, Lei; Wang, Shuai y Liu, Bing: «Deep learning for sentiment analysis: A survey». *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018, **8(4)**, p. e1253.

[61] Zhao, Wei; Ye, Jianbo; Yang, Min; Lei, Zeyang; Zhang, Suofei y Zhao, Zhou: «Investigating capsule networks with dynamic routing for text classification». *arXiv preprint arXiv:1804.00538*, 2018.

Chapter 7

Trust Based Fuzzy Linguistic Recommender Systems as Reinforcement for Personalized Education in the Field of Oral Surgery and Implantology

In this chapter we include the following paper:

- Trust Based Fuzzy Linguistic Recommender Systems as Reinforcement for Personalized Education in the Field of Oral Surgery and Implantology.
  - Authors: C. Porcel, J. Herce-Zelaya, J. Bernabé-Moreno, A. Tejeda-Lorente, E. Herrera-Viedma.
  - Journal: International Journal of Computers Communications & Control, 15(3), article number: 3858, 2020. ISSN 1841-9844.
  - DOI: `https://doi.org/10.15837/ijccc.2020.3.3858`
  - Impact factor source: Web Of Science - Journal Ctitation Report
  - Impact factor: 2.293 (year 2020)
  - Category: Computer Science, Information System.
  - Quartile: Q3.
  - Ranking: 98 of 162.

Abstract

The rapid advances in Web technologies are promoting the development of new pedagogic models based on virtual teaching. In this framework, personalized services are necessary.

Recommender systems can be used in an academic environment to assist users in their teaching-learning processes. In this paper, we present a trust based recommender system, adopting a fuzzy linguistic modeling, that provides personalized activities to students in order to reinforce their education, and applied it in the field of oral surgery and implantology. We don't take into account users with similar ratings history but users in which each user can trust and we provide a method to aggregate the trust information. This system can be used in order to aid professors to provide students with a personalized monitoring of their studies with less effort. The results obtained in the experiments proved to be satisfactory.

Keywords: Recommender system, e-learning, fuzzy linguistic modeling, oral surgery.

7.1 Introduction

The great advances in Web technologies are promoting the development of new pedagogic models that complement the present education [11]. The new technologies improve the teaching-learning processes, aiding the information broadcasting in an efficient and easy manner, and providing tools for the personal and global communications that allow encouraging the collaborative learning [3, 19]. In this academic scope, personalized education [16] can be very helpful aiding students to reinforce the areas where it is necessary some help as well as maximizing those where they have potential. Education must also have the ability to adapt itself to the necessities of the student dynamically.

Recommender systems seek to discover information items that are valuable to the users. They may be considered personalized services because they have an independent profile for each user [2, 4, 20]. Therefore these systems need some information about every user, such as the ratings provided by the users about the analyzed items. This need for information introduces the requirement for the system to maintain users' profiles containing the users' preferences or needs. Another aspect to take into consideration is which additional information is required by the system, and how this information is processed and managed to generate a list of personalized recommendations. One of the mostly used methods to generate recommendations is the *collaborative approach* in which the recommendations provided to a particular user are based upon the ratings provided by those users with similar profiles. In this sense, we can use the inherent connectivity from an educational community to support collaborative approach, where students rate resources and these ratings are shared with a large community [3, 4].

One key disadvantage of this approach consists of the need of many ratings to obtain a good performance. But users typically provide just a few ratings, so the systems have difficulties to compute the similarity between two users [14]. Therefore, collaborative approaches tend to fail in generating recommendations since they usually fail at obtaining groups of users with similar preferences. Thus, some improvements need to be introduced to overcome this situation and one promising direction is to focus on *trust*, which plays a crucial role in on-line social networks [5], so widespread and popular today. People tend to rely upon recommendations

received from trusted users, such as friends, more than those generated by automatic systems [18]. In the literature, we can find some proposals about the incorporation of trust models in recommendation systems [9, 21]. In these systems, the recommendation engine uses the trusted network between users.

The aim of this article is to present a new fuzzy linguistic recommender system that incorporates the concept of trust in the recommendations generation engine. This new system is also adapted to students of Dentistry from the Dentistry School of the University of Granada [1] (Spain). The major innovations and contributions of the system include:

1. A method to estimate the trust score between two users, because the trusted network can be huge and most users do not know each other.

2. The provision of reliable personalized information by using a recommendation approach in which users with similar ratings history or pedagogical needs are not considered, but users in which each user can trust.

3. Its user-friendly nature, using a multi-granular fuzzy linguistic modeling to improve the representation of user preferences and facilitate user-system interactions [10, 12].

4. The ability to use it in any place and at any time, providing to students the necessary freedom to organize their schedules.

5. The reliability of the information offered and the selection of exercises, endorsed by a team of experts in oral surgery from the Dentistry School of the University of Granada.

The paper is structured as follows. In Section 7.2, the preliminaries are presented. Next in Section 7.3, we describe our proposal. Section 7.4 addresses the evaluation of the system and finally in Section 7.5 we throw our conclusions.

7.2 Preliminaries

7.2.1 Basis of recommender systems

Recommender systems try to guide the user in a personalized way towards suitable tasks among a wide range of possible options [2, 20]. In order to generate personalized recommendations that are tailored to the user's preferences or needs, recommender systems must collect personal preference information. Taking into account the knowledge source, different recommendations generation methods can be distinguished [1, 14]. Each approach has certain advantages and disadvantages, depending on the scope settings. One solution is to use a *hybrid strategy* combining different approaches in order to reduce the disadvantages of each one of them and to exploit their benefits[1]. Moreover, the recommendation activity is followed by a feedback phase in which the users provide the system with their satisfaction evaluations about the recommended items and the system uses these evaluations to automatically update user profiles.

[1] www.ugr.es

7.2.2 Fuzzy linguistic approach

The fuzzy linguistic approach is a tool based on the concept of linguistic variable proposed by Zadeh [22]. This theory has given very good results to model qualitative information and it has been proven to be useful in many problems.

In [6] is proposed a continuous model of information representation based on **2-tuple fuzzy linguistic modelling**. To define it both the 2-tuple representation model and the 2-tuple computational model to represent and aggregate the linguistic information have to be established. Let $S = \{s_0, ..., s_g\}$ be a linguistic term set with odd cardinality. We assume that the semantics of labels is given by means of triangular membership functions and consider all terms distributed on a scale on which a total order is defined. If a value $\beta \in [0, g]$, and $\beta \notin \{0, ..., g\}$, we can represent β as a 2-tuple (s_i, α_i), where s_i represents the linguistic label, and α_i is a numerical value expressing the value of the translation between numerical values and 2-tuple: $\Delta(\beta) = (s_i, \alpha)$ and $\Delta^{-1}(s_i, \alpha) = \beta \in [0, g]$ [6]. In order to establish the computational model negation, comparison and aggregation operators are defined. Using functions Δ and Δ^{-1}, any of the existing aggregation operators (such as arithmetic mean, weighted average operator or linguistic weighted average operator) can be easily be extended for dealing with linguistic 2-tuples without loss of information [6].

When different experts have different uncertainty degrees on the phenomenon or when an expert has to evaluate different concepts, several linguistic term sets with a different granularity of uncertainty are necessary. To manage this situation, in [7] a *multi-granular 2-tuple fuzzy linguistic modelling* based on the concept of linguistic hierarchy is proposed. A *Linguistic Hierarchy* LH, is a set of levels $l(t, n(t))$, where each level t is a linguistic term set with different granularity $n(t)$. In [7] a family of transformation functions between labels from different levels was introduced. To establish the computational model we select a level that we use to make the information uniform and thereby we can use the defined operator in the 2-tuple model. This result guarantees that the transformations between levels of a linguistic hierarchy are carried out without loss of information.

7.2.3 Trust networks

Trust networks are social networks in which users can explicitly assign trust scores to rate other users. But trust networks are usually very large and therefore a lot of users don't even know the vast majority of other users. For this reason, we need to use a method to estimate the trust degree between two users. The idea is to search for a path between the two users and propagate the trust degrees found along the path. Usually, we can find several paths between two users, so we may select the most relevant and aggregate the propagated trust degrees into the trust degree estimation. An upper path length limit is typically imposed what is known as *horizon*, H, and typical values for H are 2 or 3.

To aggregate the propagated trust degrees of the paths found, we use MILOWA, a majority guided linguistic operator [8]. With respect to the variable inducing the reordering of the set of values to be aggregated, in a trust network we work with information about reliability, so that we use the average global trust of all users of each of the founded path. To compute the global

trust of a user we use PageRank [13] because is one of the most widely used global trust metric.

7.3 Trust based recommender system to assist dentistry students in the field of oral surgery and implantology

In this section we present the Web system to assist students from the Dentistry School of University of Granada. Initially it is oriented to students from these subjects: *Oral Surgery I*, *Oral Surgery II* and *Implantology*. The system has three main components: videos and resources, student profiles and the method for generating recommendations.

7.3.1 Information representation

In order to allow for higher flexibility in the communication processes of the system, different label sets $(S_1,\ S_2, ...)$ are used are selected from among those that compose a LH, i.e., $S_i \in LH$. The different concepts assessed in the system are the following:

- *Degree of trust* of a student relative to another, which is labelled in S_1.
- The predicted *degree of relevance* of a resource for a student, which is labeled in S_2.
- The *degree of satisfaction* with a recommended resource expressed by a student, which is labeled in S_3.
- *Membership degree* of a resource scope or student needs with respect to each of the defined reinforcing subgroup, which is labelled in S_4.

We use 5 labels to represent the degrees of trust, satisfaction and membership to reinforcing subgroup ($S_1 = S^5$, $S_3 = S^5$ and $S_4 = S^5$) and 9 labels to represent the predicted relevance degrees ($S_2 = S^9$). The linguistic terms in each level are the following ones:

- $S^5 = \{b_0 = None = N,\ b_1 = Low = L,\ b_2 = Medium = M,\ b_3 = High = H,\ b_4 = Total = T\}$
- $S^9 = \{c_0 = None = N,\ c_1 = Very\_Low = VL,\ c_2 = Low = L,\ c_3 = More\_Less\_Low = MLL,\ c_4 = Medium = M,\ c_5 = More\_Less\_High = MLH,\ c_6 = High = H,\ c_7 = Very\_High = VH,\ c_8 = Total = T\}$

7.3.2 Resource representation

A multimedia database was developed and contained videos with a wide set of different oral surgeries or implants for all possible needs inside the subjects covered. Also, different sets of scientific papers or class notes are introduced into the system. All videos, papers and notes can be combined among different subgroups, called activities, in the construction of a customized program for each student. Videos were recorded on real surgeries produced in dentistry's offices or in university's labs. Those activities are the items to be recommended by our system. Each

combination of videos, notes or papers make the different activities suitable for a student with a specific pedagogical need.

Once a teacher creates a new activity into the system, he/she provides the activity with an internal representation that is mainly based on its appropriateness for each reinforce subgroup. An activity i is represented as a vector $VR_i = (VR_{i1}, VR_{i2}, ..., VR_{i4})$, where each component $VR_{ij} \in S_4$ is a linguistic assessment that represents how appropriate is the activity i with respect to the reinforcing subgroup j.

7.3.3 Student profiles

The student profiles are represented by three components: their needs, their degrees of trust in other students and the satisfaction degrees with the recommended resources. Students must complete their profiles with the grades obtained in previous subjects related with this oral surgery and implantology. They have to periodically carry out different test to be able of evaluate their abilities. After obtaining the test results, the teachers assess the membership of the student need in each one of the four reinforcing subgroups. A student i is represented as a vector $VS_i = (VS_{i1}, VS_{i2}, ..., VS_{i4})$, where each component $VS_{ij} \in S_4$ is a linguistic assessment that represents the degree of how appropriate i is for each reinforcing subgroup j. Since student are performing test over the whole semester, their membership to the different subgroup will be changing together with their new results. Besides, the students explicitly specify their degree of trust on other students using the level S_1 of the LH, i.e., using one the 5 labels of S^5. Finally, as students receive recommendations they are asked to assess their satisfaction with the recommendations ($rc \in S_3$).

7.3.4 Recommendation approach

As a large number of users have not supplied the trust degrees to many other users, to generate recommendations, we need a method to estimate the trust degree between users. Then, in order to estimate the level in which a user u trust in other user v, $\tau_{u,v}$, the MLIOWA operator (see Section 7.2.3) is applied to aggregate the global trust of all users found in the several paths between u and v, according to the majority [8].

Then, if we wish to estimate or upgrade the relevance of a item i for a user u, the following steps are performed:

1. Identify the set of trusted users of u, Γ_u. To do that, we estimate the trust between u and all other users taking into account the selected horizon, i.e. $\tau_{u,v}\ \forall v \in \Upsilon$ with $v \neq u$ and Υ the set of users. As $S_1 = S^5$, we consider that the user v is a trusted user of u if the trust degree is higher than the mid linguistic label.

2. To recovery the assessments provided by the trusted users of u over the item i, i.e., the linguistic satisfaction assessments $sat(y, i) \in S_3, \forall y \in \Gamma_u$.

3. The item i is recommended to u with a predicted relevance degree $p_{rel}(u, i) \in S_2 \times [-0.5, 0.5]$ which is calculated as follows:

$$p_{rel}(u,i) = \overline{x}_l^w((TF_{S_2}^{S_3}(sat(y_1,i),0), TF_{S_2}^{S_1}(\tau_{u,y_1})), \ldots, (TF_{S_2}^{S_3}(sat(y_n,i),0), TF_{S_2}^{S_1}(\tau_{u,y_n}))), \tag{7.1}$$

where $y_1, \ldots, y_n \in \Gamma_u$, $\overline{x}_l^w$ is the linguistic weighted average operator and $TF_{t'}^t$ is the transformation function between a 2-tuple that belongs to level t and another 2-tuple in level $t' \neq t$.

7.4 System evaluation

7.4.1 Validating the system utility

In the first place, we performed a practical study where a group of 50 volunteers students tested the system during one semester in different subjects: *Oral surgery I*, *Oral surgery II* and *Implantology I*. Then, we evaluated the utility of the system based on the results obtained by the students and the feedback provided. After one semester of usage, the results of the students who used the system were in average a 15.5% better than the users that only assist to lessons. The linguistic weighted average of the feedback provided by users on the improvement received by the recommendations were $(b_0^5, 7)$, that is between medium and high. Therefore, the results demonstrate that the website is not only positively perceived by its users but also it increases their results compared to the ones who did not use it.

7.4.2 Recommendation approaches evaluation

We also validated the proposed approach with off-line tests, where the Epinions dataset [2] [9] was used. Due to performance reasons, we reduced the Epinion dataset containing a sample of the first 1000 users and 2000 items. The ratings subset for this reduced subset still contained 7841 ratings and 52548 trust statement values.

Experiments description

To develop the experiments we implemented the approach considering different values for the horizon, i.e., $H = 2$, $H = 3$ and $H = 4$ (called *Trust-H2*, *Trust-H3* and *Trust-H4*). In order to compare the results, we also implemented some collaborative approaches. We implemented both *item-based* and *user-based* approaches [1] with different configurations. We've also analysed the impact of having a different number of neighbours on the similarity computation, achieved by using different values for the variable K, which represents the most similar k-users. Specifically, we used following values for K: 10, 20 and 50. In addition, we combined these values with two different similarity metrics, such as *Cosine* and *Pearson*. We name *Col-It-* or *Col-Us-* the algorithms *item* and *user* based respectively; likewise we added to the algorithm name the suffixes Cs or Ps for Cosine or Pearson metrics; the last suffix corresponds to the

[2] http://www.trustlet.org/wiki/Epinions

number of neighbours ($K10$, $K20$ or $K50$). Then, we carried out a *5-fold cross validation* process [15] and to measure the accuracy, we adopted the *Mean Absolute Error (MAE)* [17].

On the other hand, we also analysed the *coverage* achieved with each approach [17], i.e., the proportion of ratings of the validation set the system can generate a prediction for, classifying users and items into different types [9]. The users are classified in these types: *cold start users* who provided from 1 to 4 ratings, *heavy raters* who provided more than 10 ratings, *opinionated users* who provided more than 4 ratings and whose standard deviation is greater than 1.5, *black sheep users* who provided more than 4 ratings and for which the average distance of their rating on corresponding item with respect to mean rating of the item is greater than 1. The items are classified in: *niche items* which received less than 5 ratings and *controversial items* which received ratings whose standard deviation is greater than 1.5.

Experiments results

Figure 7.1 shows respectively the MAE and coverage for cosine/Pearson similarity measure for item-based and users-based collaborative approaches for the different user groups. The MAE obtained is similar for all combinations (except for small differences), but the coverage is better with the user-based collaborative approach. In both cases, the higher the number of neighbours, the better the results, especially in coverage.

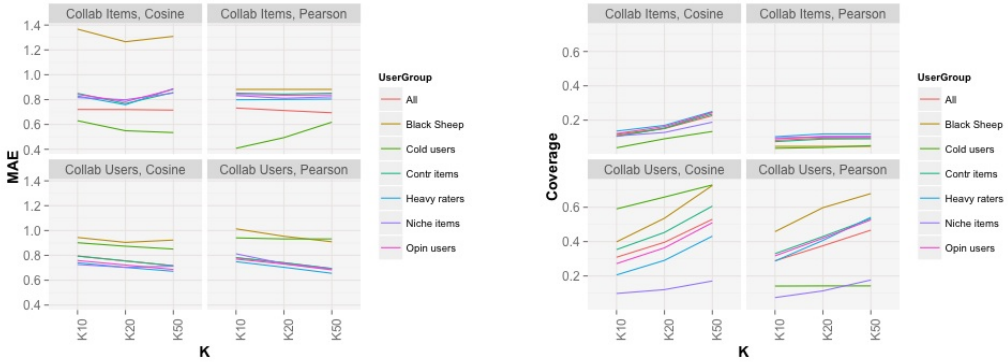


Figure 7.1: MAE and Coverage obtained with differents configurations of collaborative approach.

Now we analyse the results obtained with the approach based on trust. Figure 7.2 shows respectively the MAE and coverage obtained with the new proposal for different horizons. These figures show that a higher horizon value does not guarantee better results in MAE terms (in fact, the better MAE is obtained with $H = 3$), but in coverage. Moreover, a higher horizon value penalizes the time to results as it implies much higher execution time.

Figure 7.3 show the results of the comparison. We can see that the better MAE is obtained with item-based collaborative implementation for cold users, but only for very specific situations. However, in general terms, we see that the new proposal based on trust clearly outperforms the other approaches; specifically, we have achieved an improvement of 2.71%. But, the best results of our proposal manifest in terms of coverage because it outperforms the other methods.

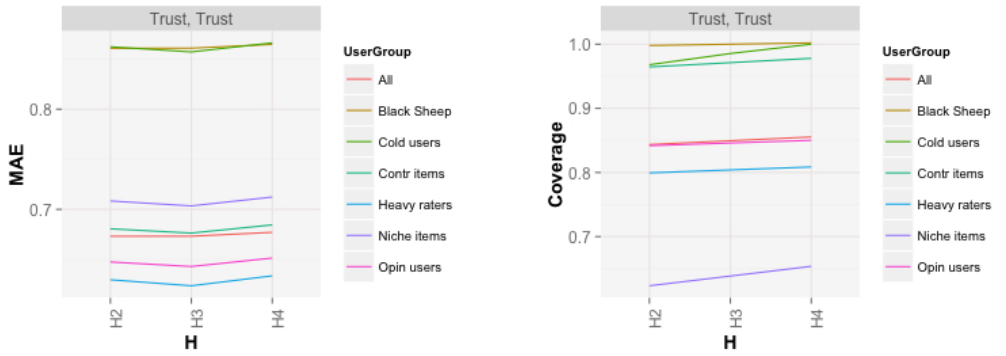


Figure 7.2: MAE and Coverage obtained with our suggested approach for differents horizons.

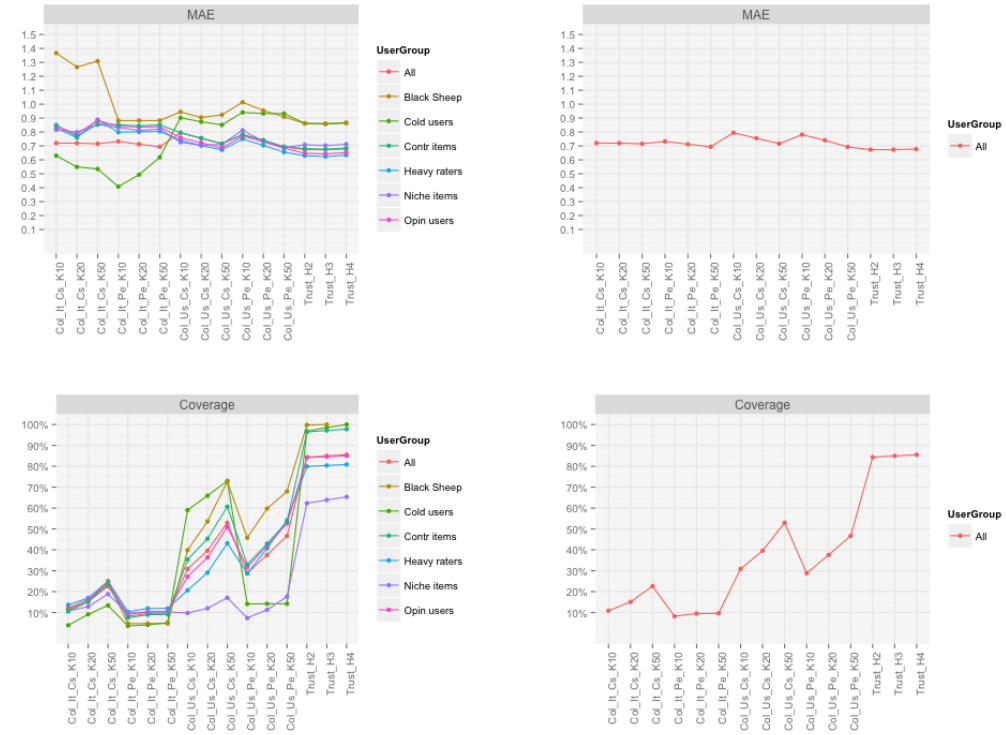


Figure 7.3: Comparison of MAE and coverage between new and the previous schemes.

7.5 Concluding remarks

In this paper, we present a trust based fuzzy linguistic recommender system, to provide personalized activities to students in order to reinforce their individualized education. The main idea consists of not taking into account students with similar requirements or rating history, but rather trustworthy students. To achieve this, we have proposed a method to estimate the trust score between a pair of students. This system is applied in a real environment, providing personalized activities to students in the subjects Oral surgery I and II and Implantology II of Dentistry degree in the University of Granada (Spain). The main benefits of this system are the increase of the personalize degree of the education received by the students that also have the

possibility of following the activities anywhere and anytime. We have evaluated the proposal, and the experimental results demonstrate the good results of the usage of the system as well as the perception by the students by enhancing the effectiveness of professors dealing with large group of students.

As future work, we consider to study the possibility of automatize the creation of activities by the system, based on individual feedback provided by the student of each component of the activities, as well as let the students create their own activities. Other proposal might be to focus on applying specific measures of the social networks analysis, exploiting the information represented in the trust network.

Acknowledgments

This paper has been developed with the FEDER financing under Project TIN2016-75850-R.

Bibliography

[1] Burke, R.: «Hybrid Web Recommender Systems». *P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS*, 2007, **4321**, pp. 377–408.

[2] Burke, R.; Felfernig, A. y Göker, M.H.: «Recommender systems: An overview». *Artificial Intelligence Magazine*, 2011, **32**, pp. 13–18.

[3] Dascalu, M.I.; Bodea, C.N.; Moldoveanu, A.; Mohora, A.; Lytras, M. y Ordoñez de Pablos, P.: «A recommender agent based on learning styles for better virtual collaborative learning experiences». *Computers in Human Behavior*, 2015, **45**, pp. 243–253.

[4] Goga, M.; Kuyoro, S. y Goga, N.: «A recommender for improving the student academic performance». *Procedia - Social and Behavioral Sciences*, 2015, **180**, pp. 1481–1488.

[5] Golbeck, J.A.: *Computing and applying trust in web-based social networks*. Tesis doctoral, 2005.

[6] Herrera, F. y Martínez, L.: «A recommender for improving the student academic performance». *A 2-tuple fuzzy linguistic representation model for computing with words*, 2000, **8**, pp. 746–752.

[7] ——: «A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making». *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 2001, **31(2)**, pp. 227–234.

[8] Herrera-Viedma, E.; Pasi, G.; López-Herrera, A.G. y Porcel, C.: «Evaluating the Information Quality of Web Sites: A Qualitative Methodology Based on Fuzzy Computing With Words». *Journal of the American Society for Information Science and Technology*, 2006, **57(4)**, pp. 538–549.

[9] Massa, P. y Avesani, P.: *Computing with Social Trust*. capítulo Trust metrics in recommender systems, pp. 259–285. Springler, 2009.

[10] Mata, F.; Martínez, L. y Herrera-Viedma, E.: «An Adaptive Consensus Support Model for Group Decision Making Problems in a Multi-Granular Fuzzy Linguistic Context». *IEEE Transactions on Fuzzy Systems*, 2009, **17(2)**, pp. 279–290.

[11] Money, W.H. y Dean, B.P.: «Incorporating student population differences for effective online education: A content-based review and integrative model». *Computers & Education*, 2019, **138**, pp. 57–82.

[12] Morente-Molinera, J.A.; Pérez, I.J.; Ureña, R. y Herrera-Viedma, E.: «On multi-granular fuzzy linguistic modelling in group decision making problems: a systematic review and future trends». *Knowledge Based Systems*, 2015, **74**, pp. 49–60.

[13] Page, L.; Brin, S.; Motwani, R. y Winograd, T.: «The pagerank citation ranking: Bringing order to the web». *Informe técnico*, Technical report, Stanford, USA, 1998.

[14] Porcel, C.; Ching-López, A.; Lefranc, G.; Loia, V. y Herrera-Viedma, E.: «Sharing notes: An academic social network based on a personalized fuzzy linguistic recommender system». *Engineering Applications of Artificial Intelligence*, 2018, **75**, pp. 1–10.

[15] Refaeilzadeh, P.; Tang, L. y Liu, H.: «Cross-Validation», 2008. `http://www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf`

[16] Segal, D.; Gal, K.; Shani, G. y Shapira, B.: «Sharing notes: An academic social network based on a personalized fuzzy linguistic recommender system». *A difficulty ranking approach to personalization in E-learning*, 2019, **130**, pp. 261–272.

[17] Shani, G. y Gunawardana, A.: *Recommender Systems Handbook*. capítulo Evaluating Recommendation Systems, pp. 257–298. Ricci, F. and Rokach, L. and Shapira, B. and Kantor, P.B. Eds. (Springer), 2011.

[18] Sinha, R.R. y Swearingen, K.: «Comparing Recommendations Made by Online Systems and Friends». En: *In DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, pp. –1–1, 2001.

[19] Tan, H.C.: «Using a structured collaborative learning approach in a case-based management accounting course». *Journal of Accounting Education*, 2019, **49**, p. 100638.

[20] Tejeda-Lorente, A.; Porcel, C.; Peis, E.; Sanz, R. y Herrera-Viedma, E.: «A quality based recommender system to disseminate information in a University Digital Library». *Information Science*, 2014, **261**, pp. 52–69.

[21] Victor, P.; Cornelis, C.; DeCock, M. y Pinheiro da Silva, P.: «Gradual trust and distrust in recommender systems». *Fuzzy Sets and Systems*, 2009, **160(10)**, pp. 1367–1382.

[22] Zadeh, L.A.: «The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Part I, Information Sciences 8 (1975) 199-249, Part II, Information Sciences 8 (1975) 301-357, Part III, Information Sciences 9 (1975) 43-80», 1975.

Chapter 8

Introducing CSP dataset, a dataset optimized for the study of the cold start problem in recommender systems

In this chapter we include the following paper:

- Introducing CSP dataset, a dataset optimized for the study of the cold start problem in recommender systems.
  - Authors: J. Herce-Zelaya, C. Porcel, A. Tejeda-Lorente, J. Bernabé-Moreno, E. Herrera-Viedma.
  - Journal: Information, 2023, 14, 19. Special Issue on "Information Retrieval, Recommender Systems and Adaptive Systems". ISSN: 2078-2489
  - DOI: `https://doi.org/10.3390/info14010019`
  - Impact factor source: Journal Ctitation Reports - Journal Citation Indicator (JCI)
  - Impact factor: 0.62 (year 2021)
  - Category: Computer Science, Information System.
  - Quartile: Q3.
  - Ranking: 128 of 246.

Abstract

Recommender systems are tools that help users in the decision-making process of choosing items that may be relevant for them among a vast amount of other items. One of the main problems of recommender systems is the cold start problem, which occurs when either new items or new users are added to the system and, therefore, there is no previous information about them. This article presents a multi-source dataset optimized for the study and the alleviation of the cold start problem. This dataset contains info about the users, the items

(movies), and ratings with some contextual information. The article also presents an example user behavior-driven algorithm using the introduced dataset for creating recommendations under the cold start situation. In order to create these recommendations, a mixed method using collaborative filtering and user-item classification has been proposed. The results show recommendations with high accuracy and prove the dataset to be a very good asset for future research in the field of recommender systems in general and with the cold start problem in particular.

Keywords: Recommender systems, datasets, cold start problem, new user problem.

8.1 Introduction

Nowadays, recommender systems play the role of experts on a matter, helping users find items that are tailored to their tastes. Most recommender systems use previous user information to generate recommendations [5, 17]. For example, the recommender systems may use the previous user's rating of similar items to create the recommendation for similar items (Content-based methods) or they may find similar users, take items that those users rated most positively, and then recommend those items to the target user (Collaborative filtering).

However, sometimes there are no previous data from the user because the user is new to the system and, therefore, due to the lack of data, it is not possible to generate tailored recommendations for the user. This situation is widely known in the recommender systems, and it is called the cold start problem [2, 11, 26]. The most common methods to palliate this problem are either asking users about their interests and tastes or asking the users directly to rate some items in order to have some data on which the recommender systems will be based. The main drawback of these methods is that they require time and effort from the user, and, therefore, those methods are ignored.

The methods proposed in this work take a slightly different approach: instead of asking users to provide explicit data, the recommender system is taking implicit data, so the user does not have to actively provide any information. To be more precise, these implicit data will be taken from the user's social media stream [8, 10, 18, 6]. These data will be used to generate a user profile with a series of features that can eventually be used to classify the users and, therefore, create predictions building a recommender system.

One of the main problems when building algorithms that aim to alleviate the cold start problem is that it is difficult to find a dataset that has a rich user profile that can be used to overcome the problem. Although currently, it has become standard to use Movielens (`https://grouplens.org/datasets/movielens/`) to evaluate and benchmark the recommender system models, when it comes to the case of cold start problem, there is not a clear dataset that would allow the evaluation of a user-feature-aware algorithm design as described above.

In this work will be introduced a new dataset optimized to be used for alleviating the cold

start problem. This dataset has three tables:

- Movies: contains info about the items (movies in this case) that can be used to extract features out of it.
- User profiles: contains info about the users. Some of them are already feature like, and others can be used to create features out of it.
- Ratings: contains the ratings from the users for the items.

This dataset has been crafted using data from two sources: Filmaffinity (`https://www.filmaffinity.com`, accessed on 19 April 2022) for the ratings and movies tables and Twitter (`https://twitter.com`, accessed on 19 April 2022) for the user profile one. Due to the duality of the dataset (user and item data), this dataset is an optimal asset that can be leveraged to create models for recommender systems under the cold start problem situation, making it easier to create connections between the user profile and item ratings.

Although nowadays, there are several public datasets available for recommender systems, as indicated in Section 8.2.2, these datasets lack quality user data, such as the behavioral user data CSP is providing. Some of these datasets have some demographic data about the users, such as sex or age (this is the case of the LDOS–CoMoDa dataset), but the rest of the variables are either item variables or contextual data (i.e., date of the rating, weather at the time of the rating). This contextual data can be a good asset but the usage of user-related features is a requirement to obtain more tailored recommendations. Therefore, CSP is an optimal dataset for cold start situations since it provides up to 12 behavioral variables that enable much more powerful and accurate decision-making models that leverage implicit data from users, which is a key aspect of the models that aim to alleviate the cold start problem.

This work also describes an example of the usage of this dataset by performing some data cleaning, feature selection, and the design and evaluation of a recommender system model that uses this dataset.

This model follows a mixed approach between collaborative filter and content based. The evaluation of the model proves that the model is very accurate.

The rest of this work is organized as follows. Section 8.2 shows the main concepts of the recommendation systems specifying solutions for cold start problems. Section 8.3 provides a description of the dataset and the designed algorithm. Section 8.4 provides the results from the previously mentioned model. Section 8.5 provides the conclusion of this work.

8.2 Background

8.2.1 Recommender Systems

Nowadays, recommender systems are present on nearly every website since the content on the web is increasingly growing and, therefore, the decision-making process is more difficult. These recommender systems aim to assist the user in the decision-making process and provide users with items that might be of interest for them [3, 5].

This process is replacing the classic expert recommendation but with two main differences: there is no expert needed for the creation of the recommendations since the process is fully automated and the recommendations are tailored to the user's taste.

Recommender systems are mainly divided into two groups: content-based algorithms and collaborative filtering.

Content-Based Algorithms

Content-based filtering utilizes the features of the items to be able to recommend similar items to the items the user has positively rated or interacted with.

In [5], the above-explained approach is described and shows its various usages in different domains.

Collaborative Filtering

In order to address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items to create recommendations. This enables recommendations based on the ratings of similar users.

In [4], the problem of online and interactive collaborative filtering is considered focusing on finding out the query that maximizes the quality of the created recommendations.

8.2.2 Datasets for Recommender Systems

The most widely used dataset for recommender systems is the Movielens dataset (`https://grouplens.org/datasets/movielens/25m/`), which contains many ratings for movies. Another very popular dataset is the Jester dataset (`https://goldberg.berkeley.edu/jester-data/`), which is a set of anonymous ratings from jokes. The Netflix Prize (`https://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a`) dataset is also popular in the recommender system scientific area and has been used in many scientific studies. In [1], the Netflix challenge is described, and related work and efforts are reviewed, summing up the progress made so far. The LDOS-CoMoDa dataset (`https://www.lucami.org/en/research/ldos-comoda-dataset/`) is a context-rich dataset that is often used for algorithms that try to alleviate the cold start problem. The dataset offers information about the context in which the ratings were provided.

Even though all the previous datasets are often used for models that aim to solve the cold start problem, these datasets do not contain significant contextual data or user data. The only exception could be the LDOS-CoMoDa dataset which has rich contextual data. Nevertheless, they do not provide any user-related data, apart from a couple of demographic variables (i.e., sex and age).

8.2.3 Cold Start Problem

The predictions from recommender systems, in most cases, are fetched from previous ratings from the user or from ratings from similar users. The cold start problem occurs when either new items or users are added to the system, and then it is not possible to create the predictions.

The cold start problem can appear in two situations:

- New users cold start problem [2]: It occurs when a new user joins the system, and then there is no data provided to the system, and the recommender can not provide any recommendation since the user can not be compared to any user nor the system can find similar items to the actual items liked by the user due to the lack of ratings from the user.

- New items cold start problem [21, 25]: It occurs when a new item is added to a system. Since there are no ratings for the item, it can not be recommended to anyone using the collaborative filtering approach because nobody has rated it yet.

There are many approaches in the literature that have alleviated this problem to some extent [11, 18].

Regarding the new users' cold start problem, in [14], a comparative study from different approaches is exposed, some of which will be discussed here. In particular, in [2], the authors present a new optimized similarity measure through neural networks. In [24], an approach with association rules, probability-based metrics, and own users' context are exposed in order to solve the cold start problem. In [13], a probabilistic model based on rules is used. In [15], a study is presented in which classification algorithm C.4.5 and Naive Bayes is leveraged with diverse similarities and prediction techniques in order to alleviate the problem. In [7], an approach that uses a trust and distrust network to find trustworthy users and use the preferences of these users to create recommendations. In the concrete case of using social network information from the user to palliate this problem, more recently, in [11], a revision is presented about how it has been working precisely with information extracted from social networks, studying some published articles between 2011 and 2017. Further, in [12], a new technique is presented using data extracted from social networks in order to create similarities between users and, afterward, create recommendations for alleviating the cold start problem.

On the other side, in order to deal with the new items cold start problem, the number of academic work is not as extensive, although it is worth citing two interesting articles. In [25], the authors present a system in which they obtain item features using deep learning, and these features are leveraged by incorporating them in a collaborative schema. Additionally, in [21], the user's created tags and features from the items are leveraged for creating matrix factorization, which is utilized in order to generate the knowledge. There are also studies in the literature regarding the idea of using information from social media streams and converting it to valuable data to create recommendations. In [26], a method is introduced where temporal data and social relations are leveraged to alleviate the cold start problem by utilizing Markov Chains. In [10], an ontology-based advertisement recommendation system that uses the data produced by users in social media is suggested. In [8], the social context is used to create recommendations for tourist attractions based on the similarity of users.

In a more recent study [19], the works from the last decade are reviewed, covering the different techniques that have been used in order to palliate the cold start problem. In [20], a collaborative filtering method is presented where they connect users with few ratings with other users with more ratings and create the recommendations accordingly. Another interesting systematic literature review in [23] presents state-of-the-art publications and techniques about

the cold start problem, and they stress the lack of rich datasets for working on alleviating the cold start problem. In [16], the authors propose a comprehensive autoencoder-based approach to handle both the cold and warm start problem, making use of ratings of users on items as well as metadata from users and items. In [9], the authors present a method that combines Probabilistic Matrix Factorization and a pairwise ranking-oriented approach of Bayesian Personalized Ranking.

8.3 Materials and Methods

This section will present the crafted dataset, explain the sources as well as the extraction methods, and a use case of leverage of this dataset for supporting decision-making by showing a recommender system model. The code used for performing the data cleaning, aggregation, and creating the models can be found in a Jupyter notebook inside this Github repository (`https://github.com/lynchblue/csp-dataset-in-action`, accessed on 19 April 2022).

8.3.1 Dataset

The dataset has been hosted in Github (`https://github.com`) in this public repository (`https://github.com/lynchblue/movie-rating-dataset`, accessed on 19 April 2022). The URL of the repository is: "https://github.com/lynchblue/movie-rating-dataset".

Dataset Sources

The dataset has been extracted from Filmaffinity and Twitter. From Filmaffinity, we have extracted the items table (movies.csv (`https://raw.githubusercontent.com/lynchblue/movie-rating-dataset/main/data/movies.csv`, accessed on 19 April 2022)) and the ratings table (ratings.csv (`https://raw.githubusercontent.com/lynchblue/movie-rating-dataset/main/data/ratings.csv`, accessed on 19 April 2022)). From Twitter, we have extracted the user table (user_profiles.csv (`https://raw.githubusercontent.com/lynchblue/movie-rating-dataset/main/data/user_profiles.csv`, accessed on 19 April 2022)). The main reason for choosing these data sources is the possibility of mapping rating data from the user and data about the user itself (that can be leveraged to elaborate a user profile) very easily, since, within the Filmaffinity portal, on the profile page, there is an option for the users to add their Twitter account.

Filmaffinity, a Database Movie Portal

Filmaffinity is a web portal that serves as a movie database where users can check many details about every movie, such as the year of publication, title, genre, cast director, or writer. The users can also provide ratings for the movies, write reviews for the movies, create lists, and also check other users' ratings, among many other features.

As mentioned before, the tables for items and ratings are extracted from the Filmaffinity portal. Due to the lack of a public API, all the information has been fetched with the leverage of Web Scraping with Python (`https://www.python.org/doc/`) and the lxml library (`https://lxml.de/`).

Web scraping is a technique for extracting data from a website. It consists on programmatically calling websites, parsing their content and inspecting the elements in the Dom, using

locators (i.e., through ids, classes or xpath) in order to identify the elements that are relevant, and lastly, storing this information in some files or databases.

Next, the fetching process for the data from the Filmaffinity platform will be described in detail:

Step 0: Fetch all users

Since there is no single page where all the users are listed, the process of fetching users is iterated over a big set of movies from the Top Filmaffinity (`https://www.filmaffinity.com/es/topgen.php`) page. For every movie, the rating page will be open, and from there, the users that have provided a review for this movie can be seen so the scraper can collect the user ids.

Note that this has a drawback since only users that have provided at least one review can be collected.

Step 1: Decide which users to use

Since now the user ids are collected, the Filmaffinity profile page (`https://www.filmaffinity.com/es/userratings.php?user_id=333743`, accessed on 19 April 2022) from the users can be visited for every one of these users. On this profile page, the users have the option to add links to their blogs and social media profiles (Twitter and Facebook). Then all the users that do not have their Twitter profile provided in their Filmaffinity profile page will be filtered out since, for them, there is no possibility to map social streams with rating data and, therefore, the rating data has no interest for the purposes of the study.

As part of this process, the URL of the Twitter profile is gathered in order to be able to fetch data from the Twitter social stream in the coming steps.

Note that the fact that only users that have provided a Twitter URL in their profile are considered again reduces the number of users that could be used.

Step 2: Fetch all rating data from the selected users

On the Filmaffinity profile page of every user, all the ratings that the user has ever provided are listed and paginated. Therefore, the scraper will iterate over every page, and inside, the page will iterate over all movies rated and will fetch the rating for every movie.

Step 3: Gather item data for the movies table

After all the ratings have been retrieved for every relevant user, the scraper will iterate over all movies in the rating table and then will navigate to the movie page (`https://www.filmaffinity.com/en/film682814.html`, accessed on 19 April 2022) and from there, all the required info will be fetched.

Twitter, a Social Media Platform

Twitter is a social networking portal where users communicate in short messages that are called tweets. These tweets can contain up to 280 characters (until late 2018, the maximum allowed size was 140 characters). Users can follow other users. In the user's timeline, the user will see all the tweets from the users the user is following. Users can retweet other users' tweets, and then these tweets will appear in the user's profile and in the user's followers' timelines.

All the Twitter-related data are fetched with Python through the Twitter public API (`https://developer.twitter.com/en/docs/twitter-api`, accessed on 19 April 2022).

Since in the previous step all the Twitter profile's URLs were gathered, now this list of URLs can be used for fetching user's related data from Twitter.

Step 1: Fetch info from every user

The overall information about every user is fetched through Twitter's Users lookup API (`https://developer.twitter.com/en/docs/twitter-api/users/lookup/introduction`). This overall information encompasses data, such as the number of likes, number of followers, or year's account creation.

Step 2: Fetch tweets from every user

For fetching new tweets, the Tweets lookup method (`https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/introduction`) has been used.

Step 3: Process these tweets and create a user profile

Now that all the raw information has been gathered, these data can be aggregated and processed to generate features for the user profile.

For example, using the date of the tweets from the users, some time-related features can be created. For example, there is a feature called *night_owl*, which is set to true if the user usually writes tweets during night-time.

Dataset Description

As previously mentioned, the dataset is hosted in Github, in this public repository (`https://github.com/lynchblue/movie-rating-dataset`, accessed on 19 April 2022). The dataset has three tables that are stored in CSV format (movies.csv, ratings.csv and user_profile.csv) that are described as follows:

Item Table or Movies.csv

This table contains info about the items (movies in this case), and the fields included are the following:

- id: unique identifier of the movie.
- main_title: title of the movie in Spanish.

- year_published: year the movie was released.
- duration: movie duration in minutes.
- country_name: name of the country of the movie.
- country_code: code for the country of the movie (i.e., ES for Spain).
- original_title: original title of the movie.
- directors: name/s of the director/s of the movie (separated by the "|" character).
- actors: cast of the movie (separated by the "|" character).
- genres: genre/s of the movie (i.e., Thriller).
- plot: the plot of the movie (in Spanish).
- script: writer/s of the movie's script (separated by the "|" character).
- producer: producer/s of the movie (separated by the "|" character).
- music: music composer/s (separated by the "|" character).
- photography: director/s of photography (separated by the "|" character).
- rate: average rate of the movie.
- topics: Optional field for topics such as "World War II" or "Terrorism" (separated by the "|" character).

Rating Table or Ratings.csv

This table contains the ratings of the users for the movies. The fields included are described below:

- id: unique identifier of the user.
- rate: rating provided by the user to the movie.
- movie_id: unique identifier of the movie.
- date: date on which the rating was provided by the user in the following format: YYYY-MM-DD (i.e., 2021-12-29).

User Table or User_Profile.csv

This table contains information about the user extracted from Twitter. The fields included are described below:

- id: unique identifier of the user.
- account_creation_year: rating provided by the user for the movie.

- friends_count: indicates the number of users that the user follows.
- twitterName: Twitter name.
- preferred_hour: preferred hour where the user tweets.
- weekend_tweeter: flag to indicate whether the user tends to predominantly tweet on weekends.
- preferred_weekday: day of the week the user mostly writes their tweets.
- early_bird: flag to indicate whether the user tends to tweet early in the morning.
- night_owl: flag to indicate whether the user tends to tweet late in the night.
- geo_enabled: flag to indicate whether the user has enabled the geo-location.
- week_tweeter: flag to indicate whether the user tends to predominantly tweet on weekdays.
- favourites_count: number of tweets that the user has marked as favorite.
- followers_count: number of user's followers.
- number_of_tweets: total number of tweets.

8.3.2 Methods

The technology used for the import, cleaning, and processing of the data is a Jupyter notebook (`https://jupyter.org/`) written in Python using pandas (`https://pandas.pydata.org/`) and NumPy (`https://numpy.org/`) libraries, among others.

Jupyter notebook is a web application that allows the interactive creation of computational documents. It offers flexibility since the code can be executed as it is written and also provides the possibility to create graphs and diagrams and embeds them in the same document. Because of that and its ease in sharing documents, the chosen platform for writing the code was Jupyter.

Python was the choice for writing the code due to its broad applicability to data science and its rich ecosystem of libraries.

Pandas is built on top of Python, and it is a data analysis and transform tool that is powerful, fast, and flexible.

Now, the different steps for the model creation will be defined:

Data Cleaning

After importing and loading the three tables from the dataset presented in this work, the first step is to make some changes, aggregation, and cleaning of the raw data. This process is defined as follows:

User Profile Table

Although in this dataset, there are some fields that are already feature-ready, some of the others can still be optimized in order to transform them into feature-like fields. In order to perform this, some numeric fields will be categorized by being transformed into flag fields by setting a threshold, and if the numerical field is greater than the threshold, the flag field will be set to true, and if it is equal to or lower than the threshold, the flag field value will be set to zero.

As an example of that, in the raw data, there is a field called *number_of_tweets*, which is a numerical field representing the number of total tweets of the user. From this numerical value, a flag-like value called *heavy_tweeter* will be created, which is set to 1 if the value of *number_of_tweets* is greater than a threshold and set to 0 if the value is equal or lower than the threshold. In this case, the threshold is set to 5000, which means that users that have more than 5000 tweets are classified as *heavy tweeters*.

After the creation of these new features, the fields that are not relevant (or the ones that are not feature-like) will be removed.

Ratings Table

For the rating table, the first thing that will be done is to add a flag field based on the date field. The field will be called weekend, and the value will be 1 if the rate was provided on the weekend, and 0 if it was not.

After that, all rates for movies that have less than a number of ratings will be removed from the table (this number is set to 100).

Movies Table

The first thing to do is to remove the movies that have less than a certain number of ratings from this table.

After that, a clean-up of some string-based fields will be performed in order to remove special characters.

Lastly, new fields based on some existing fields that are list-like will be created. As an example, there is the genre field, which will be then converted into as many fields as there are genres; that is, a Drama field, which is a flag field set to 1, will be added, among others, if the movie has Drama among its genres.

This will be performed for the genres, topics, and country code fields.

The cleaning function from above is called for the required fields.

After the cleaning, some new fields will be added based on duration and year published fields, and, lastly, the not needed fields will be removed.

Statistics

A total of 1.172.038 ratings for 78.628 movies, rated by 481 users can be found in the dataset.

Movies Table

In Figure 8.1, the distribution of the movie's country is shown.

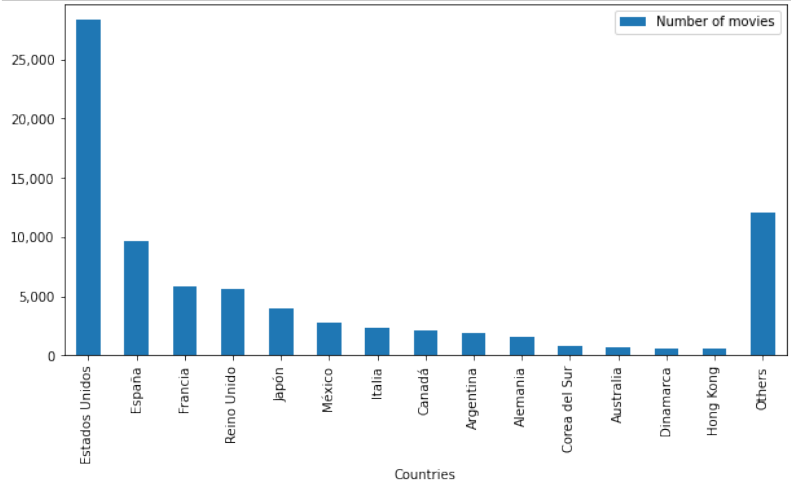


Figure 8.1: Movie country distribution.

Following, in Figure 8.2, the distribution of the movie's duration is shown.

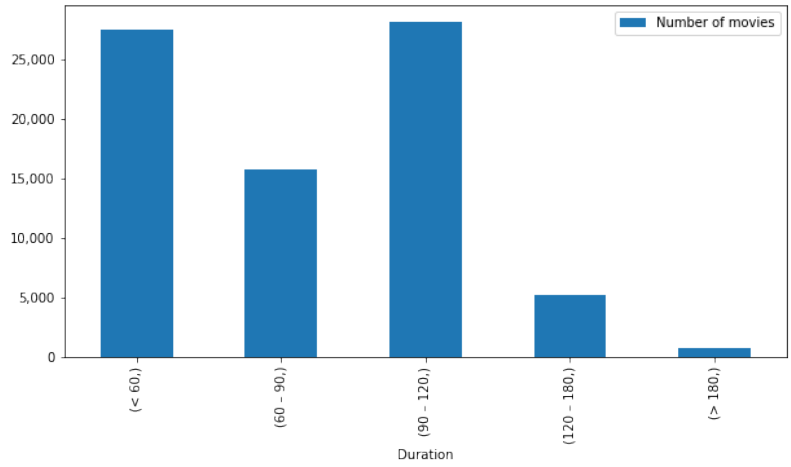


Figure 8.2: Movie duration distribution.

Furthermore, in Figure 8.3, the distribution of movie genres is shown.

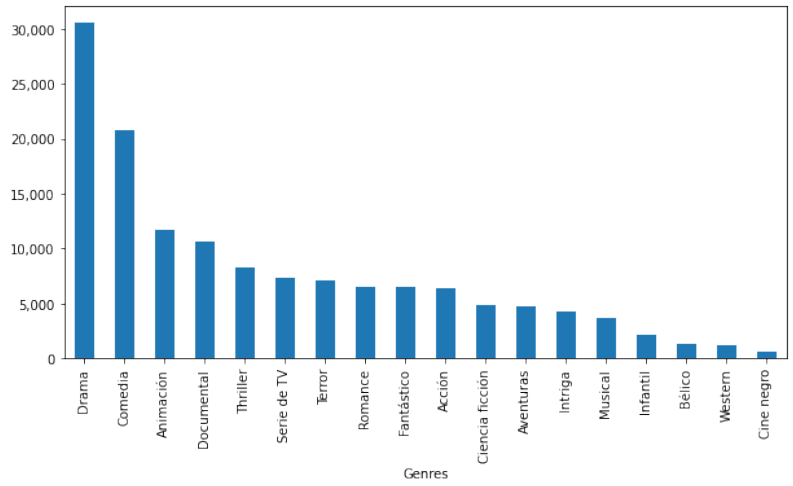


Figure 8.3: Movie genre distribution.

User Profile Table

In Figure 8.4, the distribution of the user profile features is shown.

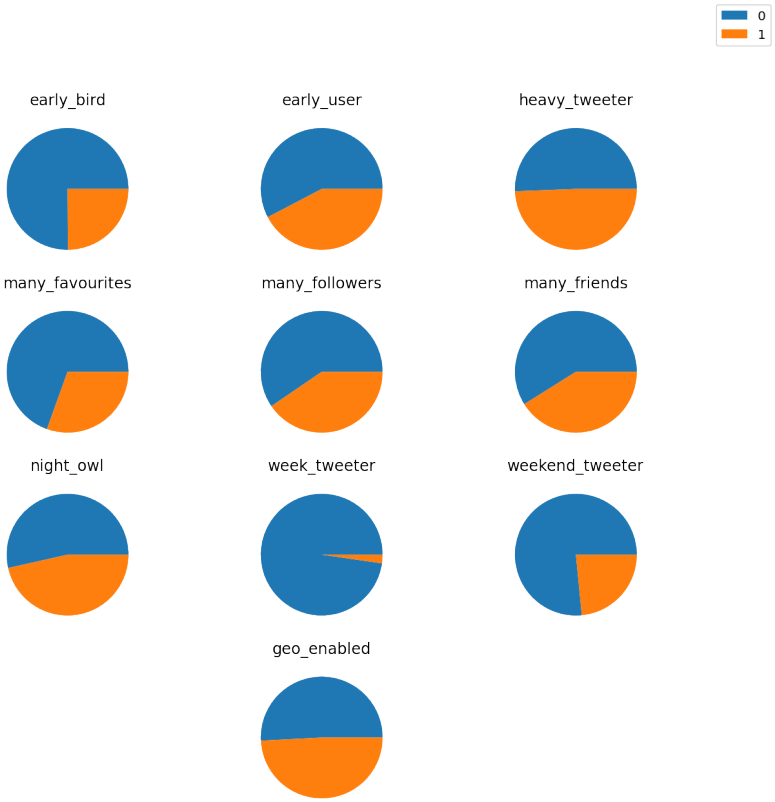


Figure 8.4: All user feature distribution.

Ratings Table

The total average rating from all users for all movies is 6.041 (out of 10).

In Figure 8.5, the distribution of the ratings is shown.

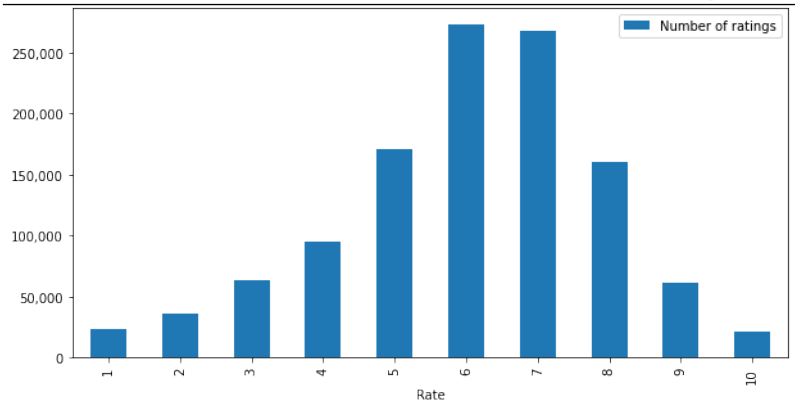


Figure 8.5: Rating distribution.

In Figure 8.6, the distribution of the number of ratings per user is shown.

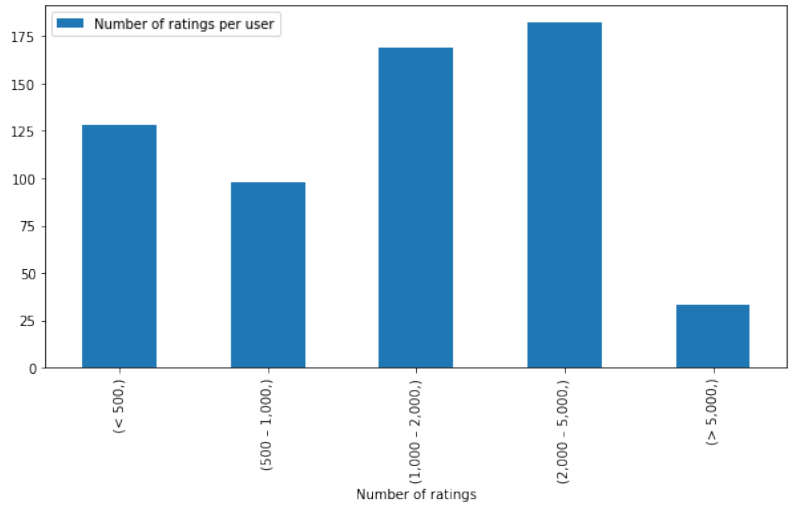


Figure 8.6: Number of rating per user distribution.

Model Description

The model used for creating the predictions and, thus, the recommendations is a mix of two approaches. On the one hand, there is a content-based approach, where the features of the users are classified and mapped to the product features. On the other hand, there is a collaborative filtering approach where the prediction is obtained from the rates of similar users. The split for the training–test data is 80%–20%. This split has been made based on user ids with the idea that users in the test data do not appear in the training data, emulating a new user's cold-start scenario.

The model is exhaustively explained in the following subsections. However, the main idea is described in aggregated Formula (**??**) of the two models. The content-based prediction model is described in Formula (**??**), and the high-affinity prediction model is described in Formula (**??**). Both these models are explained in the next subsections.

$$\hat{y} = \frac{\hat{y}_{cb} + \hat{y}_{affinity}}{2} \tag{8.1}$$

where $\hat{y}_{cb}$ is the movie rating prediction vector using the content-based approach and $\hat{y}_{affinity}$ is the movie rating prediction vector using the high-affinity approach.

$$\hat{y}_{cb} = \frac{U_{ij} \cdot P_{jk}}{\sum_{i=0}^{N} \sum_{k=0}^{M} U_{ij} \cdot P_{jk}} \cdot P_{jk}^{T} \tag{8.2}$$

where U is a matrix with the user features and the movie's rating average for every feature, P is a matrix with the movie features and the movie rates average for every feature, N is the number of user features and M is the number of movie features.

$$\hat{y}_{affinity} = \frac{\sum_{i=0}^{F} R_{ij}}{F} \tag{8.3}$$

where R is the rating matrix with only users with high affinity with the user the recommendations are created for, high-affinity users are those with more than 80% coincidence in the user features and F is the number of high-affinity users.

Product-User Feature Matrix (Content-Based)

The first step would be to create a rating matrix where the x-axis is the user ids, and the y-axis is the movie ids. This matrix will be enriched with the user profiles. For each value from every feature from the user profile, the rating will be calculated for every movie. For example, for the feature "weekly_tweeter" and the value: 0, the rates are aggregated and reduced by calculating the average. The result would be a table, as described in Table 8.1.

In parallel, the preprocessed movies table will be used for creating the predictions. This table is shown in Table 8.2.

Then, the user for which the predictions will be generated is chosenm and their user profile is fetched. Based on this user profile, the user-feature matrix from the image above will be filtered out, and, as a result, no relevant features will be dropped. The result is shown in Table 8.3.

Table 8.1: Dataframe feature ratings matrix, including the average rating per movie for all different user feature values. The shape of the matrix is (20, 2831), where 20 is the different user feature values, and 2831 is the number of movies after filtering movies without enough ratings.

Feature_key	Movie_id Feature_value	100072	100408	100958	...
early_bird	0	6.463918	3.902174	8.345865	...
	1	6.545455	4.000000	8.550000	...
early_user	0	6.486111	3.957143	8.401869	...
	1	6.482759	3.863636	8.378788	...
geo_enabled	0	6.349206	4.166667	8.379310	...
	1	6.611940	3.700000	8.406977	...
heavy_tweeter	0	6.555556	4.000000	8.218391	...
	1	6.396552	3.839286	8.569767	...
many_favourites	0	6.521739	3.860759	8.362069	...
	1	6.394737	4.057143	8.456140	...
many_followers	0	6.629630	3.840000	8.336735	...
	1	6.244898	4.076923	8.466667	...
many_friends	0	6.600000	4.060606	8.377551	...
	1	6.327273	3.729167	8.413333	...
night_owl	0	6.470588	3.746032	8.384615	...
	1	6.500000	4.137255	8.402439	...
week_tweeter	0	6.488372	3.911504	8.390533	...
	1	6.000000	5.000000	8.500000	...
weekend_tweeter	0	6.395833	3.962500	8.411348	...
	1	6.735294	3.823529	8.312500	...

Table 8.2: Dataframe movies, including all the movies with all movie features. The shape is (2831, 412), where 2831 is the number of movies, and 412 is the number of features.

id	Thriller	Drama	Romance	...	20 s	Short	Long
100072	0	1	0	...	0	0	0
100408	0	0	0	...	0	1	0
100958	0	0	0	...	0	0	1
...	...	...	...	...	...	...	...

After that, both matrices (feature ratings and movies) will be multiplied, and then a user feature–movie feature will result from it. Then, the table values will be normalized. The result will be a normalized array of user movie features that reflects the mapping between the user

features and the movie features. The shape is (10, 412), where 10 is the number of user features, and 412 is the number of movie features.

Then, the result will be multiplied by the movie matrix transposed, and the result will be the data frame with the prediction for the movie rate per user feature. The shape is (10, 2831), where 10 is the number of features, and 2831 is the number of movies.

Table 8.3: Dataframe matrix as the result of filtering Table 8.1 with the features from the user selected. The resulting shape is (10, 2831), where 10 is all the user feature values from selected users, and 2831 is the number of movies.

	Movie_id	**100072**	**100408**	**100958**	**...**
Feature_key	**Feature_value**				
early_bird	0	6.463918	3.902174	8.345865	...
early_user	1	6.482759	3.863636	8.378788	...
geo_enabled	1	6.611940	3.700000	8.406977	...
heavy_tweeter	0	6.555556	4.000000	8.218391	...
many_favourites	0	6.521739	3.860759	8.362069	...
many_followers	0	6.629630	3.840000	8.336735	...
many_friends	0	6.600000	4.060606	8.377551	...
night_owl	1	6.500000	4.137255	8.402439	...
week_tweeter	0	6.488372	3.911504	8.390533	...
weekend_tweeter	1	6.735294	3.823529	8.312500	...

Lastly, this table will be reduced for every movie, calculating the average of every feature key and feature value combination for the movie recommendation. This will result in a data frame with the predictions of every movie for the selected user. The shape is (2831), where 2831 is the number of movies.

Collaborative Filtering with Users with High Affinity (Collaborative Filtering)

For the collaborative filtering model, the approach is to find users with high affinity with the user the recommender is creating the recommendations for. In order to do that, the recommender will search for users with more than a number of user features in common (for the current experiment, it was set to 8), and from these users, a movie rating matrix will be generated. This matrix is shown in Table 8.4. The table displays the normalized ratings from every user for every movie. The NaN values mean that the user has not rated the movie. No rated movies will be ignored for calculating the total average rating for every movie. Movies with no provided rating from any high-affinity user will not be taken into consideration for the recommendations.

Table 8.4: Dataframe matrix for affinity showing the ratings from the users with high affinity with the selected user. The shape is (15, 2831), where 15 is the number of users with high affinity and 2831 is the number of movies.

Movie_id User_id	100072	100408	...	998393	999360
122203	NaN	NaN	...	NaN	NaN
175298	NaN	NaN	...	0.67	0.50
204280	NaN	0.50	...	0.83	1.00
...	...	...	...	...	...
825186	1.00	NaN	...	0.83	0.50
871105	0.50	0.50	...	0.50	NaN
976346	0.50	NaN	...	1.00	0.50

From this matrix, it will be reduced by calculating the rating average per movie, obtaining an array of recommendations for every movie, as shown in Table 8.5.

Table 8.5: Array with movie predictions for the selected user according to users with high affinity. The shape is (2831), where 2831 is the number of movies.

Movie_id	Rate
100408	0.733333
100958	0.666667
101022	0.833333
...	...

Merging Approaches and the Creation of Predictions

In order to merge the two previous approaches, the first step will be to create an evaluation table where the real ratings for the user are provided. Both arrays of recommendations (content-based and collaborative filtering) will be added to the table as new columns. Then, all movies that have not been rated will be filtered out. Lastly, a new column will be generated with the normalized average of both predictions.

In Table 8.6, an example of the final outcome can be found, with the actual ratings from a certain user for every movie and the predictions from both models together with the aggregated prediction.

Table 8.6: Dataframe predictions for a user where the predictions for the selected user are displayed. The shape is (585, 4), where 585 is the number of movies rated by the user from the movie set, and 4 is the number of columns added for predictions.

Movie_id	y	yhat	yhat_affinity	yhat_total
107060	7.0	0.566104	0.887597	0.726850
108145	7.0	0.606314	0.790698	0.698506
109220	7.0	0.712769	0.848837	0.780803
...	...	...	...	...

8.4 Results

The resulting outcome table can be sorted by higher predicted value; that is, the items that the recommender thinks are more likely to be liked by the user. The result is shown in Table 8.7.

Table 8.7: Dataframe predictions for the user where the predictions for the selected user are displayed sorted by prediction (highest prediction rank on top). The shape is (585, 4), where 585 is the number of movies rated by the user from the movie set, and 4 is the number of columns added for predictions.

Movie_id	y	yhat	yhat_affinity	yhat_total
745751	10.0	0.963304	0.915282	0.939293
655275	8.0	0.917696	0.939276	0.928486
624827	9.0	0.834351	0.998339	0.916345
459936	9.0	0.950638	0.872689	0.911664
252628	7.0	0.926104	0.887597	0.906850
370639	9.0	0.923677	0.876586	0.900131
...	...	...	...	...

8.4.1 Metrics

For evaluating the results, the top N values will be chosen (5 in our example), and some metrics will be used to evaluate how the model performed.

Accuracy

This metric is directly calculated from the sklearn library (`https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html`). It is called the Accuracy classification score, and it compares the predicted items with the actual items. The formula of

this metric is defined in Formula (**??**).

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \tag{8.4}$$

where $1(x)$ is the indicator function.

Mean Reciprocal Rank

The mean reciprocal rank is a metric that checks that the items are recommended in the same order that they were actually rated. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q. The latter is described in Formula (**??**).

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{8.5}$$

where $rank_i$ is the position of the document for query.

The average results from these two metrics applied to the whole dataset are an MRR of 0.457 and an accuracy of 60%.

Recommended Items Average

Lastly, there is the recommended item's average metric, which is just calculating the average of the real rating of the top N recommended items. The results are shown in Table 8.8.

Table 8.8: Results of the recommended item average.

Average Rating of Recommended Items	Average Rating from User	Improvement over Average Rating
8.6 (out of 10)	7.38 (out of 10)	16.53%

Figure 8.7 displays the recommended items average and the average rating of the user together with the rating distribution to stress the quality of the recommendations.

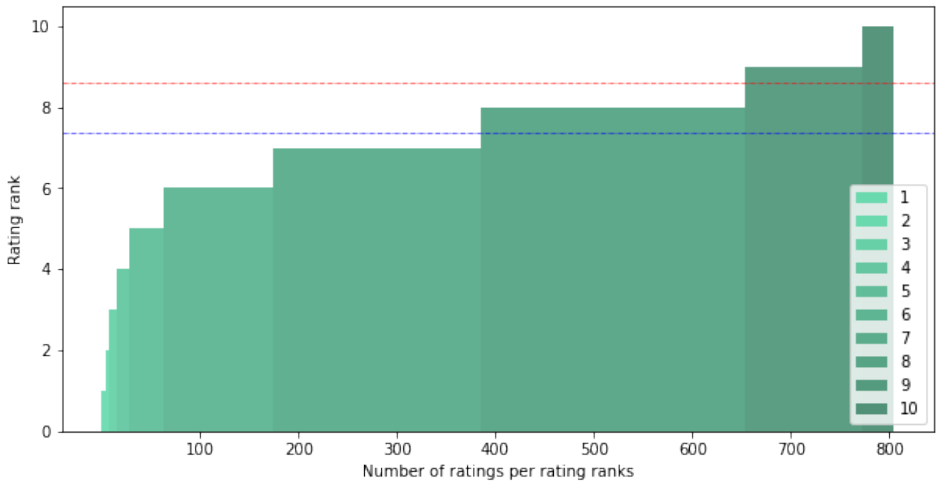


Figure 8.7: User rating distribution in comparison with the average of the recommended items (red) and the item rating average from the user (blue).

8.4.2 Baseline

For the baseline, we have chosen the recent work on recommendations for items set completion [22]. In order to choose a similar scenario to the user's cold start problem covered in this work, the chosen scenario is the predicted subject labels for the EconBiz Dataset, where the partial set of items along with the title is given and is used with the SVD model. This scenario, which is comparable to the cold-start problem that is faced in this work, obtains an MRR of around 45%. On the other hand, the proposed model of our work with the CSP-Dataset obtains an MRR of 0.457%.

8.5 Discussion

The main limitation of the dataset, the algorithm itself, and potential models that leverage the datase, is that in order to be executed on real users, it would require those users to have a Twitter account (or any other kind of social stream data) that will be used to elaborate their behavioral profile. This could be seen as a drawback. However, it is the backbone of the whole work: the usage of social media data to palliate the cold start problem.

The main benefit of CSP-Dataset is the fact that the dataset offers two different tables for the same individual, representing, on one side, the behavior of the user and, on the other side, the ratings for the movies. Moreover, the dataset also includes a comprehensive table for the characteristics of the items (movies). Therefore, accurate predictions could be created by extrapolating the features from one table (behavioral data from Twitter) toward the other (rating data), creating correlation connections between the behavior features and item features.

Then, the presented dataset can be used to craft models that could be leveraged for operational applications. For example, these models could be used by streaming applications, such as Netflix or Spotify, by asking their users to grant temporary access to their social stream (i.e., Twitter), and then instant and tailored recommendations would be provided to the users in

a matter of seconds without the need to perform any rating or manually providing user data.

This work provides a dataset of high interest due to the scarcity of datasets providing extensive items and user behavioral features. The dataset will enable researchers to create their models in the future with cutting-edge algorithms (i.e., Neural Networks) and, therefore, the dataset is a very good candidate to become the standard dataset for the cold start problem due to the fact that it contains many user's behavioral data.

Moreover, the results of the experiments support the hypothesis that the presented dataset can be used for creating recommendations for users without having any previous rating information from them. The extrapolation of user classification data to the rating behavior has, therefore, also been confirmed to be a valid approach. Thus the crafted dataset can be used by other researchers to create other studies that focus on the alleviation of the cold start problem. This dataset, because of its duality of user-item information, is a candidate to become one of the standards for the cold start problem.

The algorithm used has proven to create very good results, even though many other techniques can be leveraged to improve the accuracy of the recommendations even more. Moreover, other features could be created out of the raw data and be leveraged for future work.

Acknowledgments

This research was partially supported by the Spanish State Research Agency through the project PID2019-103880RB-I00 and the Andalusian Agency project P20_00673.

Bibliography

[1] Bennett, J. y Lanning, S.: «The Netflix Prize». En: *In Proceedings of the KDD Cup and Workshop*, pp. 3–6, 2007.

[2] Bobadilla, J.; Ortega, F.; Hernando, A. y Bernal, J.: «A collaborative filtering approach to mitigate the new user cold start problem». *Knowledge-Based Systems*, 2012, **26**, pp. 225–238.

[3] Bobadilla, J.; Ortega, F.; Hernando, A. y Gutiérrez, A.: «Recommender systems survey». *Knowledge-Based Systems*, 2013, **46**, pp. 109–132.

[4] Boutilier, C.; Zemel, R.S. y Marlin, B.: «Active Collaborative Filtering». En: *In Proceedings of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 98–106, 2003.

[5] Burke, R.; Felfernig, A. y Göker, M.H.: «Recommender systems: An overview». *AI Magazine*, 2011, **32**, pp. 13–18.

[6] Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R. y García-Sánchez, F.: «Social knowledge-based recommender system. Application to the movies domain». *Expert Systems with Applications*, 2012, **39**, pp. 10990–11000.

[7] Chien, C.; Yu-Hao, W.; Meng-Chieh, C. y Yu-Chun, S.: «An effective recommendation method for cold start new users using trust and distrust networks». *Information Sciences*, 2013, **224**, pp. 19–36.

[8] Esmaeili, L.; Mardani, S.; Golpayegani, S.A.H. y Madar, Z.Z.: «A novel tourism recommender system in the context of social commerce». *Expert Systems With Applications*, 2020, **149**, p. 113301. doi: https://doi.org/10.1016/j.eswa.2020.113301.

[9] Feng, J.; Xia, Z.; Feng, X. y Peng, J.: «RBPR: A hybrid model for the new user cold start problem in recommender systems». *Knowledge-Based Systems*, 2021, **214**, p. 106732. doi: https://doi.org/10.1016/j.knosys.2020.106732.

[10] García-Sánchez, F.; Colomo-Palacios, R. y Valencia-García, R.: «A social-semantic recommender system for advertisements». *Information Processing and Management*, 2020, **57**, p. 102153. doi: https://doi.org/10.1016/j.ipm.2019.102153.

[11] Gonzalez Camacho, L.A. y Nice Alves-Souza, S.: «Social network data to alleviate cold-start in recommender system: A systematic review». *Information Processing and Management*, 2018, **54**, pp. 529–544.

[12] Herce-Zelaya, J.; Porcel, C.; Bernabé-Moreno, J.; Tejeda-Lorente, A. y Herrera-Viedma, E.: «New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests». *Information Sciences*, 2020, **536**, pp. 156–170. doi: https://doi.org/10.1016/j.ins.2020.05.071.

[13] Hernando, A.; J., Bobadilla.; Ortega, F. y Gutiérrez, A.: «A probabilistic model for recommending to new cold-start non-registered users». *Information Sciences*, 2017, **376**, pp. 216–232.

[14] Hoang-Son, L: «Dealing with the new user cold-start problem in recommender systems: A comparative review». *Information Systems*, 2016, **58**, pp. 87–104.

[15] Lika, B.; Kolomvatsos, K. y Hadjiefthymiades, S.: «Facing the cold start problem in recommender systems». *Expert Systems with Applications*, 2014, **41**, pp. 2065–2073.

[16] Majumdar, A. y Jain, A.: «Cold-start, warm-start and everything in between: An autoencoder based approach to recommendation». En: *In Proceedings of the 2017 International Joint Conference on Neural Networks*, pp. 3656–3663, 2017. doi: https://doi.org/10.1109/IJCNN.2017.7966316.

[17] Martínez-Cruz, C.; Porcel, C.; Bernabé-Moreno, J. y Herrera-Viedma, E.: «A Model to Represent Users Trust in Recommender Systems using Ontologies and Fuzzy Linguistic Modeling». *Information Sciences*, 2015, **311**, pp. 102–118. doi: doi:10.1016/j.ins.2015.03.013.

[18] Natarajan, S.; Vairavasundaram, S.; Natarajan, S. y Gandomi, A.H.: «Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data». *Expert Systems With Applications*, 2020, **149**, p. 113248. doi: https://doi.org/10.1016/j.eswa.2020.113248.

[19] Panda, D.K. y Ray, S.: «Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review». *Journal of Intelligent Information Systems*, 2022, **59**, pp. 341–366.

[20] Ramezani, M.; Akhlaghian Tab, F.; Abdollahpouri, A. y Abdulla Mohammad, M.: «A new generalized collaborative filtering approach on sparse data by extracting high confidence relations between users». *Information Sciences*, 2021, **570**, pp. 323–341. doi: https://doi.org/10.1016/j.ins.2021.04.025.

[21] Sahu, A.K.; Dwivedia, P. y Kant, V.: «Tags and Item Features as a Bridge for Cross-Domain Recommender Systems». *Procedia Computer Science*, 2018, **125**, pp. 624–631.

[22] Vagliano, I. y Galke, L.: «Recommendations for item set completion: On the semantics of item co-occurrence with data sparsity, input size, and input modalities». *Information Retrieval Journal*, 2022, **25**, pp. 269–305. doi: https://doi.org/10.1007/s10791-022-09408-9.

[23] Viktoratos, I. y Tsadiras, A.: «Personalized Advertising Computational Techniques: A Systematic Literature Review, Findings, and a Design Framework». *Information*, 2021, **12(11)**, p. 480. doi: https://doi.org/10.3390/info12110480.

[24] Viktoratos, I.; Tsadiras, A. y Bassiliades, N.: «Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems». *Expert Systems With Applications*, 2018, **101**, pp. 78–90.

[25] Wei, J.; He, J.; Chen, k.; Zhou, Y y Tang, Z.: «Collaborative filtering and deep learning based recommendation system for cold start items». *Expert Systems With Applications*, 2017, **69**, pp. 29–39.

[26] Zhang, Y.; Shi, Z.; Zuo, W.; Yue, L. y Li, X.: «Joint Personalized Markov Chains with social network embedding for cold-start recommendation». *Neurocomputing*, 2020, **386**, pp. 208–220. doi: https://doi.org/10.1016/j.neucom.2019.12.046.

Chapter 9

Concluding remarks

To finalize, in this chapter we present the concluding remarks from the development of the studies included in this Doctoral Thesis. The present Thesis provides methodologies and datasets that can be leveraged from further research studies or from operational applications that aim to solve real life problems with recommender systems. These datasets and systems can be enhanced with new and more complex techniques like the usage of neural networks or further feature engineering.

As general conclusion, we have surpass in accuracy state-of-the-art algorithms by using contextual data and data from other domains, extrapolating it to the target domain. As future work, we might want to keep exploiting extracted data by increasing the number of features by use of feature engineering. Another objective from future research might be the leverage of sentiment analysis for the datasets obtained in previous works in order to create even more features.

In the following sections, the conclusions for every specific study are described.

9.1 Study 1

This study presents a recommendation approach that is based on a prediction model. The approach makes use of information that is extracted from social media to classify users based on their profiles. To achieve this, we utilized machine learning techniques, such as classification trees and random forest, which helped us to assign users a flag for each item that indicates whether it is suitable to be recommended or not. By doing this, users do not need to provide any personal information other than the source of their social media, which helps to alleviate the cold start problem. Implicit data obtained from social media helped us to bridge the information gap for new users of the system.

We tested the proposal in a movie recommendation environment and achieved satisfactory results for the suggested predictions, indicating that the information we have in social media is a valuable source of information that can be used to give recommendations for items. The limitations of this research work include the fact that, in order to build prediction models, we need the connection between two sources of data: item rating data and social media data. That

was possible due to the site sources we used to extract the data, but it is not a normal pattern in some other data sets. Another limitation factor occurs when the user either does not have social media data or these data are not meaningful enough to create a user profile with some degree of confidence.

Future work could be focused on the extraction of other features by running sentiment analysis on the tweets from the user, adding in this way more features to the profile.

9.2 Study 2

This study introduces a trust-based fuzzy linguistic recommender system that provides personalized activities to students, thereby reinforcing their individualized education. The system aims to recommend activities to students based on trust relationship rank instead of those with similar requirements or rating history. To achieve this, the study proposes a method to estimate the trust score between a pair of students. The system was applied in a real environment, specifically in the subjects Oral surgery I and II and Implantology II of Dentistry degree at the University of Granada in Spain. The benefits of this system include an increase in the personalization of education received by students, who can also access activities anytime and anywhere. The proposal was evaluated, and the experimental results demonstrated good results and positive feedback from students, indicating an enhancement in the effectiveness of professors dealing with large groups of students.

As future work, we might consider to study the possibility of automate the creation of activities by the system, based on individual feedback provided by the student of each component of the activities, as well as let the students create their own activities. Other proposal might be to focus on applying specific measures of the social networks analysis, exploiting the information represented in the trust network.

9.3 Study 3

This study introduces a new method for automatically generating a polarity dictionary using the stock market as a reference domain without the need for human intervention. The system uses price changes of particular stocks over time as a guiding polarity value, attributing the magnitude of price variation to financial news about the stock during that period. This information is used to create a working corpus, from which a binned corpus is built and the TF-IDF algorithm is applied to compute the signed guiding polarities for each term. These values are then disseminated within the neighbourhood of each term based on embeddings-enabled cosine distance, and mapped to fuzzy linguistic labels with an indicator showing how reliable the scores are based on its distribution of occurrences in the corpus.

To demonstrate the effectiveness of the approach, it was implemented for the Euro Stoxx 50 from January 2018 to March 2019, and the resulting fuzzy polarity dictionary was made available. This approach solves three typical issues of classic polarity dictionary building methods: 1) providing fuzzy linguistic sets instead of crisp polarity values to solve the scale and thresholding

problem, 2) avoiding human bias by inferring polarity values without human intervention and providing an indicator for reliability, and 3) contextualizing polarity values to a specific domain.

Further research work could focus on the impact of using of n-grams instead of mono-grams as well as the extension to further Part of Speech label (adverbs, etc). In addition, techniques to transfer the polarity dictionary to a different domain might also pave the way towards a multi-domain generic approach. Last but not least, we'd like to point to all the operationalization of the polarity dictionary to compute sentiment using fuzzy linguistic arithmetic operations.

9.4 Study 4

The CSP-Dataset is unique because it offers two distinct tables for the same individual, one detailing the user's behavior and the other providing ratings for movies. In addition, the dataset includes a comprehensive table that describes the characteristics of the items (movies). By extrapolating features from one table to another, correlation connections can be created between behavioral and item features, leading to accurate predictions.

This dataset can be used to create models that could be used for operational applications, such as Netflix or Spotify, to provide instant and tailored recommendations to users without requiring rating or manual user data. This dataset is of great interest due to the scarcity of datasets providing extensive items and user behavioral features. Researchers can use the dataset in the future to create cutting-edge algorithms (e.g., Neural Networks) for the cold start problem, and it is a strong candidate to become the standard dataset for this scenario.

The results of the experiments support the hypothesis that the dataset can be used to create recommendations for users without prior rating information, and the extrapolation of user classification data to the rating behavior is a valid approach. Therefore, the dataset can be used by other researchers to create studies that focus on alleviating the cold start problem. Although the algorithm used has shown very good results, other techniques can be utilized to further improve recommendation accuracy.

Further research could be the usage of other techniques, i.e. Neural Networks, that can be used to improve the accuracy of the recommendations even more. Moreover, other features could be created out of the raw data and be leveraged for future work.

9.5 Future trends

Recommender systems have become an integral part of our daily lives, providing personalized suggestions and recommendations for a variety of products and services. However, these systems face several challenges, including the cold-start problem (which arises when there is insufficient data about a new user or item), the need for explainability and interpretability, and the issue of privacy and fairness. To overcome these challenges, researchers are exploring new techniques and approaches that leverage advances in artificial intelligence, data science, and human-computer interaction.

In this context, several emerging trends are shaping the future of recommender systems. These trends are described next.

1. Development of new recommendation methods supported by AI techniques such as machine learning, soft computing, or deep learning.
2. Improving the automatic establishment of user profiles to reduce the impact of the cold-start problem:
   - Extraction of information from social networks.
   - Opinion analysis.
3. Working on new recommendation techniques that include ethical guidelines for trustworthy AI:
   - Explainability and interpretability: the ability of systems to explain their internal functioning principles and the human decisions that could lead to the generated output.
   - Ethical considerations: a commitment to ensuring that recommendations do not include biases or discrimination towards certain user groups.
4. Analysis of specific domains and environments to propose new applications:
   - Advancing the application of these techniques to improve the quality of life for older people through physical exercise.
   - Adapting and applying RecSys in the field of smart cities to assist in decision-making associated with the services offered: monitoring of noise, waste, traffic, smart parking, smart lighting, automation of public buildings, urban sustainability, etc.
   - Other sectors: energy (networks and infrastructure), human resources, tourism, personalized nutrition, etc.