UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: A Recommendation System Implementation For E-commerce Web Sites With Implicit Feedback Data Sets: An Ecuadorian Enterprise Case Study

Trabajo de integración curricular presentado como requisito para la obtención del título de
Ingeniero en Tecnologías de la Información

Autor:

Román Eras Osiris Anael

Tutor:

Ph.D Peluffo Ordoñez Diego Hernán

Urcuquí, marzo 2020

UNIVERSIDAD YACHAY TECH

Urcuquí, 25 de marzo de 2020

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2020-00020-AD

A los 25 días del mes de marzo de 2020, a las 10:00 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dra. GUACHI GUACHI, LORENA DE LOS ANGELES , Ph.D.
Miembro No Tutor	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.
Tutor	Dr. PELUFFO ORDONEZ, DIEGO HERMAN , Ph.D.

El(la) señor(ita) estudiante **ROMAN ERAS, OSIRIS ANAEL**, con cédula de identidad No. **1724771645**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **A Recommendation System Implementation For Ecommerce Web Sites With Implicit Feedback Datasets: An Ecuadorian Enterprise Case Study**, previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN.**

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. PELUFFO ORDONEZ, DIEGO HERMAN , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Miembro Tribunal De Defensa	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.	9,0
Tutor	Dr. PELUFFO ORDONEZ, DIEGO HERMAN , Ph.D.	10,0
Presidente Tribunal De Defensa	Dra. GUACHI GUACHI, LORENA DE LOS ANGELES , Ph.D.	9,5

Lo que da un promedio de: **9.5 (Nueve punto Cinco)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

ROMAN ERAS, OSIRIS ANAEL
Estudiante

Dra. GUACHI GUACHI, LORENA DE LOS ANGELES , Ph.D.
Presidente Tribunal de Defensa

Dr. PELUFFO ORDONEZ, DIEGO HERMAN , Ph.D.
Tutor

Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.
Miembro No Tutor

MEDINA BRITO, DAYSY MARGARITA
Secretario Ad-hoc

AUTORÍA

Yo, **Osiris Anael Román Eras**, con cédula de identidad 1724771645, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autora (a) del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, marzo y 2020.

Anael R.

Osiris Anael Román Eras
CI: 1724771645

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Osiris Anael Román Eras,** con cédula de identidad 1724771645, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior

Urcuquí, marzo y 2020.

Anael R.

Osiris Anael Román Eras
CI: 1724771645

Dedicatoria

Dedicado a mi familia, por ser mi soporte e inspiración para seguir adelante...

Osiris Anael Román Eras

Agradecimiento

Quiero empezar agradeciendo a Joyas Nereyda, una empresa nacida en Santo Domingo y actualmente presente en otras provincias del Ecuador. Sus datos aportados, fueron de gran importancia durante el proceso investigativo en este proyecto de titulación. Esta empresa ya se ha destacado en años anteriores por ayudar a estudiantes universitarios en sus procesos de investigación y ahora también forman parte de este estudio.

Agradezco a todo el conjunto de docentes de la Universidad Yachay Tech, pues han sabido trazar un buen camino académico para que estudiantes como yo puedan desarrollar sus capacidades y conocer su verdadero potencial.

Siempre he pensado que el proceso que atraviesa una persona para alcanzar una meta es de igual o mayor importancia que la meta misma. Por lo tanto, quiero agradecer a todas las personas que en algún punto formaron parte de mi vida universitaria. Los momentos vividos, las experiencias y pensamientos que compartí con ustedes, han podido formar el profesional y la persona que soy hoy en día. En un intento por mencionar a las personas que mi memoria recuerda con cariño, por ser personas con quienes compartí tardes de estudio, risas o simplemente alguna salida buscando relajar la mente del ámbito académico, quiero nombrar a: Marlon Gancino, Pablo Contreras, Winter Morejón, Sergio Hidalgo, Paul Silva, William Arellano, Kevin Chamorro, María Fernanda Molina, Emil Vega, Jordy Cevallos, Mayra Jiménez, Horus Román y Krishna Román. Los recuerdos que tengo con ustedes harán que mi etapa universitaria sea una de las mejores en mi vida.

Estoy agradecido con mis padres Carlos Román y Lorgia Eras, porque sus enseñanzas y comentarios de apoyo a lo largo de mi carrera universitaria me han permitido ver las cosas desde una perspectiva diferente intentando siempre pensar fuera de la caja. Los logros en mi vida no son otra cosa sino una muestra del esfuerzo y sacrificio de ustedes por sacar lo mejor que existe dentro de mí.

Finalmente, quiero agradecer a Diego Peluffo dirigente de SDAS Research Group y mi tutor de tesis, porque ha sabido guiarme y apoyarme en la construcción de este proyecto.

Osiris Anael Román Eras

Resumen

Hoy en día, internet se ha convertido en una herramienta muy importante y ampliamente utilizada en diversas actividades diarias de la humanidad. El comercio electrónico es uno de los sectores que está siendo impulsado por internet, permitiéndole a las personas adquirir productos o servicios de una manera más fácil. Debido a la sobrecarga de información que existe en la actualidad, las empresas se encuentran inmersos en la búsqueda constante de herramientas que les permitan ofrecer sus productos correctamente a la variedad de usuarios que visitan su tienda en línea.

Esta investigación destaca la relevancia que existe en el conjunto de datos de retroalimentación implícita al construir un sistema de recomendación. Al mismo tiempo, explica el estado del arte sobre los sistemas de recomendación y menciona los beneficios existentes de usar este tipo de herramientas en las empresas. En este proyecto, se implementan y evalúan diferentes técnicas de recomendación basadas en la factorización matricial utilizando dos conjuntos de datos. Uno de estos proviene de Retail Rocket, un sitio web de comercio electrónico anónimo del mundo real que ha recopilado datos implícitos de sus usuarios y ha decidido compartirlo para fines investigativos. Todos los modelos implementados aquí se compararán utilizando dos métricas de evaluación muy comunes en el campo de los sistemas de recomendación. Finalmente, los modelos implementados se aplican al conjunto de datos reales provistos por una empresa ecuatoriana. Este conjunto de datos se utilizó para revelar cómo se comportarían los diferentes modelos al ser usado con datos de una empresa ecuatoriana similar.

Palabras Clave: Comercio Electrónico, Sistemas de Recomendación, Feedback Implícito, Machine Learning, Factorización Matricial.

Abstract

Nowadays, the internet has become a very important and widely used tool in several human daily activities. E-commerce is one of the sectors being powered by the internet, enabling people to purchase products or services more easily. Due to information overload, enterprise actors are constantly immersed in the search for tools that allow them to offer correctly their products to the great variety of users that visit their e-commerce.

This research emphasizes the relevance of the implicit feedback data set when building a recommendation system. Likewise, it explains the state of the art about recommendation systems and mentions the benefits of using this kind of tool in companies. Besides, this degree thesis document implements and evaluates various models of recommendation techniques based on matrix factorization using two data sets. One of these datasets comes from Retail Rocket, a real anonymous e-commerce website which has collected implicit data from customers and has decided to share for research purposes. All the models here-implemented are evaluated and compared regarding two evaluation metrics commonly used in the recommendation systems field. Finally, the models are implemented with the Ecuadorian real data set. This data set was used to reveal how the distinct models might behave under real data provided by an Ecuadorian similar enterprise.

Key Words: Ecommerce, Recommender Systems, Implicit Feedback, Machine Learning, Matrix Factorization.

Contents

Agradecimientos **ii**

Resumen **iii**

Abstract **iv**

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Document Organization . . . 2
- 1.3 Objectives . . . 4
  - 1.3.1 General Objective . . . 4
  - 1.3.2 Specific Objectives . . . 4

2 Overview Background **5**
- 2.1 E-commerce . . . 5
  - 2.1.1 International Context . . . 5
  - 2.1.2 Latin America and Ecuador Context . . . 6
- 2.2 Information Filtering . . . 8
- 2.3 Recommendation Systems . . . 10
  - 2.3.1 State of the Art . . . 11
  - 2.3.2 How Does a Recommendation Engine Work? . . . 12
  - 2.3.3 Type of Feedback Data Sets . . . 13
    - Explicit Feedback . . . 13
    - Implicit Feedback . . . 15
  - 2.3.4 Classification of Recommendations Techniques . . . 16
    - Content-Based Recommending Systems . . . 16
    - Collaborative Filtering Recommending Systems . . . 18
    - Hybrid Filtering Recommending Systems . . . 21
  - 2.3.5 Evaluation Metrics . . . 22
    - Hit Rate . . . 22
    - Normalized Discounted Cumulative Gain (NDCG) . . . 23
- 2.4 Machine Learning . . . 26

3 Methodology **28**
3.1 Introduction . . . 28
3.2 The Data . . . 29
3.3 Modelling . . . 32
3.3.1 Experiment 1: Matrix Factorization . . . 33
Implementation . . . 33
3.3.2 Experiment 2: Non-Negative Matrix Factorization . . . 35
3.3.3 Experiment 3: Neural Matrix Factorization Approach . 35
3.3.4 Experiment 4: Implicit Feedback Matrix Factorization Approach . . . 36
3.4 Experimental Setup . . . 37
3.4.1 Experimental Settings . . . 37
3.5 Results of the Generic Data . . . 39
3.5.1 Matrix Factorization Result . . . 39
3.5.2 Non-Negative Matrix Factorization . . . 39
3.5.3 Neural Matrix Factorization . . . 40
3.5.4 Implicit Feedback Matrix Factorization . . . 41
3.5.5 Results Summary . . . 42

4 Analysis of results **43**
4.1 Ecuadorian Enterprise Case Study . . . 43
4.1.1 Training-Test Splitting . . . 43
4.1.2 Matrix Factorization . . . 44
4.1.3 Non-Negative Matrix Factorization . . . 45
4.1.4 Neural Network . . . 45
4.1.5 Implicit Feedback . . . 46
4.1.6 Ecuadorian Case Study Results Summary . . . 47

5 Final remarks **48**
5.1 Conclusions . . . 48
5.2 Recommendations . . . 49
5.2.1 Future Research . . . 50

A Python Implementation of the Different Models **51**
A.1 Experiment #1 Matrix Factorization . . . 51
A.2 Experiment #2 Non-Negative Matrix Factorization . . . 52
A.3 Experiment #3 Neural Network Matrix Factorization . . . 53
A.4 Experiment #4 Implicit Feedback Matrix Factorization Approach 54

List of Figures

2.1 Countries ranked by retail e-commerce sales [8]. 6
2.2 Latin America retail e-commerce sales [9]. 7
2.3 E-commerce purchasing in Ecuador [12]. 8
2.4 Information filtering process [15]. 10
2.5 Explicit feedback [21]. 14
2.6 Implicit feedback [16]. 15
2.7 General recommendation techniques classification retrieved from [16]. 16
2.8 Content-based recommendation example [16]. 17
2.9 Collaborative filtering recommendation [16]. 18
2.10 Collaborative filtering classification techniques [16]. 20
2.11 Hybrid recommendation system [16]. 21
2.12 Hybrid recommendation system black box [16]. 22

3.1 Methodology structure. 29
3.2 Number of events registered. 30
3.3 Events table. 31
3.4 Graphical representation of user events data set. 31
3.5 User events data set after cleaning. 32
3.6 Training and testing data set splitting. 33
3.7 Final architecture [32]. 36
3.8 HR, NDCG and MSE during training evolution. 39
3.9 HR, NDCG and MSE during training evolution. 40
3.10 HR, NDCG and MSE during training evolution. 41
3.11 Implicit feedback matrix factorization evolution. 41

4.1 Training and testing data set splitting. 44
4.2 HR and NDCG training evolution. 44
4.3 HR and NDCG training evolution. 45
4.4 HR and NDCG training evolution. 46
4.5 Implicit feedback matrix factorization evolution. 46

List of Tables

2.1 DCG calculation example. 25

3.1 Variable setup of the different experiments. 38
3.2 Comparison between MF models through evaluation metrics - Retail Rocket Case Study. 42

4.1 Comparison between MF models through evaluation metrics - Ecuadorian Case Study. 47

List of Abbreviations

INEC: Instituto Nacional de Estadística y Censos.

ARCOTEL: Agencia de Regulación y Control de las Telecomunicaciones.

PMF: Probabilistic Matrix Factorization.

NMF: Non-Negative Matrix Factorization.

SVD++: Singular Value Decomposition for Recommender Systems.

RLFM: Regression-based Latent Factor Model.

RMSE: Root Mean Square Error.

MAE: Mean Absolute Error.

NDCG: Normalized Discounted Cumulative Gain.

DCG: Discounted Cumulative Gain.

CG Cumulative Gain.

HR: Hit Rate.

MF: Matrix Factorization.

GMF: General Matrix Factorization.

MLP: Multi Layer Perceptron.

Chapter 1

Introduction

Currently, it is undeniable the fact that several enterprises are intended to become increasingly active in the broad range of possibilities on the internet much beyond an enterprise web page. The importance of e-commerce web pages has increased rapidly since they are capturing a significant place as a tool in everyday human being activities. In these times, the adaptation of companies to a market that is constantly changing in a local and an international context represents a key aspect to take into account.

Thanks to the easy accessibility to portable communication devices (smartphones and tablets) and the evolution of the internet, the use of e-commerce has exponentially increased in Latin America in recent years. Currently, Ecuador has been experienced growth in internet usage. According to a study made by INEC in 2012-2014. It is observed that 16.8% of enterprises in Ecuador have made at least one online transaction in 2012 vs the 17.1% in 2014 [1]. Besides, 81% of the Ecuadorian population make use of the internet [2], making Ecuadorian people able to buy or purchase products or services via the internet. Latin America remains a potential market that has even captivated the giant Amazon, which is considered the best e-commerce business in the world. The company announce the opening of a modern center dedicated to customer service in Bogota-Colombia. Some news like the previous one, let us know that Latin America has evolved enough to attract first world countries which can perceive in Latin America an excellent opportunity to develop the e-commerce market.

1.1 Motivation

Just to persuade yourself, if you have ever looked for products on Amazon, watched movies on Netflix or even looking for a web page on Google, you might have used a recommendation system without realizing it. Currently, given the evolution of the internet, there are huge volumes of information generated daily causing information overload. Such a massive amount of information is often distracting and can negatively impact decision-making. Recommendation systems assist users to avoid the information overload problem acting as information filters when it comes in the form of suggestions. In the e-commerce field, product recommendation systems are a great way to improve user experience bringing customers the relevant products they want or need.

Nowadays a recommendation system represents an essential tool in many popular big tech industries like Amazon, Google, Facebook, Alibaba, Netflix, Uber and so forth. The main goal is to offer the user what they are looking for even before they know it. Some of the benefits of using a recommendation system for products are:

- Increase sales and average order value,
- Create a consistent brand experience, and
- Avoid customer frustration.

In the case of Ecuadorian enterprises having no enough experience on e-commerce platforms, an application just as the one here proposed may represent a meaningful contribution. It could enable the possibility to raise strategies to increasing sales while offering a better shopping experience to the consumers.

From the obtained results, different data analysis tools can be developed to be adaptable to gain a more proper understanding of the Ecuadorian consumer. One of the main goals of e-commerce businesses is to offer a personalized sale experience especially at the specific moment of purchasing a product via the internet.

In this research, there is, of course, a substantial interest to work with real data from an Ecuadorian enterprise. In addition to achieving widespread results that can be extended to other companies with e-commerce platforms. The interaction of a client in the webshop can help to predict future behaviors about the consumer [3].

1.2 Document Organization

This Document is formed by four chapters. This chapter, the first one, makes an introduction to the importance and motivation behind this research project.

In this chapter you will find the objectives that wants to be achieved at the end of the project.

The rest of this thesis is organized as follows. Chapter 2 provides an overview of the background that exists behind recommendation systems. Section 2.1 offers an overview of e-commerce in an Ecuadorian and international context. In Section 2.2, information filtering is presented as a broad field where recommendation systems can be found as a subclass. The next Section 2.3 introduces the core of this research project as recommendation systems and its state of the art. Here can be found examples of current researches and applications in this field. The varied types of data used in recommendation systems are also explained in this section as the different types of recommendation systems that exist. You can also see some common evaluation metrics used to measure the performance of a recommendation system and an introduction to Machine learning and its classification.

Chapter 3 carries you through the different experiments and how they were performed. Section 3.1 and 3.2 provide you an introduction and explanation about the generic data used and where it can be obtained. Section 3.3 tries to explain in an easygoing way the different matrix factorization models that have been implemented for the recommendation system and how they work and differ between them. The next Section 3.4 provides you the experimental settings used in each of the models. Finally, in Section 3.5 you will observe the graphical results where the different models show the efficiency as recommenders.

In Chapter 4 you can discover the application of the previous models over a data set provided by the Ecuadorian enterprise case study. The results of the experiments can be found in Section 4.1.

Chapter 5 is the concluding chapter of this thesis project. Some recommendations for subsequent research are gathered in Section 5.1, and the conclusion of the project is drawn in Section 5.2.

Appendix A at the end of the document presents the link to the project website and the codes used to implement the various models written in Python programming language.

1.3 Objectives

1.3.1 General Objective

- To implement a recommendation system based on implicit feedback data sets and test it using a real data set taken from an Ecuadorian enterprise.

1.3.2 Specific Objectives

- To select and characterize the generic data set about customer shopping web behavior to build the input data set.
- To design the recommendation system in both mathematical and programming terms.
- To design an experimental setup to improve the recommendation technique approach through a machine learning technique.
- To validate the designed experimental setup in an Ecuadorian enterprise case study.

Chapter 2

Overview Background

2.1 E-commerce

E-commerce was born along with the internet opening announcement to the public. From this moment, thousands of websites for commercial usage have been registered (.com domain registering). Even when the dot-com bubble led many companies to bankruptcy, others survived and many others were born from the ashes of this period [4]. E-commerce has evolved and currently is classified in several ways. Mentioning two of these we have:

- **Depending on the seller or customer:** Business to Business (B2B), Business to Consumer (B2C), Business to Business to Consumer (B2B2C), Customer to Customer (C2C), and so forth [5].

- **Depending on the business model:** By subscription, by affiliation, online stores, and so forth.

Given that in this project we focus on B2C online stores here are some of the advantages of this type of e-commerce [6]:

- An easy search through a large database of products or services.

- Reduced maintenance costs.

- Geographical barrier removal.

- 24/7 Availability, and so forth.

2.1.1 International Context

The dot-com bubble made the financial markets present an evolution that emerged from the development of new computer technologies [4]. This progress was made thanks to the first web browsers, the advance of HTML 3.0, and so forth. All of this made rise the globalization and intercommunication of markets in real-time. Even with the experience acquired by the USA about e-commerce, currently, it is China that leads the top of the biggest e-commerce markets worldwide. It might look impossible but, China's e-commerce sales surpassed USAs in 2013 for the first time [7]. Since then, China's e-commerce market has been growing faster until reach 54.7% of global e-commerce sales [8]. China billed about 1934 billion dollars while the next five biggest competitors countries together reach around half of this with 1026 billion dollars as can be seen in Figure 2.1.

Top 10 Countries, Ranked by Retail Ecommerce Sales, 2018 & 2019
billions and % change

	2018	2019	% change
1. China*	$1,520.10	$1,934.78	27.3%
2. US	$514.84	$586.92	14.0%
3. UK	$127.98	$141.93	10.9%
4. Japan	$110.96	$115.40	4.0%
5. South Korea	$87.60	$103.48	18.1%
6. Germany	$75.93	$81.85	7.8%
7. France	$62.27	$69.43	11.5%
8. Canada	$41.12	$49.80	21.1%
9. India	$34.91	$46.05	31.9%
10. Russia	$22.68	$26.92	18.7%

*Note: includes products or services ordered using the internet via any device, regardless of the method of payment or fulfillment; excludes travel and event tickets, payments such as bill pay, taxes or money transfers, food services and drinking place sales, gambling and other vice good sales; *excludes Hong Kong*
Source: eMarketer, May 2019

T10308 www.eMarketer.com

FIGURE 2.1: Countries ranked by retail e-commerce sales [8].

2.1.2 Latin America and Ecuador Context

Latin America's e-commerce market is in its initial stage. However, given its growing, it is expected that the market reaches 82.33 billion dollars in 2022 as can be seen in Figure 2.2.

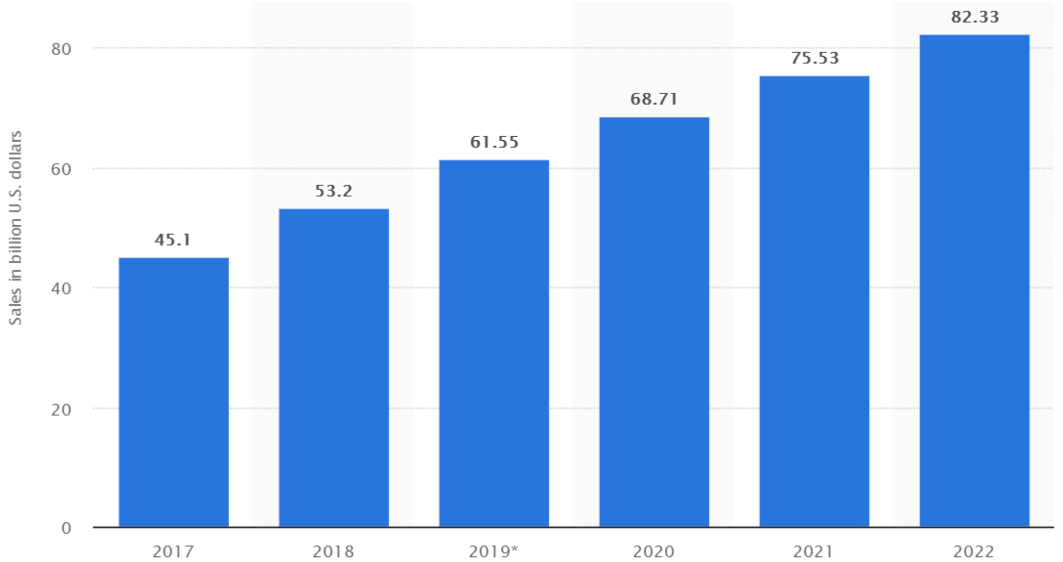

FIGURE 2.2: Latin America retail e-commerce sales [9].

In Ecuador the e-commerce has become immensely significant indeed, the e-commerce day event has been held in Ecuador year after year. Once occurred this event, a lot of information and statistical data on how to improve the Ecuadorian e-commerce are released. According to ARCOTEL in 2019, there were 11,5M of accounts that represent 67,07% of Ecuadorians [10]. This kind of information helps companies to prepare their staff to make internet business. Statistical information from INEC reports that the 55,32% of the Ecuadorians greater than 5 years old have used the internet at least one time in 2017 [11]. In the e-commerce day event of 2018, a research carried out by the UES University was presented. This study shows that, 35% of people in Ecuador utilize the internet to purchase products as can be seen in Figure 2.3.

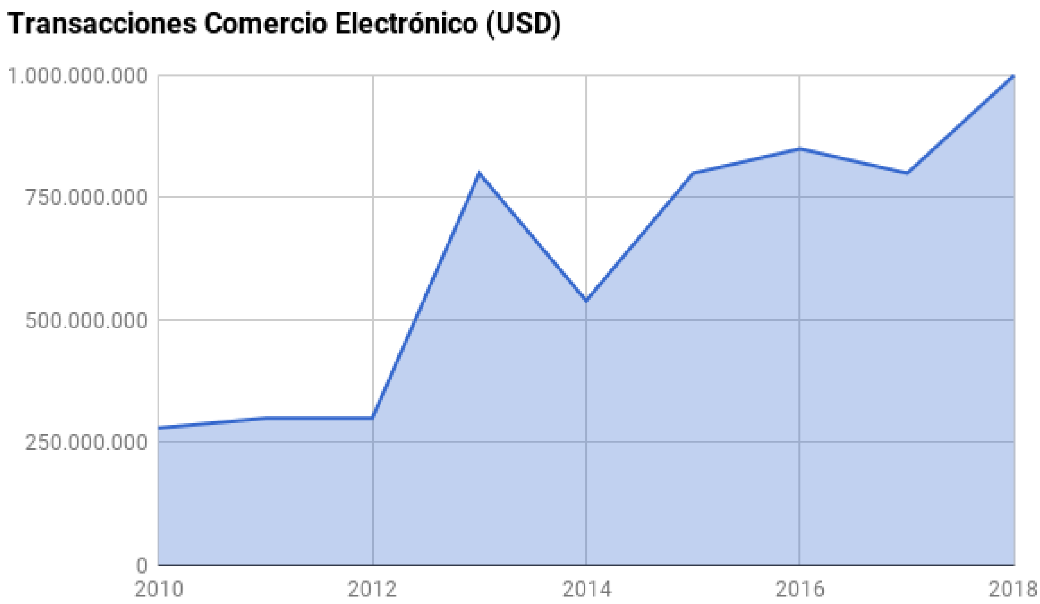

FIGURE 2.3: E-commerce purchasing in Ecuador [12].

2.2 Information Filtering

The information revolution is one of the newest, historical periods [13]. Currently, in the post-industrial society (information society) there is an excessive quantity of material on the internet. When people want to obtain relevant information or some data about any topic they navigate the web using a search engine through a browser. Nevertheless, nobody can read all the information available on all web pages that exist on the internet. Here filtering techniques help to detect the most relevant material for the users. A filter consists essentially of searching and selecting the most significant information for each user. This will reduce the time people spend searching by allowing them to use their time in other areas in a more productive way. These filters remove unrelated documents ordering and organizing redundant data in terms of relevancy. Its fundamental goal is to effectively manage the information overload. This can be achieved by making use of computerized techniques as decision trees, support vector machines, neural networks, Bayesian networks, linear discriminators, logistic regression and so forth. The rapid increase in the amount of information published on the web has coined terms such as Data Science, Big Data, Data Mining, Data Cleaning, etc.

Information filtering approaches are widely used (Eg. Tapestry was a mail system developed by Xerox in Palo Alto which used filters to avoid spam [14].) by enterprises according to their needs. These enterprises typically fall in the field of e-commerce, mail systems, search engines or music/video/movies streaming companies and others. It is worth mentioning that some technical research articles use information retrieval as an information filtering synonym. Nevertheless, even when both terms are highly related, there are some little differences that exist between these terms due to information filtering represent a specific type of information retrieval [15]. Nevertheless, generally the process of information filtering systems involve some common stages presented in Figure 2.4:

- **Searching:** The process begins with the user performing some searching (product search, web search, mail search, etc.) through some graphical interface provided.
- **Query Representation:** The search is transformed into a query, which is addressed to a database.
- **Information Database:** The database executes the query and retrieves all the resultant information.
- **Profile Matching:** A user profile that has the best match with the actual user is selected (Some systems use other types of matching to select an appropriate filter).
- **Filter Assignment:** Based on the previous match, the system selects an appropriate filter.
- **Information Filtering:** The information provided by the database is filtered or sorted using some algorithm present inside the filter.
- **Output:** The most relevant information is ranked and presented to the user.
- **Feedback:** Positive or negative feedback serves the system to evolve and improve itself through a learning and adaptation process making it more effective in future searches.

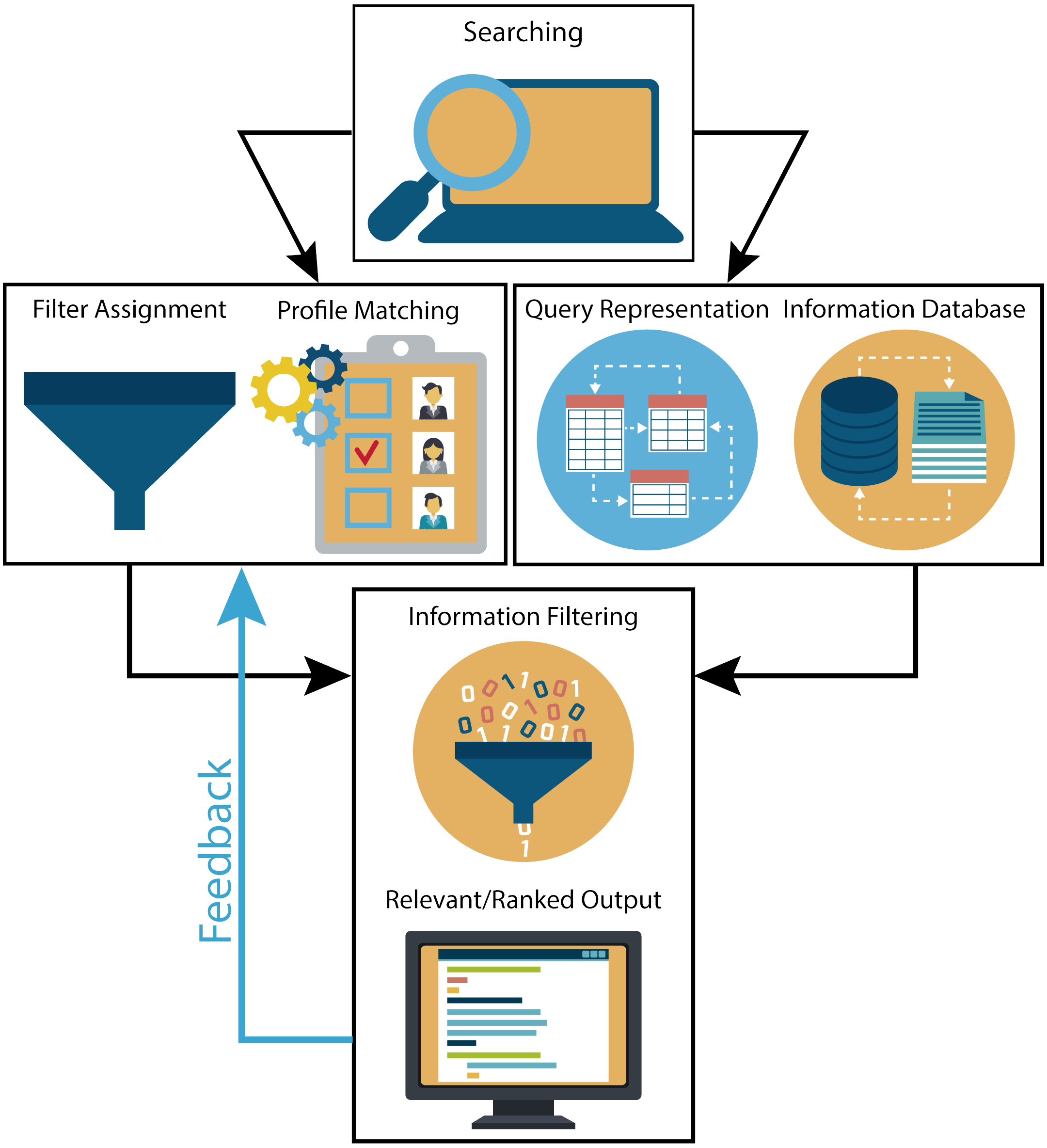

FIGURE 2.4: Information filtering process [15].

2.3 Recommendation Systems

In general, people face many decisions they must make in several aspects of their lives. Currently, with the growth of the internet the decision-making process has been unable to properly handle the massive volume of available data. Recommendation engines are tools employed to filter information for customers. This encourages them to focus on their interest avoiding large volumes of data that could cause them to lose what they might desire [16]. Recommendation engines help enterprises to increase average order value offering people products or services that they should acquire boosting sales and revenues for online merchants. Personalized interactions offered by recommendation engines generate a sense of customer satisfaction suggesting relevant products at the same time the customer shops.

2.3.1 State of the Art

- **Digital Guide Map Using Tiled Display and Recommendation Function.**

  This research published in 2011 built a digital guide map that could be used when people went shopping in a mall or a street. To achieve an effective digital map for public places, the authors implemented a recommendation system without needing pre-identification for public users. With this characteristic, the digital guide did not need to record users and recommend the user based on a probability density distribution. The distribution could be constructed from fragmentary usage historical data to calculate a similarity score between places [17].

- **Ecosystem of a shopping mall robot - Neel.**

  An article published in 2012 explains about an assistant robot for customers in a shopping center. The robot should help customers obtain offers and discounts based on historical data about their preferences. At the same time, the robot could connect and bring together high-potential customers and other distributors present in the mall [18].

- **Health-aware Food Recommender System.**

  Some people are trying to transform their life, improving their health. Food consuming habits are key aspects that non-healthier people have to change. This paper explains about a mobile app that recommends recipes to users using a recommender system. The recommendations are made based on user's preferences taking into account its health, diseases or allergies. A recommendation engine like this can be used to reduce the risk of diabetes, obesity, and so forth [19].

- **The Netflix Recommender System: Algorithms, Business Value, and Innovation.**

  In 2007, Netflix created a public challenge for those who could improve their platform by developing a better recommendation system. The prize for the winning team would be $1 Million. In 2008 one of the teams invented an algorithm that improved the Netflix recommendation system by 10.06% and they were the winners of the grand prize. Nevertheless, Netflix did not implement the algorithm (commercial strategies were taken into account to make this decision) due to the implementation did not appear to justify the engineering effort required to bring it into a production environment. Other algorithms were implemented into the Netflix's recommendation system and this article discusses several of them describing its business purpose [20].

2.3.2 How Does a Recommendation Engine Work?

Recommendation engines build user behavior models about the consumer types through some common phases:

- **Collection phase:** Data are the main component in a recommendation system. Here is where all the data generated by the users as preferences or behavior within the website is collected and saved in a relational or not relational database. During this process, the information collected is evaluated and classified according to its relevance to the project. Next, a standardization technique must be applied to the data. If a normalization is required to adapt the data and pass to the subsequent phase, then here is where the standardization is executed.

- **Learning phase:** This phase seeks to employ some learning paradigms present in the field of machine learning. Supervised, unsupervised and reinforcement learning attend the three main groups where the different learning algorithms are classified. The objective of these algorithms is to transform the perceived information into valuable knowledge. A learning algorithm captures the data, filter it, and extract the relevant information to produce an appropriate recommendation based on user interests or likes. The learning algorithm is implemented according to the recommendation technique used in the recommendation system.

- **Prediction/Recommendation:** Finally, the recommendation is presented to the user and depending on a received feedback the algorithm is improved for future recommendations.

All recommendation systems work focused on the same objective. They can differ depending on how they implement their distinct phases. Firstly, depending on the data used the recommendation system can be based on implicit or explicit feedback data set. Next, the recommendation systems could differ according to the learning algorithm implemented. Content-based, collaborative, and hybrid filtering are the main models used in the learning phase.

2.3.3 Type of Feedback Data Sets

Feedback word is a synonym of reaction or opinion about a topic. The main objective of the feedback is to use the result obtained from an activity in a system to modify and correct the system. Given that feedback analysis and usage has been constantly improving, companies have laid their interests on it. This has allowed them to save costs and increase their productivity. Feedback can be positive or negative and both are very important to improve a system [21]. Some of the principal advantages to use feedback are loyalty, retention, and comprehension.

The feedback can be labeled as positive when the system is growing correctly and continuously. This type of feedback allows the system knows that it is following the appropriate path and that users like it and the way how it is growing. It is worth to mention that positive feedback should not be considered a synonym of a perfect system. The improvement process is a very long way walk and new features or experimenting is the correct way to follow when all look well done. On the other hand, the feedback can also be negative. This type of feedback can be employed as support to correct the system and improve it until achieving an objective or positive feedback.

To build an accurate recommendation system, relevant data about the preferences of the users need to be collected. There are two particular approaches in which we can base the collected data to employ in a recommendation system. These approaches are implicit or explicit data sets. It is important to know both types of data, their characteristics, pros, and cons to know how or when to apply depending on the situation. Generally, the majority of recommending system implementations rely on explicit data sets while implicit data sets are rarely used [22].

Explicit Feedback

Explicit feedback represents all the information that the user provides consciously through comment reviews, internal messages, star rating, thumb rating and so forth. The objective of this type of feedback is to let the user express its opinion about a product or service to improve it for future situations. Explicit data are information that a customer directly communicates to a website. It can go from the customer's age or sex to specific comments or opinions on a product. Some of the most traditional techniques to collect explicit feedback can be identified in Figure 2.5.

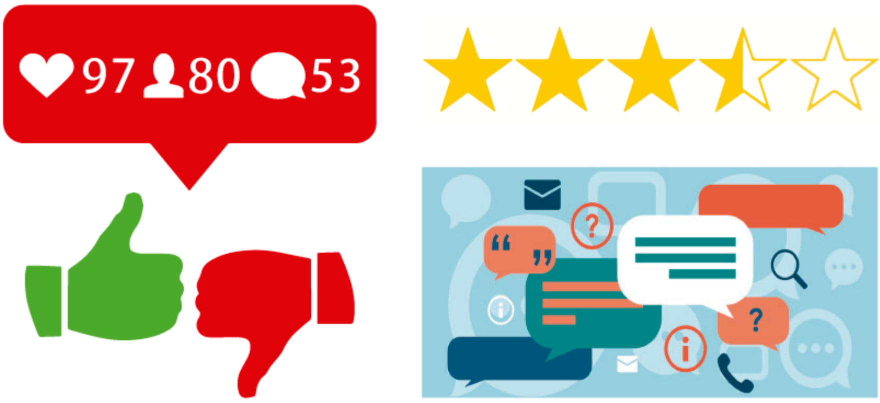

FIGURE 2.5: Explicit feedback [21].

- **Star Rating:** The star rating enables users to provide their opinion implementing visual support. This type of feedback collection tries to be accurate giving the user the chance to specify how much a product/service like it in a specific way. Finally, an average result is calculated and the system can employ it to improve its accuracy.

- **Thumbs:** This is a binary thumbs-up, thumbs-down rating system. This type of rating tries to avoid complexity in decision making, focused only on better results about what users like or dislike regarding a product/service.

- **Comment Reviews:** Given that the previous rating systems sometimes do not express the genuine opinion of users about a product/service. In that case, comments feedback could represent a proper chance to approach what exactly people are thinking to prevent marketers to guess. Marketers indeed make guesses based on data, but data is not the same as an actual comment from the customer about the product/service. Indeed consumers read online reviews when purchasing, then review comments means a better chance of users purchasing.

- **Hearts:** This type of rating is based on let companies know that customers like products even if they are unable to buy these at the moment of the rating. A notable application of this type of rating is the wish list which can be found in many e-commerce websites. These act as a middle ground between purchasing and forgetting, reminding users that they maintain favorite products/services for purchasing on the next visits to the store.

Implicit Feedback

On many occasions, it is very difficult to obtain explicit feedback due to the user's unwillingness. Nevertheless, here is where implicit feedback data sets help to model user's behavior. Implicit feedback data sets do not depend on customer opinions about a product or anything else, but on customer behavior on a website [16]. Implicit feedback is based on extrapolations about purchases, visited products, abandoned shopping carts, spending time on a product, search history and so forth. That is why implicit feedback overcome hugely to explicit feedback data sets [23].

Youtube, Spotify, Netflix, and other content-based companies possess very well built recommendation systems based on videos in which people interact. Depending on if a user watches a video completely or partially a recommendation score is given to this action. If users send, embed, download, comment or share a video in a social media network the recommendation system assigns a different punctuation to the action. At the end, the system computes the sum of all the actions according to their respective importance and makes a recommendation prompted to the screen all in real-time. High scores imply high possibilities that other users would like to see the video. Implicit feedback can be very useful, besides there are a lot of other kinds of applications based on mouse movements, screen touch, time spent on a given screen and so forth as can be seen in Figure 2.6. It is important to make a reflection about the worth behind these kinds of systems. Users commonly feel annoying to be asked on completing surveys or making raking scoring to develop their preferences. For these reasons implicit feedback is the option that was chosen to apply to this research.

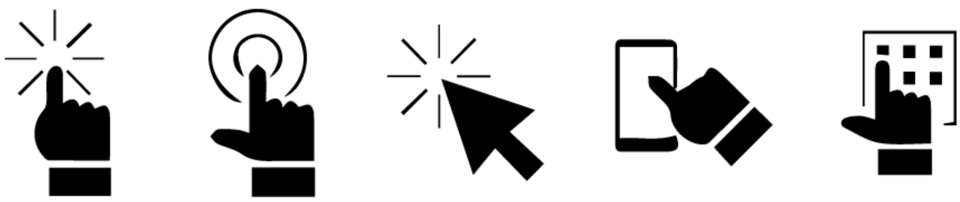

FIGURE 2.6: Implicit feedback [16].

2.3.4 Classification of Recommendations Techniques

It is necessary to know where the recommendation system are applied. Depending on the context, a recommendation model that perfectly fits the problem to be solved can be chosen. Current recommendation systems are classified into three main groups Content-based, Collaborative and Hybrid filtering. The classification tree for the recommendation systems classification based on these algorithms is presented in Figure 2.7.

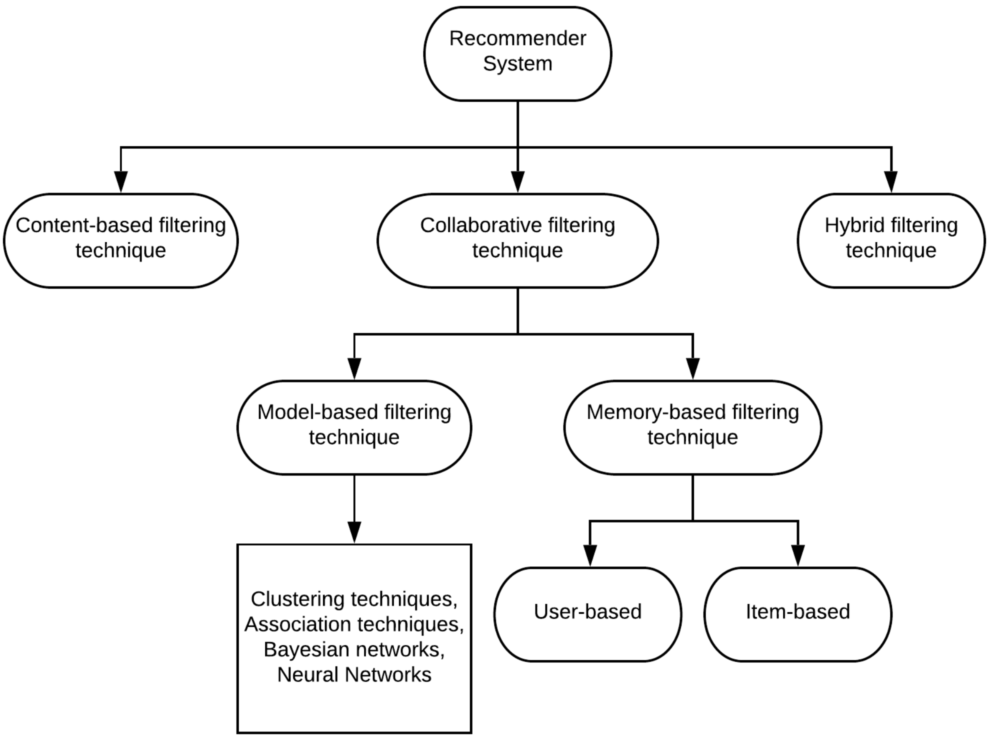

FIGURE 2.7: General recommendation techniques classification retrieved from [16].

Content-Based Recommending Systems

This type of recommendation is based on similarities between features of the items and characteristics of the users. Most of them use item keywords to generate recommendations that a user might like based on items that the user did consume previously. For each user, the recommendation is personalized to its preferences. To build a strong content-based recommendation system it is essential to possess a well-structured database of products, classified by its features. It is equally important to collect preferences or user behavior. Measuring the relationship level between products the recommendation could achieve excellent accuracy as can be seen in the example in Figure 2.8. Nevertheless, even when the content-based looks like a logical approach (because it considers several aspects of each user or each item) it is very difficult to implement because of the item/user representation problem. For example, how do we calculate the similarity score of distinct products like a sofa vs a chair or sofa vs a bed?.

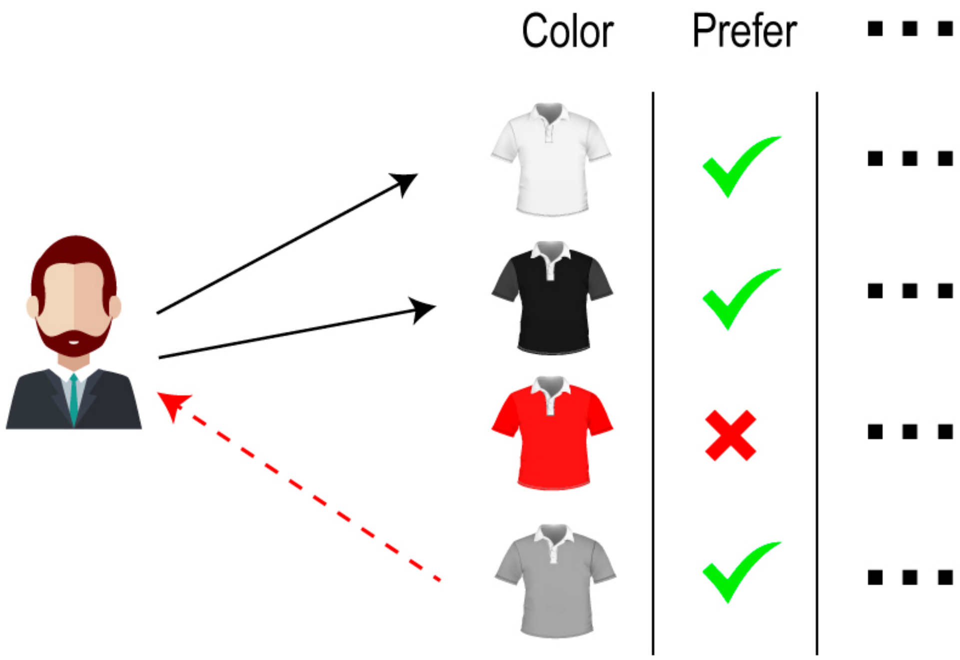

FIGURE 2.8: Content-based recommendation example [16].

As can be seen, there are several things to take into account before to select this type of recommender system.

A summary of its pros and cons can be seen below:

Pros

- New items are easier to recommend because the recommendation was trained based on features.
- Content representations are varied and allow us to use diverse techniques like text processing, semantic information, and inference.
- Recommendation can be explained easier because these are based on content, keywords, and features.

Cons

- Diverse recommendation is affected because the system just recommends items similar to previously consumed by the user. This creates a filter bubble with over-specialization recommendations.
- Without enough information between users and products, the recommending system could not be precise.

Collaborative Filtering Recommending Systems

Notwithstanding this is an old-school algorithm for recommending, it is still a very effective one. This type of recommending system is based on the idea that items must be recommended according to similarities in purchasing between users. Similar ratings are assigned to consumed items and then users who share common purchasing behaviors are grouped as can be seen in Figure 2.9. Collaborative filtering uses neighborhood user-user or item-item profiles' distance to recommend based on these distances. Closer users might share preferences and items that one user has purchase must be recommended to its neighborhood who has not purchase that item yet [24].

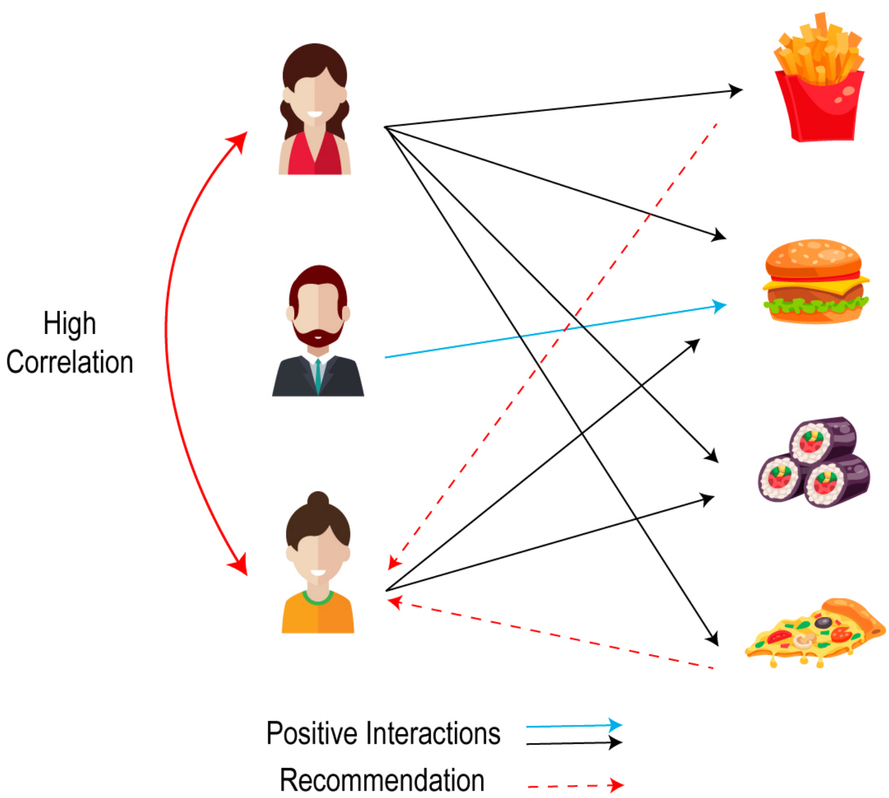

FIGURE 2.9: Collaborative filtering recommendation [16].

Pros

- Very effective with a sufficient amount of data is available.
- Neither products' information nor the users' profile is required for building recommendations.
- Inference comes from user's behavior.
- Two flavors: Product-based or user-based.
- Avoidance of filter bubble problem.

Cons

- Sparsity: This type of recommendation system is usually used with ranking or purchasing. With a lot of items and too many users, it is unlikely that most users have classified or purchased a large part of the items, then it is highly probable that just a few items have been tried by a few users causing a huge lack of information making the recommendation not to be precise.
- Scalability: As it increases the number of users, the computational cost to find the nearest neighbors also increases. Million of users with thousands of products could be very difficult to analyze.
- Cold Start: New users without information can not be compared with the rest of the users because of a lack of neighborhood.
- New-Item: This problem is extremely similar to the previous one. New items will lack of purchasing. Then it is very difficult for this item to be offered to users taking into account that nobody has purchased it already.
- Users with peculiar purchasing behaviors are very complex to be assigned to a neighborhood that fits perfectly with recommendations for him.

As can be seen in Figure 2.10 there are two types of collaborative filtering approaches: **memory-based** and **model-based** approaches.

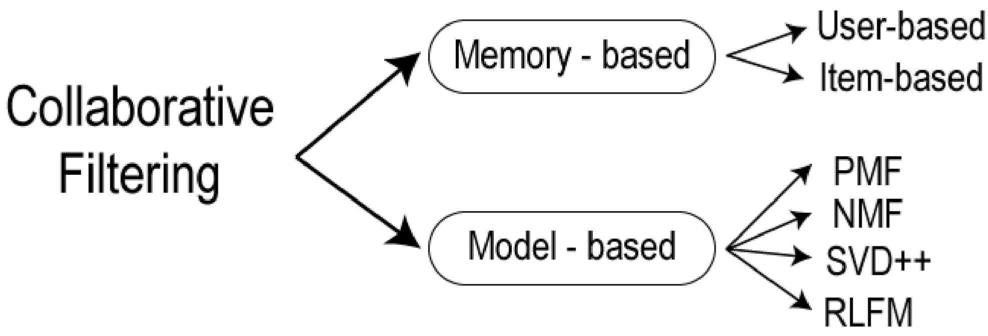

FIGURE 2.10: Collaborative filtering classification techniques [16].

Memory-Based Approach

Suppose user A has purchased the items with id: 1, 2, 3, 4, 5, and 6. User B has purchased the items with id: 1, 2, 3, 5 and 6, but not 4. Because both users purchased five of the same six items (items with id: 1,2,3,5,6), it can be said that they share some purchasing preferences. User A liked item 4, then, probably, user B would also like item 4 if the user were aware of its existence. This is where the recommendation engine shows its value, because it informs User B about item 4, taking into account the interest of the user.

Memory-based filtering follows the intuitive way, finding similarities in items or users consuming or rating behavior using neighborhood. But this approach has several challenges to overcome and a higher computational cost.

Model-Based Approach

Model-based model collaborative filtering approach came to rescue overpassing the challenges existing on the memory-based models. This new approach follows the matrix factorization instead of neighborhood solution. A model-based approach use historical data to establish a model first and then do inference when recommending.

Some of the model-based techniques can be:

- Probabilistic matrix factorization (PMF): This is a classic probabilistic linear model with Gaussian observation noise.

- Non-negative matrix factorization (NMF): This fits the low-rank matrix factorization framework with additional non-negativity constrains.

- Singular Value Decomposition (SVD++): This makes uses of implicit feedback to refer to any kind of user's history information that can help to indicate the preference of users.

- Regression-based latent factor model (RLFM): This technique uses the side information. For example. Demographic data and items features.

Hybrid Filtering Recommending Systems

Given that the recommendation system is applied depending on its context, data or the problem to be solved. Some of the strongest and precise recommendation systems chose to develop a suitable combination of recommendation models that fits perfectly with their companies. They use a type of recommendation system depending on the situation and its structure finally might look like the one depicted in Figure 2.11.

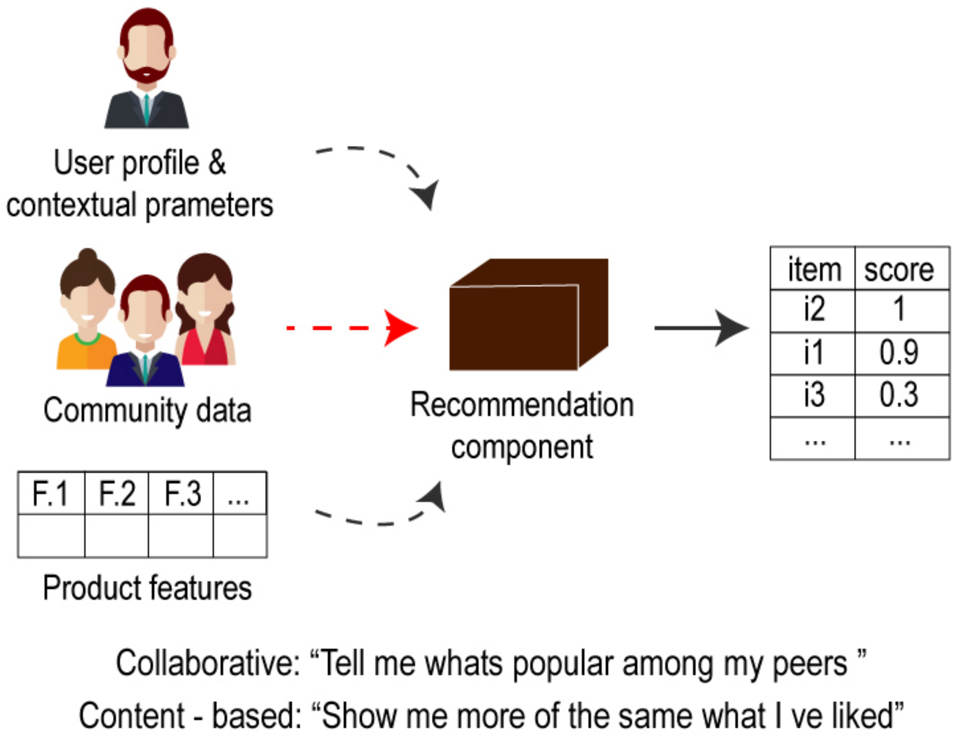

FIGURE 2.11: Hybrid recommendation system [16].

The implementation of this kind of recommendation systems is extremely data dependant. In most cases (due to the vastly alternative approaches that can be implemented), the recommendation component can be seen as a black box where both algorithms are mixed to make a proper recommendation to each user, as can be seen in Figure 2.12.

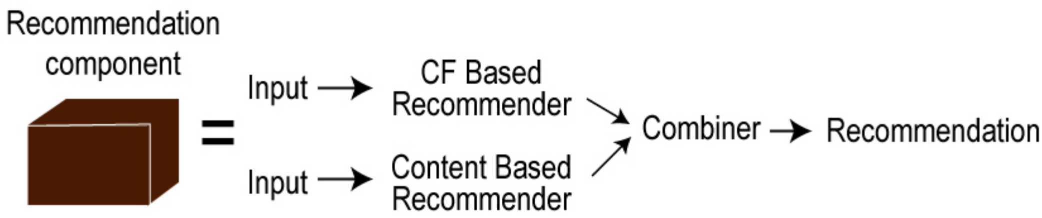

FIGURE 2.12: Hybrid recommendation system black box [16].

2.3.5 Evaluation Metrics

Companies interested in recommender systems frequently worry about, how well would work a recommendation system built for them? Evaluation constitutes an essential part of the developing process inside a recommending system. There are several types of evaluation techniques that can be used to measure the accuracy of a recommending system. It will depend on the data set, type of recommending system or the recommendation result. The most typical type of evaluation takes a set of user preferences and divides them into a training and a testing set. Given the training set as input, the objective of the recommending system is to return a new list of recommended products that a user might like. Comparing the generated recommendation with the user preferences in the testing set, it is straightforward to tell how much a user could appreciate the recommendation. The better the recommender predicts the product that appears in the test set for a given user, the more confidence that the recommender is generating good quality recommendations.

Most recommender systems that work with explicit data set use evaluation metrics as Root Mean Square Error (RMSE) or Mean Absolute Error (MAE). These metrics are used when predicts the possible rating assigned by a user for a determined product. Traditionally, explicit ratings go from zero to five expressing how much the user enjoyed the product. But for implicit feedback and top-N recommendations as the one proposed in this research, other two techniques to measure the efficiency of the recommenders are available.

Hit Rate

Hit rate evaluation is a remarkably simple evaluation method that can be used to identify how good a top-n recommender system is [25]. This evaluation is based on the idea that if a product preferred by the user appears on the top-n recommended items list, then it is considered a "Hit".

Hit Rate Algorithm:

- Generate top-n recommendations for all the users who appear in the testing data set.
- If something that the user rated from the testing set is in the top-10 recommendation list then this success is considered a hit.
- To calculate the whole hit rate of the system it is necessary to just add the total of hits and divide it for the number of users.
  $HitRate = \#hits/\#users$

As can be seen, the hit rate can be easy to understand but it just measures the train/test for individual ratings. To evaluate the quality of a top-n recommendation list for individual users, the previous implementation is not enough. A better way to evaluate the top-n list it is necessary to also use a method called leave-one-out cross-validation.

The algorithm now could be as next:

- Compute the top-n recommendations for each user in the training data set.
- Discard one of these items (This technique is called leave-one-out cross-validation.) totally from the training set.
- Use all other items to feed the recommender and ask for a top 10 recommendation list for each user again. Then, check that the recommender system has recommended the item that was left out from the first generated list in the pre-training phase.
- If the item discarded appears in the top 10 recommendation list then record it as a "hit" otherwise, this is considered a "miss".

Now the evaluation metric provides a more accurate result. To calculate the final hit rate, we just divide the number of hits over the number of unique users in the testing set.

Normalized Discounted Cumulative Gain (NDCG)

NDCG is an evaluation metric from the information retrieval field and is based on measure the performance of a ranking evaluation [26] ("In a recommendation list, products must appear sorted according to their relevance.").

NDCG is a variant of "Discounted Cumulative Gain" (DCG). The last one measure the usefulness of a product based on its position in a recommendation list. DCG accumulates the gain for a top-n recommendation from the top result to the bottom and the gain of each result is discounted at lower ranks. For example:

Suppose there is a list of products. Then, for a given user each product has its importance rank. This rank possesses an importance scale where zero means that the product is not relevant for the user and five means higher importance for the user. Let a set of recommended products to be:

$$RL = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\}, \tag{2.1}$$

where the relevance score given by a user for each product is:

$$RL = \{3, 4, 1, 3, 5, 1, 4, 1, 0, 1\}. \tag{2.2}$$

The Cumulative Gain (CG) formula for a top-n list is given by

$$CG_n = \sum_{i=1}^{n} rel_i. \tag{2.3}$$

Then, the CG for our recommendation list is:

$$CG_{10} = \sum_{i=1}^{10} rel_i = 3 + 4 + 1 + 3 + 5 + 1 + 4 + 1 + 0 + 1 = 23. \tag{2.4}$$

As can be seen, the cumulative gain does not depend on the order that the items have into the list. For this reason, DCG uses a logarithm scale to cause reduction depending on the order list as explained next.

$$DCG_n = \sum_{i=1}^{n} \frac{rel_i}{log_2(i+1)}. \tag{2.5}$$

Then the DCG for our recommendation list is calculated and appear on the Table 2.1:

i	rel_i	$log_2(i+1)$	$\frac{rel_i}{log_2(i+1)}$
1	3	1	3
2	4	1.585	2.524
3	1	2	0.5
4	3	2.322	1.292
5	5	2.585	1.934
6	1	2.807	0.356
7	4	3	1.425
8	1	3.17	0.333
9	0	3.322	0
10	1	3.459	0.301
DCG_{10}			**11.665**

TABLE 2.1: DCG calculation example.

Easily can be seen that if we change the position of the elements on the list, the total discounted cumulative gain will also change. But the main topic was the normalized version of the DCG metric. This is achieved using the ideal order to decrease the sort of relevance score. The DCG is divided by the ideal DCG and the final value will appear normalized following the next formula:

$$nDCG_n = \frac{\sum_{i=1}^{n} DCG_i}{IDCG_n}. \tag{2.6}$$

The ideal DCG for the proposed example is:

$$RL = \{P_1 = 5, P_2 = 4, P_3 = 4, P_4 = 3, P_5 = 3, P_6 = 1, P_7 = 1, P_8 = 1, P_9 = 1, P_{10} = 0\}, \tag{2.7}$$

Finally the $nDCG$ is calculated as follows:

$$nDCG_{10} = \frac{11.665}{13.292} = 0.878 \tag{2.8}$$

2.4 Machine Learning

Currently, machine learning has achieved extraordinary popularity [27] given that emerging technologies are employing it into several applications [28]. The possibility of being using it in one way or the other even without noticing is high. Pattern recognition and learning from data have resulted in better decision making and autonomous processes [29]. Some popular emerging technologies making use of machine learning as part of their systems are:

- Self-driving vehicles from companies as Waymo.
- Virtual personal assistants as Siri.
- Face recognition as the one provided by Facebook.
- Email spam filtering from popular email services as Gmail, etc.

As can be seen, there are a lot of applications where machine learning has been properly applied. Huge product/service companies as Amazon, Netflix, Walmart, and others have did not think twice to also implement it.

The broad range of machine learning algorithms can be divided into three main groups [30]:

- **Supervised Learning:**

  In this type of algorithms, a predictive model is generated based on its input/output data. The supervised part on its name comes from the idea that these types of algorithms need to be previously labeled and classified [30]. This set of labeled data will be the training set used to adjust the initial model. Through several iterations the algorithm adjusts its values, learning to classify and predict comparing its result with the label provided.

- **Unsupervised Learning:**

  Unsupervised learning algorithms work a little differently than the previously explained. In this case, the labels on the data are not necessary, and feedback to adjust the model is not used. This type of algorithm group/segment provided data taking into account similarities and differences between the entries [31]. These types of algorithms are used frequently to discover unknown patterns and relationships between data points in data set.

- **Reinforcement Learning:**

  This learning approach defines a model based on test error, trying to maximize the "reward" system given by the training environment. The model differentiates itself from the supervised approach because the previous one employs a set of labeled training data whereas reinforcement is about making decisions sequentially. The model selects the most efficient path to achieve a result based on reinforcement and maximizing the reward [31].

Chapter 3

Methodology

3.1 Introduction

The kind of research applied in this thesis document is a comparative study. The research begins collecting generic data from the internet, to then structure and clean it. Several recommendation algorithms are studied and selected for the comparison study. The different algorithms are implemented using Python programming language and with the help of Keras machine learning libraries. The different algorithms are also measured using HR and NDCG testing. In the end, the case study data is used on the same algorithms previously developed for the generic data set. The research analyzes the behavior with the real data providing a brief explanation about it. Some recommendations for future improvements are also provided. Taking into account the previous explanation, the phases for the development of the project are briefly stated in Figure 3.1.

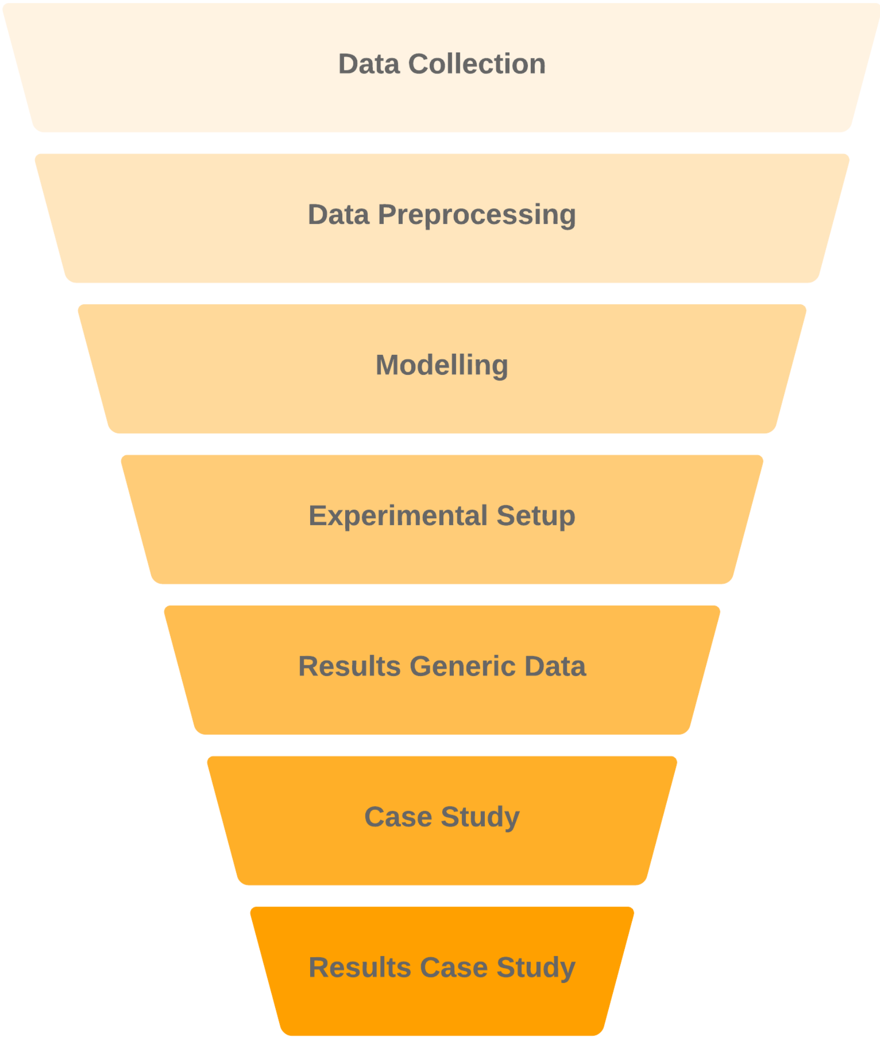

FIGURE 3.1: Methodology structure.

3.2 The Data

This data set was taken from the Retail Rocket Recommender System post that can be found on the Kaggle web page: https://www.kaggle.com/. It consists of four files:

- `events.csv` is the file that stores the user behavior.
- `item_properties_part1.sv` and `item_properties_part2.sv` are files that describes all products.

- `category_tree.sv` is the file which describe the relationship between products.

The data has been collected from a real-world e-commerce website. It is raw data without any content transformations, however, all values are hashed due to confidential issues. The behavior data, i.e. events like clicks, add to carts, transactions, represent interactions that were collected over a period of 4.5 months. A visitor can produce three types of events, namely "view", "addtocart" or "transaction". In total there are 2 664 312 views, 69 332 add to carts and 22 457 transactions produced by 1'407.580 unique visitors as can be seen in Figure 3.2. Given that the recommending system makes use of an implicit data set, then the project must focus its attention on the `events.csv` file. In the first place, it is necessary to do an exploratory data analysis. Let's analyze the `events.csv` file to structure and clean it.

```
events['event'].value_counts()
```

```
view           2664312
addtocart        69332
transaction      22457
Name: event, dtype: int64
```

FIGURE 3.2: Number of events registered.

`events.csv` file is composed of 2 756 101 events divided in five columns: "timestamp", "visitorid", "event", "itemid" and "transactionid" as shown in Figure 3.3.

```
events.head(3)
```

	timestamp	visitorid	event	itemid	transactionid
1459312	2015-09-18 02:59:47	1287495	view	98299	NaN
1454122	2015-09-18 02:59:41	622226	view	345308	NaN
1456783	2015-09-18 02:59:34	255126	view	47467	NaN

+ Code + Markdown

```
events.shape
```

```
(2756101, 5)
```

FIGURE 3.3: Events table.

Figure 3.4 graphically shows you the number of implicit events registered in the data set previously explained.

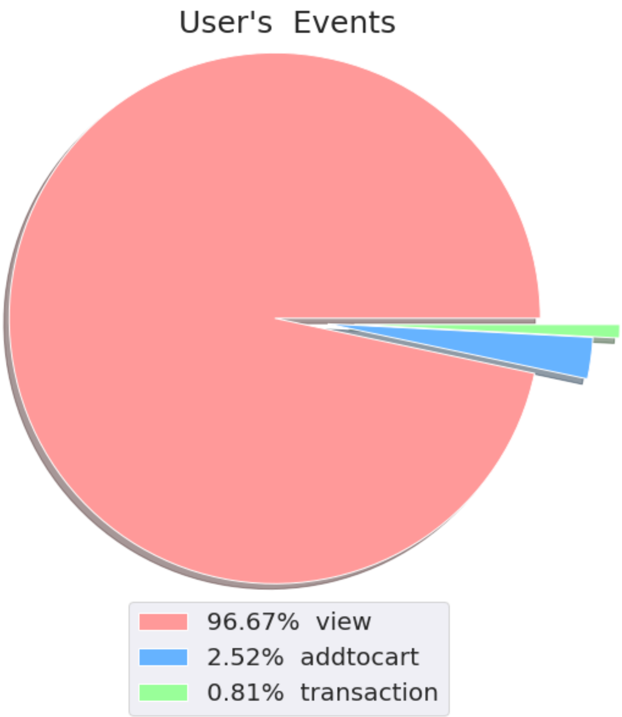

FIGURE 3.4: Graphical representation of user events data set.

As can be seen, the number of "view" events dominates over add to cart and transaction events. To avoid useless data it is necessary to do some cleaning over the data set. To achieve this it is necessary to filter all the users who just have viewed or added products to the cart. Cleaning the data in this way makes the recommender system have better accuracy. Now the recommendation is based just on users who have purchased items (transaction event = 1). The new plot can be seen in Figure 3.5.

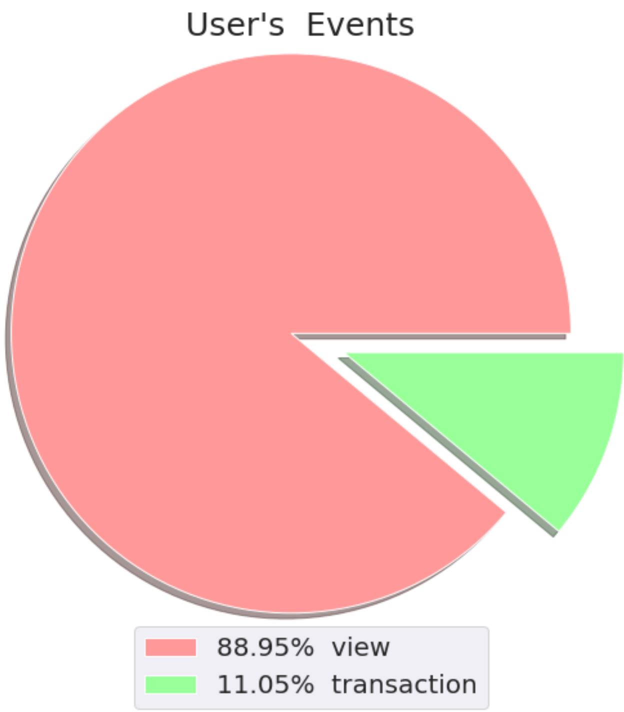

FIGURE 3.5: User events data set after cleaning.

3.3 Modelling

In the first place, it is necessary to choose a type of recommender system. Given that the data used are based on implicit feedback, the proper option for this project is a recommender system based on collaborative filtering. The project adopted the memory-based way because of its easy to implement with great accuracy results. In this part, the project explores some matrix models as Matrix Factorization, Non-negative Matrix Factorization, Neural Network Factorization and a model developed as an API for Implicit feedback. As previously explained there are several types of evaluation methods. Given that most websites do not offer a single recommendation but lists of recommendations, the evaluation methods used in this project are focused on accuracy and precision. Then, Hit Ratio and NDCG, both evaluation metrics previously explained are implemented. For the testing set, just users with more than two purchases are selected as testing users. The number of users who have purchased more than two items are 8,76%. The training-test splitting can be seen in Figure 3.6.

The item used for evaluation will be the last item these users have purchased. For example: Items purchased by user **953371** are **[270383, 277943, 173433, 9975, 293687]** then the testing item should be **293687** while the rest will belong the training set. A success event or a "hit" will be counted if the test item is provided in the top 10 recommendation list. The 'ndcg' on the other hand will evaluate the position of the item inside the recommendation list.

```
df_train,df_test, retrieve = train_test_split_time(usefulData,'visitorid','itemid','numevent')
```

```
train_test_split succeed!! with df_train shape:(200635,3), df_test shape:(2576,3)
```

FIGURE 3.6: Training and testing data set splitting.

3.3.1 Experiment 1: Matrix Factorization

Matrix factorization or MF for short is a simple and classical algorithm used in recommendation systems. MF represents users' feedback collected in the form of a matrix. In this matrix, each row represents a single user while the columns represent the different products. Matrix factorization makes inferences about user's preference over items from other users' preferences. If the estimated preference is high then the system recommends the product to the user. Matrix factorization works by decomposing the user-item matrix into the product of two rectangular matrices. Then dot product between these matrices provides us an approximation (the preference score) for all missing values of each user in the user-item matrix. In this project, an iterative optimization based on gradient descent to minimize the error in the prediction is applied. The iterative process shows that the recommender system achieved a 99% accuracy in the Hit ratio test and 89% possibilities that the ndcg value is close to the ideal value.

Implementation

Let $\mathbf{X_{up}}$ be the $user - item$ data matrix, where x_{ij} is the representative value of the $i - th$ user given to the $j - th$ product:

$$X_{up} = \begin{pmatrix} x_{11} & & \dots & & x_{1m} \\ & \ddots & & & \\ \vdots & & x_{ij} & & \vdots \\ & & & \ddots & \\ x_{n1} & & \dots & & x_{nm} \end{pmatrix}_{n \times m} . \tag{3.1}$$

It should be noticed that not all entries ij of $\mathbf{X}$ are filled since the users are not necessarily providing preference values for all the products. This means that some x_{ij} are unavailable.

Let $\{\mathbf{W_u^{(l)}}\}_{\mathbf{l=1}}^{\mathbf{n}}$ and $\{\mathbf{W_p^{(k)}}\}_{\mathbf{k=1}}^{\mathbf{m}}$ be the set of weighting vectors related to users and products respectively, defined as follows:

$$\begin{Bmatrix} \mathbf{w_u^1} \\ \mathbf{w_u^2} \\ \vdots \\ \mathbf{w_u^n} \end{Bmatrix}, \tag{3.2}$$

and

$$\begin{Bmatrix} \mathbf{w_p^1} \\ \mathbf{w_p^2} \\ \vdots \\ \mathbf{w_p^n} \end{Bmatrix}, \tag{3.3}$$

where $\mathbf{w_u^l}, \mathbf{w_p^k} \in \mathbb{R}^{\mathbf{d}}$ and d is empirically setted.

Accordingly, $\hat{\mathbf{X}}$ data matrix approximation can be written as follows:

$$\hat{\mathbf{X}}_{up} = \begin{pmatrix} \hat{x}_{11} & & \dots & & \hat{x}_{1m} \\ & \ddots & & & \\ \vdots & & \hat{x}_{ij} & & \vdots \\ & & & \ddots & \\ \hat{x}_{n1} & & \dots & & \hat{x}_{nm} \end{pmatrix}_{n \times m} , \tag{3.4}$$

where $\hat{x}_{ij} = \mathbf{w_u^i} \cdot \mathbf{w_p^j}$ and $\cdot$ stands for dot product.

Data values hold in $\mathbf{X}$ are compared in an iterative process with their respective values in $\hat{X}$. Such an iterative process follows the Adam optimizer approach to minimize the mean square error between $\hat{x}_{ij}$ and x_{ij} for the available values in X.

3.3.2 Experiment 2: Non-Negative Matrix Factorization

Non Negative Matrix Factorization follows the same process as the previous experiment but with a non-negative constraint over the weighting vectors to avoid negative values in the approximation:

$$\begin{Bmatrix} \mathbf{w_u^1} \\ \mathbf{w_u^2} \\ \vdots \\ \mathbf{w_u^n} \end{Bmatrix} > 0, \tag{3.5}$$

and

$$\begin{Bmatrix} \mathbf{w_p^1} \\ \mathbf{w_p^2} \\ \vdots \\ \mathbf{w_p^n} \end{Bmatrix} > 0, \tag{3.6}$$

As might be expected, usually it is impossible to reconstruct the initial matrix precisely. But in this case "as close as possible" to the initial matrix also provide valuable information. At the very end is easy to select a user and sort its item values to recommend those with the highest values. In this experiment, the Hit ratio demonstrated a more precise accuracy than the previous case while the ndcg was not as good as the simple matrix factorization algorithm.

3.3.3 Experiment 3: Neural Matrix Factorization Approach

In this experiment, a neural network matrix factorization approach has been implemented. The method was taken from another recommendation system paper [32]. The author decided to implement and improve the first matrix factorization technique previously explained by making a fusion between the general matrix factorization and a neural network architecture. The chosen neural network architecture was a multi layer perceptron. From the paper, we take that the general matrix factorization model applies a linear kernel to model the latent feature interactions while the multi-layer perceptron uses a non-linear kernel to learn the interactions from the data.

The first part comes from embedding the user and item respectively. The General Matrix Factorization (GMF) part uses the element-wise product between the latent representations. The multi-layer perceptron architecture try to learn the relationship between user and item embeddings. At the very end, the neural network concatenates the GMF result with the MLP result and try to predict the score value between a user and an item. In this way, the Matrix factorization technique has been modified to adapt the two methods into one. This new approach could try to learn the linear and non-linear interactions between users and items. The final Neural Matrix Factorization model architecture can be seen in Figure 3.7

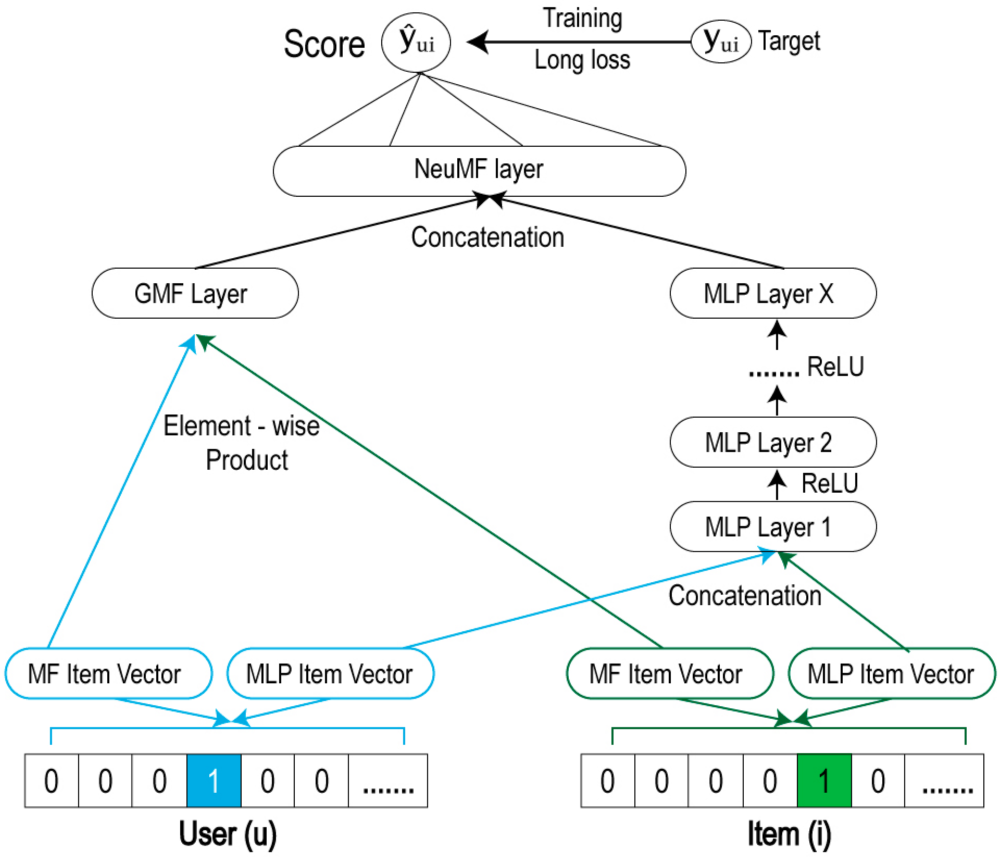

FIGURE 3.7: Final architecture [32].

3.3.4 Experiment 4: Implicit Feedback Matrix Factorization Approach

In this experiment, another similar matrix factorization technique was applied. Most of the implementations of matrix latent factorization are applied to explicit feedback data sets. In these experiments the model tries to avoid over-fitting through the following regularizer:

$$\min_{\mathbf{x}_*,\mathbf{y}_*} \sum_{r_{ui} \text{ is known}} \left(r_{ui} - \mathbf{x}_{\mathbf{u}}^{\mathbf{T}}\mathbf{y}_{\mathbf{i}}\right)^2 + \lambda \left(\sum_u \|\mathbf{x}_{\mathbf{u}}\|^2 + \sum_i \|\mathbf{y}_{\mathbf{i}}\|^2\right), \tag{3.7}$$

where λ is a data dependant variable used for regularizing the model and the parameters are learned by stochastic gradient descendant.

A group of researchers took the previous idea and modified it. They adapted for implicit feedback data sets. The paper "Collaborative Filtering for Implicit Feedback Datasets" [33] explains that recommendation systems based on explicit data indicate preference while the ones based on implicit feedback indicate confidence. Bearing this in mind the new cost function to minimize -in this case- can be expressed as:

$$\min_{\mathbf{x}_*,\mathbf{y}_*} \sum_{u,i} c_{ui} \left(p_{ui} - \mathbf{x}_\mathbf{u}^\mathbf{T} \mathbf{y}_\mathbf{i} \right)^2 + \lambda \left(\sum_u \|\mathbf{x}_\mathbf{u}\|^2 + \sum_i \|\mathbf{y}_\mathbf{i}\|^2 \right), \tag{3.8}$$

where c_{ui} measures the confidence in the preference relationship p_{ui} between the user u and the item i. The $\mathbf{x}_\mathbf{u}^\mathbf{T}$ and $\mathbf{y}_\mathbf{i}$ are the user and item vector representations, while the second term $\lambda \left(\sum_u \|\mathbf{x}_\mathbf{u}\|^2 + \sum_i \|\mathbf{y}_\mathbf{i}\|^2 \right)$ is a necessary term for regularizing the model to avoiding the over-fitting while training data.

3.4 Experimental Setup

The various experiments were implemented in Python. This programming language was chosen because it provides several libraries which are specialized in algebra and machine learning. Nowadays `Keras`, `Scipy` and `Tensorflow` are well-known and widely used in tech companies for data analysis. They provide flexible tools to create and setup math procedures or to create neural and deep network architectures. In this thesis, Keras and Scipy libraries were used to train and develop the various experiments.

3.4.1 Experimental Settings

Table 3.1 summarizes the used techniques and algorithms settings.

Experiment	**Settings**
• Matrix Factorization • Non-Negative MF • Neural MF	• topK = 5 • verbose = 0 • latent_dim = 15 • epochs = 100 • batch_size = 4096 • evaluation_threads = 1 • best_hr, best_ndcg = -1, -1 • best_iter = -1 • learning_rate = 0.001 • patience = 10 • early_stop =True • optimizer=Adam • loss='mse'
• Non-Negative MF	• embeddings_constraint = non_neg()
• Neural Matrix Factorization	• layer_units = [30,1] • layer = Dense(activation('relu')) • layer = Dense(activation('selu')
• Implicit Feedback Approach	• alpha = 195 • implicit library • factors = 15 • topK = 5

TABLE 3.1: Variable setup of the different experiments.

3.5 Results of the Generic Data

The experiments explained in the methodology section were applied to the `Retail Rocket data set`. The evaluation metrics Hit ratio and Normalized Discounted Cumulative Gain allow us to easily evaluate the performance of the different models. This allows us to compare the different methods in an easy graphical way.

3.5.1 Matrix Factorization Result

As can be seen in Figure 3.8 this approach reaches the optimal training after a few iterations. The maximum is reached in the 23rd iteration. The maximum Hit Ratio achieved during this experiment was 0.9445 while the NDCG achieved in this iteration was 0.8378. This is a good result since the recommendation accomplished by this experiment was correct in the 94% of the cases, with an 84% of accuracy to get the optimal recommendation list.

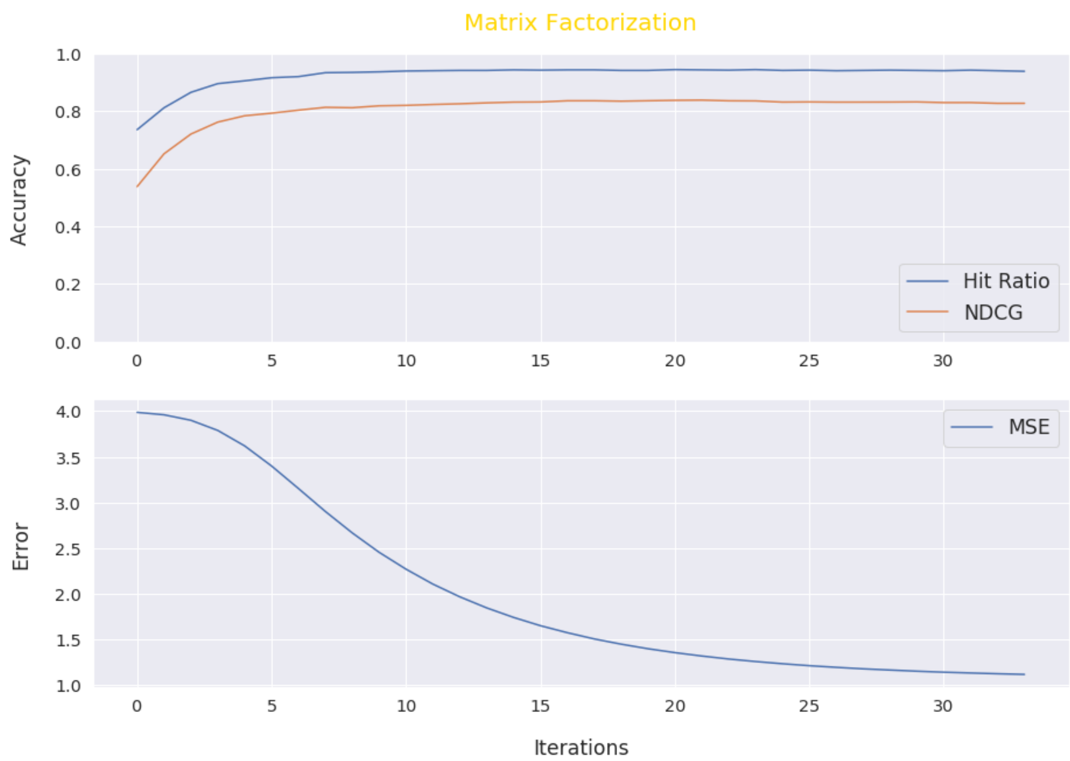

FIGURE 3.8: HR, NDCG and MSE during training evolution.

3.5.2 Non-Negative Matrix Factorization

Unlike the previous approach, in this case, the Hit ratio reaches the optimal training in the 11th iteration but, the NDCG evaluation metric did not reach an acceptable value during all the training processes as can be seen in Figure 3.9. It is necessary to say that after several repetitions of this experiment the NDCG value never reached an acceptable value. The maximum Hit ratio achieved was 0.8413 while the NDCG during this iteration was 0.5784.

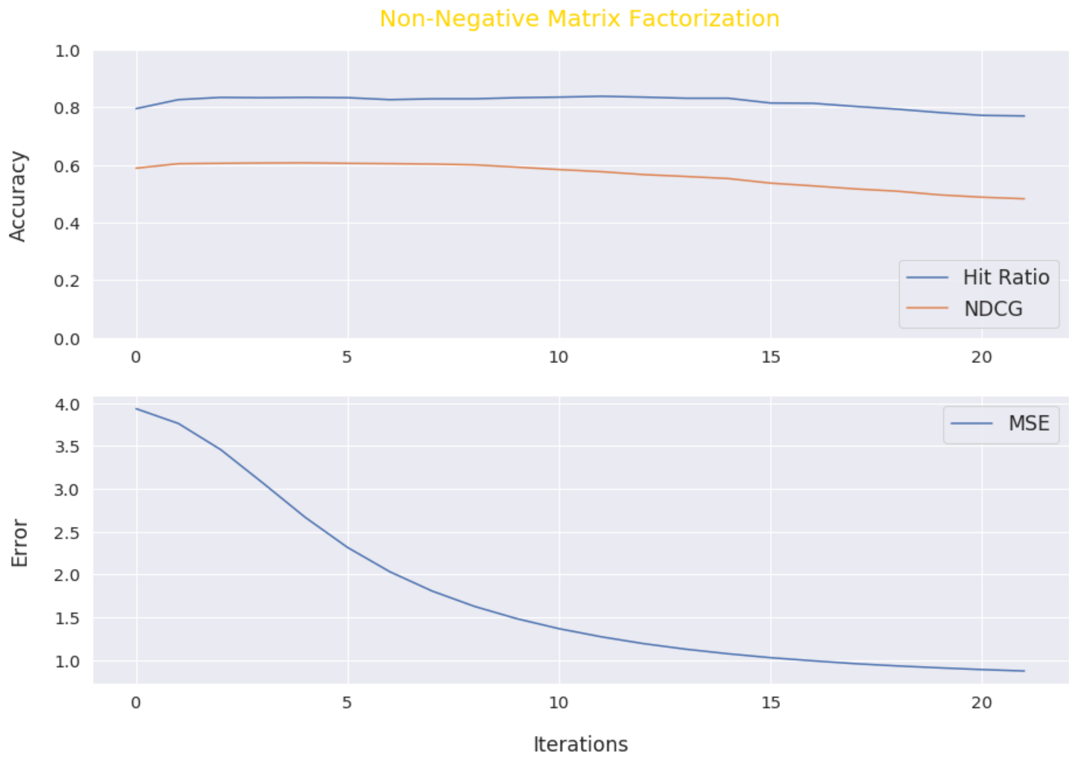

FIGURE 3.9: HR, NDCG and MSE during training evolution.

3.5.3 Neural Matrix Factorization

The matrix factorization through the neural network architecture obtained a not very good approximation by reaching 0.8383 in the hit ratio while the NDCG only reached a 0.6006. These results suggest that the recommendation could be acceptable while the recommended item does not appear in the high places of the list the ones with significant importance for the user. The results are graphically depicted in Figure 3.10.

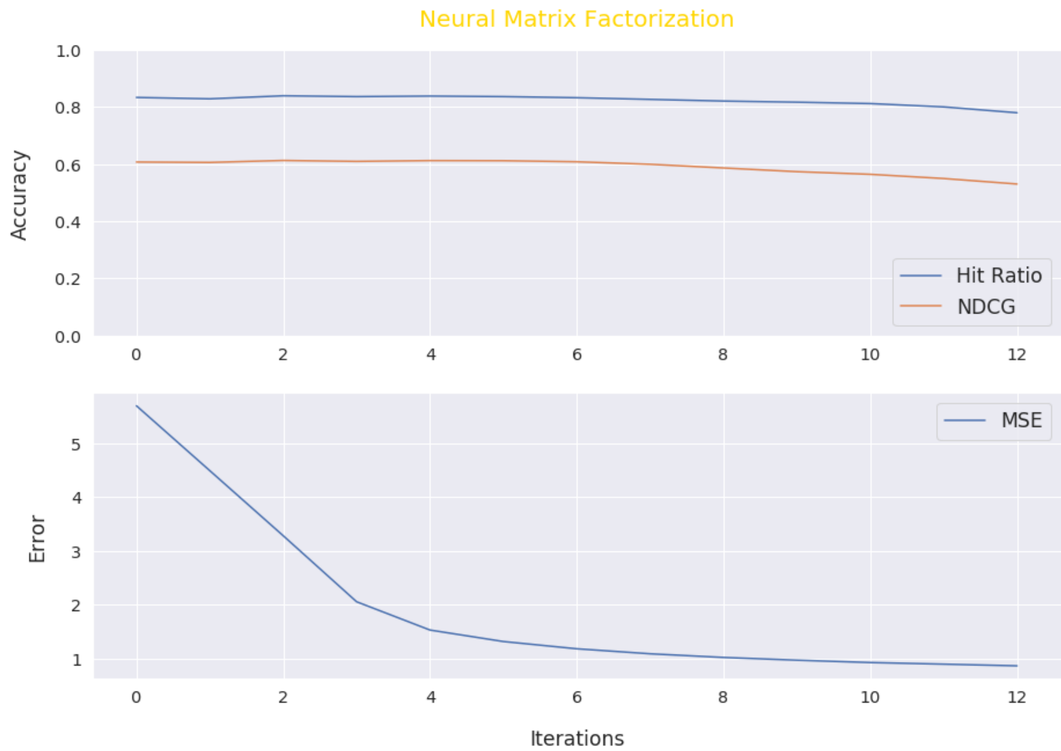

FIGURE 3.10: HR, NDCG and MSE during training evolution.

3.5.4 Implicit Feedback Matrix Factorization

Finally, the matrix factorization for implicit feedback data sets shows that this is a powerful implementation for recommendation systems. This approach got a high hit rate with a remarkably good NDCG making both evaluation metrics have reached more than 90% of accuracy. This model achieved a 0.9523 on the Hit Ratio metric while the NDCG metric achieved a 0.8920. This result reveals that the recommendation made by this experiment was correct in the 95% of the cases with 89% of accuracy to get the optimal recommendation list. Results can be appreciated in Figure 3.11.

```
userItemMatrix shape :  (35992, 11719)

100% 15/15 [00:16<00:00, 1.07s/it]

HR = 0.9523, NDCG = 0.8920, alpha = 195
```

FIGURE 3.11: Implicit feedback matrix factorization evolution.

3.5.5 Results Summary

As can be seen in the Table 3.2, among all the implemented methods, the implicit feedback approach has shown to be most suitable, and then it has been selected as the "Winner". This implementation let the recommendation system to predict the correct product to a given user the 95% of the times. At the same time, this model places the recommended product in the highest places (the most important ones to the user) in the recommendation list. At the very end, this model reflects the most appropriate HR-NDCG relationship. It is necessary to mention that depending on the data used the evaluation metrics resultant values could improve. This is the reason because it is necessary to apply all the methods to the enterprise case study as can be seen in the next Chapter 4.

Experiment	Hit Ratio	NDCG	Winner
Matrix Factorization	0.9445	0.8378	
Non-Negative MF	0.8413	0.5784	
Neural Network MF Approach	0.8382	0.6006	
Implicit Feedback Approach	0.9523	0.8920	**X**

TABLE 3.2: Comparison between MF models through evaluation metrics - Retail Rocket Case Study.

Chapter 4

Analysis of results

4.1 Ecuadorian Enterprise Case Study

Joyas Nereyda is a company dedicated to the wholesale and retail sale of gold-filled, sterling silver and steel jewelry. They also sell crowns, watches, and accessories. Joyas Nereyda is an Ecuadorian enterprise from 1994 and is present in four Ecuadorian provinces. Over time they have understood the needs of customers, generating a constant level of growth in the Ecuadorian jewelry and accessories market. Joyas Nereyda is an enterprise who has been worried about the perception that customers have about their business. In this way, they have built and strong and confident brand which customers can trust. To offer great customer satisfaction they have decided to invest in social media, online advertising, trade shows, and several techniques. Joyas Nereyda to increase its revenues and its customers' satisfaction has decided to be a candidate to test a recommendation system. This could offer these benefits to its business providing a real data set with the sales made from august 2018 to august 2019.

The data set provided contains information of more than fifty thousand sales. The matrix data set is formed by four columns: idcliente, idproducto, comprado and cantidad. All of these values are secured under a unique code due to confidential reasons. To clean the data and reproduce similar characteristics to the generic data set it was necessary to remove duplicate user-item pairs. Next, we merely select just the necessary columns as idcliente, idproducto, and comprado. Finally, it is necessary to remove the sales where the invoice was registered as a final consumer with none identification id.

4.1.1 Training-Test Splitting

The training-test splitting was made as in the previous experiment. In this case, there were 702 users in the testing data set as can be seen in Figure

4.1. This is because the Ecuadorian case study posses a smaller data set of information collected about the users in comparison with the Retail Rocket generic file.

```
train_test_split succeed!! with df_train shape:(63947,3), df_test shape:(702,3)
```

FIGURE 4.1: Training and testing data set splitting.

4.1.2 Matrix Factorization

As can be seen in Figure 4.2 this approach reaches the optimal training after a few iterations. The local maximum is achieved in the 36th iteration. The maximum Hit Ratio achieved during this experiment was 0.8050 while the NDCG achieved in this iteration was 0.6378. This is a satisfactory result because it tells that the recommendation made by this experiment was correct with 82% accuracy. The recommended product appeared at the middle of the recommendation list because the NDCG value was 63%. Figure 4.2 shows the model can be trusted because through the training process the accuracy is maintained after achieve the optimal point.

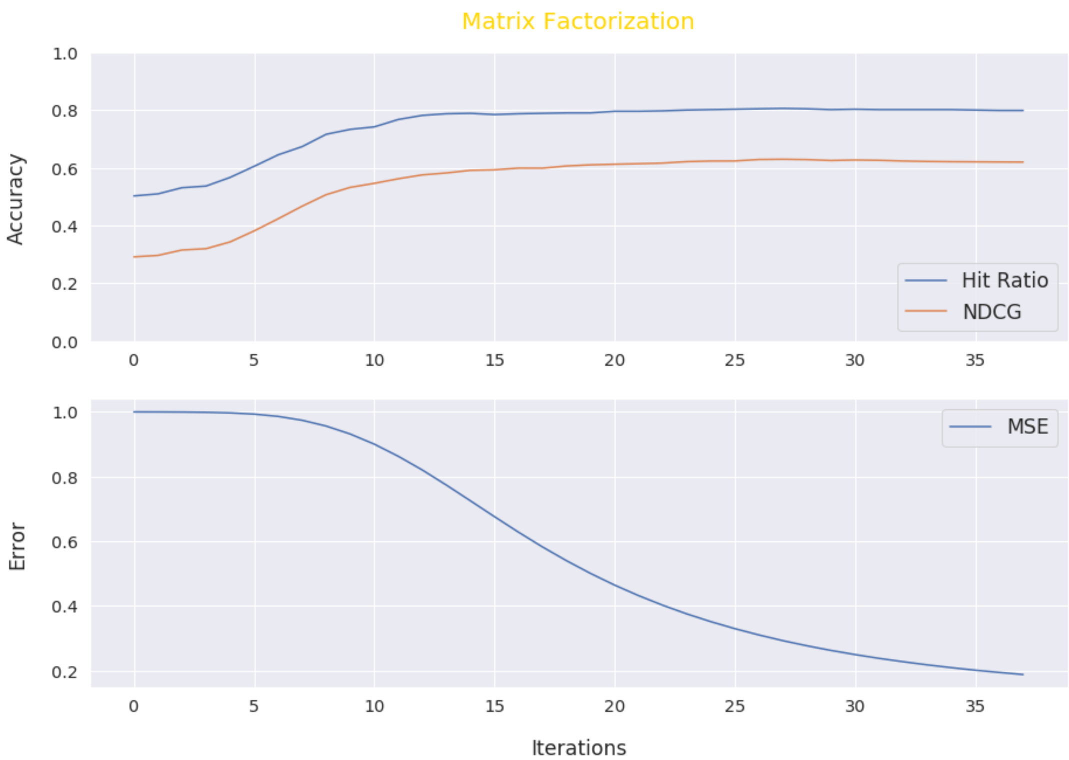

FIGURE 4.2: HR and NDCG training evolution.

4.1.3 Non-Negative Matrix Factorization

In this case, the non-negative matrix factorization approach shows an intriguing result because at the 25th iteration the training achieves the optimal result. Even though the evaluation metrics are strikingly similar to the previous experiment. Figure 4.3 shows that the maximum hit ratio value reached was 0.8236 at the 25th iteration while the NDCG value was 0.5929.

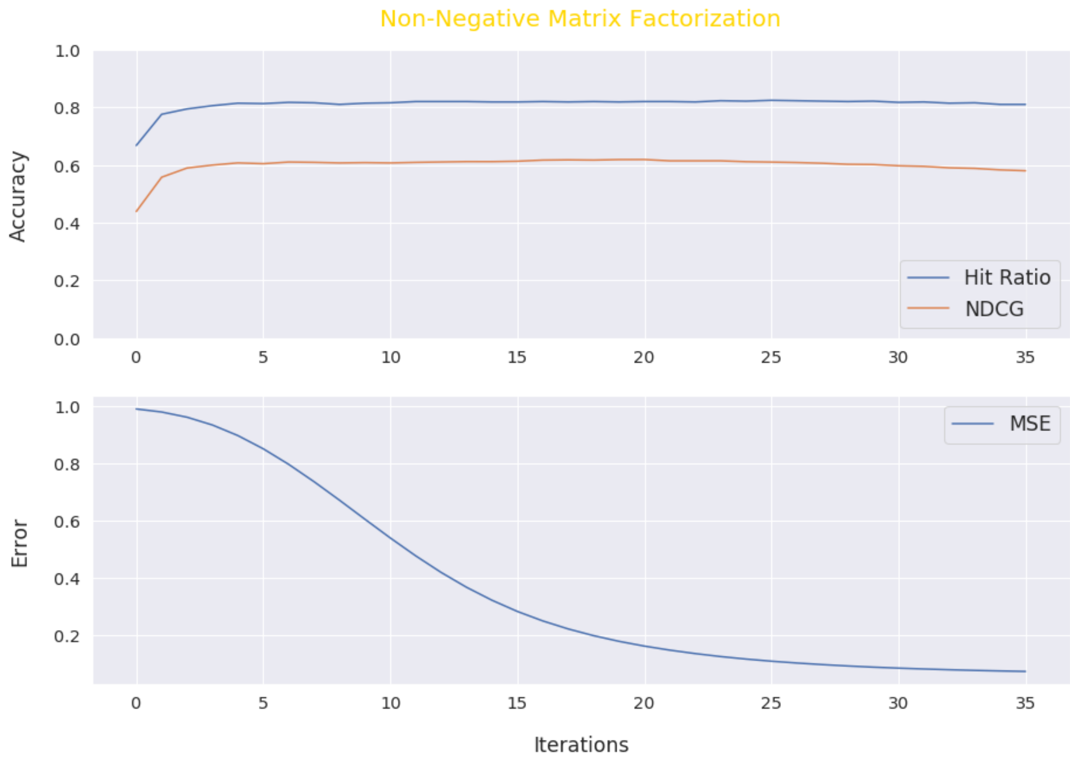

FIGURE 4.3: HR and NDCG training evolution.

4.1.4 Neural Network

The neural network approach has shown a remarkably stable graph. Figure 4.4 shows that the hit ratio has reached 0.8435. Then the accuracy of the recommendation engine at the moment of proposing some item to a given user is 84%. The NDCG has reached 0.6493 this value tells that the score produced by the recommendation list is about 60% of the optimal. Subsequently, the item appears close to the middle position in the recommendation list. The optimal solution appeared in the 15th iteration.

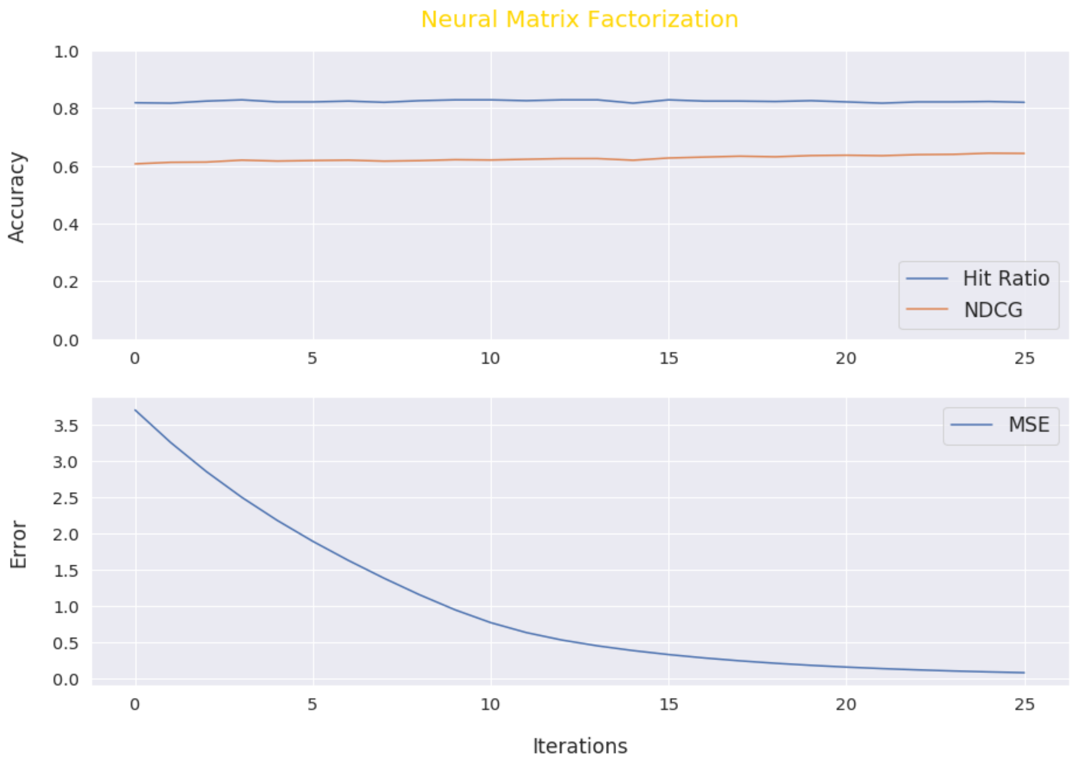

FIGURE 4.4: HR and NDCG training evolution.

4.1.5 Implicit Feedback

This model was the winner in the previous experiment and now has reached a hit ratio of 0.7738 as shown in Figure 4.5. Then the accuracy of making a recommendation to a user about an item is more inferior than the rest of the models. The NDCG value has reached a value of 0.6392 an extremely similar value than the rest of the implementations.

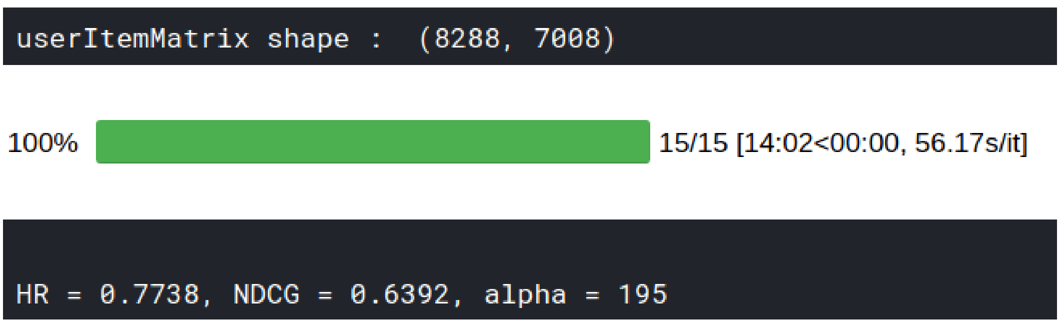

FIGURE 4.5: Implicit feedback matrix factorization evolution.

4.1.6 Ecuadorian Case Study Results Summary

As can be seen, the evaluation metrics do not show very good results as in the Retail Rocket data set. It might be caused because of the lack of information present in the Joyas Nereyda data set. Nevertheless, the Neural Network approach was the winner in this case. This model has reached the most superior accuracy making a recommendation about an item for a given user. It also got the most proper NDCG evaluation metric making the list of recommended items for a given user. This is a reliable list even when the appropriate item does not appear at the beginning of it.

Experiment	**Hit Ratio**	**NDCG**	**Winner**
Matrix Factorization	0.8050	0.6378	
Non-Negative MF	0.8236	0.5929	
Neural Network MF Approach	0.8435	0.6493	**X**
Implicit Feedback Approach	0.7738	0.6392	

TABLE 4.1: Comparison between MF models through evaluation metrics - Ecuadorian Case Study.

Chapter 5

Final remarks

5.1 Conclusions

Next, the conclusions reached by this document:

- Enterprises that decide to adopt the use of recommendation systems must know that the performance could be improved depending on the quantity and quality of the information provided. In this work, the retail rocket generic data set has demonstrated to represent a splendid example of information collection, since its data characterization and study is feasible.

- The mathematical representation of the recommendation systems has indicated how the relationship between users and products is built. The code for each model can be found in the Appendix A for a better understanding. Machine learning techniques as can be seen in the experiment results improved in a considerable way the Matrix Factorization recommendation technique approach.

- The experiments made with the generic data and Ecuadorian case study showed that machine learning can improve the recommendation techniques. Accuracy metrics showed that the implicit feedback approach and neural matrix factorization can provide more accurate recommendations. Thanks to the technology advancements, tools like the ones used in this research as python, TensorFlow or Keras, machine learning is within everyone's reach. Combining machine learning with modeling of user behavior, preferences can be faced offering a deeper point derived from user's interaction within a website and visited products.

- The winner recommendation system based on implicit feedback implemented in this project has shown good performance in a real environment as the Ecuadorian enterprise case study. Nevertheless, it was not enough, and neural matrix factorization showed better accuracy at the moment of recommending. The recommending systems field is a very extensive but worthy topic to research to change and improve the productive matrix of Ecuador.

- The implementation of a recommender system based on implicit feedback data set through a generic and Ecuadorian case study information has let us learn about the importance of the recommendation system. These proved to be an effective way of marketing through personalizing the customer's shopping experience. There are a lot of recommendation systems techniques to test and apply. Even when a type of recommender system does not satisfy a company's requirements, it can be solved by combining several types of recommender systems or feeding some combination of an explicit and implicit data set. As has been discussed throughout this research, recommendation systems are very useful tools for enterprises holding an e-commerce website. Information overload can readily be handled employing recommendation systems improving at the same time the user's experience.

5.2 Recommendations

From experiments, it can be observed that the amount of information is a quality factor at the moment of building a recommendation engine. It is suggested to build a strong data set collecting more user behavior information. The retail rocket data set posses "view" and "transaction" events. These two user data could be taken into account for future improvements over the implementations. Given that the provided data set contains sales from different Ecuadorian provinces. It is also suggested that the Ecuadorian case study data set must be segmented based on some type of marketing suggestion. With this approach, the recommendation engine could be evaluated to discover if this characteristic affects the recommendation result.

The machine learning matrix factorization models showed to be strong and confident than the basic one. It is suggested that these can be implemented for any enterprise interested in knowing the benefits of a recommendation system in its business. After seeing the recommendation results, the enterprises could bet for more complex recommendation models.

The recommendation engine can be rebuilt using any other kind of evaluation and regularization techniques. It is suggested to improve or replace cross-validation, hit rate accuracy or NDCG to analyze the recommendation results from a different perspective.

5.2.1 Future Research

From this thesis research, three lines of interest have emerged for future research: mouse tracking, hybrid filtering, and explicit vs implicit feedback.

- Mouse Tracking: It can provide a deeper perspective on the user's preferences over products posted on an e-commerce web page. Knowing what a user clicked on, time spent seeing a product profile and so forth could provide valuable information to model user's preferences. These can provide a pattern between the users who decided to purchase a product from the ones who decided to abandon the shopping without purchasing. Some researches have use mouse tracking to evaluate the effectiveness of a design interface [34]. These can be used as a starting point to develop mouse tracking behavior that helps to recommend products in a more personalized way.

- Hybrid Filtering: This kind of recommendation technique has shown in the literature section to be a powerful approach model [35]. The way of combining explicit and implicit data is an absorbing topic to investigate. Content-based filtering combined with collaborative filtering provides a system that could take advantage of both the representation of the content as well as the similarities among users. The combination of information filters could improve the personalizing of the recommendation modeling more effectively the user's preference.

- Explicit vs Implicit data: Several Recommendation systems works depending on the type of feedback provided by users. But combining both types of feedbacks and use all information generated by the users could improve the recommendation engine results [22]. Modeling the user's preferences through all the information provided could be more effective. This research could take into account some customer segmentation to detect purchasing patterns accurately.

Appendix A

Python Implementation of the Different Models

In this appendix are placed the different code implementations of the mathematical explanations made in the Section . For the complete code you can visit the web page of the complete project https://sites.google.com/site/degreethesisdiegopeluffo/a-recommendation-system-for-e-commerce

A.1 Experiment #1 Matrix Factorization

```
def matrix_factorization(usersNumber, itemsNumber,
                         latentDims, lambda_reg=0):
    ### define placeholder.
    user_id_input = Input(shape=[1], name='user')
    item_id_input = Input(shape=[1], name='item')

    ### define embedding size and layers.
    user_embedding = Embedding(output_dim = latentDims,
                               input_dim = usersNumber,
                               input_length=1,
                               name='user_embedding',
            embeddings_regularizer = l2(lambda_reg)
                              )(user_id_input)
    item_embedding = Embedding(output_dim = latentDims,
                               input_dim = itemsNumber,
                               input_length=1,
                               name='item_embedding',
            embeddings_regularizer = l2(lambda_reg)
                              )(item_id_input)

    user_vecs = Reshape([latentDims])(user_embedding)
    item_vecs = Reshape([latentDims])(item_embedding)

    # The prediction, which we calculate the loss
    # function with ground truth and optimize.
    y_hat = Dot(1, normalize=False)([user_vecs,
```

```
                                          item_vecs])

    model = Model(inputs=[user_id_input, item_id_input],
                  outputs=y_hat)

    return model
```

A.2 Experiment #2 Non-Negative Matrix Factorization

```
def nonNegMatrFact(usersNumber, itemsNumber,
                   latentDims, lambda_reg=0):
    ### define placeholder.
    user_id_input = Input(shape=[1], name='user')
    item_id_input = Input(shape=[1], name='item')

    ### define embedding size and layers.

    user_embedding = Embedding(output_dim = latentDims,
                                input_dim = usersNumber,
                                input_length=1,
                                name='user_embedding_NMF',
                    embeddings_constraint = non_neg(),
    embeddings_regularizer = l2(lambda_reg))(user_id_input)

    item_embedding = Embedding(output_dim = latentDims,
                                input_dim = itemsNumber,
                                input_length=1,
                                name='item_embedding_NMF',
                        embeddings_constraint = non_neg(),
    embeddings_regularizer = l2(lambda_reg))(item_id_input)

    user_vecs = Reshape([latentDims])(user_embedding)
    item_vecs = Reshape([latentDims])(item_embedding)

    # The prediction, which we calculate the loss
    #function with ground truth and optimize.
    y_hat = Dot(1, normalize=False)([user_vecs,
                                          item_vecs])

    model = Model(inputs=[user_id_input,
                          item_id_input],
                  outputs=y_hat)

    return model
```

A.3 Experiment #3 Neural Network Matrix Factorization

```
def neuralNetworkMF(usersNumber, itemsNumber,
                    latentDims, layers_units,
                    lambda_reg=0,
                    reg_layers = [0,0]):
    ### define placeholder.
    #Number of layers in the MLP
    num_layer = len(layers_units)
    # Input variables
    user_input = Input(shape=(1,), dtype= 'int32',
                       name='user_input')
    item_input = Input(shape=(1,), dtype= 'int32',
                       name='item_input')

    # Embedding layer
    MF_Embedding_User = Embedding(input_dim=usersNumber,
                                  output_dim=latentDims,
                                name='mf_embedding_user',
        embeddings_initializer = normalKerasInitializer,
                    embeddings_regularizer = l2(lambda_reg),
                            input_length=1)(user_input)
    MF_Embedding_Item = Embedding(input_dim=itemsNumber,
                                  output_dim=latentDims,
                                name='mf_embedding_item',
        embeddings_initializer = normalKerasInitializer,
                embeddings_regularizer =l2(lambda_reg),
                            input_length=1)(item_input)

    MLP_Embedding_User = Embedding(input_dim=usersNumber,
        output_dim=int(layers_units[0]/2),
                        name='mlp_embedding_user',
        embeddings_initializer = normalKerasInitializer,
                embeddings_regularizer = l2(lambda_reg),
                        input_length=1)(user_input)
    MLP_Embedding_Item = Embedding(input_dim=itemsNumber,
        output_dim=int(layers_units[0]/2),
                        name='mlp_embedding_item',
        embeddings_initializer = normalKerasInitializer,
                embeddings_regularizer =l2(lambda_reg),
                        input_length=1)(item_input)

    # MF part

    mf_user_latent = Reshape([latentDims]
                            )(MF_Embedding_User)
    mf_item_latent = Reshape([latentDims]
                            )(MF_Embedding_Item)
```

```

    # Element-wise product of user and item embeddings
    mf_vector = multiply([mf_user_latent,
                          mf_item_latent])

    # MLP part
    concatenated = Concatenate()([MLP_Embedding_User,
                                  MLP_Embedding_Item])
    mlp_vector = Flatten()(concatenated)

    # MLP layers
    for layerNumber in range(0, num_layer):
        layer = Dense(layers_units[layerNumber],
        kernel_regularizer= l2(reg_layers[layerNumber]),
                      activation='relu',
                name = 'layer%d' %layerNumber,)
        mlp_vector = layer(mlp_vector)

    predict_vector =  Concatenate()([mf_vector,
                                     mlp_vector])

    # Final prediction layer
    prediction = Dense(1, activation='selu',
                    kernel_initializer='RandomNormal',
                name = "prediction")(predict_vector)

    model = Model(inputs=[user_input,
                          item_input],
                  outputs=prediction)

    return model
```

A.4 Experiment #4 Implicit Feedback Matrix Factorization Approach

```
nb_users = len(useridDict.items())
nb_articles = len(prodidDict.items())
userItemMatrix = np.zeros((nb_users, nb_articles),
                          dtype=np.float32)
userItemMatrix[trainTestDF.visitorid,
        trainTestDF.itemid] = trainTestDF.numevent

from scipy import sparse as sp
alpha = 195
userItemMatrix = userItemMatrix +
                    userItemMatrix * alpha
userItemMatrix = sp.csr_matrix(userItemMatrix.T)
```

```python
# The API needs the rows to be items and the
# columns to be users
print('userItemMatrix shape : ',
      userItemMatrix.shape)

from implicit.als import AlternatingLeastSquares

model = AlternatingLeastSquares(factors=15,
                                use_gpu=False)
model.fit(item_users = userItemMatrix)

assert(model.user_factors.shape[0] == len(useridDict))
assert(model.item_factors.shape[0] == len(prodidDict))

### Evaluating
topK = 5
evaluation_threads = 1
testRatings, testNegatives = newTestUserItemsList,
                             newNegativeItemsList

(hits, ndcgs) = model_evaluation(model,
                        newTestUserItemsList,
                        newNegativeItemsList,
                        topK,
                        evaluation_threads,
                        eval_mode = 'ALS',
                userItemMatrix = userItemMatrix)

hr, ndcg = np.array(hits).mean(),
            np.array(ndcgs).mean()

print('HR = %.4f, NDCG = %.4f, alpha = %d' % (hr,
                                              ndcg,
                                              alpha))
```

Bibliography

[1] I. N. de Estadística, *El 17,1% de las empresas realizan comercio electrónico en ecuador*, 2016. [Online]. Available: http://www.ecuadorencifras.gob.ec/el-171-de-las-empresas-realizan-comercio-electronico-en-ecuador/.

[2] UESS, *Antecedentes y Situación del E-commerce en Ecuador*. 2017. [Online]. Available: https://drive.google.com/file/d/1zA1A8R3yq--2dv2qztMx18Z8afERgldi/view.

[3] T. Lang and M. Rettenmeier, "Understanding consumer behavior with recurrent neural networks", 2017.

[4] H. Hwang and C. Stewart, "Lessons from dot-com boom and bust", in. Jan. 2006. DOI: 10.4018/9781591407997.ch112.

[5] P. T. Joseph *et al.*, *E-commerce: An Indian perspective*. PHI Learning Pvt. Ltd., 2019.

[6] A. G. Khan, "Electronic commerce: A study on benefits and challenges in an emerging economy", 2016.

[7] Y. Hongfei, "National report on e-commerce development in china", *Inclusive and Sustainable Industrial Development Working Paper Series WP17*, 2017.

[8] A. Lipsman, *Global ecommerce 2019*, 2019. [Online]. Available: https://www.emarketer.com/content/global-ecommerce-2019.

[9] S. Chevalier, *Latin america: Retail e-commerce sales 2017-2022*, 2019. [Online]. Available: https://www.statista.com/statistics/445860/retail-e-commerce-sales-latam/.

[10] A. de Regulación y Control de las Telecomunicaciones, *Abonados y usuarios de internet fijo y móvil*. [Online]. Available: http://www.arcotel.gob.ec/servicio-de-acceso-a-internet-sai2/.

[11] *Tecnologías de la Información y Comunicación - ENEMDU - TIC 2017*. INEC, 2018. [Online]. Available: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2017/Tics2017_270718.pdf.

[12] *Comercio electrónico en el ecuador*, 2018. [Online]. Available: https://seoquito.com/comercio-electronico-en-ecuador/.

[13] P. E. Thomas, M. Srihari, and S. Kaur, *Handbook of research on cultural and economic impacts of the information society*. Information Science Reference, 2015.

[14] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, ISSN: 0001-0782. DOI: 10.1145/138859.138867. [Online]. Available: http://doi.acm.org/10.1145/138859.138867.

[15] N. Belkin and W. Croft, "Information filtering and information retrieval: Two sides of the same coin?", *Commun. ACM*, vol. 35, pp. 29–38, Dec. 1992. DOI: 10.1145/138859.138861.

[16] F. Isinkaye, Y. Folajimi, and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation", *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261 –273, 2015, ISSN: 1110-8665. DOI: https://doi.org/10.1016/j.eij.2015.06.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1110866515000341.

[17] T. Ogi, Y. Tateyama, and J. Kawasaki, "Digital guide map using tiled display and recommendation function", in *2011 14th International Conference on Network-Based Information Systems*, 2011, pp. 432–437. DOI: 10.1109/NBiS.2011.72.

[18] R. Vijay, A. Kapuria, C. Datta, G. Dubey, C. Sharma, and G. Taank, "Ecosystem of a shopping mall robot - neel", in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pp. 120–125. DOI: 10.1109/URAI.2012.6462949.

[19] M. Ge, F. Ricci, and D. Massimo, "Health-aware food recommender system", in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys '15, Vienna, Austria: ACM, 2015, pp. 333–334, ISBN: 978-1-4503-3692-5. DOI: 10.1145/2792838.2796554. [Online]. Available: http://doi.acm.org/10.1145/2792838.2796554.

[20] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation", *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, 13:1–13:19, Dec. 2015, ISSN: 2158-656X. DOI: 10.1145/2843948. [Online]. Available: http://doi.acm.org/10.1145/2843948.

[21] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service", Jan. 2010. DOI: 10.1145/1869446.1869453.

[22] S. Liu, X. Tu, and R. Li, "Unifying explicit and implicit feedback for top-n recommendation", in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, 2017, pp. 35–39. DOI: 10.1109/ICBDA.2017.8078860.

[23] E. Núñez Valdez, D. Quintana, R. Gonzalez Crespo, P. Isasi, and E. Herrera-Viedma, "A recommender system based on implicit feedback for selective dissemination of ebooks", *Information Sciences*, vol. 467, pp. 87–98, Oct. 2018. DOI: 10.1016/j.ins.2018.07.068.

[24] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: A literature survey", *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016, ISSN: 1432-1300. DOI: 10.1007/s00799-015-0156-0. [Online]. Available: https://doi.org/10.1007/s00799-015-0156-0.

[25] X. Wang, Y. Guo, and C. Xu, "Recommendation algorithms for optimizing hit rate, user satisfaction and website revenue", in *IJCAI*, 2015.

[26] Y. Wang, L. Wang, Y. Li, D. He, T.-Y. Liu, and W. Chen, "A theoretical analysis of ndcg type ranking measures", *Journal of Machine Learning Research*, vol. 30, Apr. 2013.

[27] M. Jordan and T. Mitchell, "Machine learning: Trends, perspectives, and prospects", *Science (New York, N.Y.)*, vol. 349, pp. 255–60, Jul. 2015. DOI: 10.1126/science.aaa8415.

[28] S. A. Bini, "Artificial intelligence, machine learning, deep learning, and cognitive computing: What do these terms mean and how will they impact health care?", *The Journal of Arthroplasty*, vol. 33, no. 8, pp. 2358 –2361, 2018, ISSN: 0883-5403. DOI: https://doi.org/10.1016/j.arth.2018.02.067. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0883540318302158.

[29] C. Coglianese and D. Lehr, "Regulating by robot: Administrative decision making in the machine-learning era", *Georgetown Law Journal*, vol. 105, pp. 1147–1223, Jan. 2017.

[30] F. C. Pereira and S. S. Borysov, "Machine learning fundamentals", in *Mobility Patterns, Big Data and Transport Analytics*, Elsevier, 2019, pp. 9–29.

[31] M. Pecht, “Prognostics and health management of electronics”, *Encyclopedia of structural health monitoring*, 2009.

[32] X. He, L. Liao, and H. Zhang, “Neural collaborative filtering”, *Proceedings of the 26th International Conference on World Wide Web*, Aug. 2017.

[33] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets”, in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 263–272. DOI: 10.1109/ICDM.2008.22.

[34] F. Mueller and A. Lockerd, “Cheese: Tracking mouse movement activity on websites, a tool for user modeling”, Jan. 2001. DOI: 10.1145/634067.634233.

[35] F. Strub, J. Mary, and R. Gaudel, *Hybrid collaborative filtering with autoencoders*, 2016. arXiv: 1603.00806 [cs.IR].