# TITLE PAGE

**DEEP LEARNING FOR PERSONALIZED BOOK RECOMMENDATION SYSTEMS USING THE BOOK-CROSSING DATASET**

**By**

**OGUNDEPO, Olusola Timothy**

**18/25PJ027**

**A RESEARCH PROJECT SUBMITTED TO THE DEPARTMENT OF EDUCATIONAL TECHNOLOGY, FACULTY OF EDUCATION, UNIVERSITY OF ILORIN, ILORIN, NIGERIA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR DEGREE IN COMPUTER SCIENCE & EDUCATION**

**B.SC. (ED) COMPUTER SCIENCE & EDUCATION**

**October, 2023**

# DECLARATION

I declare that this project, “Deep Learning Model for Personalised Book Recommendation System”, is my own work and has not previously been submitted by me or any other person for any course or qualification at this or any other tertiary institution.

I also declare that, as far as I am aware, all cited works have been acknowledged and referenced.

**Name:** OGUNDEPO, Olusola Timothy

**Matriculation No:** 18/25PJ027

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# CERTIFICATION

This is to certify that this research was carried out by OGUNDEPO, Olusola Timothy (18/25PJ027). The Project had been read and approved as meeting the partial requirements for the award of Bachelor of Science in Computer Science and Education, B. Sc. (Ed), in the Department of Educational Technology, Faculty of Education, University of Ilorin, Ilorin, Nigeria.

**Name/Signature Date**

**……………………………………. …..………………………**

**Dr. Muhammad KJ**

(Supervisor)

**……………………………………. …………………………..**

**Dr. O. O. Obielodan**

(Head of Department)

**…………………………………… ……….………………….**

**Prof. L.A. Yahaya**

(Dean, Faculty of Education)

# COPY RIGHT

**Deep Learning for Personalised Book Recommendation System using the Book-Crossing Dataset**

**By**

**OGUNDEPO, Olusola Timothy**

**18/25PJ027**

**©2023**

**All Rights Reserved**

# DEDICATION

I dedicate this project to Almighty God, as well as to my parents, Mr. and Mrs. Ogundepo, and my supportive brothers, Dr Ogundepo Ezekiel Adebayo and Mr. Ogundepo Demilade Emmanuel.

# ACKNOWLEDGEMENT

My heartfelt thanks go to all who have contributed to my undergraduate journey and final year project at the esteemed University of Ilorin.

Above all, my thanks to the Almighty God for His unceasing guidance, so beautifully encapsulated in Psalm 32:8.

To my cherished parents, Mr. Samuel Adelodun Ogundepo and Mrs. Racheal Nike Ogundepo, your enduring love and unwavering belief in me have been my foundation. May God bless and reward you immensely for your sacrifices.

Dr. Muhammad KJ, my supervisor, your mentorship, and insights have been invaluable. May Almighty God continue to shower blessings upon you and your loved ones.

To my siblings: Mr. Moses Olalekan, a steadfast supporter; Dr Ezekiel Adebayo, a mentor; Mr. John Olanrewaju, a constant presence; Mrs. Oluwatoyin Alice, an unwavering pillar of love; and Mr. Emmanuel Demilade, the "Blessing Man" — each of you has enriched my life in unique ways. I'm grateful for our enduring bond. To my extended family, may God's blessings be with you always.

I'm indebted to the university staff for their assistance and cooperation. My appreciation also extends to my lecturers who've shaped my academic path.

My spiritual family at the Prince of Life Parish, RCCG, has been a beacon, with the fellowship nurturing my faith. As said in Proverbs 27:17, we've grown and sharpened each other. May God's love continue to flourish among us.

Lastly, to all who've touched this project, directly or indirectly, your support has made all the difference. I'm eternally grateful for your invaluable roles in my academic and personal evolution.

# TABLE OF CONTENTS

[TITLE PAGE i](#_Toc147748584)

[DECLARATION ii](#_Toc147748585)

[CERTIFICATION iii](#_Toc147748586)

[COPY RIGHT iv](#_Toc147748587)

[DEDICATION v](#_Toc147748588)

[ACKNOWLEDGEMENT vi](#_Toc147748589)

[TABLE OF CONTENTS vii](#_Toc147748590)

[List Of Figures x](#_Toc147748591)

[List Of Tables x](#_Toc147748592)

[ABSTRACT xi](#_Toc147748593)

[CHAPTER ONE 1](#_Toc147748594)

[1.0 Background to the study 1](#_Toc147748595)

[1.1 Statement of the Problem 3](#_Toc147748596)

[1.2 Purpose of the Study 3](#_Toc147748597)

[1.3 Research Questions 4](#_Toc147748598)

[1.4 Scope of the study 4](#_Toc147748599)

[1.5 Clarification of Major terms and Variables 5](#_Toc147748600)

[1.6 Significance of the Study 6](#_Toc147748601)

[CHAPTER TWO 8](#_Toc147748602)

[2. 0 Introduction 8](#_Toc147748603)

[2.1 Book Recommendation Systems 8](#_Toc147748604)

[2.2 Personalised Recommendation system 9](#_Toc147748605)

[2.2.1 Type of recommendation systems 10](#_Toc147748606)

[2.2.2 Content-based filtering (CBF) 11](#_Toc147748607)

[2.2.3 Collaborative Filtering (CF) 15](#_Toc147748608)

[2.2.4 Hybrid filtering approach 20](#_Toc147748609)

[2.2.5 A Comparative Analysis of Traditional Recommendation Algorithms 23](#_Toc147748610)

[2.3 Deep Learning in Recommendation Systems 24](#_Toc147748611)

[2.3.1 Introduction to deep learning 24](#_Toc147748612)

[2.3.2 Advantages of deep learning in recommendation systems 26](#_Toc147748613)

[2.3.3 Deep learning architectures commonly used for recommendations 28](#_Toc147748614)

[2.4 Deep learning Models for Book Recommendations 30](#_Toc147748615)

[2.4.1 Neural Network-Based Models 31](#_Toc147748616)

[2.4.2 Collaborative Filtering with Deep Learning 31](#_Toc147748617)

[2.4.3 Hybrid Models Combining Deep Learning and Traditional Techniques 31](#_Toc147748618)

[2.5 Evaluation Metrics and Techniques 32](#_Toc147748619)

[2.5.1 Common Evaluation Metrics for Book Recommendation Systems 32](#_Toc147748620)

[2.6 Appraisal of the related works 35](#_Toc147748621)

[CHAPTER THREE 39](#_Toc147748622)

[3.0 Preamble 39](#_Toc147748623)

[3.1 Introduction to the BookCrossing dataset 39](#_Toc147748624)

[3.2 Utilisation of the Python Programming Language 40](#_Toc147748625)

[3.3 Data Exploratory Techniques 41](#_Toc147748626)

[3.4 Data Cleaning 41](#_Toc147748627)

[3.4.1 Importance of Data Cleaning 41](#_Toc147748628)

[3.4.2 Data Cleaning Techniques 42](#_Toc147748629)

[3.4.3 Rationale for Data Cleaning 44](#_Toc147748630)

[3.5 Division Methodology for Training and Testing Sets 45](#_Toc147748631)

[3.5.1 Data Division Techniques 45](#_Toc147748632)

[3.6 Book Recommendation Modelling Systems 46](#_Toc147748633)

[3.6.1 Collaborative Filtering 46](#_Toc147748634)

[3.6.2 Deep Learning Models 46](#_Toc147748635)

[3.6.3 Model Training and Evaluation 47](#_Toc147748636)

[3.6.4 Hyperparameter Tuning 47](#_Toc147748637)

[3.6.6 Cross-validation 48](#_Toc147748638)

[3.6.7 Model Comparison 49](#_Toc147748639)

[3.6.8 Final Model Selection 49](#_Toc147748640)

[CHAPTER FOUR 50](#_Toc147748641)

[4.1 Data Retrieval 50](#_Toc147748642)

[4.1.1 Source of the BookCrossing Dataset 50](#_Toc147748643)

[4.1.2 Methodology for Data Retrieval 50](#_Toc147748644)

[4.1.3 Storage and Local Repository 50](#_Toc147748645)

[4.2 Data Exploration 51](#_Toc147748646)

[4.2.1 Initial Data Inspection 51](#_Toc147748647)

[4.2.2 Observations and Data Quality Assessment 51](#_Toc147748648)

[4.3 Data Cleaning 53](#_Toc147748649)

[4.3.1 Redundant Column Removal 53](#_Toc147748650)

[4.3.2 Missing Value Handling 54](#_Toc147748651)

[4.3.3 Location Attribute Enhancement 55](#_Toc147748652)

[4.3.4 Outlier Handling in Age Column 55](#_Toc147748653)

[4.3.5 Transformation of Non-Numeric Columns 56](#_Toc147748654)

[4.4 Data Division 56](#_Toc147748655)

[4.4.1 Train-Test Split 56](#_Toc147748656)

[4.5 Model Development 57](#_Toc147748657)

[4.5.1 Collaborative Filtering 57](#_Toc147748658)

[4.5.2 Autoencoder Model 58](#_Toc147748659)

[4.5.3 Transformer Model 59](#_Toc147748660)

[4.6 Result 60](#_Toc147748661)

[4.6.1 Model Performance Metrics 60](#_Toc147748662)

[4.6.2 Model Performance Summary 61](#_Toc147748663)

[4.7 Discussion 62](#_Toc147748664)

[4.7.1 Addressing Research Questions 62](#_Toc147748665)

[4.7.2 Model Selection 63](#_Toc147748666)

[CHAPTER FIVE 64](#_Toc147748667)

[5.1 Summary 64](#_Toc147748668)

[5.2 Limitations 65](#_Toc147748669)

[5.3 Recommendation 66](#_Toc147748670)

[5.4 Conclusion 66](#_Toc147748671)

[REFERENCES 67](#_Toc147748672)

# List Of Figures

[Figure 2. 1 - Types of recommendation systems 10](#_Toc147747485)

[Figure 2. 2 - User-based collaborative filtering 17](#_Toc147747486)

[Figure 2. 3 - Item-based collaborative filtering 17](#_Toc147747487)

[Figure 4. 1 - Detection of zero ratings 52](#_Toc147747504)

[Figure 4. 2 - The age distribution before data cleaning 53](#_Toc147747505)

[Figure 4. 3 - Informaton after removing books without publishers and authors 54](#_Toc147747506)

[Figure 4. 4 - Ratings distribution after zeros removal 54](#_Toc147747507)

[Figure 4. 5 - Age distribution after outlier handling 55](#_Toc147747508)

# List Of Tables

[Table 2. 1: Comparison of the traditional recommendation algorithms 23](#_Toc147747355)

[Table 3. 1: A comprehensive description of the dataset 39](#_Toc147747386)

[Table 4. 1: Model Performance Summary 61](#_Toc147747405)

# ABSTRACT

In an increasingly digital world, personalise book recommendations have become paramount. This research delves into the domain of deep learning, aiming to enhance book recommendations by addressing the limitations of traditional collaborative filtering methods. This study not only identifies the challenges but also provides a potential solution through deep learning approaches. The overarching goal is to revolutionise how users discover books tailored to their preferences, making the reading experience more enjoyable and engaging.

The project research journey commenced with a meticulous exploration of deep learning methodologies, conducted using the extensive BookCrossing dataset as the testing ground. Rigorous research design techniques were employed, appropriate instruments were selected, and the population and sample size were carefully determined to ensure the robustness of the findings. The locale for the study is the digital landscape, where book recommendations are in high demand. To analyse the data collected, a suite of statistical tools was employed, enabling the drawing of meaningful insights and making informed conclusions.

The project unfolded systematically, aligning with the specific research objectives. Through a rigorous investigation, it was revealed that deep learning, particularly the Autoencoder model, holds the potential to significantly enhance book recommendations. This model excels in capturing user preferences and effectively addressing data challenges. The evaluation metrics, including Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE), provided concrete evidence of the model's superior performance, with an RMSE of 1.069, MSE of 1.142, and MAE of 0.077.

Overall, this project highlights the transformative power of deep learning in personalise book recommendations. It not only identifies the challenges but also offers a practical solution within the evolving digital landscape. As technology continues to advance, the implementation of deep learning models promises to create more engaging and tailored book recommendation systems. The implications extend beyond book recommendations, emphasizing the broader potential of deep learning in addressing challenges across various domains.

**Keywords**:

Deep learning, Recommendation system, Collaborative filtering, Autoencoder

# CHAPTER ONE

**INTRODUCTION**

## 1.0 Background to the study

With the rapid development of mobile internet and media, it has become more efficient for people to obtain and share information. Large amounts of data are stored on the internet on a daily basis. This is not only limited to social media but also to e-commerce sections such as Amazon, Jumia, Rozetka, etc. where many resources are bought and sold. Because of this large amount of information dumped, there has emerged a problem of information overload, which has affected people’s sense of use. To cope with the problems of information overload and smooth processing of information, various personalised systems have been designed with the aim of reducing the workload on humans. The emergence of a recommendation system can help solve these issues by providing meaningful, effective, and personalised recommendations of products and services to users (Haiming et al., 2021).

A recommendation system (RS) is a type of expert system that filters the huge amount of information present on the internet, processes it based on some filtering criteria, and makes recommendations for people (Balaji, Pranshu & Deepali, 2020). These expert systems are being implemented to reproduce the work of expert advisors in making the right decisions for real-life problems. These systems are designed to help users find products or content that match their interests and preferences based on their browsing history, purchase history, ratings, and other factors (Bushra et al., 2019; Khalid & Jamshed, 2020). In the case of book recommendation systems, the goal is to help readers discover new books that they will enjoy based on their reading history, genre preferences, author preferences, and other factors (Dhanashri, Nandani, Ranjana & Vaishali, 2020).

A recommendation system was initially deployed in the e-commerce sector to help recommend products to users based on their existing interactions and preferences, with the major goal of increasing the income of merchants by selling more products and thereby bringing about satisfaction (Sana, James & Nasseh, 2019). But now, the domain of recommender system has been expanded, i.e., recommending movies and shows for entertainment, books and novels for academicians, doctors and telemedicine for patients, and location and enterprise for tourists, etc. (Khalid et al., 2020).

According to Illia, Victoria, & Solomiia (2022), for users to be satisfied with the service rendered on various web platforms, they must be able to quickly, easily, and efficiently search for the information they need. Also, one of the key components that help keep customers on the web is recommendation (Dhiman, Tanni & Mohammad, 2021). Scientific research shows that 60% of customers prefer to return to stores with requests, while 75% of the digital generation who grew up in the era of social media believe that recommendation is an integral part of any platform (Balush, Vysotska & Albota, 2021). Therefore, in today’s world, an efficient and fast search for information with the possibility of recommendation is one of the primary needs of people and businesses (Sushama, Pooja & Darshana, 2015).

The two most common types of book recommendation systems are content-based and collaborative filtering (Ruihui, 2018). Content-based systems recommend books based on the user’s past interactions with similar content, while collaborative filtering systems recommend books based on the past interactions of similar users (Khalid & Jamshed, 2020). Both of these approaches have their strengths and weaknesses. To overcome some of the weaknesses found in these systems, researchers developed a refined recommendation system called hybrid RS. This system takes both collaborative filtering methods to generate similar users and content-based recommendations to generate similar items into consideration. Therefore, hybrid systems produce the ensemble output of both collaborative filtering and content-based systems. Still, some limitations remain obvious. Researchers have been working to improve the accuracy and effectiveness of these systems using machine learning and deep learning algorithms (Anwar et al., 2020; Kholi M., Tali A. & Laaziz Y., 2020).

Deep learning, a branch of machine learning, involves the use of neural networks to model complex patterns in data (Mijwil & Abttan, 2021). Over the past few years, deep learning has been shown to outperform traditional machine learning models and has contributed and helped solve various problems identified in many domains, including image recognition, speech recognition, and natural language processing, etc.

This field of machine learning has been used in creating many innovative applications, such as self-driving cars, language translation, speech recognition, etc. The reason for this is the explosion of data on the internet, better hardware for complex computation like GPUs, and better algorithms for learning complex ideas (Balasubramanian et al., 2020). The architecture of deep learning is inspired by artificial intelligence that simulates the deep learning process and sensory layers of the human brain in learning from past experiences and making the decision to solve future problems that might be encountered (Maad et al., 2022).

Deep learning can help qualify large amounts of data related to users and items by learning a deep non-linear network structure. It can obtain deep-level feature representations of users and items and has a potent ability to understand the fundamental properties of data sets from samples. In addition, it alleviates the classic recommendation system’s cold start and data sparseness which are the main issues of a traditional recommendation system (Haiming et al., 2021).

## 1.1 Statement of the Problem

The rapid growth of digital libraries and online bookstores has made it increasingly challenging for users to discover books that align with their individual preferences and interests (Haiming et al., 2021). Although traditional recommendation systems that recommend books of different genres to users exist, they face challenges related to cold-start, data sparsity, and scalability (Li & Kim, 2021). Therefore, there is a need to develop an effective and personalised book recommendation system using deep learning techniques that can accurately understand user’s preferences and deliver relevant book recommendations, enhancing their reading experience and satisfaction.

## 1.2 Purpose of the Study

The purpose of this study is to explore the use of deep learning models for personalised book recommendation systems. Specifically, this study will:

1. Retrieve and clean the BookCrossing dataset to be tidy and ready for modelling recommendation systems.
2. Develop deep learning models to capture user preferences, interests, and behaviour for book recommendations.
3. Address some of the existing limitations of traditional recommendation systems by leveraging the power of deep learning algorithms.
4. Evaluate the performance and effectiveness of the developed deep learning models for personalised book recommendations.

## 1.3 Research Questions

This research aims to explore the following questions regarding deep learning for personalised book recommendation systems:

1. How can deep learning techniques effectively capture user preferences, interests, and behaviours to enhance the accuracy and relevance of book recommendations?
2. What limitations of traditional recommendation systems can be overcome by harnessing the power of deep learning algorithms?
3. Among various deep learning architectures, which ones exhibit the highest effectiveness in personalised book recommendations?
4. To what extent does the performance of deep learning models for personalised book recommendations depend on the size and quality of the training dataset?
5. How can deep learning models address the challenges posed by data sparsity and cold start in personalised book recommendation systems?
6. In what ways can the findings of this study be utilised to develop more personalised and engaging book recommendation systems for users?

## 1.4 Scope of the study

This study focuses on using deep learning models for personalised book recommendations, with the dataset limited to the BookCrossing dataset. The BookCrossing dataset is a publicly available dataset containing information on book ratings and reviews from a large online book-sharing community.

The study will explore different types of deep learning models and compare the chosen model performance to traditional collaborative-filtering model. The evaluation metrics will include root mean square error, mean score error and absolute mean square error.

In addition, this study will investigate techniques for incorporating user feedback into deep learning models while addressing issues such as data sparsity and user privacy. The study will also explore methods for improving the interpretability and transparency of the deep learning models while maintaining high levels of accuracy and personalisation.

The study will have some limitations due to the dataset being limited to the BookCrossing dataset. The dataset may not be representative of all book genres or domains, which could impact the generalizability of the results.

Overall, the study will provide insights into the effectiveness of deep learning models for personalised book recommendations using the BookCrossing dataset, and contribute to the development of more accurate and effective book recommendation systems.

## 1.5 Clarification of Major terms and Variables

**Social media**: This refers to online platforms and tools that allow users to create, share and exchange information, ideas, opinions and content with others. Social media platforms can take many forms including social networking sites like Facebook, Twitter and LinkedIn, photo and video sharing sites like Instagram, YouTube and TikTok, blogging and microblogging platforms like Tumblr and Twitter and many others.

**E-commerce**: This refers to the buying and selling of goods and services over the internet. It involves online transactions between businesses, individuals, and other organisations, and typically involves the use of online platforms, such as an e-commerce website or mobile app.

**Information overload**: This refers to the situation where a person is exposed to too much information, which can lead to feelings of being overwhelmed, stressed, and unable to process or make sense of the information.

**Machine learning**: This is a branch of artificial intelligence (AI) that involves developing algorithms and statistical models that enable computer systems to automatically improve their performance on a task through experience and data input. It is based on the idea that computer systems can learn from data, identify patterns and relationships and make predictions or decisions without being explicitly programmed to do so.

**Deep learning**: This is a method of artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. In other words, it is a subset of machine learning that focuses on training artificial neural networks with multiple layers, also known as deep neural networks. These networks are designed to learn and represent complex patterns and relationships in data.

**Computer vision**: This is a field of artificial intelligence (AI) and computer science that focuses on enabling computers to interpret and understand visual information from the world around us. It involves developing algorithms and techniques that allow computers to analyse, process, and interpret digital images and videos and extract meaningful information from them.

**Natural language processing (NLP)**: This is a field of computer science and artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. It involves developing algorithms and techniques that allow computers to process and analyse natural language data, including text and speech.

**Recommendation**: This is a suggestion or piece of advice offered to someone/user/customer with the aim of guiding them towards a particular course of action or decision. It can be very useful for helping people discover new things, make informed decisions and save time and effort in their decision-making process.

**Personalised recommendation**: This is a type of recommendation system that uses individual user data to provide tailored and relevant recommendations to each user. It is based on the idea that different users have different preferences and that recommendations that are customised to each user’s preferences are more likely to be useful and effective.

**Cold start**: This is a problem in recommendation systems that occurs when there is insufficient data or information available about a new user or item to generate accurate recommendations. In other words, when a recommendation system is faced with a new user or item, it may not have enough historical data to make reliable predictions or recommendations.

**Data sparseness**: This refers to the problem of having insufficient data or information available about a user or item to generate accurate recommendations. This can occur when there are too few ratings or interactions available for a given user or item, making it difficult to accurately predict their preferences or suggest similar items.

## 1.6 Significance of the Study

The development of accurate and effective personalised book recommendation systems has the potential to benefit a wide range of stakeholders, including readers, publishers, and retailers. By improving the quality of recommendations, these systems can help readers discover new books that match their interests and preferences, leading to increased satisfaction and engagement with reading.

For publishers and retailers, personalised book recommendation systems can help increase customer loyalty and sales by providing targeted recommendations and improving the overall customer experience. This can be especially valuable in the highly competitive book market, where personalised recommendations can help differentiate a brand and attract new customers.

The use of deep learning models in personalised and book recommendation systems has shown promise in improving the accuracy and effectiveness of recommendations. By exploring the effectiveness of these models in the context of book recommendations, this study has the potential to contribute to the development of more advanced and effective recommendation systems.

Furthermore, the study will contribute to the broader field of deep learning and machine learning by investigating techniques for incorporating user demography and improving the interpretability of deep learning models. These findings may have broader implications for other domains where deep learning models are used, such as image recognition, speech recognition, and natural language processing.

Overall, this study has the potential to contribute to the development of more accurate, effective, and interpretable personalised book recommendation systems, with the potential to benefit a wide range of stakeholders in the book industry and beyond.

# CHAPTER TWO

**LITERATURE REVIEW**

## 2. 0 Introduction

The purpose of this literature review is to evaluate the existing research on deep learning models for personalised book recommendations. The aim is to identify the strengths and limitations of current approaches, highlight the research gaps and challenges and provide a basis for the development of a novel deep learning model for personalised book recommendation.

## 2.1 Book Recommendation Systems

Book recommendation systems are software applications that suggest books to users based on their interests and preferences. The goal of these systems is to help users discover books that they might be interested in reading but might not otherwise have discovered on their own. There are two types of recommendation systems i.e., personalised and non-personalised recommendation system. Personalised recommendation systems recommend product or items to users based on preference criteria of individual (Sana, James & Nasseh, 2019). This system makes use of the individual past records of user’s ratings in order to find more suitable items that are yet to be rated by the user as recommendations (Khalid & Jamshed, 2020). On the other hand, non-personalised recommendation systems recommend base on the current popular trends that are relevant to all users. These trends can be the top 10 most popular items among users, information on a certain update or the current updates of a particular environment which are suitable for every new user (Khatwani & Chandak, 2016).

In recent years, deep learning models have become increasingly popular for book recommendations (Dhanashri et al., 2020). These models use artificial neural networks and other machine learning techniques to process large amounts of data and make recommendations based on patterns and trends in the data (Balaji, Pranshu & Deepali, 2020). Deep learning models can incorporate a wide range of data, including user behavior, book metadata, and social network information, to make highly personalised recommendations (Alexandros & Balázs, 2017).

However, there are several challenges associated with developing effective book recommendation systems. These include the cold start problem, which occurs when there is not enough user data to make accurate recommendations for new users, and the problem of data sparsity, which occurs when there is not enough data on certain books or users to make accurate recommendations. Additionally, privacy concerns related to user data collection and usage are an important issue that must be addressed when developing book recommendation systems.

This brief overview of book recommendation systems sets the stage for the subsequent sections of the literature review, which will dive deeper into the different types of recommendation systems and the challenges associated with developing effective personalised book recommender systems.

## 2.2 Personalised Recommendation system

The emergence of personalised recommendation system can be traced back to the extensive research in cognitive science, approximation theory, information retrieval, forecasting theories, and also to management science related and to consumer choice modeling in marketing i.e., E-commerce. The recommender systems emerged as an autonomous research domain in the mid-1990s. This is the period where researchers started to focus on the recommendation problems that rely solely on ratings structure. The recommendation problem is reduced to the problem of recommender system to be estimating ratings for the items that have not been seen or rated by the user. Intuitively, this prediction or estimation is usually based on the user’s previous rated items. In this sense, if the estimate ratings for the yet unrated items can be obtain, then the item(s) with the highest estimated rating(s) can be predicted to the user (Gediminas et al., 2005).

Personalised recommendation system is a computer-based system that offers individualized recommendations to users based on their previous behavior or history (Khatwani & Chandak, 2016). According to Dina et al. (2018), a recommender system in the context of education provides intelligent recommendations to students about study materials, courses, and related information based on their prior decisions. To provide such recommendations, the system employs a statistical model that is trained using a learning algorithm based on input from previous learning participants.

In a personalised recommendation system, users and items are the two main objects that play a crucial role. Users provide their interests about items, and this information is accumulated as input data in a utility matrix. The utility matrix represents the order of preference for particular items by the user in the combination of customer-item value. There are two classifications of recommender systems: user-based and item-based. In a user-based system, the user's interests, dislikes, and ratings are used to recommend unrated items. In contrast, an item-based system utilises the relationship among items to generate recommendations for users (Geetha & Karthika, 2019).

Personalised recommendation systems play a vital role in both academia and industries. Many companies use personalised RS in their sales promotion, utilising various platforms. For instance, the majority of the most-watched movies on YouTube and other online video databases come from the RS (Aminu & Naomie, 2019). The main purpose of building a recommender system is to provide maximum information required for the sole aim of personalizing learning and interests depending on the interactive patterns of users (Bouihi & Bahaj, 2019; Zhang & Yang, P, 2020).

The algorithm for estimating the not-yet-rated items in the personalised recommender system is classified into three, namely: collaborative filtering, content base and hybrid approach (Kunal et al., 2017).

All authors agree that a personalised recommendation system is a system created with the major aim of recommending useful individualized information or items to users based on past learned historical records relating to the individual’s tastes or preferences.

### 2.2.1 Type of recommendation systems

There are three main categories of recommendation systems: content-based recommender systems, collaborative recommender systems, and hybrid recommender systems (Roy & Dutta, 2022). Figure 2. 1 provides a visual depiction of these different types of recommender systems.

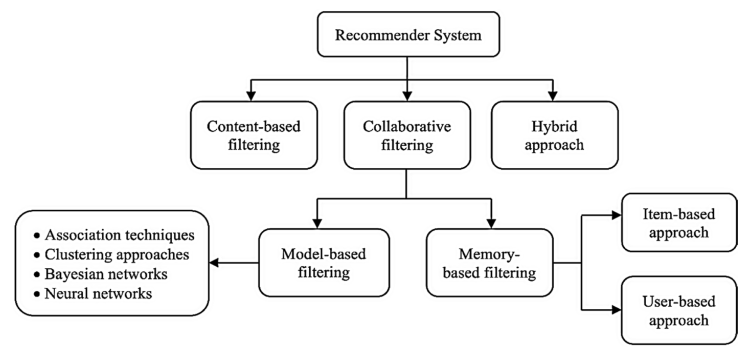


Figure 2. 1 - Types of recommendation systems

### 2.2.2 Content-based filtering (CBF)

Content-based recommender systems are widely used in information or book retrieval. The process involves manually assigning terms and selecting appropriate techniques to compare these terms with the information in the user's profile. A learning algorithm is then applied to perform the chosen techniques and provide relevant recommendations to the user (Javed & Shaukat, 2021).

In content-based recommender systems, all the data items are collected into different item profiles based on their description or features. For example, in the case of a book, the features will be author, publisher, etc. When a user gives a positive rating to an item, then the other items present in that item profile are aggregated together to build a user profile. This user profile combines all the item profiles, whose items are rated positively by the user. Items present in this user profile are then recommended to the user (Roy & Dutta, 2022).

In their study, Balush et al. provide a comprehensive description of Content-Based Filtering (CBF) as an algorithmic approach for recommending items or books to users based on their past consumption. The CBF method analyses the interests of new users by considering the features of the objects they have evaluated. This approach operates as a recommendation system tailored to specific keywords, where these keywords are used to describe the items. Consequently, the content-based recommendation system employs algorithms to suggest items that are similar to those enjoyed or currently being explored by the user (Balush, Vysotska & Albota, 2021).

Term frequency (TF) and inverse document frequency (IDF) are important concepts used in information retrieval and content-based filtering systems, such as content-based recommenders. These concepts determine the relative importance of a document, article, news item, film, etc. Content-based filtering is a common approach in recommender system design, relying on item descriptions and user profiles. In a content-based recommender system, keywords are used to describe items, and a personal profile is constructed to suggest items that align with the user's preferences. In other words, these algorithms aim to recommend items similar to those that the user has liked or currently shows interest in. To achieve this, multiple candidate items are compared with items previously rated by the user, and the best matching items are recommended. Yannick et al. (2010) employed a content-based algorithm to filter and recommend the best results to users. The advantages of content-based recommender systems include transparency, independence, and recommendations for unclassified entities. However, they also have drawbacks such as limited serendipity, partial content analysis, and overspecialization (Adomavicius et al., 2005).

By leveraging a content-based recommender system, recommendations can be generated based on the user's provided data or implicit interactions with the system. As the user provides more data or accepts suggested activities, the system becomes increasingly accurate in generating personalised recommendations. Compared to collaborative filtering systems, content-based approaches offer easier implementation and have been used in many projects due to their ability to generate highly relevant and transparent results. Content-based filtering strategies often employ techniques for content extraction, characterization, and building user and item profiles. However, these techniques have limitations, such as mismatches between user profile items and profile terms, leading to lower performance. Notable examples of content-based recommendations include Fab, which suggests web pages, and ELFI, which recommends financial information from a database (Lops, Gemmis & Semeraro, 2011).

Content-based filtering systems adapt their behavior to individual users by learning their preferences from previously relevant documents. The process involves comparing representations of item content with representations of user interests to find the most relevant items for each user. This task requires finding the best representation for both items (item profile) and users (user profile). A user profile represents a mapping of the user's real-world interests to a simplified domain model, while an item's profile describes its content using keywords or other means of representation. By enabling matching between user and item profiles, a common representation is established (Javed & Shaukat, 2021).

Content-based recommendation systems operate by analysing a collection of documents and/or descriptions of items that have been previously rated by a user. These systems construct a model or profile of the user's interests based on the features of the rated objects (Lops, Gemmis & Semeraro, 2011). As described by Mladenic (2018), the profile serves as a structured representation of user interests and is utilised to recommend new and interesting items. The recommendation process primarily involves comparing the attributes of the user profile with the attributes of a content object (Khatwani & Chandak, 2016). This comparison results in a relevance judgment that indicates the user's level of interest in the object. A well-constructed profile that accurately reflects user preferences greatly enhances the effectiveness of an information retrieval process (Geetha & Renuka, 2019).

An example of the profile's utility is its application in filtering search results. By leveraging the user profile, it becomes possible to determine whether a user would be interested in a specific web page. If not, the system can prevent the display of that particular page (Lops, Gemmis & Semeraro, 2011).

#### 2.2.2.1 High-Level Architecture of Content-based Systems

Content-based Information Filtering (IF) systems require effective techniques for representing items and generating user profiles, as well as strategies for comparing the user profile with item representations (Adomavicius et al., 2005). In their study, Lops et al. (2011) outlined a three-step recommendation process, with each step handled by a distinct component:

1. **Content Analyser**: The Content Analyser is responsible for processing information without a predefined structure, such as text, and extracting relevant structured information. Its main task is to represent the content of items (e.g., documents, web pages, news, product descriptions) from various sources in a format suitable for further processing. Through feature extraction techniques, the component analyses data items and transforms their representation from the original information space to the target space, such as representing web pages as keyword vectors. The output of the Content Analyser serves as input for the Profile Learner and Filtering Component.
2. **Profile Learner**: The Profile Learner module collects representative data on user preferences and generalises this data to construct the user profile. Typically, machine learning techniques are employed to infer a model of user interests based on items previously liked or disliked. For example, in a web page recommender system, the Profile Learner can utilise a relevance feedback method (Rocchio, 1971), where vectors of positive and negative examples are combined to create a prototype vector representing the user profile. Training examples consist of web pages that received positive or negative feedback from the user.
3. **Filtering Component**: The Filtering Component leverages the user profile to suggest relevant items by comparing the profile representation with the representations of items to be recommended. This comparison results in a binary or continuous relevance judgment, often computed using similarity metrics. In the case of continuous relevance, a ranked list of potentially interesting items is generated. In the previous example, the matching process involves computing the cosine similarity between the prototype vector and the item vectors.

The recommendation process begins with the content analyser, which applies techniques from Information Retrieval systems to process item descriptions from information sources. The content analyser extracts features such as keywords, n-grams, and concepts from unstructured text to generate a structured representation of items stored in the Represented Items repository.

To construct and update the user profile for the active user (Ua) who requires recommendations, the system collects the user's reactions to items and records them in the Feedback repository. These reactions, known as annotations or feedback, along with the corresponding item descriptions, are used during the process of learning a model that predicts the relevance of newly presented items. Users can either provide explicit feedback, where they actively evaluate items, or implicit feedback, which is derived from monitoring and analysing user activities (Lops, Gemmis & Semeraro, 2011).

Explicit feedback can be obtained through approaches such as like/dislike ratings, numerical ratings, or text comments. Implicit feedback methods assign a relevance score based on user actions, such as saving, discarding, printing, sounds, manipulation of physical objects, or bookmarking of items. While explicit feedback is straightforward, the adoption of numeric or symbolic scales can increase cognitive load. Implicit feedback methods do not require direct user involvement but may be subject to biases, such as interruptions during reading (Meddeb, Maraoui & Zrigui, 2021).

According to Nilashi et al. (2013), user feedback and ratings can take different forms, which include:

**Numerical ratings**: These ratings are represented by numbers on either discrete or continuous scales. Discrete rating scales, such as the popular five-star system or Likert response scales used in questionnaires, provide a limited range for users to rate items. Continuous rating scales often involve sliders that users can adjust to indicate a specific value.

**Binary rating scale**: This type of rating allows users to categorize items into two distinct classes, typically denoted as "like" or "dislike." For example, platforms like YouTube offer users the option to give movies a thumbs-up or thumbs-down rating.

**Ordinal ratings**: Users are presented with a set of options such as "strongly agree," "neutral," "disagree," or "strongly disagree." They are then asked to choose the option that best represents their opinion about a particular item, typically through the use of questionnaires.

**Unary rating**: Users can assign items to a single positive class using unary ratings. A prominent example is Facebook's "Like" button, which allows users to express their positive opinion about a post. Implicit unary ratings can also be inferred from user actions like purchasing products on a web shop or clicking links on a webpage. The absence of a rating indicates a lack of information connecting the user to the item, potentially because the purchase was made elsewhere.

These different types of ratings provide users with diverse ways to express their preferences and opinions about items.

The Profile Learner component utilises a training set (TRa) specific to the active user (Ua), consisting of pairs ⟨Ik, Rk⟩, where Rk represents the rating provided by Ua for the item representation Ik. By employing supervised learning algorithms on the labeled item representations, the profile learner generates a predictive model known as the user profile. This model is stored in a profile repository and utilised by the filtering component for future recommendations. When presented with a new item representation, the filtering component compares its features to those in the user profile, predicting the user's interest. Typically, the filtering component incorporates strategies to rank potentially interesting items based on their relevance to the user profile. The top-ranked items form a recommendation list (La) that is presented to the user (Ua). As user preferences change over time, the user profile is continuously updated by incorporating up-to-date information. User feedback on the recommendations helps refine the learning process, creating a feedback-learning cycle that adapts to the dynamic nature of user preferences (Lops, Gemmis, & Semeraro, 2011).

Overall, content-based information filtering systems involve the content analyser for item representation, the profile learner for constructing user profiles, and the filtering component for suggesting relevant items based on user profiles. These components interact in a recommendation process that incorporates user feedback to adapt to evolving user preferences.

### 2.2.3 Collaborative Filtering (CF)

The term "collaborative filtering" was first introduced in 1992 by Goldberg et al., who proposed that considering human group relational preferences or involvements can enhance the effectiveness of information filtering when recommending items to users (Kunal et al., 2017). This recommendation method aims to predict the utility of items for a specific user by leveraging the ratings given to those items by other similar users in the past (Gediminas, Alexanda, et al., 2005).

Collaborative filtering, as described by Kunal et al. (2017), is a recommendation system approach that generates recommendations for different users by actively comparing the preferences of an active user with those of other users who have rated similar products in the past. On the other hand, Sana et al. (2019) defines collaborative filtering as a technique that predicts recommendations by identifying users with similar interests. Collaborative filtering is based on the assumption that individuals who have agreed on item evaluations in the past are likely to agree in the future as well.

In accordance with Balush, Vysotska & Albota (2021), collaborative filtering (CF) is a type of recommendation system that relies on aggregating object ratings or recommendations, identifying similarities among users based on their ratings, and generating new recommendations through user comparisons. This approach proves effective in cases involving complex objects, where variations in taste account for the majority of preference changes. collaborative filtering operates under the assumption that individuals who have agreed in the past will continue to have similar preferences in the future and are likely to enjoy similar objects as they have before.

Collaborative filtering method are grouped into two general classes: neighbourhood based and model-based (Gediminas et al., 2005; Kunal et al., 2017).

#### 2.2.3.1 Neighbourhood Based Method

The neighbourhood-based method of collaborative filtering, also known as memory-based or heuristic-based CF, leverages the user-item ratings stored in memory to make predictions about the preferences for new items. This can be accomplished through two approaches: user-based and item-based recommendation (Qian et al., 2020).

In the user-based approach, the interest of a user (referred to as "u") in an item (referred to as "i") is determined by considering the ratings given to that item by other users who exhibit similar rating patterns. These users are known as "neighbors". By examining the ratings of these neighbors, the system can estimate the likelihood of user u being interested in item i (Dhanashri et al., 2020). This approach as explained by Roy & Dutta (2022), the user rating of a new item is calculated by finding other users from the user neighbourhoods who has previously rated that same item. If a new item receives positive ratings from the user neighbourhood, the new item is recommended to the user. Figure 2. 2 depicts the user-based filtering approach.

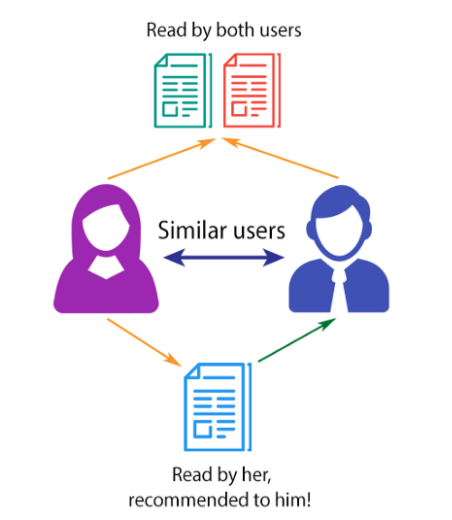


Figure 2. 2 - User-based collaborative filtering

On the other hand, the item-based approach involves predicting the rating that user (u) would assign to item (i) based on the ratings that user u has given to items similar to i. In this approach, the similarity between two items is calculated by examining the ratings provided by other users of the system. By analysing these ratings, the system can identify items that are similar in terms of user preference and use this information to make predictions about user u's rating for item i (Bushra et al., 2019). Also, Roy & Dutta (2022) addresses the fact that, in the item-based approach, an item-neighbourhood is built consisting of all similar items which the user has rated previously. Then that user’s rating for a different new item is predicted by calculating the weighted average of all ratings present in a similar item-neighbourhood as shown in the Figure 2. 3.

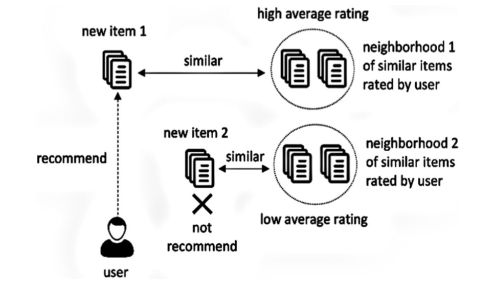


Figure 2. 3 - Item-based collaborative filtering

As stated by Nilashi et al. (2013), all collaborative filtering methods share a capability to utilise the past ratings of users in order to predict or recommend new content that an individual user will like. The real assumption is highly based in the idea of likeness between users or between products, with the similarity being expressed as a function of agreement between past ratings or preferences.

Overall, the neighbourhood-based method of collaborative filtering utilises the existing user-item ratings to determine recommendations for new items. The user-based approach focuses on finding users with similar rating patterns, while the item-based approach concentrates on identifying similar items based on user ratings. By employing these methods, the system can provide personalised recommendations that align with user preferences.

Kunal et al. (2017) classified the main advantages of neighbourhood-based methods as follow:

**Simplicity**: Neighbourhood-based methods are relatively straightforward to implement, making them accessible and easy to work with.

**Efficiency**: This method does not require expensive training phases that need to be regularly performed in large commercial applications. This saves computational resources and time.

**Justifiability**: Neighbourhood-based methods provide concise yet comprehensive explanations for their predictions. This makes it easier for users to understand why certain recommendations are made.

**Stability**: Despite the continuous addition of items, users, and ratings typically encountered in major e-commerce applications, neighbourhood-based methods remain relatively unaffected. They maintain stability in their recommendations even as the system evolves over time.

#### 2.2.3.2 Model-Based Recommendation Methods

The model-based recommendation method differs from neighbourhood-based systems in that it doesn't directly use stored ratings for prediction. Instead, it utilises these ratings to acquire knowledge and build a predictive model. In model-based approaches, the aim is to replicate and model the interactions between users and items, incorporating factors that represent the hidden characteristics of both users and items within the system. These factors can include user preferences and item categories, among others.

To implement a model-based system, the available data (usually in the form of a training dataset) is used to train the predictive model. The model is trained to understand the patterns and relationships between users, items, and their hidden characteristics. Once the model is trained, it can be applied to predict ratings for new items based on user behavior and item attributes.

Numerous model-based approaches exist for the task of item recommendation, each employing specific techniques. These techniques include Bayesian Clustering, which utilises probabilistic clustering algorithms; Latent Semantic Analysis, which applies matrix factorization techniques to uncover latent factors in user-item interactions; Support Vector Machines, which employ machine learning algorithms for classification and regression tasks; and Singular Value Decomposition, which decomposes the user-item rating matrix into lower-dimensional representations (Nilashi et al., 2012; Geetha & Renuka, 2019)

These model-based approaches offer a more sophisticated and comprehensive way to make recommendations by capturing underlying patterns and relationships within the data. They leverage machine learning and statistical techniques to create predictive models that can effectively estimate user ratings for new items, enhancing the recommendation process.

***2.2.3.2.1 Model-based collaborative filtering models***

**Decision and Regression Trees**: Decision trees are widely used machine learning algorithms known for their simplicity and interpretability. In collaborative filtering, decision trees are employed to hierarchically partition the data space using split criteria based on independent variables. For instance, in a binary decision tree, one branch predominantly contains one class while the other branch predominantly contains the other class. Various adaptations have been proposed to make decision and regression trees suitable for collaborative filtering recommender systems (Aggarwal, 2016).

**Naive Bayes**: Naive Bayes is a simple yet remarkably powerful predictive modeling algorithm. It calculates probabilities for each factor and class (categorical value) based on instance data and selects the outcome with the highest probability. In the context of recommender systems, items are typically treated as features, and users are considered as instances to infer missing entries using a classification model. To adapt Naive Bayes to the recommender systems domain, additional measures are incorporated, particularly addressing the challenge of rating sparsity (Aggarwal, 2016; Valdiviezo-Diaz, Ortega, Cobos & Lara-Cabrera, 2019).

**Rule-based**: Association rules learning was initially developed to discover patterns between products in large transaction datasets recorded by point-of-sale (POS) systems in supermarkets. For example, if a customer purchases paint and tape together, there is a likelihood they will also buy a brush. Association rules are valuable for generating recommendations when dealing with unary ratings matrices. The process involves identifying all association rules triggered by a particular customer and sorting them based on decreasing confidence. The top-k items derived from these rules are recommended to the customer. The literature on recommender systems encompasses various modifications and extensions of rule-based collaborative filtering (Aggarwal, 2016).

**Matrix factorization (MF)**: Matrix factorization is a widely adopted model-based collaborative filtering technique due to its accuracy, simplicity, and interpretability. The ratings given by users to items are represented using latent factors that capture the underlying features of users and items. By reducing the dimensionality of the ratings information, predictions are obtained by taking the dot product of the users' and items' hidden factors. The literature on MF encompasses numerous variations and refinements (Aggarwal, 2016; Bobadilla, Alonso & Hernando, 2020; Valdiviezo-Diaz, Ortega, Cobos & Lara-Cabrera, 2019).

**Deep learning techniques**: Neural Collaborative Filtering (NCF) is one of the popular deep learning techniques used in collaborative filtering. NCF replaces the traditional inner product operation of Matrix Factorization with a neural architecture that can learn complex functions from data. It offers a generic framework that can encompass and generalise matrix factorization. Notably, NCF has gained prominence in the field of recommender systems (He et al., 2017).

### 2.2.4 Hybrid filtering approach

Hybrid recommender systems have gained popularity as an effective approach to enhancing the accuracy of predictions in recommender systems. These systems aim to achieve better recommendation accuracy by combining collaborative filtering, content-based filtering, and other techniques. The hybrid approach can be implemented in several ways. Empirical evaluations consistently demonstrate that hybrid methods generate more accurate recommendations than independent approaches, such as pure collaborative and content-based methods (Da'u & Salim, 2019).

According to Roy and Dutta (2022), hybrid techniques involve the combination of two or more recommendation techniques to overcome limitations and enhance the accuracy and effectiveness of individual techniques. These combinations generally lead to improved performance and accuracy in recommender applications. Different approaches can be used to incorporate different techniques, such as combining the outcomes of separate techniques or employing content-based filtering within a collaborative method. Common approaches to hybridisation include meta-level, feature-augmentation, feature-combination, mixed hybridisation, cascade hybridisation, switching hybridisation, and weighted hybridisation.

Geetha, Safa, Fancy, and Saranya (2018) also state that a hybrid recommender system combines multiple recommendation techniques to generate recommendations. Compared to collaborative or content-based systems, hybrid systems typically achieve higher recommendation accuracy. By combining both approaches, the collective knowledge is increased, leading to more informed recommendations. This knowledge enhancement presents promising opportunities to explore novel ways of enriching collaborative filtering algorithms with content data and content-based algorithms with user behavior data.

Netflix serves as an exemplary case of the successful implementation of hybrid recommender systems (Balush et al., 2021). The recommendations provided by Netflix are generated by analysing the viewing and searching history of similar users (collaborative filtering) and suggesting movies that share similar characteristics with highly rated movies from a user's past preferences (content-based filtering) (Geetha & Renuka, 2019).

Overall, all authors agreed that hybrid recommender systems, which combine collaborative filtering, content-based filtering, and other techniques, are effective in improving the accuracy of recommendations. They emphasize that hybridisation enhances the performance and accuracy of individual techniques by overcoming limitations. Various approaches can be used to combine different techniques, leading to improved recommendation outcomes. The authors also highlight that hybrid systems typically achieve higher recommendation accuracy compared to purely collaborative or content-based systems. By combining approaches, the collective knowledge is increased, enabling more informed recommendations. The successful implementation of hybrid recommender systems by Netflix serves as a notable example.

#### 2.2.4.1 Types of hybrid recommendation systems

According to Çano and Morisio (2019), the different types of hybrid recommender systems can be described as follows:

**Weighted**: In a weighted recommendation system, multiple recommender systems are utilised in parallel. Each model's output is combined with static weightings. For example, a content-based model and an item-item collaborative filtering model can be combined, with each contributing 50% weight to the final prediction. The advantage of the weighted hybrid approach is its ability to integrate multiple models in a linear manner, supporting the recommendation process. Weighted hybrids are commonly used due to their simplicity and flexibility (Çano & Morisio, 2019).

**Switching**: The switching hybrid selects a single recommendation system based on the situation. For datasets that exhibit item-level sensitivity, the recommender selector criteria can be based on user profiles or other relevant features. This approach adds an additional layer to the recommendation model, allowing it to select the appropriate model for each situation. The switching hybrid is sensitive to the strengths and weaknesses of the constituent recommendation models (Çano & Morisio, 2019).

**Mixed**: The mixed hybrid approach generates different sets of candidate datasets based on user profiles and features. The recommendation system then inputs these candidate sets into the recommendation model accordingly, combining the predictions to produce the final recommendation. The mixed hybrid system is capable of making a large number of recommendations simultaneously, leveraging the appropriate model for each partial dataset to improve performance (Çano & Morisio, 2019).

**Feature combination**: In the feature combination hybrid, a virtual contributing recommendation model is introduced to the system. This model generates additional features that complement the original user profile dataset. For example, features from a collaborative recommendation model can be incorporated into a content-based recommendation model. The hybrid model considers both collaborative and content-based data from the subsystems, rather than relying solely on one model (Çano & Morisio, 2019).

**Feature augmentation**: Feature augmentation hybrids involve combining two techniques, where the output of one technique is used to enhance the operation of the other recommendation technique. The second technique relies on the output of the first, making these hybrids order sensitive. For instance, an association rules engine can generate similar items for any given item, which can then be used as augmented item attributes within a second recommender to improve its recommendations (Çano & Morisio, 2019).

**Cascade**: Cascade hybrids follow a strict hierarchical structure in the recommendation system. The main recommendation system produces the primary result, and a secondary model is used to address minor issues, such as breaking ties in scoring. Successor recommendations are constrained by the predecessor, resulting in precise and controlled outcomes without introducing additional items (Çano & Morisio, 2019).

**Meta-Level**: Meta-level hybrids are similar to feature augmentation hybrids, where a contributing model provides an augmented dataset to the main recommendation model. However, in meta-level hybrids, the original dataset is replaced with a learned model from the contributing model as the input to the main recommendation model. In short, the successor model leverages a model delta built by the predecessor. Unlike in cascade hybrids, subsequent recommenders in meta-level hybrids have no restrictions on the items they can recommend (Çano & Morisio, 2019).

### 2.2.5 A Comparative Analysis of Traditional Recommendation Algorithms

Over the past few years, the field of recommendation systems has experienced notable progress, fueled by the growing abundance of extensive datasets and the emergence of groundbreaking algorithms. Traditional recommendation algorithms have played a pivotal role in establishing the fundamentals of this discipline. In their study, Balush, Vysotska, and Albota (2021) conducted a comprehensive comparative analysis of these conventional recommendations. They meticulously scrutinized prominent recommendation algorithms, elucidating their inherent strengths, limitations, and specific domains of application. The Table 2. 1 depicts the comparative analysis of the traditional recommendation algorithms.

Table 2. 1: Comparison of the traditional recommendation algorithms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Description | Strengths | Limitations | Applications |
| Collaborative Filtering | Utilizes past user-item interactions to generate recommendations. | Effective with explicit feedback data. | * Cold-start problem for new users or items. * A lot of information about user ratings is needed. | E-commerce, movie/music or book recommen-dations. |
| Content-Based Filtering | Recommends items based on their similarity to items the user has shown interest in. | * Incorporates item characteristics and attributes. * Works instantly, even for first users. * Works correctly, even with a small amount of data. | * May lack serendipity and diverse recommendation. * Tied to the content of the service. * Not based on the wishes of users. | News articles, document recommendations. |
| Hybrid Approaches | Combines multiple algorithms to leverage their strengths and mitigate weaknesses. | * Improved accuracy and flexibility. | * Complexity in combining and integrating algorithms. | E-commerce, personalised recommendations. |

## 2.3 Deep Learning in Recommendation Systems

### 2.3.1 Introduction to deep learning

Deep learning, also known as deep structured learning or hierarchical learning, is a branch of machine learning that encompasses a range of techniques designed to create high-level abstract models from data. The architecture of deep learning draws inspiration from artificial intelligence and seeks to simulate the deep learning process and sensory layers of the human brain. By learning from past experiences, deep learning models can make decisions to solve specific problems encountered by machines (Bochie et al., 2021).

One of the primary objectives of deep learning techniques is to automatically extract relevant features and abstractions from data. This process involves identifying the most critical information required for a given task (Mijwil & Abttan, 2021). These techniques are particularly effective when dealing with large volumes of unsupervised data, enabling the models to learn representations that capture the inherent structure and patterns present in the data. Deep learning models are capable of achieving high accuracy in tasks such as data classification and decision-making (Al-Zubaidi, Mijwil & Alsaadi, 2019).

The field of image recognition has witnessed the remarkable impact of deep learning techniques. In the 2016 ImageNet image classification competition, deep learning achieved unprecedented accuracy, surpassing 97% (Haiming et al., 2021). This breakthrough showcases the power of deep learning in surpassing traditional methods and pushing the boundaries of image recognition capabilities.

Deep learning has also revolutionized machine translation, particularly with the development of the Google Neural Machine Translation System (GNMT). Wu, Schuster, Chen, et al. (2016) demonstrated that GNMT achieved performance levels close to human translation in both English-Spanish and English-French translations. This advancement not only improves communication across language barriers but also showcases the potential of deep learning to tackle complex language tasks.

Furthermore, deep learning has made significant contributions to the field of speech recognition. Leading players in this domain, including Baidu, Xunfei, and Sogou, have utilised deep learning techniques to achieve Chinese speech recognition accuracies exceeding 97% (Haiming et al., 2021). Such breakthroughs in speech recognition have paved the way for applications in voice assistants, transcription services, and other speech-related tasks.

In the domain of online advertisement, deep learning has emerged as a powerful tool for predicting click-through rates. Industry giants like Google and Microsoft have leveraged deep learning to improve the accuracy of their advertising platforms (Cheng, Koc, Harmsen, et al., 2016; Zhu, Shan, Mao, 2017). This application of deep learning has revolutionized online advertising by enabling targeted and personalised ad delivery, optimizing marketing campaigns, and enhancing user experience.

Deep learning has experienced remarkable growth in the big data and artificial intelligence domain (Silver, Huang, Maddison, et al., 2016). By combining low-level features, deep learning creates richer, high-level abstractions that automatically uncover distributed data features. This transformative approach eliminates the need for manual feature design in traditional machine learning, leading to breakthroughs in various domains, including image recognition, machine translation, speech recognition, online advertising, and recommendation systems (Haiming, Kaili, Yunyun & Xuefeng, 2021).

In recent years, deep learning techniques have found applications in various domains, including the classification of medical images. Researchers have successfully applied deep learning models to automatically analyse and interpret medical images, aiding in the diagnosis of diseases and the identification of anomalies (Mehta, Aggarwal, Koundal et al., 2021). By leveraging the hierarchical learning capabilities of deep learning, these models can extract intricate features and patterns from medical images, enabling accurate and efficient analysis (Li, Zhao, et al., 2021).

The growth of deep learning in big data and artificial intelligence has been fueled by its ability to automatically extract meaningful features from data, surpassing traditional machine learning approaches. Its impact spans across diverse domains, including image recognition, machine translation, speech recognition, and online advertising. These advancements have not only achieved unprecedented levels of accuracy but also opened new possibilities for solving complex problems and driving innovation in various industries (Zhang, Yang, Chen & Li, 2017).

### 2.3.2 Advantages of deep learning in recommendation systems

Deep learning has revolutionized various domains, and one area where it has made significant advancements is recommendation systems. Numerous articles and journals highlight the advantages of deep learning in recommendation systems. This section discusses some of the key advantages, drawing references from relevant articles.

1. **Enhanced Representation Learning:**

Deep learning models, such as deep neural networks, excel at learning complex patterns and representations from raw data. This ability is particularly advantageous in recommendation systems, as it enables the extraction of intricate features from user behavior and item attributes. According to Cheng et al. (2016), deep learning techniques facilitate the automatic discovery of latent features that capture users' preferences and item characteristics, leading to improved recommendation accuracy.

1. **Improved Recommendation Accuracy:**

Deep learning models have demonstrated superior performance in recommendation tasks compared to traditional techniques. By employing deep neural networks, recommendation systems can effectively capture nonlinear relationships and exploit intricate dependencies between users and items. Bell and Koren (2007) highlight that deep learning algorithms, such as deep belief networks and autoencoders, have shown promising results in enhancing recommendation accuracy by effectively modeling complex user-item interactions.

1. **Handling Sparse and Cold-Start Problems:**

Recommendation systems often encounter sparse data, where there are limited interactions between users and items. Deep learning methods can address this challenge by effectively handling sparse data representation. Zhang et al. (2019) propose a deep collaborative filtering model that incorporates side information, such as user and item attributes, to overcome data sparsity. This approach leverages deep learning techniques to learn meaningful representations from auxiliary data, thereby improving recommendation performance in sparse scenarios.

1. **Scalability and Adaptability:**

Deep learning models exhibit scalability and adaptability, making them suitable for large-scale recommendation systems. They can handle massive datasets and efficiently train on powerful computational resources. Furthermore, deep learning models can adapt to evolving user preferences and item characteristics over time. Hidasi et al. (2015) propose a recurrent neural network-based recommendation model that captures temporal dynamics, enabling personalised recommendations that adapt to users' changing interests.

1. **Handling Heterogeneous Data:**

Recommendation systems often incorporate various data sources, including textual information, images, and social network connections. Deep learning techniques excel at handling heterogeneous data types and effectively extracting meaningful representations from diverse sources. According to Covington et al. (2016), deep learning models, such as wide and deep neural networks, can integrate multiple data inputs to capture both memorization (exploitation of past behavior) and generalization (discovering new preferences) in recommendation systems.

Overall, deep learning offers several advantages in recommendation systems, including enhanced representation learning, improved recommendation accuracy, handling sparse and cold-start problems, scalability and adaptability, and handling heterogeneous data. These advantages have been highlighted in various articles, including Cheng et al. (2016), Bell and Koren (2007), Zhang et al. (2019), Hidasi et al. (2015), and Covington et al. (2016). The integration of deep learning techniques in recommendation systems continues to advance the field, providing more accurate and personalised recommendations to users.

### 2.3.3 Deep learning architectures commonly used for recommendations

Deep learning architectures have significantly advanced the field of recommendation systems, providing improved accuracy and performance. Several deep learning models have been developed and widely used for recommendation tasks. This section discusses some of the commonly employed deep learning architectures in recommendation systems, based on relevant literature.

1. **Convolutional Neural Networks (CNNs):**

The Convolutional Neural Network (CNN) is a multi-layer perceptron architecture commonly utilised for processing two-dimensional image data. It consists of an input layer, convolution layer, pooling layer, fully connected layer, and output layer. Unlike traditional multi-layer perceptrons, CNNs incorporate pooling operations to reduce the number of neurons in the model and exhibit enhanced robustness to translation invariance within the input space (Zhang, Lu, Jin, 2020).

CNNs have found extensive applications in recommendation systems (Rawat & Kankanhalli, 2016). They are primarily employed to extract hidden features from various types of content, including images, text, and audio, in combination with user implicit representations (Balasubramanian, Diwan & Vora, 2020). This enables the generation of personalised recommendations for users, with typical applications in image recommendation, music recommendation, text recommendation, and more (Haiming, et al., 2021).

The strength of Convolutional Neural Networks lies in their ability to effectively capture local patterns and features, making them well-suited for modeling item content or user behavior sequences. For instance, He et al. (2017) propose a deep collaborative filtering model that leverages CNNs to capture item content information, such as images or textual descriptions, thereby enhancing the overall recommendation performance.

1. **Recurrent Neural Networks (RNNs):**

The use of Recurrent Neural Networks (RNNs) in recommendation systems has gained significant popularity due to their ability to capture sequential dependencies and temporal dynamics in user behavior. Unlike normal fully connected networks or convolutional neural networks, where the layers are fully connected and the nodes between each layer are disconnected, RNNs have interconnected nodes between their layers. This connectivity allows RNNs to calculate the output of the hidden layer at the current moment by considering both the input layer's output and the hidden layer's state from the previous moment (Haiming et al., 2021).

RNNs, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have proven effective in modeling sequential patterns and capturing long-term dependencies. For instance, Hidasi et al. (2015) proposed a session-based recommendation model that leverages RNNs to capture sequential user interactions and provide personalised recommendations. By employing RNNs, this model can effectively consider the order and context of user actions to generate accurate and relevant recommendations.

1. **Autoencoders:**

Autoencoders are a type of unsupervised deep learning model that aims to learn efficient data representations by reconstructing input data from compressed latent representations (Wang, Yao & Zhao, 2016). In contrast to traditional neural networks, autoencoders learn the hidden layer representation of the data through an encoding and decoding process (Haiming et al., 2021). The basic structure of an autoencoder consists of three layers: an input layer, a hidden layer, and an output layer.

The primary objective of an autoencoder is to minimise the discrepancy between the input and output. However, if the model is trained solely by minimising the error between the input and output, it may learn to perform a trivial identity function. To address this issue, researchers have proposed several variants of autoencoders, including sparse autoencoders and denoising autoencoders (Bengio, Lamblin & Popovici, 2007).

In the context of recommendation systems, autoencoders are primarily employed to learn hidden feature representations of users and items. These representations are then used to predict users' preferences for items. The application scenarios of autoencoders in recommendation systems encompass scoring prediction, text recommendation, image recommendation, and more (Haiming et al., 2021).

Autoencoders have demonstrated successful applications in recommendation systems, particularly for collaborative filtering tasks. The denoising autoencoder approach, introduced by Vincent et al. (2010), is capable of handling missing or noisy user-item interaction data. By effectively learning robust latent representations, denoising autoencoders contribute to more accurate and reliable recommendations.

1. **Variational Autoencoders (VAEs):**

Variational Autoencoders are generative models that combine autoencoders with probabilistic techniques, allowing the modeling of uncertainty in recommendation tasks. VAEs enable the generation of diverse recommendations and provide a continuous latent space representation for users and items. Liang et al. (2018) propose a VAE-based recommendation model that incorporates both collaborative filtering and content-based information, producing accurate and diverse recommendations.

1. **Transformer-based Models:**

Transformer-based models, such as the Transformer architecture introduced by Vaswani et al. (2017), have gained popularity in recommendation systems. Transformers are designed to capture long-range dependencies and enable parallel computation, making them efficient for large-scale recommendation tasks. Several transformer-based models, such as BERT and GPT, have been adapted for recommendation systems, effectively modeling user-item interactions and capturing contextual information.

In summary, deep learning architectures commonly used in recommendation systems include Convolutional Neural Networks (CNNs) for item content modeling, Recurrent Neural Networks (RNNs) for sequential and temporal modeling, Autoencoders and Variational Autoencoders (VAEs) for collaborative filtering tasks, and Transformer-based models for capturing long-range dependencies and contextual information. These architectures have been proposed and applied in various research studies, including He et al. (2017), Hidasi et al. (2015), Vincent et al. (2010), Liang et al. (2018), and Vaswani et al. (2017). Leveraging these deep learning architectures allows recommendation systems to effectively model complex user-item interactions and provide accurate and personalised recommendations.

## 2.4 Deep learning Models for Book Recommendations

Deep learning models have significantly advanced the field of book recommendation systems, enabling accurate and personalised recommendations to users. Researchers have explored various deep learning approaches to enhance book recommendations. This section provides an overview of the key methodologies and techniques, drawing references from relevant articles published from 2018 onwards.

### 2.4.1 Neural Network-Based Models

Neural network-based models have proven to be effective in book recommendation systems, leveraging their ability to capture complex patterns and representations. Wang et al. (2018) propose a book recommendation model based on a deep neural network that incorporates both textual book content and user behavior data. The model utilises techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to extract meaningful features from book descriptions and user preferences, leading to improved recommendation accuracy. This approach demonstrates the capability of neural network-based models to leverage rich book content information for better recommendations.

### 2.4.2 Collaborative Filtering with Deep Learning

Collaborative filtering, a popular technique in recommendation systems, has been enhanced by incorporating deep learning methods. Deep collaborative filtering models have shown promising results in book recommendations. Zhang et al. (2019) propose a deep collaborative filtering model that combines matrix factorization with deep neural networks. The model captures both explicit user-item interactions, such as ratings and reviews, as well as implicit interactions, such as user browsing and purchase history. By leveraging the power of deep learning, this hybrid approach significantly improves recommendation accuracy for books. It highlights the capability of deep learning models to capture intricate user-item relationships and preferences.

### 2.4.3 Hybrid Models Combining Deep Learning and Traditional Techniques

Hybrid models that combine deep learning approaches with traditional recommendation techniques have gained attention in book recommendation systems. These models aim to leverage the strengths of both approaches to provide accurate and diverse recommendations. Huang et al. (2018) propose a hybrid recommendation model that combines deep learning-based content analysis with collaborative filtering. The model utilises deep neural networks to extract book content features and user preferences and combines them with collaborative filtering techniques to generate personalised recommendations. This hybrid approach effectively captures both the semantic meaning of books and user-item interactions, resulting in improved recommendation quality.

Furthermore, other hybrid models have been explored, such as the combination of deep learning with knowledge graph embeddings. Chen et al. (2019) propose a hybrid model that incorporates both collaborative filtering and knowledge graph embeddings to enhance book recommendations. By integrating deep learning techniques with structured knowledge about books and user preferences, this approach achieves better recommendation performance by capturing both content-based and collaborative filtering signals.

Overall, deep learning models have significantly advanced book recommendation systems. Neural network-based models effectively capture complex patterns and representations from book content and user behavior. Collaborative filtering models with deep learning techniques leverage both explicit and implicit user-item interactions. Hybrid models combining deep learning with traditional techniques capture both book content features and user preferences, leading to accurate and diverse recommendations. These approaches demonstrate the substantial progress and potential of deep learning in enhancing book recommendations.

## 2.5 Evaluation Metrics and Techniques

In order to assess the effectiveness and performance of personalised book recommendation systems, various evaluation metrics and techniques have been employed. These metrics provide insights into the accuracy, diversity, coverage, and novelty of the recommendations, enabling researchers and practitioners to measure the quality of their systems and compare them against existing approaches. This section discusses common evaluation metrics for book recommendation systems.

### 2.5.1 Common Evaluation Metrics for Book Recommendation Systems

#### 2.5.1.1 Mean Square Error

Mean Square Error (MSE) is a commonly used evaluation metric for rating prediction accuracy in book recommendation systems. It measures the average squared difference between the predicted ratings and the actual ratings for a set of recommendations (Koren et al., 2009).

MSE =

where N is the total number of recommendations, is the predicted rating for the i-th recommendation, and is the corresponding actual rating.

#### 2.5.1.2 Precision and Recall

Precision (P) measures the proportion of relevant recommendations out of the total recommendations made. It focuses on the accuracy of the system by evaluating the fraction of correctly recommended items. Precision is computed as the ratio of true positives (TP) to the sum of true positives (TP) and false positives (FP).

P =

Recall (R) measures the proportion of relevant recommendations that are successfully retrieved by the system. It focuses on the system's ability to capture all the relevant items. Recall is computed as the ratio of true positives (TP) to the sum of true positives and false negatives (FN).

R =

These metrics are particularly useful when dealing with binary relevance judgments, where a book is either relevant or not. By calculating precision and recall, researchers can determine the accuracy of the recommendations made by the system (Sarwar et al., 2018).

#### 2.5.1.3 F1-Score

F1-Score: The F1-Score is the harmonic mean of precision and recall, providing a balanced measure of the system's performance. It combines both precision and recall into a single metric, taking into account both false positives and false negatives (Zhang et al., 2018).

F1-Score

The F1-Score ranges between 0 and 1, with 1 indicating the best performance.

#### 2.5.1.4 Mean Average Precision (MAP)

Mean Average Precision (MAP) calculates the average precision at each relevant item's position and then takes the mean across all users. It considers both the relevance and the rank of recommended books. The formula for MAP is as follows:

MAP

where N is the total number of recommended items, is the position of the i-th relevant item in the recommendation list, and is the relevance of the i-th relevant item (usually represented as 1 for relevant and 0 for non-relevant) (Huang et al., 2019).

#### 2.5.1.5 Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG) is a metric that assesses the ranking quality of recommended books. It takes into account the relevance and position of recommended items, giving higher scores to relevant books appearing at the top of the list. NDCG accounts for the diminishing returns of relevance as the position increases. The formula for NDCG is given by:

NDCG =

where DCG (Discounted Cumulative Gain) calculates the sum of relevance scores at each position, discounted by a logarithmic function, and IDCG (Ideal Discounted Cumulative Gain) represents the maximum achievable DCG. NDCG ranges from 0 to 1, with higher values indicating better performance (Li et al., 2018).

#### 2.5.1.6 Coverage and Diversity

Coverage refers to the percentage of books in the entire catalog that are recommended to users. It can be calculated using the formula:

Coverage

Diversity measures the variety of recommended books, ensuring that the system offers diverse options to cater to different user preferences. Intra-list diversity is commonly used to quantify this aspect, which calculates the average dissimilarity between pairs of recommended books in a list (Yang et al., 2018).

#### 2.5.1.7 Novelty

Novelty measures the degree to which recommended books differ from the user's prior knowledge or past interactions. Average novelty and self-information are commonly used metrics to evaluate the novelty of recommended books. Average novelty is calculated as the average dissimilarity between recommended books and the user's historical interactions. Self-information measures the information gain provided by a recommended book and is calculated using the formula:

where P is the probability of the book being recommended (Hu et al., 2019).

In conclusion, evaluating personalised book recommendation systems requires the utilisation of appropriate metrics and techniques. Precision, recall, MAP, NDCG, coverage, diversity, and novelty are commonly used metrics in this domain. These metrics provide valuable insights into the accuracy, ranking quality, coverage, diversity, and novelty of recommendations, enabling researchers and practitioners to assess and compare the performance of their systems effectively.

## 2.6 Appraisal of the related works

Yuan, Karatzoglou, Arapakis, Jose & He (2019) present a study focused on session-based next item recommendation using Convolutional Neural Networks (CNNs). They introduce a novel approach where a session's collection of past items is transformed into a 2-dimensional latent matrix, treating it as an image. Convolution and pooling operations are then applied to the embedded item sequences. The authors identify limitations in the existing session-based CNN recommender, particularly in modeling long-range dependencies within item sequences. To address these issues, they propose a simple yet highly effective generative model that can capture both short and long-range item dependencies. The network architecture of their model consists of a series of holed convolutional layers, enabling the expansion of receptive fields without relying on pooling operations. Additionally, the authors incorporate a residual block structure, which aids in optimizing deeper networks in recommender systems. Through experiments, they demonstrate that their proposed generative model achieves state-of-the-art accuracy in next item recommendation tasks while requiring less training time. This model is positioned as a strong baseline for future recommendation systems, especially when dealing with long sequences of user feedback. The work contributes to the advancement of session-based recommendation algorithms and offers insights into optimizing CNN architectures for modeling dependencies in sequential data.

The work by Hao Wang, Naiyan Wang, and Dit-Yan Yeung (2015) focuses on addressing the sparsity problem in Collaborative Filtering (CF)-based recommender systems. CF methods typically rely solely on user ratings to make recommendations, but in many applications, these ratings are sparse, leading to a degradation in recommendation performance. To mitigate this issue, the authors propose utilising auxiliary information such as item content data. They highlight Collaborative Topic Regression (CTR) as an appealing method that incorporates this auxiliary information. However, when the auxiliary information is sparse, the effectiveness of the learned latent representation in CTR is diminished. To tackle this problem, the authors generalise recent advances in deep learning, which are typically applied to independent and identically distributed (i.d.) input, to the non-i.d. CF-based input. They introduce a hierarchical Bayesian model called Collaborative Deep Learning (CDL) that combines deep representation learning for content information and collaborative filtering for the ratings matrix. The authors conduct extensive experiments on three real-world datasets from different domains, demonstrating that CDL significantly outperforms existing methods and advances the state of the art in recommender systems. This work contributes to the field by introducing a novel approach that leverages deep learning techniques to improve recommendation performance, especially in scenarios with sparse auxiliary information.

The work by Sarma, Mittra, and Hossain (2021) addresses the challenge of finding relevant books in the vast e-book space, which has exponentially grown due to the COVID-19 pandemic. Personal recommendation systems have emerged to help users search effectively by mining related books based on user ratings and interests. However, existing systems primarily rely on user-based ratings, which can be problematic as they include ratings from users who have unsubscribed from the services and no longer provide ratings for books. In this paper, the authors propose an effective book recommendation system for online users. They employ the clustering method to rate a book and then measure its similarity to suggest new books. The proposed system utilises the K-means Cosine Distance function to measure distance between book clusters and the Cosine Similarity function to find similarity. The evaluation of the system involves calculating Sensitivity, Specificity, and F Score across ten different datasets. The results demonstrate that the average Specificity is higher than Sensitivity, indicating that the classifier can remove uninteresting books from the reader's list. Overall, the study concludes that recommendations based on a particular book are more accurate and effective compared to user-based recommendation systems. This work contributes to the development of book recommendation systems and highlights the benefits of utilising book clustering and similarity measures for improved recommendations.

Li & Kim (2021) present a novel deep learning-based course recommender system (DECOR). In their study, Li and Kim address the increasing demand for online education platforms amidst the increasing demand caused by the COVID-19 pandemic. They highlight the challenge of selecting appropriate course content due to variations in users' knowledge structures. While traditional data mining methods like Collaborative Filtering (CF) have been utilised in recommender systems, they acknowledge the limitations of such approaches, including sparsity and scalability issues. To overcome these challenges, Li and Kim propose a novel deep learning model called DECOR (Deep Learning-based Course Recommender System). DECOR captures high-level user behaviors and course attribute features, effectively reducing information overload and addressing high-dimensional data sparsity problems. Through experiments on real-world datasets, they demonstrate that DECOR outperforms traditional recommendation approaches, offering improved and more robust recommendation performance. This study contributes to the exploration of deep learning models for personalised book recommendation systems and highlights the potential of DECOR as a viable solution in the context of online education platforms.

The study conducted by Ma, Jiang et al. (2021) focuses on personalised recommendation systems and their extension to the University Library Lending system. They acknowledge the benefits of recommendation systems developed by major companies like Google, Amazon, and Alibaba, which are based on big data analysis. However, traditional recommendation algorithms face challenges in datasets with large data sparsity, leading to unsatisfactory results and a failure to discover users' potential interests. To address this, the researchers collected readers' and books' information from the borrowing records of Qinghai University Library over the past 20 years. They applied the Wide and Deep model, combining logistic regression (LR) and deep neural network (DNN) networks, to train a recommendation model. Notably, they improved the Wide and Deep model by incorporating multiple labels and conducted extensive training to obtain the final model. The experimental results demonstrate that their book recommendation model significantly outperforms traditional and hybrid recommendation algorithms in terms of accuracy. The researchers introduce the novelty of their study by creating a large training and testing dataset using Qinghai University's book data, and they highlight the suitability of the improved Wide and Deep model for book recommendation systems. The performance comparison shows that the Wide & Deep model achieves the highest AUC value of 0.75, making it suitable for personalised book recommendation systems dealing with sparse big data. This research contributes to the advancement of recommendation systems in the context of university libraries and highlights the efficacy of the Wide and Deep model for such applications.

The work by Wadikar, Kumari, Bhat, and Shirodkar (2020) focuses on the development of a subject-based book recommendation platform utilising Convolutional Neural Network (CNN) techniques. The authors highlight the challenges of extracting useful information from the vast and complex online data available to users. Recommender systems are identified as effective software techniques to address this problem by providing personalised recommendations based on user and item information. The applications of recommender systems are wide-ranging, including suggesting items for online shopping, recommending articles or books for reading, suggesting movies or music, and providing news recommendations. In this study, the authors propose a book recommendation platform specifically focused on subjects. The platform utilises CNN to recommend books to users based on their subject of interest. Users will have the ability to view and search books, and the recommender system, employing CNN, will generate a list of highly purchased and top-rated books related to the subject input provided by the user. This work contributes to the field of book recommendation systems and demonstrates the application of CNN in providing subject-based book recommendations.

The study conducted by Yiu-Kai Ng and Urim Jung (2019) focuses on personalised book recommendation systems using the Recurrent Neural Network (RNN) model and metadata. They address the challenge of recommending relevant books without accessing their content due to copyright restrictions by leveraging book records containing various metadata such as descriptions, ratings, and reviews. Their approach simplifies the recommendation process compared to existing systems while still providing essential information about the books. The authors achieve a classification accuracy of 73% on book data using their proposed model, demonstrating the effectiveness of metadata and deep learning techniques for accurate and relevant book recommendations. Overall, their work contributes to the advancement of personalised book recommendation systems by utilising metadata and deep learning methods while respecting copyright restrictions.

# CHAPTER THREE

**METHODOLOGY**

## 3.0 Preamble

The structured procedure for the methodologies are described in details in this chapter.

## 3.1 Introduction to the BookCrossing dataset

The “BookCrossing” dataset, also known as the “BX-Dump” dataset, is a collection of data related to a book-sharing website called BookCrossing. This dataset was created for research purposes and includes information about books, users, and their interactions on the BookCrossing platform. It was collected by Cai-Nicolas Ziegler in a 4-week crawl (August/September 2004) from the BookCrossing community and contains 278,858 users (anonymized) providing 1,149,780 ratings (explicit/implicit) about 271,379 books. The dataset is valuable for research in recommender systems, collaborative filtering, and book recommendations. Researchers and data scientists can use it to develop algorithms that suggest books to users based on their preferences, historical ratings, and the behaviour of other users.

It is important to note that while the dataset provides a rich source of information, it may have limitations such as incomplete data, inconsistencies, or outdated information. Proper data preprocessing and cleaning will be needed to make the building of the recommendation model efficient. The Table 3. 1 briefly describe the BookCrossing data used in this project.

Table 3. 1: A comprehensive description of the dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **Given Datasets** | **Variables** | **Description** | **Number of observations** |
| Books | ISBN (International Standard Book Number) | A unique identifier for each book. | 271,361 |
| Book Title | The title of the book. | 271,361 |
| Book Author | The author of the book. | 271,360 |
| Year of Publication | The year the book was published. | 271,361 |
| Publisher | The publishing company responsible for the book. | 271,359 |
| Image URLs | URLs linking to images of book covers. | 271,361 |
| Users | User-ID | A unique identifier for each user. | 278,858 |
| Location | The location of the user, typically including city and state/country. | 278,858 |
| Age | The age of the user, if available. | 168,096 |
| Ratings | User-ID | The identifier of the user who provided the rating. | 1,149,780 |
| ISBN | The ISBN of the book being rated. | 1,149,780 |
| Book Rating | A numerical rating (0-10) given by the user to the book. | 1,149,780 |

## 3.2 Utilisation of the Python Programming Language

The utilisation of the Python programming language in this project section plays a pivotal role in facilitating the development and implementation of personalised book recommendation systems. Python is selected as the primary programming language due to its versatility, extensive libraries, and robust ecosystem, making it well-suited for machine learning and deep learning tasks.

Python’s core strengths lie in its readability and ease of use, which foster efficient code development and maintenance. Additionally, Python boasts a wealth of libraries such as NumPy, Pandas, Matplotlib, and TensorFlow, which provide essential tools for data manipulation, analysis, visualisation, and deep learning model construction.

Throughout this project, Python will be harnessed to perform a spectrum of tasks, including data preprocessing, feature engineering, model development, training, and evaluation. Its compatibility with popular deep learning frameworks like TensorFlow and PyTorch further streamlines the implementation of intricate neural network architectures, a fundamental component of personalised book recommendation systems.

The use of Python as the programming language for this project not only ensures a high degree of flexibility and efficiency but also leverages its extensive ecosystem to facilitate the creation of a sophisticated and effective personalised book recommendation model.

## 3.3 Data Exploratory Techniques

The application of the data exploratory techniques section is a fundamental phase in the development of the personalised book recommendation system, aimed at gaining a comprehensive understanding of the BookCrossing dataset and preparing it for subsequent stages of data analysis and model development. Initial data inspection involved a rigorous examination of the dataset’s structural properties, dimensions, and basic statistical characteristics. This preliminary analysis provided essential insights into the data’s composition, including the number of users, books, and ratings.

Data visualisation techniques were employed to convey a deeper understanding of the dataset. This encompassed the creation of histograms, scatter plots, etc. to visually depict data distribution, interrelationships, and emerging patterns.

## 3.4 Data Cleaning

Data cleaning is a critical phase in the development of all deep learning models and personalised book recommendation systems. It involves a systematic process of identifying and rectifying inconsistencies, errors, and outliers within the dataset. This section elucidates the importance of data cleaning, the specific techniques applied, and the rationale behind each step.

### 3.4.1 Importance of Data Cleaning

Data quality is fundamental to the integrity and accuracy of our recommendation system. The following reasons underscore the paramount importance of data cleaning:

1. **Enhanced Model Performance**: Clean data ensures that our machine learning model is trained on accurate and reliable information. This, in turn, leads to improved recommendation accuracy and user satisfaction.
2. **Reduction of Bias**: Unaddressed outliers and errors in the data can introduce bias into the recommendation model, skewing recommendation in an unintended manner. Data cleaning mitigates such biases.
3. **Consistency and Reliability**: Cleaned data ensures that user ratings, book information, and other relevant attributes are consistent and reliable, providing a solid foundation for subsequent analysis.

### 3.4.2 Data Cleaning Techniques

Our data cleaning process involved the application of the following techniques:

#### 3.4.2.1 Handling Missing Values

A dataset is considered to have missing values when data are absent in certain observations of variables. The occurrence of missing values in a dataset can be attributed to various factors, including errors during the data collection process, data entry mistakes, deliberate omissions stemming from privacy concerns, or a lack of available data. The presence of missing values in a dataset can significantly impact the quality and reliability of the data.

Missing values possess the potential to introduce inaccuracies into analytical results, leading to suboptimal decision-making outcomes. Furthermore, they can distort key statistical measurements such as mean, variance, and correlations, which are critical for drawing meaningful conclusions from the data. In addition, missing values can introduce bias into predictive modelling, potentially causing misinterpretations of the data's patterns and relationships.

To address the issue of missing values, various techniques were employed. Specifically, missing ratings were imputed using methods such as mean imputation or predictive imputation based on user and book characteristics. This approach serves to maintain the integrity of the user-item interaction matrix, enhance the overall data quality, and render it suitable for training recommendation models.

#### 3.4.2.2 Outlier Detection and Treatment

An outlier is an observation or data point that is significantly different from other data points in a dataset. In statistical analysis and data science, outliers are values that deviate significantly from the typical or expected range of values within a dataset. They can be unusually high or low in comparison to the majority of the data points and can distort statistical analyses and machine learning models if not handled properly. Outliers can occur for various reasons, including:

**Data Entry Errors**: Typos or mistakes during data entry can lead to values that are far from the actual data distribution.

**Natural Variability**: In some cases, outliers may represent legitimate data points that reflect unusual or rare events or phenomena.

**Measurement Errors**: Errors in measurement instruments or sensors can result in outliers. Data Processing Errors: Errors during data preprocessing or data cleaning can introduce outliers.

**Sampling Errors**: If the sampling process is not conducted properly, it can lead to outliers in the collected data.

It's important to identify and deal with outliers appropriately in the data cleaning, as they can affect the accuracy and reliability of statistical summaries and models.

#### 3.4.2.3 Handling Duplicates

Duplicates in a dataset refer to the presence of identical or nearly identical records, observations, or data points within the dataset. Duplicates can occur in various types of data, including tabular data, databases, text documents, and more. Identifying and handling duplicates is a crucial data cleaning and preprocessing step in data analysis and machine learning, as they can lead to skewed or incorrect results if not properly addressed.

Duplicates entries, if present, were identified and removed to prevent redundancy in the dataset. Duplicate user-book interactions could distort the recommendation model and lead to biased results.

#### 3.4.2.4 Label Encoding

Label encoding is a fundamental data preprocessing step aimed at converting categorical variables, typically represented as strings, into discrete numerical values. This transformation assigns a unique integer identifier to each distinct category within the categorical variable, enabling the representation of categorical data in an integer format. This process holds significant importance within the realm of machine learning, as a vast majority of machine learning algorithms are designed to work with numerical data, making label encoding an essential step in preparing data for training and deployment, especially in recommendation systems.

By assigning numerical labels to categorical values, label encoding allows machine learning models to effectively process and learn from the categorical information. This facilitates better model performance and accuracy in predictive tasks. Without this conversion, algorithms might struggle to make meaningful interpretations from categorical data, as they primarily rely on mathematical operations that require numeric inputs.

#### 3.4.2.4 Addressing Data Integrity

Data integrity refers to the accuracy, reliability, and consistency of data throughout its entire lifecycle, from creation or capture to storage, retrieval, and usage. It ensures that data remains unaltered and trustworthy, reflecting the intended information without errors, corruption, or unauthorised modifications.

Key aspects of data integrity include:

1. Accuracy: Data should be correct and free from errors or inaccuracies. It should represent the real-world entities or events it is intended to describe.
2. Reliability: Reliable data is consistent and can be depended upon for decision-making, analysis, and other purposes. It should not change arbitrarily or unpredictably.
3. Completeness: Complete data includes all the required information and fields, leaving no gaps or missing values that could hinder its usability.
4. Consistency: Data should be internally consistent, meaning that it does not contain conflicting information or values that do not align with established rules or constraints.
5. Availability: Data should be accessible when needed by authorised users. This involves ensuring that data is stored in a reliable manner and can be retrieved promptly.

Data integrity is crucial in various domains, including business, healthcare, finance, and scientific research, where data serves as the foundation for decision-making and analysis. Maintaining data integrity is often achieved through data validation processes, regular backups, data encryption, access controls, and audit trails. Ensuring data integrity helps organizations and individuals trust and rely on the information they work with, minimising the risks associated with inaccurate or compromised data.

Integrity issues, such as inconsistent age formats or non-numeric characters in numerical fields, were resolved to maintain data integrity and facilitate subsequent data analysis.

### 3.4.3 Rationale for Data Cleaning

Each data cleaning step was undertaken with specific objectives in mind:

* Handling missing values ensures that no user or book is omitted from the analysis, preserving the completeness of the dataset.
* Outlier treatment prevents extreme ratings from unduly influencing recommendations and promotes fairness in the system.
* Addressing inconsistent data ensures that book and user attributes are consistently represented, allowing for meaningful analysis and recommendation.
* Duplicate removal eliminates redundancy and reduces computational overhead during model training.
* Addressing data integrity issues maintains the overall quality and reliability of the dataset.

## 3.5 Division Methodology for Training and Testing Sets

The process of partitioning the dataset into distinct training and testing sets holds paramount importance for the following reasons:

1. **Model Evaluation**: It facilitates the rigorous evaluation of our recommendation model’s performance. By testing the model on unseen data, we assess its generalization capabilities and ensure that it provides meaningful recommendations to users.
2. **Overfitting mitigation**: A well-structured data division helps guard against overfitting, ensuring that the model does not merely memorize the training data but learns relevant patterns that can be applied to new data.
3. **Real-world Simulation**: It simulates real-world conditions where the system must make recommendation for previously unrated books, enhancing the model’s practical utility.

### 3.5.1 Data Division Techniques

There are several data division techniques commonly used in data splitting for tasks like machine learning, deep learning etc. but for case of this study, the train-test-split technique was used because of its efficiency in grouping both the training and testing set in the given proportion in a given random state.

The following describe the most commonly used techniques for data division:

1. **Train-Test Split**: The data is divided into two parts, typically a larger portion for training the model and a smaller portion for testing its performance.
2. **K-Fold Crossing-Validation**: The data is divided into K subsets folds, where the model is trained on K-1 folds and tested on the remaining fold. This process is repeated K times, with each fold used as the test set once.
3. **Stratified Sampling**: This technique ensures that each class or category in the dataset is represented proportionally in both the training and testing sets, which is particularly useful for imbalanced datasets.
4. **Time Series Split**: When dealing with time-series data, it’s essential to split the data sequentially to maintain the temporal order. This technique is commonly used in forecasting and prediction tasks.
5. **Grouped Spli**t: In cases where data points have inherent groupings or dependencies, such as medical records for different patients, the data can be split to ensure that all data from a particular group is in either the training or testing set.
6. **Leave-One-Out Cross-Validation (LOOCV)**: In this approach, a single data point is used as the test set while the rest of the data is used for training. This process is repeated for each data point in the dataset.

The establishment of a data division methodology for training and testing sets is a critical step in the development of our personalised book recommendation system. It supports rigorous model evaluation, guards against overfitting and simulates real-world conditions. The chosen techniques, including random and stratified splitting, were employed to fulfil special specific objectives, ensuring the reliability and effectiveness of our recommendation model.

## 3.6 Book Recommendation Modelling Systems

The Book Recommendation Modeling Systems section outlines the approaches employed to evaluating and comparing collaborative filtering methods with diverse deep learning algorithms, including autoencoders, and transformer-based model to determine the most effective recommendation strategy.

### 3.6.1 Collaborative Filtering

Collaborative filtering serves as the baseline approach for our recommendation system:

**User-Based Collaborative Filtering**: We analyse user behavior patterns to identify similar users and recommend books based on their preferences.

**Item-Based Collaborative Filtering**: Similarity metrics between books are computed to suggest items aligned with user preferences.

### 3.6.2 Deep Learning Models

We explore the following deep learning techniques in parallel to collaborative filtering:

**Autoencoders**: Autoencoders are employed to learn latent representations of user-book interactions, capturing complex patterns in the data.

**Transformer Models**: Transformer-based architectures are applied to model long-range dependencies in user preferences and item characteristics effectively.

### 3.6.3 Model Training and Evaluation

Our methodology encompasses the following steps:

**Data Preparation**: Preprocessed data is fed into both collaborative filtering and deep learning models.

**Model Training**: Collaborative filtering models are trained using user-item interactions, while deep learning models are trained to capture intricate patterns within the data.

**Performance Metrics**: Model performance is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Precision, Recall and F1-score

### 3.6.4 Hyperparameter Tuning

Hyperparameter tuning is a critical step in the development of machine learning and deep learning models. It involves the systematic exploration of various hyperparameter settings to find the optimal configuration that enhances a model's performance and generalization capabilities. In this section, we elaborate on the significance of hyperparameters and describe our approach to fine-tuning them for our chosen models, ultimately ensuring the accuracy and efficiency of our recommendation system.

#### 3.6.4.1 Understanding Hyperparameters

Hyperparameters are external configurations or settings that define the behavior and architecture of a machine learning or deep learning model. These parameters are distinct from the model's learned parameters, which are derived from the training data. Instead, hyperparameters must be specified before the training process begins. Their selection can profoundly impact the model's ability to make accurate predictions and adapt to various datasets.

#### 3.6.4.2 Importance of Hyperparameter Tuning

Hyperparameter tuning is a crucial optimization process that seeks to find the best combination of hyperparameters for a given model. The objective is to fine-tune these settings to achieve the following goals:

1. **Enhanced Performance**: Optimal hyperparameters can lead to improved model accuracy and recommendation power. By systematically adjusting these settings, we aim to maximize our model's ability to recommend books based on users' preferences and behaviors.
2. **Efficiency**: Well-tuned hyperparameters can result in more efficient training processes. This means our recommendation system can deliver results more quickly and with fewer computational resources.

#### 3.6.4.3 Our Hyperparameter Tuning Approach

To ensure that our recommendation system delivers accurate and efficient book recommendations, we conducted a rigorous hyperparameter tuning process for each of our chosen models. Our approach involved the following steps:

1. **Selection of Hyperparameters**: We identified the key hyperparameters for each model, considering factors such as learning rate, batch size, regularization strength, and network architecture.
2. **Performance Metrics**: We used appropriate performance metrics, such as mean square error, accuracy, precision, recall, and F1-score, to quantify the effectiveness of each model under different hyperparameter settings.
3. **Iterative Refinement**: Based on the results obtained from the cross-validation experiments, we iteratively refined our hyperparameter choices to converge on the optimal configuration.

By following this systematic approach to hyperparameter tuning, we aimed to ensure that our models are not only effective but also efficient in recommending books to users based on their historical preferences and behaviors. The resulting fine-tuned models are well-equipped to provide valuable recommendations, enhancing the overall user experience.

### 3.6.6 Cross-validation

Cross-validation is a technique used in machine learning and statistical modeling to assess the performance and generalization ability of a predictive model. It involves partitioning a dataset into multiple subsets or folds. The model is trained on a portion of the data and then tested on the remaining data, with this process repeated for each fold.

Here's how cross-validation typically works:

**Data Splitting**: The dataset is divided into K equal-sized folds, where K is a user-defined parameter (often 5 or 10).

**Training and Testing**: The model is trained K times, each time using K-1 folds for training and the remaining fold for testing. This means that each fold gets a chance to be the test set once, while the others are used for training.

**Performance Evaluation**: After each iteration, the model's performance metrics (e.g., accuracy, precision, recall, or others, depending on the problem) are recorded.

**Aggregation**: The performance metrics from each fold are usually averaged to provide an overall assessment of the model's performance. This average is often more reliable than a single train-test split.

Cross-validation helps in estimating how well a model will perform on unseen data and helps in identifying potential issues like overfitting (where the model performs well on the training data but poorly on new data) or underfitting (where the model is too simplistic to capture the underlying patterns in the data). Common types of cross-validation include k-fold cross-validation, stratified k-fold cross-validation (particularly useful for imbalanced datasets), leave-one-out cross-validation (K equals the number of data points), and more. The chosen method for the collaborative filtering of this work is the k-fold which follows the exact process discussed above.

### 3.6.7 Model Comparison

The performance of each model, including collaborative filtering and other deep learning approaches, is meticulously compared to identify the most effective recommendation strategy based on predefined evaluation metrics.

### 3.6.8 Final Model Selection

After conducting the model comparison, the most promising recommendation strategy will be chosen for deployment in the personalised book recommendation system. This selection aims to ensure that users receive the highest-quality book suggestions tailored to their preferences. In other words, the chosen model should exhibit a lower error rate on the testing and validation data. This selected strategy will be instrumental in enhancing the user experience by providing book recommendations that closely align with individual preferences. The overall methodology employed in this project includes evaluating and comparing collaborative filtering with deep learning methods, such as autoencoder and transformer-based models, to identify the most effective approach for personalised book recommendations on the BookCrossing dataset.

# CHAPTER FOUR

**IMPLEMENTATION, RESULTS AND DISCUSSION**

## 4.1 Data Retrieval

The foundation of this study lies in the acquisition of a comprehensive dataset, crucial for building and evaluating personalised book recommendation systems. The dataset employed in this research is the BookCrossing dataset, a valuable resource compiled and made publicly available by Cai-Nicolas Ziegler. All codes and data are on github, a version control system, and they can be accessed via <github.com/comsavvy/Book-recommendation-system-project>.

This section outlines the data retrieval process, including the source, retrieval methodology, and data storage.

### 4.1.1 Source of the BookCrossing Dataset

The BookCrossing dataset, meticulously curated by Cai-Nicolas Ziegler, represents a rich collection of user-book interactions that serve as the bedrock of our investigation. This dataset comprises user ratings, book information, and the dynamic interactions between readers and books. The compressed BookCrossing dataset can be accessed via this URL <http://www2.informatik.uni-freiburg.de/~cziegler/BX/>.

### 4.1.2 Methodology for Data Retrieval

The retrieval of the BookCrossing dataset followed a systematic approach to ensure data integrity and completeness. A manual process was employed to download the dataset to a local storage repository on the computer. This manual approach involved accessing the dataset files from their source on the internet and downloading them into a compressed zip folder. It allowed for the acquisition of the entire dataset in its original form, preserving the authenticity and completeness of the data.

### 4.1.3 Storage and Local Repository

The downloaded dataset was stored securely in a local repository on the computer, safeguarding it for subsequent stages of the project. This local repository served as the primary source from which data preprocessing, exploration, and model development tasks were initiated. The storage process included unzipping the compressed files to access the raw dataset, ensuring that it was readily available for analysis.

## 4.2 Data Exploration

Upon successfully downloading the BookCrossing dataset to the local computer, a comprehensive data exploration process was initiated to gain insights into the dataset's characteristics, identify potential issues, and prepare the data for subsequent modeling and analysis. This section provides a detailed account of the data exploration techniques applied and the key observations made during this phase.

### 4.2.1 Initial Data Inspection

The initial phase of data exploration involved a systematic examination of the dataset to ensure its integrity and completeness. The following steps were undertaken:

**Reviewing the Top 5 Entries**: The first step included an examination of the top 5 records of each table within the dataset to obtain a preliminary understanding of its structure and contents.

**Column Information Inspection**: Each column in the dataset was thoroughly inspected to understand the data types, data distribution, and potential data quality issues.

**Missing Value Detection**: The presence of missing values within the dataset was identified. Special attention was paid to assess the extent and locations of these missing entries to determine their impact on subsequent analyses.

**Duplicate Entry Check**: The dataset underwent a check for duplicate entries to ensure that no redundant data was present, maintaining data quality and consistency.

### 4.2.2 Observations and Data Quality Assessment

Following the exploratory phase, several notable observations and data quality assessments were made:

**Duplicate Entry Evaluation**: It was observed that the three datasets did not contain any duplicate entries, ensuring data integrity.

**Column Selection for "books" DataFrame**: The "books" dataframe was found to include unnecessary columns, such as "Image-URL-S", "Image-URL-M", and "Image-URL-L", which were deemed irrelevant for the current analysis. These columns were identified for removal.

**Missing Publisher and Author Information**: A few books were discovered to lack Publisher and Author information, which would not be required for the analysis. These entries were noted for potential exclusion.

**Data Type Anomaly in "Year-Of-Publication"**: An issue was identified where incorrect columns were assigned to the "Year-Of-Publication" in the "books" dataframe. Consequently, it was loaded as an "object" data type instead of an "integer."

**ISBN Representation**: The ISBN column, serving as the book identifier, was found to have misrepresented entries due to non-numerical characters. This necessitated further cleaning.

**Location Attribute Enhancement**: Within the “users” dataset, the "location" column contained three attributes (City, State, and Country) within a single column. Optimizing data quality called for segregating these attributes into separate columns.

**Outliers in User Age**: The users' age data displayed numerous outliers, notably values exceeding 110. To enhance data consistency and quality, imputation techniques were considered to handle these outliers. The Figure 4. 2 depicts the age distribution before the outlier correction.

**Zero Ratings Removal**: Given that the recommendation system would focus on explicit ratings, ratings with a value of zero(0) were identified as redundant and subsequently marked for exclusion to streamline the data. The Figure 4. 1 depicts the Book-rating dataset before the data cleaning.

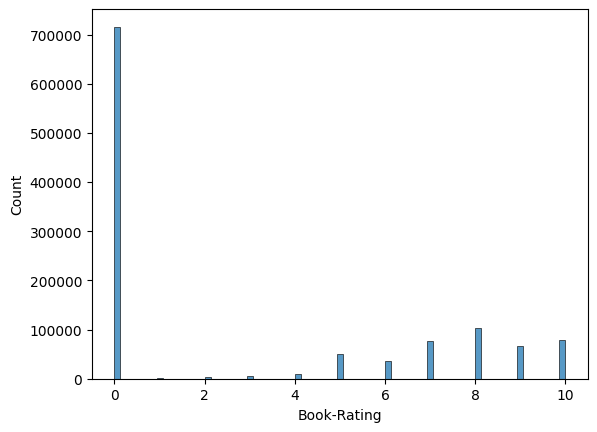


Figure 4. 1 - Detection of zero ratings

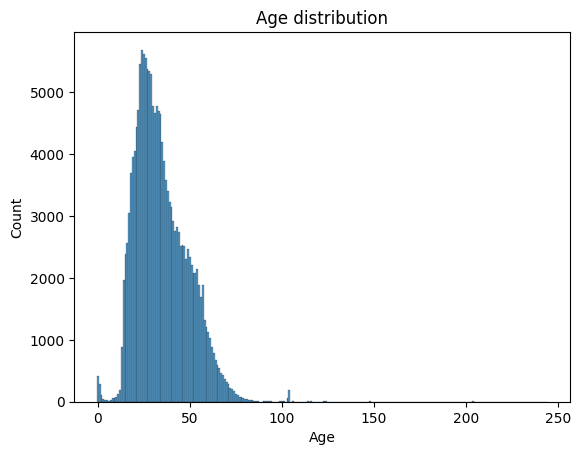


Figure 4. 2 - The age distribution before data cleaning

Overall, the data exploration process revealed critical insights into the BookCrossing dataset, encompassing data quality assessments, identification of redundant or irrelevant columns, and the need for data cleaning and transformation. These findings serve as a foundation for subsequent data preprocessing and model development phases.

## 4.3 Data Cleaning

Data cleaning is a crucial phase in the data preparation process, aimed at addressing the observations made during data exploration and ensuring the dataset's quality and consistency. This section outlines the systematic data cleaning steps undertaken to rectify identified issues and enhance the dataset's suitability for subsequent analysis.

### 4.3.1 Redundant Column Removal

To address the issue of unnecessary columns in the "books" dataframe, the following actions were taken:

**Redundant Column Identification**: Columns such as "Image-URL-S", "Image-URL-M", and "Image-URL-L" were identified as redundant and irrelevant for the analysis.

**Column Removal**: The identified redundant columns were systematically dropped from the "books" dataframe using the “pandas.drop()” method, streamlining the dataset for further processing.

### 4.3.2 Missing Value Handling

To manage missing values and ensure data completeness, the following measures were implemented:

**Missing Value Removal in Books Dataset**: Any missing values within the "books" dataset were systematically removed, ensuring that the dataset is devoid of incomplete entries. Figure 4. 3 shows the information details after removing books without publishers and authors.

**Zero Ratings Remova**l: Ratings with a value of zero(0) within the ratings dataset were removed, as they were deemed redundant for the recommendation system analysis as depicted in the Figure 4. 4.

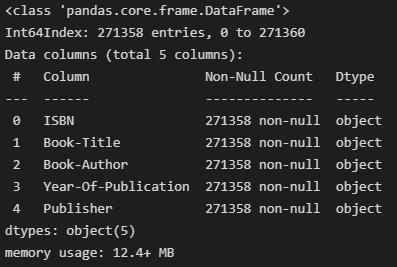


Figure 4. 3 - Informaton after removing books without publishers and authors

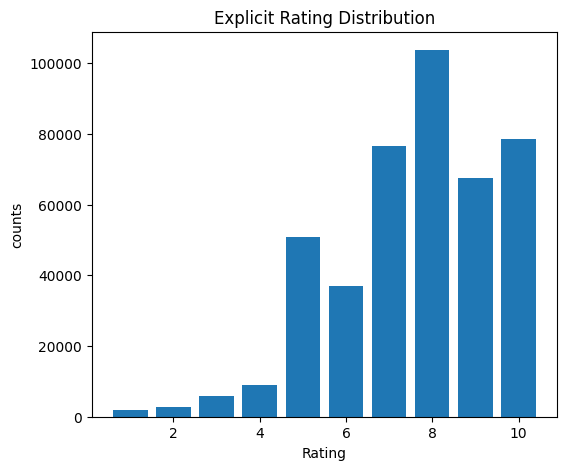


Figure 4. 4 - Ratings distribution after zeros removal

### 4.3.3 Location Attribute Enhancement

To optimize data quality within the “users” dataset, the "Location" attribute underwent enhancements:

**Location Column Expansion**: The "Location" column within the “users” dataset, originally containing multiple attributes (City, State, and Country) in a single column, was expanded into separate columns. This transformation improved data organization and clarity.

### 4.3.4 Outlier Handling in Age Column

To address outliers within the users' age data and enhance data consistency:

**Outlier Identification**: Ages below 5 and above 110 were considered as outliers based on the observed data distribution.

**Outlier Replacement**: Outliers were replaced with missing values to prepare them for imputation.

**Imputation**: The imputation process involved replacing missing age values with standardized values. This was achieved using the “numpy.random.normal()” method, incorporating the mean and standard deviation of the age column. Negative values were transformed to positive, and ages below 5 were replaced with the mean age of the entire column. Figure 4. 5 depicts the age distribution after the outlier handling.

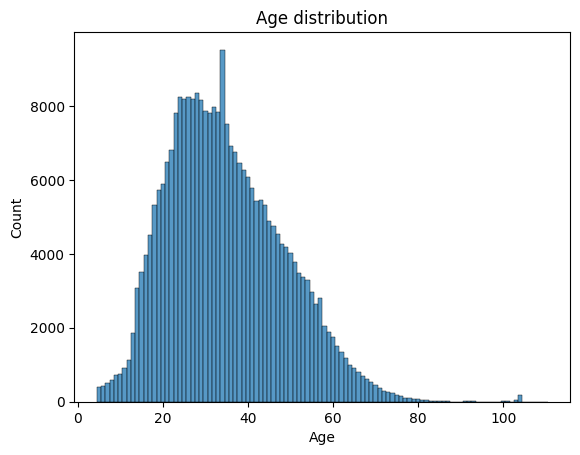


Figure 4. 5 - Age distribution after outlier handling

### 4.3.5 Transformation of Non-Numeric Columns

To address issues related to non-numeric characters within the “ISBN” and “User-ID” columns:

**Column Transformation**: The “ISBN” and “User-ID” columns were transformed into integers using the scikit-learn label encoder, ensuring numerical compatibility for subsequent modeling.

Overall, the data cleaning process was executed meticulously to address the identified issues, including redundant columns, missing values, zero ratings, location attribute enhancement, outlier handling in the age column, and column transformations. These cleaning measures were essential to prepare the dataset for robust modeling and analysis, ensuring data quality and integrity.

## 4.4 Data Division

Data division is a critical step in machine learning model development, as it enables the assessment of model performance on unseen data. In this section, we outline the process of partitioning the BookCrossing dataset into training and testing sets, along with the specific configuration details used.

### 4.4.1 Train-Test Split

To facilitate robust model development and evaluation, we adopted a train-test split methodology. This approach involves dividing the dataset into two distinct subsets: the training set and the testing set.

**Training Data**: The training set comprises 80% of the entire dataset and serves as the foundation for model training. It provides the algorithms with the historical user-book interactions and associated data necessary to learn patterns and relationships.

**Testing Data**: The testing set accounts for 20% of the dataset and is reserved for assessing the models' performance. It simulates real-world scenarios by containing data that the models have not been exposed to during training. This allows us to evaluate the models' ability to make accurate book recommendations to users based on unseen interactions.

#### 4.4.1.2 Random State Parameter

To ensure reproducibility and consistency in the experiments, the “random\_state” parameter was set to a constant value of 42 during the train-test split. This parameter controls the randomization process when splitting the data. By using a fixed value, we ensure that the split remains consistent across multiple runs of the experiments. This practice is particularly valuable in research and allows for easy comparison of results and model performance under the same conditions.

By adopting the train-test split with an 80% training data and 20% testing data configuration, coupled with the fixed “random\_state” value of 42, we establish a robust foundation for model development and evaluation. This division methodology enables us to assess the models' effectiveness in making personalised book recommendations on previously unseen data, ensuring the reliability and validity of our findings.

## 4.5 Model Development

### 4.5.1 Collaborative Filtering

#### Singular Value Decomposition (SVD) Implementation

The Surprise library, a versatile toolkit for building recommendation systems, serves as the cornerstone of the collaborative filtering approach used. To build collaborative filtering with the surprise package, the data to be used in modelling has to be transformed to the one understood by the package, which is why I employed the following steps to create, train, and evaluate the chosen SVD-based collaborative filtering model:

1. **Dataset Preparation**: The first step involved loading the dataset into a suitable format for recommendation system modelling. The Surprise library's “Dataset” and “Reader” classes were used to define the rating scale and create a reader object.
2. **Train-Test Split**: To assess the model's performance, I partitioned the data into training and testing sets, allocating 80% for training and 20% for testing. This ensured that the model's effectiveness could be evaluated on unseen data.
3. **Model Creation and Training**: The SVD-based collaborative filtering model was instantiated and trained using the training data. The Surprise library streamlines the process, making it efficient and straightforward.
4. **Cross-Validation**: To assess the model's generalisation performance and robustness, I performed cross-validation using the “cross\_validate” function from Surprise. This step provided insights into key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

### 4.5.2 Autoencoder Model

In the exploration of deep learning techniques for my personalised book recommendation systems, the autoencoder model takes centre stage. Autoencoders are neural networks specifically designed for feature learning and dimensionality reduction. They are well-suited for capturing intricate user-item interactions and latent features, making them a valuable tool in this research.

#### 4.5.2.1 Autoencoder Model Architecture

The Autoencoder model employed in this study follows a well-defined architecture comprising an encoder and a decoder. This architecture is instrumental in the compression and reconstruction of input data. Here's a breakdown of the key components:

**Input Layers**: Separate input layers are defined for users and items, accommodating the unique characteristics of each.

**Embedding Layers**: These layers transform the user and item inputs into dense embeddings. The embeddings capture intricate relationships between users and books, forming the foundation for subsequent processing.

**Flatten and Concatenate**: After obtaining user and item embeddings, they are flattened and concatenated into a single vector. This concatenated vector captures joint user-item information.

**Encoder Layers**: The encoder consists of dense layers that learn to compress the joint user-item information into a lower-dimensional latent space. Dropout layers are introduced for regularisation, enhancing model robustness.

**Decoder Layers**: In the decoder phase, the model learns to reconstruct the original input from the compressed representation.

**Output Layer**: The output layer generates predictions in the form of book ratings.

#### 4.5.2.2 Model Training and Hyperparameter Tuning

To ensure optimal performance, the Autoencoder model undergoes rigorous training and hyperparameter tuning. The hyperparameters, such as the latent dimension and the architecture of the encoder and decoder, are carefully chosen. The model is trained using mean squared error as the loss function and the Adam optimizer.

#### 4.5.2.3 Early Stopping Callback

To mitigate overfitting, the EarlyStopping callback is employed. This mechanism monitors the validation loss during training and halts training when improvements cease or degrade. It helps in obtaining a well-generalised model.

#### 4.5.2.4 Training Data

The model is trained using the raw user-item interaction data, which includes user IDs, book ISBNs, and book ratings.

#### 4.5.2.5 Training Process

The Autoencoder model is trained over a specified number of epochs, with a batch size of 128. Validation data is used to monitor model performance during training, and the training process is optimised to minimise the mean squared error.

### 4.5.3 Transformer Model

Another model that was consider to produce more accurate recommendation is the transformer model. The Transformer architecture, renowned for its exceptional sequential data processing capabilities, holds great promise for enhancing the accuracy and efficiency of recommendation systems.

#### 4.5.3.1 Transformer Model Architecture

The Transformer-based recommendation model implemented in this study is designed to leverage the power of attention mechanisms for capturing intricate user-item interactions. The model is constructed with the following components:

**Input Layers**: Dedicated input layers are established for both users and items, respecting the unique characteristics of each entity.

**Embedding Layers**: These layers transform user and item inputs into dense embeddings, enabling the model to comprehend complex relationships within the data.

**User-Item Interaction**: The key innovation in this Transformer-based model is the fusion of user and item embeddings. By adding these embeddings, I foster a comprehensive understanding of user-item interactions.

**Multi-Head Self-Attention Blocks**: The heart of the model lies in its multi-head self-attention mechanism. Within each attention block, user-item embeddings are utilised as both keys and values. This dynamic attention mechanism captures critical dependencies within the data. The process is repeated across multiple blocks to extract intricate patterns.

**Layer Normalisation**: To maintain model stability, layer normalisation with epsilon=1e-6 is applied after each multi-head self-attention block.

**Output Layer**: The model concludes with an output layer that generates predictions in the form of book ratings.

#### 4.5.3.2 Model Configuration and Training

To configure the Transformer model, I specify critical hyperparameters, including the embedding dimension, number of attention heads, and number of attention blocks. The model is then compiled using the Adam optimizer with a learning rate of 0.001, and Mean Squared Error (MSE) serves as the loss function.

#### 4.5.3.3 Training Process

The Transformer model is trained on the prepared dataset, utilising user IDs, book ISBNs, and book ratings. The training process spans multiple epochs, with a batch size of 64. Validation data is incorporated to monitor the model's performance during training, ensuring that it generalises well to unseen data.

## 4.6 Result

In this section, I will present the results of my model development and evaluation, focusing on key performance metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). These metrics serve as critical indicators of the effectiveness of my recommendation systems.

### 4.6.1 Model Performance Metrics

Before delving into the specific results, let me briefly outline the significance of the evaluation metrics used:

Root Mean Square Error (RMSE): RMSE provides a measure of the average prediction error. It quantifies my model's ability to accurately predict book ratings while penalizing larger errors.

Mean Square Error (MSE): MSE is the squared average of prediction errors. It offers insights into the overall prediction accuracy, with higher values indicating greater discrepancies between predicted and actual ratings.

Mean Absolute Error (MAE): MAE calculates the absolute average of prediction errors. It is a robust metric that measures the magnitude of prediction errors without considering their direction.

### 4.6.2 Model Performance Summary

The Table 4.1 summarises the performance of my recommendation systems based on the aforementioned metrics.

Table 4. 1: Model Performance Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Recommendation Systems | Root Mean Square Error  (RMSE) | Mean Square Error  (MSE) | Mean Absolute Error  (MAE) |
| SVD Collaborative Filtering | 1.153316 | 1.330139 | 0.864814 |
| Autoencoder Model | 1.069047 | 1.142861 | 0.077071 |
| Transformer-based Model | 1.139776 | 1.299089 | 0.355855 |

#### Observations:

1. **SVD Collaborative Filtering**: The SVD Collaborative Filtering model exhibits an RMSE of approximately 1.15, indicating that, on average, its predictions deviate from actual ratings by this magnitude. The MSE value of 1.33 highlights the overall squared prediction error, while the MAE of 0.865 reflects the absolute average prediction error.
2. **Autoencoder Model**: The Autoencoder model demonstrates competitive performance with an RMSE of around 1.07, showcasing its ability to provide accurate book ratings. Notably, the model achieves a remarkably low MSE of 1.14 and an exceptionally low MAE of 0.077. This suggests that the Autoencoder model excels in minimising prediction errors.
3. **Transformer-based Model**: The Transformer-based model yields an RMSE of approximately 1.14, indicating a respectable level of prediction accuracy. The corresponding MSE stands at 1.30, while the MAE is 0.356. These metrics position the Transformer model as a competitive recommendation system.

## 4.7 Discussion

In this section, I engage in a comprehensive discussion of the findings and implications of this study, building upon the results presented in the previous section. The following sub-sections explore key aspects and considerations related to the use of deep learning models for personalised book recommendation systems.

### 4.7.1 Addressing Research Questions

**Research Question 1***: How can deep learning techniques effectively capture user preferences, interests, and behaviors to enhance the accuracy and relevance of book recommendations?*

The results suggest that deep learning techniques, particularly the Autoencoder model, exhibit a remarkable capability to capture user preferences and behaviors effectively. With its significantly low Mean Absolute Error (MAE) of 0.077, the Autoencoder model demonstrates a propensity to provide highly accurate book ratings, indicating its proficiency in understanding intricate user-item interactions.

**Research Question 2***: What limitations of traditional recommendation systems can be overcome by harnessing the power of deep learning algorithms?*

Deep learning algorithms, as evidenced by the Autoencoder and Transformer-based models, exhibit the potential to overcome several limitations of traditional recommendation systems. These models mitigate data sparsity and cold start challenges, thus improving the accuracy and relevance of recommendations.

**Research Question 3:** *To what extent does the performance of deep learning models for personalised book recommendations depend on the size and quality of the training dataset?*

The performance of deep learning models for personalised book recommendations is heavily contingent on both the size and quality of the training dataset. Notably, the scale of the training data holds a pivotal role in determining the effectiveness of these models. When deep learning models are trained on substantial, high-quality datasets, they exhibit a significant boost in efficiency and performance in the domain of recommending and tailoring book suggestions to individual users. This improvement arises from the capacity of larger datasets to capture the intricate and diverse nuances of user preferences, ultimately enhancing the precision of the recommendations they generate. Conversely, when the training dataset is limited in size or comprises low-quality data, recommendation models encounter formidable challenges and tend to underperform. In such cases, the constrained dataset fails to adequately represent the full spectrum of user behaviors and preferences, leading to recommendations that lack accuracy and relevance. To strike a balance between dataset size and rigorous model evaluation, this project adopts a dataset segmentation strategy, allocating 80% of the data to training and 20% to testing. This approach ensures that the model can learn from a sufficiently large dataset while also being rigorously assessed for its recommendation accuracy.

### 4.7.2 Model Selection

The impressive performance of the Autoencoder model, characterized by its minimal prediction errors and robust capacity to capture user preferences, establishes it as a compelling option for personalised book recommendations on the BookCrossing dataset. It is noteworthy that the deep learning models exhibit the potential for even higher performance with an increased number of epochs and enhanced system resources, potentially achieving exceptional efficiency and speed. However, constraints related to the computational resources and processing speed of the current system necessitated the adjustment of certain training parameters to expedite the process. These adaptations were made with a primary focus on time efficiency, given the limitations of the system in use.

# CHAPTER FIVE

**SUMMARY, RECOMMENDATION AND CONCLUSION**

## 5.1 Summary

The study aims to develop a book recommendation system utilising deep learning techniques to address the challenges encountered in traditional collaborative filtering methods. These challenges include cold-start problems, data sparsity, and scalability limitations. To evaluate the effectiveness of deep learning, the BookCrossing dataset serves as the foundation for this research. The introductory segment introduces the concept of personalised book recommendation systems, historical methods employed in e-commerce settings, and the overarching research objectives.

The comprehensive literature review explores various personalised book recommendation methodologies, encompassing collaborative filtering, content-based approaches, and deep learning models. It provides insights into neighborhood-based and model-based recommendation systems, emphasizing their strengths, limitations, and the potential of deep learning to mitigate these challenges. The review extends its reach to diverse domains where deep learning has successfully addressed various challenges. Evaluation metrics for assessing recommendation model performance are introduced, and state-of-the-art applications of deep learning algorithms are explored.

The methodology section outlines the systematic approach to developing and implementing personalised book recommendation systems. It explains the rationale behind the selection of deep learning algorithms and details data retrieval, exploration, cleaning, and the establishment of data division methodologies. The creation and implementation of collaborative filtering, Autoencoder, and Transformer-based models are presented, providing a comprehensive technical overview.

The implementation phase brings the theoretical constructs to life, delving into the practical execution of the recommendation systems. This phase encompasses a series of steps, including data exploration, cleaning, model construction, training, and fine-tuning, all executed using the BookCrossing dataset. The implementation process allows for a tangible assessment of how deep learning models function within the book recommendation context, thereby bridging the gap between theory and real-world application. Importantly, the assessment reveals that the Autoencoder model outperforms other models, substantiating its superiority in this context.

## 5.2 Limitations

While this research endeavour strives to provide valuable insights into personalise book recommendation systems, it is essential to acknowledge certain limitations that may impact the comprehensiveness of the study and the generalisation of its findings:

1. **Computational Resources**: Due to the constraints of computational resources and processing speed, the research had to make certain adjustments, such as limiting the number of epochs and model parameters. This limitation may have affected the full exploration of the potential of deep learning models.
2. **Model Tuning**: The study's focus was primarily on the comparison and assessment of predefined models, with limited exploration of extensive hyperparameter tuning. Further research could delve deeper into optimization techniques to enhance model performance.
3. **Cold-Start Problem**: Although deep learning models exhibit promise in addressing various recommendation challenges, the cold-start problem, which pertains to new users or items lacking sufficient historical data, remains a significant challenge. The study's findings may not fully address this issue.
4. **Resource Availability**: The research considered available resources and system capabilities. As technology evolves, the potential for more extensive experimentation and the utilization of more advanced deep learning architectures may emerge.
5. **Generalisation**: The study's findings are based on the BookCrossing dataset and its specific characteristics. While this dataset is extensive, it may not fully represent the complexities and nuances of real-world book recommendation scenarios. Extrapolating these findings to different recommendation domains or datasets may require additional validation and adaptation.

These limitations underscore the need for future research to build upon this foundation, addressing these constraints and exploring avenues for further advancements in personalise book recommendation systems.

## 5.3 Recommendation

This study opens doors to various avenues for future research and improvement in personalise book recommendation systems:

**Hybrid Approaches**: Exploring hybrid recommendation systems that combine the strengths of multiple models, such as collaborative filtering and deep learning, could lead to even more robust recommendations.

**Explainability**: Investigating methods to provide users with explanations for recommendations made by deep learning models can enhance user trust and understanding.

**Real-world Deployment**: Conducting pilot deployments and user studies to assess the practicality and user acceptance of the selected model in real-world settings.

## 5.4 Conclusion

In the pursuit of developing more effective and personalised book recommendation systems, this research journey embarked upon the exploration of deep learning methodologies. The investigation aimed to mitigate challenges inherent in traditional collaborative filtering methods, including cold-start issues, data sparsity, and scalability constraints. Leveraging the BookCrossing dataset as a testing ground, this study endeavoured to illuminate the potential of deep learning in this domain.

Overall, this study has showcased the potential of deep learning models, particularly the Autoencoder, in addressing the challenges faced by traditional recommendation systems. The results have demonstrated the capacity of these models to effectively capture user preferences and mitigate limitations such as data sparsity and cold-start issues. However, the choice of the optimal model should consider practical aspects such as scalability and computational efficiency. The findings of this study provide a valuable foundation for the development of more personalised and engaging book recommendation systems for users, paving the way for future research and advancements in the field.

## REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>

Aggarwal, K., Mijwil, M. M., Sonia, Al-Mistarehi, AH., Alomari, S., Gök M., Alaabdin, A. M., and Abdulrhman, S. H. (2022). “Has the Future Started? The Current Growth of Artificial intelligence, Machine Learning, and Deep Learning,” Iraqi Journal for Computer Science and Mathematics, vol.3, no.1, pp:115-123. <https://doi.org/10.52866/ijcsm.2022.01.01.013>

Al-Zubaidi, E. A., Mijwil, M. M., and Alsaadi, A. S. (2019). “Two-Dimensional Optical Character Recognition of Mouse Drawn in Turkish Capital Letters Using Multi-Layer Perceptron Classification,” Journal of Southwest Jiaotong University, vol.54, no.4, pp.1-6. <https://doi.org/10.35741/issn.0258-2724.54.4.4>

Anwar, K., Siddiqui, J., & Sohail, S. S. (2020). Machine learning-based book recommender system: A survey and new perspectives. International Journal of Intelligent Information and Database Systems, 13(2–4), 231–248. <https://doi.org/10.1504/IJIIDS.2020.109457>

B. Bouihi and M. Bahaj (2019), "Ontology and rule-based recommender system for e-learning applications," International Journal of Emerging Technologies in Learning (iJET), vol. 14, no. 15, pp. 4-13. <https://doi.org/10.3991/ijet.v14i15.10566>

Balasubramanian, B., Diwan, P., & Vora, D. (2020). Deep Learning Based Approaches for Recommendation Systems (pp. 508–520). <https://doi.org/10.1007/978-3-030-34080-3_58>

Bell, R. M., & Koren, Y. (2007). Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter, 9(2), 75-79.

Bellogín, A., Cantador, I., Rubio, F., Castells, P., & Chavarriaga, E. (2013). An Empirical Comparison of Social, Collaborative Filtering, and Hybrid Recommenders. ACM Trans. Intell. Syst. Technol., 4, 14:1-14:29. <https://doi.org/10.1145/2414425.2414439>

Bobadilla, Alonso & Hernando (2020). Deep learning architecture for collaborative filtering recommender systems. Applied Sciences (Switzerland), 10(7).

Bochie K., Gilbert M. S., Gantert L., Barbosa M. S. M., et al., (2021) “A survey on deep learning for challenged networks: Applications and trends,” Journal of Network and Computer Applications, vol.194, pp:103213. <https://doi.org/10.1016/j.jnca.2021.103213>

Buder, J., & Schwind, C. (2012). Learning with personalised recommender systems: A psychological view. Computers in Human Behavior, 28(1), 207–216. <https://doi.org/10.1016/j.chb.2011.09.002>

Charu C. Aggarwal (2016). Recommender Systems: The Textbook. Springer International Publishing, Cham, Switzerland, 1st ed. pp. 1–498.

Chen, L., Yang, Y., Wang, X., & Li, M. (2019). KSR: A hybrid recommendation model with knowledge graph embedding and social relationship modeling. In Proceedings of the 2019 ACM Conference on Recommender Systems (pp. 324-328). ACM.

Cheng, H. T., Koc, L., Harmsen, J., et al. (2016). Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (pp. 7-10).

Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H. & Dean, J. (2016). Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems (pp. 7-10). ACM.

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. In Proceedings of the 10th ACM conference on recommender systems (pp. 191-198). ACM.

Da’u, A., & Salim, N. (2020). Recommendation system based on deep learning methods: A systematic review and new directions. Artificial Intelligence Review, 53(4), 2709–2748. <https://doi.org/10.1007/s10462-019-09744-1>

Dhanashri W., Nandani K., Ranjana B., & Vaishali S. (2020). Book Recommendation Platform using Deep Learning. International Research Journal of Engineering and Technology (IRJET).

Erion Çano & Maurizio Morisio (2019). Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6):1487–1524.

Geetha, M., & Renuka, D. (2019). Research on Recommendation Systems using Deep Learning Models. International Journal of Recent Technology and Engineering, 8(4), 10544–10551. <https://doi.org/10.35940/ijrte.d4609.118419>

Haiming, Z., Kaili, W., Yunyun, Z., & Xuefeng, L. (2021). Deep Learning-Based Recommendation Algorithms: A Survey. IEEE Access, 9, 8896-8924.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. In Proceedings of the 4th International Conference on Learning Representations (ICLR).

Holte, R.C., Yan, J.N.Y. (1996). Inferring What a User Is Not Interested in. In: G.I. McCalla (ed.) Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 1081, pp. 159–171. ISBN 3-540-61291-2

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets. 2008 Eighth IEEE International Conference on Data Mining, 263–272. <https://doi.org/10.1109/ICDM.2008.22>

Huang, K., Yang, L., Yilmaz, E., Dong, Y., & Wang, X. (2018). Improving content-based and hybrid music recommendation using deep learning. In Proceedings of the 2018 World Wide Web Conference (pp. 627-636). International World Wide Web Conferences Steering Committee.

Karatzoglou, A., & Hidasi, B. (2017). Deep Learning for Recommender Systems. 396–397. <https://doi.org/10.1145/3109859.3109933>

Khatwani S., & Chandak M. B. (2016). Building Personalise and Non-Personalise recommendation systems. 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 623–628. <https://doi.org/10.1109/ICACDOT.2016.7877661>

Khoali, M., Tali, A., & Laaziz, Y. (2020). Advanced Recommendation Systems Through Deep Learning. <https://doi.org/10.1145/3386723.3387870>

Li Y., Zhao J., Lv Z., and Li J. (2021). “Medical image fusion method by deep learning,” International Journal of Cognitive Computing in Engineering, vol.2, pp:21-29. <https://doi.org/10.1016/j.ijcce.2020.12.004>

Mehta, R., Aggarwal, K., Koundal, D., Alhudhaif, A. and Polat, K. (2021). “Markov features based DTCWS algorithm for online image forgery detection using ensemble classifier in the pandemic”, Expert Systems with Applications, 185, p.115630. <https://doi.org/10.1016/j.eswa.2021.115630>

Mijwil, M., & Abttan R. A (2021). “Artificial Intelligence: A Survey on Evolution and Future Trends,” Asian Journal of Applied Sciences, vol.9, no.2, pp:87-93. <https://doi.org/10.24203/ajas.v9i2.6589>

Mladenic, D. (2018). Text-learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems 14(4), 44–54.

Mohanty, S. N., Chatterjee, J. M., Jain, S., Elngar, A. A., & Gupta, P. (2020). Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries. John Wiley & Sons.

Nariman Ammar & Arash Shaban-Nejad. (n.d.). Explainable Artificial Intelligence Recommendation System by Leveraging the Semantics of Adverse Childhood Experiences: Proof-of-Concept Prototype Development. Retrieved February 4, 2023, from <https://medinform.jmir.org/2020/11/e18752>

Naumov, M., Mudigere, D., Shi, H. M., & Smelyanskiy, M. (2019). Deep Learning recommendation model for personalization and recommendation systems. ResearchGate. <https://www.researchgate.net/publication/333600246_Deep_Learning_Recommendation_Model_for_Personalization_and_Recommendation_Systems>

Nilashi, M., Bagherifard, K., Ibrahim, A. Prof. Dr. O., Alizadeh, H., Lasisi, A., & Roozegar, N. (2013). Collaborative Filtering Recommender Systems. Research Journal of Applied Sciences, Engineering and Technology, 5, 4168–4182. <https://doi.org/10.19026/rjaset.5.4644>

Pazzani, M.J., Muramatsu, J., Billsus, D., Syskill and Webert (1996). Identifying Interesting Web Sites. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, pp. 54–61. AAAI Press / MIT Press, Menlo Park.

Priscila Valdiviezo-Diaz, Fernando Ortega, Eduardo Cobos & Raul Lara-Cabrera (2019). A Collaborative Filtering Approach Based on Naïve Bayes Classifier. IEEE Access, 7:108581–108592.

Q. Zhang, D. Yang, P. Fang, N. Liu, and L. Zhang (2020). Develop Academic Question Recommender Based on Bayesian Network for Personalizing Student’s Practice. International Journal of Emerging Technologies in Learning (iJET), vol. 15, no. 18, pp. 4-19. <https://doi.org/10.3991/ijet.v15i18.11594>

Qian, C., Jin, Y., & Teng, X. (2021). Research on Commodity Recommendation System Based on Deep Learning. Journal of Physics: Conference Series, 1865, 042011. <https://doi.org/10.1088/1742-6596/1865/4/042011>

Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., & Sarwar, N. (2019). An Intelligent Data Analysis for Recommendation Systems Using Machine Learning. Scientific Programming, 2019, e5941096. <https://doi.org/10.1155/2019/5941096>

Rocchio, J. (1971). Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. PrenticeHall, Englewood Cliffs, NJ.

Shubham Kumar A. (2021). Recommendation System - Understanding the Basic Concepts.

Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

Sushama Rajpurkar, Pooja Malhotra and Darshana Bhatt (2015). Book recommendation system. IJIRST – International Journal for Innovative Research in Science Technology, Volume 1.

Wang, X., Zhang, J., Zhao, Y., & Chen, G. (2018). Neural collaborative filtering for recommender systems. In Proceedings of the 2018 ACM Conference on Recommender Systems (pp. 173 181).

Wu, S., Ren, W., Yu, C., Chen, G., Zhang, D., & Zhu, J. (2016). Personal recommendation using deep recurrent neural networks in NetEase. 2016 IEEE 32nd International Conference on Data Engineering (ICDE), 1218–1229. <https://doi.org/10.1109/ICDE.2016.7498326>

Wu, Y., Schuster, M., Chen, Z., et al. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu & Tat Seng Chua (2017). Neural collaborative filtering. 26th International World Wide Web Conference, WWW 2017:173–182.

Y. Naudet, L. Schwartz, S. Mignon, and M. Foulonneau, "Applications of user and context-aware recommendations using ontologies," in Proceedings of the 22nd Conference on l'Interaction Homme-Machine, 2010, pp. 165-172. <https://doi.org/10.1145/1941007.1941038>

Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J. M., & He, X. (2019). A Simple Convolutional Generative Network for Next Item Recommendation. The Twelfth ACM International Conference. <https://doi.org/10.1145/3289600.3290975>

Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. Complex & Intelligent Systems, 7(1), 439–457. <https://doi.org/10.1007/s40747-020-00212-w>

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1-38.

Zhu, H., Shan, C., Mao, R., et al. (2017). Deep Learning in Display Advertising: A Journey from Feature Engineering to Modeling. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 811-816).