Evaluating the Effectiveness of Neural Collaborative Filtering, Multilayer Perceptron and Matrix Factorization Techniques for Recommender Systems

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE
OF

MASTER OF TECHNOLOGY

IN

DATA SCIENCE

Submitted by:

SWAPNIL KARN

2K21/DSC/10

Under the supervision

of

Mr. SANJAY PATIDAR

(Assistant Professor)

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2023

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042

CANDIDATE'S DECLARATION

I, SWAPNIL KARN, Roll No. 2K21/DSC/10 student of M. Tech (DATA SCIENCE), hereby declare that the project Dissertation titled "Evaluating the Effectiveness of Neural Collaborative Filtering, Multilayer Perceptron, and Matrix Factorization Techniques for Recommender Systems" which is submitted by me to the Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: Delhi
Date: 31/05/23

SWAPNIL KARN
2K21/DSC/10

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Evaluating the Effectiveness of Neural Collaborative Filtering, Multilayer Perceptron, and Matrix Factorization Techniques for Recommender Systems" which is submitted by SWAPNIL KARN, 2K21/DSC/10 Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 3/6/23

SANJAY PATIDAR
SUPERVISOR
ASSISTANT PROFESSOR

ACKNOWLEDGMENT

I would like to express my deep appreciation to everyone who has contributed to the success of this project. Without their valuable assistance and support, this endeavour would not have been possible.

First and foremost, I am sincerely grateful to Mr. Sanjay Patidar, my project guide, for providing me with the opportunity to undertake this project under his guidance. His constant support, encouragement, and belief in my abilities have been instrumental in my growth throughout this journey.

I would also like to thank the esteemed panel of faculties for their guidance, supervision, and motivation during the progress evaluations. Their expertise, feedback, and support have greatly influenced the direction and quality of my work.

Furthermore, I am deeply thankful to my family for their unwavering support and understanding. Their belief in me and their constant encouragement have been the driving force behind my perseverance.

To all those who have contributed to this project, I extend my heartfelt thanks. Your assistance, guidance, and support have been invaluable, and I am truly grateful for the opportunity to work alongside such remarkable individuals. It is through our collective efforts that this project has been successfully completed, and I am honoured to have gained valuable experience and knowledge from this collaborative endeavour.

Swapnil Karn
2K21/DSC/10

ABSTRACT

In the age of digital connectivity, where a vast array of content is easily accessible, it has become increasingly difficult to find captivating and personally appealing material. Moreover, as the internet continues to expand at a rapid pace, more and more people are turning to OTT platforms and digital content providers, like e-books, to enhance their leisure activities. In response to this challenge, recommendation systems have emerged as indispensable tools that offer users customized content suggestions based on their individual preferences and interests. In this project, we aim to assess the effectiveness of three distinct methods: Neural Collaborative Filtering, Multilayer Perceptron, and Matrix Factorization. Our objective is to evaluate the performance and efficacy of these approaches in the context of our study. Neural Collaborative Filtering (NCF) is a powerful approach that combines the capabilities of neural networks with collaborative filtering techniques, enabling the generation of highly personalized recommendations. By leveraging the strengths of both neural networks and collaborative filtering, NCF excels at capturing complex patterns and relationships in user-item interactions, resulting in accurate and tailored recommendations for individual users. Matrix Factorization is a technique commonly employed in collaborative filtering-based recommendation systems. Its primary objective is to decompose a user-item interaction matrix into lower-dimensional representations. This approach assumes that the observed interactions between users and items can be effectively explained by a set of latent factors or features. By discovering these underlying factors, matrix factorization enables the system to make accurate predictions and provide recommendations based on users' preferences and historical behaviour. The Multilayer Perceptron (MLP) is an Artificial Neural Network (ANN) architecture with interconnected layers of artificial neurons. It processes information in a one-way flow from input to output, without loops. Neurons apply nonlinear functions and pass outputs to the next layer. MLPs effectively capture complex data relationships for tasks like classification, regression, and pattern recognition.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

LIST OF ABBREVIATIONS

ALS	Alternating Least Squares
ANN	Artificial Neural Network
CBF	Content-based Filtering
CF	Collaborative Filtering
MF	Matrix Factorization
MLP	Multilayer Perceptron
MSE	Mean-Squared Error
NCF	Neural Collaborative Filtering
PMF	Probabilistic Matrix Factorization
ReLU	Rectified Linear Unit
RMSE	Root Mean-Squared Error
RS	Recommender Systems
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition

CHAPTER I

INTRODUCTION

CHAPTER I

INTRODUCTION

Recommender systems[1] and good recommendations are vital in the entertainment industry, specifically for movies and TV shows. They provide personalized experiences, help users discover relevant content, boost engagement, promote diverse options, enable targeted marketing, and enhance user satisfaction and loyalty. These systems analyse user preferences and behaviour to suggest tailored content, making it easier for users to find what they like. By offering enjoyable recommendations, users are more likely to stay engaged, explore new content, and remain loyal to the platform. Overall, recommender systems play a crucial role in enhancing the entertainment experience by connecting users with the content they love.

1.1. Recommender Systems

A recommender system is an information filtering system that provides personalized recommendations to users based on their preferences, past behaviour, and other relevant data. The primary goal of a recommender system is to suggest items or content that a user is likely to be interested in, thus improving their overall experience, and helping them discover new items they might have otherwise missed.

Recommender systems are widely used in various industries, including e-commerce, streaming platforms, social media, and content providers. They utilize various algorithms and techniques to analyse user data and generate recommendations. Here are a few common types of recommender systems:

1. Content-based filtering: This approach recommends items similar to the ones a user has shown interest in previously. It analyses the content or features of items and matches them to a user's profile or preferences[2]–[4].
2. Collaborative filtering: This method recommends items based on the preferences of similar users. It considers the behaviour and preferences of a group of users to find patterns and

make recommendations. Collaborative filtering can be further divided into two types: user-based and item-based[5]–[7].

- User-based collaborative filtering: It identifies users with similar preferences and recommends items that these similar users have liked or purchased[8], [9].
- Item-based collaborative filtering: It identifies items similar to the ones a user has interacted with and recommends those similar items.

3. Hybrid recommender systems: These systems combine multiple approaches, such as content-based filtering and collaborative filtering, to provide more accurate and diverse recommendations. By leveraging the strengths of different techniques, hybrid recommender systems aim to overcome the limitations of individual methods[1], [10].

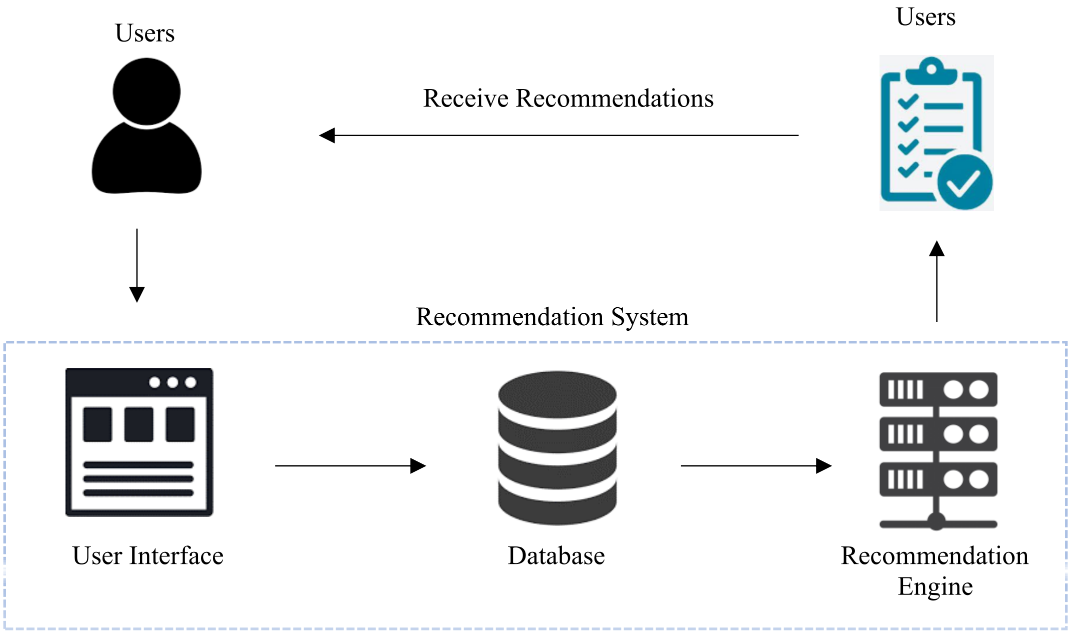


Figure 1.1 A simple Recommendation System

Recommender systems typically improve over time by collecting feedback from users and incorporating it into their recommendation algorithms. They can adapt to changing user preferences and adjust their recommendations accordingly.

Recommender systems have emerged as indispensable tools across diverse domains, owing to their profound significance and impact. These systems revolutionize the user experience by providing highly personalized recommendations that are precisely tailored to individual preferences and interests. By analysing user data and behaviour patterns, recommender systems deliver recommendations that resonate with users on a deep level, resulting in heightened engagement, satisfaction, and a sense of delight. Through their ability to suggest relevant and diverse items, these systems enable users to embark on journeys of serendipitous discovery, unveiling new and exciting content, products, or experiences they might have otherwise missed.

The influence of recommender systems extends beyond user satisfaction and engagement. In the realm of e-commerce and retail, they play a pivotal role in driving sales and generating revenue. By intelligently leveraging user browsing history and purchase behaviour, recommender systems can influence and guide users' purchasing decisions, significantly impacting sales figures. Through techniques such as cross-selling and upselling, these systems help businesses maximize the average order value and ultimately boost their bottom line.

Furthermore, recommender systems possess a profound impact on customer retention and loyalty. By consistently delivering personalized recommendations that align with individual preferences, these systems foster a deep sense of user loyalty and satisfaction. Users who experience a seamless and enjoyable recommendation process are more likely to remain committed to a platform or service, reducing churn rates and increasing customer lifetime value.

The benefits of recommender systems go beyond user-centric advantages. They contribute to efficient information filtering, particularly in platforms with vast content or product catalogues. By intelligently filtering out irrelevant or less interesting items, recommender systems assist users in navigating through the overwhelming volume of choices, saving them valuable time and effort. This streamlined and targeted approach enhances the overall user experience and alleviates the burden of information overload.

Another crucial aspect is the role recommender systems play in empowering businesses with actionable insights. By mining and analysing user data, these systems provide valuable intelligence and deeper understanding of customer preferences, trends, and behaviours. Armed with this knowledge, businesses can make informed decisions, optimize their product offerings, refine marketing strategies, and enhance the overall user experience.

In essence, recommender systems are transformative tools that deliver personalized, engaging, and relevant recommendations. They unlock the power of discovery, foster user loyalty, boost sales and revenue, streamline information filtering, and enable data-driven decision making. As the digital landscape continues to evolve, recommender systems will remain an essential component of delivering exceptional user experiences and driving business success.

1.2 Introduction to Techniques Used

In this evaluation, the effectiveness of three techniques, namely Neural Collaborative Filtering (NCF)[11], Multilayer Perceptron (MLP)[12], and Matrix Factorization[13][14], is assessed. The objective is to compare and analyse the performance of these techniques to determine their effectiveness within the context of the study.

Neural Collaborative Filtering (NCF)[11] is a powerful recommendation technique that combines neural networks with collaborative filtering. It addresses the limitations of traditional methods by using deep learning models to capture intricate user-item interactions. NCF transforms user and item features into dense embeddings, which are then processed through multi-layer perceptrons (MLPs)[12] to capture preferences and characteristics. By leveraging neural networks, NCF overcomes sparsity and scalability challenges in recommender systems, providing accurate recommendations even with limited data. Its scalability enables it to handle large-scale datasets and real-time scenarios. NCF's ability to model non-linear relationships allows it to capture complex user-item patterns, surpassing traditional methods. It has shown improved performance in tasks such as item recommendation and personalized ranking. Neural Collaborative Filtering represents a significant advancement in recommender systems, delivering accurate and personalized recommendations while overcoming challenges of sparsity and scalability.

Matrix factorization[13][14] is a popular technique used in recommendation systems to analyse and predict user-item interactions. It simplifies the recommendation process by decomposing the user-item interaction matrix into two lower-rank matrices: the user matrix and the item matrix. By representing users and items in a lower-dimensional latent space, matrix factorization captures their latent preferences and characteristics. In matrix factorization, the user-item interaction matrix is typically a sparse matrix with users as rows, items as columns, and the observed interactions as entries. The goal is to find low-rank representations of users and items that can reconstruct the original matrix as accurately as possible. This is achieved through optimization algorithms such as gradient descent, which minimize the difference between the reconstructed matrix and the observed interactions.

Matrix factorization[12] has several advantages in recommendation systems. It can handle sparsity and scalability issues commonly encountered in real-world datasets. By reducing the dimensionality of the user-item space, matrix factorization enables efficient computation and faster recommendation generation. It also provides personalized recommendations by identifying similarities between users and items based on their latent representations. Matrix factorization techniques, such as Singular Value Decomposition (SVD)[15] and Alternating Least Squares (ALS)[16], have been widely used and studied in recommendation systems. They have shown successful results in various applications, including movie recommendations, music recommendations, and personalized advertising.

To summarize, matrix factorization is a powerful approach used in recommendation systems. It involves decomposing the user-item interaction matrix into lower-dimensional representations, which brings several advantages. Firstly, it allows for efficient computation by reducing the dimensionality of the problem. Secondly, it effectively handles sparsity in the data, which is common in recommendation systems. Lastly, matrix factorization enables the generation of personalized recommendations by capturing latent preferences of users.

Multi-Layer Perceptron (MLP)[12] is a type of artificial neural network commonly used in recommender systems to capture complex patterns and make accurate predictions. MLPs consist of multiple layers of interconnected nodes, or neurons, with each neuron performing a non-linear transformation on its input.

In the context of recommender systems, MLPs are employed to model the relationships between users, items, and their features. The input layer of the MLP represents the user and item features, such as demographic information or item attributes, while the output layer represents the predicted user preferences or ratings for items. The intermediate layers, known as hidden layers, play a crucial role in learning the non-linear relationships and extracting relevant features from the input data. MLPs are advantageous in recommender systems for several reasons. First, they can handle non-linear relationships between user and item features, which is crucial for accurately capturing the complex preferences and interactions in recommendation scenarios. MLPs are capable of learning intricate patterns that may not be captured by simpler linear models. Second, MLPs are highly flexible and can handle diverse types of input data. They can accommodate different types of features, including categorical, numerical, and text-based features. This flexibility allows MLPs to effectively utilize a wide range of data sources and capture rich representations of users and items. Third, MLPs can be trained using gradient-based optimization algorithms, such as backpropagation, which enable efficient and scalable training on large-scale datasets. This makes MLPs suitable for recommendation systems with vast user and item populations. Additionally, MLPs can be enhanced with regularization techniques, such as dropout or batch normalization, to prevent overfitting and improve generalization performance. They can also incorporate techniques like residual connections or attention mechanisms to further enhance the model's capability to capture important features and user-item interactions.

Overall, MLPs provide a powerful framework for building recommender systems that can capture complex patterns, handle various types of data, and make accurate predictions. By leveraging the non-linear transformations and representation learning capabilities of MLPs, recommender systems can deliver more personalized and effective recommendations to users.

1.3 Motivation

Recommender systems play a crucial role in various domains, including e-commerce, entertainment, and personalized content delivery. These systems are designed to assist users in navigating vast amounts of information and making informed decisions by providing personalized recommendations.

While numerous techniques have been proposed for building recommender systems, three prominent methods have gained considerable attention in recent years: Neural Collaborative

Filtering, Multilayer Perceptron, and Matrix Factorization. Each technique offers unique advantages and has shown promising results in different scenarios. However, there is a need for a comprehensive evaluation and comparison of these techniques to understand their relative strengths and weaknesses in terms of recommendation accuracy, scalability, interpretability, and adaptability to various datasets. This research aims to bridge this gap by conducting an extensive empirical analysis and evaluation of these three techniques.

By investigating the effectiveness of Neural Collaborative Filtering, Multilayer Perceptron, and Matrix Factorization for recommender systems, this thesis aims to contribute to the existing body of knowledge by providing valuable insights into the performance and applicability of these techniques. The findings of this research will help researchers, practitioners, and decision-makers in selecting the most suitable technique for specific recommender system applications, ultimately enhancing user satisfaction, and improving the overall quality of recommendations.

1.4 Objective

The objective of this thesis is to evaluate and compare the hit ratio performance of Neural Collaborative Filtering, Multilayer Perceptron, and Matrix Factorization techniques for recommender systems. By calculating the hit ratio and analysing the results, this research aims to identify the strengths and weaknesses of each technique in terms of recommendation accuracy. The findings will provide valuable insights for selecting the most suitable technique based on hit ratio, enhancing the effectiveness of recommender systems. This research contributes to the existing body of knowledge by providing empirical evidence and recommendations for improving recommendation algorithms.

1.5 Thesis Outline

This research aims to assess the effectiveness of NCF, MLP, and matrix factorization techniques within the realm of recommender systems. The outcomes derived from this study hold value for researchers seeking appropriate recommender systems for their own projects or desiring to compare performance benchmarks.

The thesis is thoughtfully structured into five chapters. The synopsis of each chapter is as follows:

Chapter 1: Introduces recommender systems, their significance, an introduction of the techniques employed in the study, the motivation driving this research, and the objectives.

Chapter 2: Provides a comprehensive literature overview of recent advancements in recommender systems, with particular emphasis on NCF, MLP, and matrix factorization.

Chapter 3: Expounds on the techniques, explaining their underlying principles and discussing the methodology employed in this study.

Chapter 4: Presents the findings resulting from the application of the aforementioned techniques.

Chapter 5: Summarizes the study's conclusions and outlines potential avenues for future research.

CHAPTER 2
LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

The literature review section extensively examines the existing body of knowledge concerning recommender systems, with a particular focus on the three techniques being investigated in this study: neural collaborative filtering, multilayer perceptron, and matrix factorization. The main objective of this section is to provide a thorough understanding of the principles, methodologies, and applications of these techniques within the realm of recommender systems. By synthesizing and critically analysing previous research and scholarly works, the literature review establishes a foundational framework for comprehending the current landscape of recommender systems and identifying gaps in the existing literature. Through a systematic review of relevant literature, this section aims to establish the necessary context and theoretical foundations for evaluating the effectiveness and comparative performance of the selected techniques.

2.1 Overview of Recommender Systems

The first research paper on recommender systems is often attributed to the paper titled "GroupLens: An Open Architecture for Collaborative Filtering of Netnews"[17] published in 1994. The paper was authored by Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl from the GroupLens research project at the University of Minnesota.

In this paper, the authors introduced the collaborative filtering approach and presented a system that recommended Usenet news articles to users based on their ratings and similarities to other users. They described the architecture, algorithms, and evaluation of their collaborative filtering system. The GroupLens project was influential in laying the foundation for collaborative filtering and sparked further research and development in the field of recommender systems. The concepts and techniques presented in this paper paved the way for subsequent advancements in personalized recommendation algorithms and the widespread adoption of recommendation systems in various domains.

Girsang *et al.*[5] built a recommendation system serves as a method to understand the preferences of consumers by presenting potential objects of interest. It aims to assist consumers in finding objects that align with their preferences, including popular recommendations like anime films. In this research, the focus is on recommending anime films based on the ratings of previously watched films. Collaborative filtering is employed as a technique, involving the calculation of similarities, predictions, and recommendations. The study utilizes a dataset from Kaggle, consisting of 73,516 users and 12,294 anime. The viewing history of users is matched with the entire user history using the alternating least squares (ALS) method, enabling the recommendation of anime films based on the results. This method is expected to benefit millions of users in discovering the anime films that best match their preferences.

Singh *et al.*[18] conducted a comprehensive study on recommender systems (RS), which have gained considerable research interest due to their ability to provide tailored suggestions to users, matching their interests in finding online items. The authors examined various recommendation approaches, addressed associated issues, and explored information retrieval techniques within the RS domain. Their study encompassed more than 1,000 research papers published by prominent publishers such as ACM, IEEE, Springer, and Elsevier, spanning from 2011 to the first quarter of 2017.

The research conducted by Singh *et al.* [18] yielded valuable insights for both current and future RS researchers, helping them evaluate and shape their research roadmap. By analysing the trends and findings, the authors aimed to contribute to the ongoing development and evolution of RS. Additionally, Singh et al. [reference] presented a forward-looking perspective on the future of RS, identifying potential research directions that could drive innovation in the field. Their work serves as a valuable resource for researchers seeking to understand the current landscape of RS and explore new avenues for exploration.

2.2 Collaborative Filtering Techniques

Schafer *et al.* [6] mention that collaborative filtering emerges as a potent personalization technology that drives the adaptive web, enabling the evaluation and filtering of items based on the opinions and preferences of others. It harnesses the collective wisdom of interconnected communities on the web to handle substantial volumes of data. This exploration introduces the core concepts of collaborative filtering, highlighting its key applications in the adaptive web domain. It delves into the theory and practical aspects of CF algorithms, while also considering important factors such as rating systems and the acquisition of ratings. Evaluating CF systems and the evolution of interactive interfaces for enhanced user experiences are also discussed. Furthermore, privacy challenges unique to CF recommendation services are addressed, along with key research questions that continue to shape the field. Overall, this overview provides insights into collaborative filtering, its functionalities, and its role within the realm of personalized recommendations on the web.

In a study conducted by Mishra *et al.* [9], the primary focus was centred around recent advancements in Collaborative Filtering Approaches, specifically utilizing Matrix Factorization Models and Nearest Neighbourhood Models. The Matrix Factorization Models encompassed various techniques such as SVD and PMF, which were explored in conjunction with newer models that incorporated implicit user feedback, such as SVD++ methods. Additionally, an extension to Factor Models using time SVD++ methods was investigated to account for temporal effects. Similarly, the Nearest Neighbourhood Models were examined, with an emphasis on recent advancements. This included an extension to the Global Neighbourhood Model and the Factorized Neighbourhood Model, both aimed at enhancing predictive accuracy. Furthermore, temporal dynamics were considered to enrich the overall accuracy of the models.

The study provided a comprehensive overview of the following areas: an introduction to Collaborative Filtering Models, techniques for estimating baseline predictors with temporal effects, advanced dimensionality reduction techniques within Matrix Factorization Models, and the evolution of Nearest Neighbourhood Models incorporating temporal effects.

2.3 Neural Collaborative Filtering

Neural Collaborative Filtering was proposed in He *et al.*[11]. Neural collaborative filtering (NCF) revolutionizes the traditional collaborative filtering approach by integrating neural networks to create personalized recommendations. Collaborative filtering, a widely employed technique in recommendation systems, exploits user behaviour and preferences to generate suggestions. It operates on the assumption that users who shared similar preferences in the past will exhibit similar preferences in the future. Conventionally, collaborative filtering employs matrix factorization methods to decompose the user-item interaction matrix into lower-dimensional latent feature vectors, representing users and items. These latent vectors capture underlying characteristics or factors influencing user preferences. Recommendations are then derived by measuring similarity between users or items in this latent space.

NCF takes collaborative filtering a step further by incorporating neural networks, which are highly capable models for capturing complex patterns and nonlinear relationships in data. In NCF, neural networks are utilized to model the user-item interaction and directly learn latent representations from the raw data. This integration enables NCF to capture intricate user-item dynamics and provide more accurate and personalized recommendations.

2.4 Multilayer Perceptron

Recommender systems aim to personalize recommendations by predicting user preferences based on their past tastes. However, a common challenge faced by many recommender systems is the Cold-start problem, where the system struggles to provide recommendations for users with limited or insufficient information. In the study conducted by Alizadeh *et al.*[19], the authors propose a hybrid recommender system that combines collaborative filtering (CF) techniques and content-based filtering (CBF) using an artificial neural network (ANN). The objective is to develop a model that is suitable for both cold-start users and regular users. To enhance the model's performance, Mutual Information techniques are utilized to select the most relevant properties.

The proposed method is evaluated using MovieLens and Netflix datasets, and the results demonstrate that the prediction accuracy of the proposed approach outperforms other methods applied to the same datasets. The hybrid recommender system shows promising potential in addressing the Cold-start problem and providing more accurate recommendations for users.

In another notable study conducted by Singh *et al.* [20], the significance of the perceptron model in the field of artificial neural networks was further emphasized. The research delved into the practical applications of the perceptron, its learning mechanisms, and the impact it has had on advancing the field of machine perception and artificial intelligence. The study provided valuable insights into the concepts and workings of the perceptron, ultimately contributing to the broader understanding and development of neural networks. By incorporating the findings of Singh *et al.* [20], it becomes evident that the perceptron model continues to be a cornerstone in the field, driving advancements and enabling machines to learn and make intelligent decisions.

2.5 Matrix Factorization

In accordance with the paper published by Bell *et al.*[14] in 2009, matrix factorization techniques have emerged as a dominant methodology within collaborative filtering recommenders. The research, which drew insights from datasets like the Netflix Prize data, demonstrated that these techniques outperform classical nearest-neighbour approaches in terms of accuracy. One of the notable advantages of matrix factorization techniques is their ability to provide a compact and memory-efficient model that can be easily learned by systems. Moreover, these techniques naturally integrate crucial aspects of the data, including multiple forms of feedback, temporal dynamics, and confidence levels. This integration allows for a more comprehensive and nuanced understanding of the data, leading to improved recommendation performance.

The findings presented by Bell *et al.*[14] highlight the convenience and effectiveness of matrix factorization techniques in collaborative filtering. By leveraging these techniques, recommender systems can generate accurate and context-aware recommendations, making them valuable tools in the field of personalized recommendation systems.

Li *et al.*[13] highlights the significant emphasis placed by researchers and programmers on recommendation systems as a solution to tackle the challenges of information overload. They specifically focus on the collaborative filtering algorithm, which is widely used in this context. To enhance the algorithm's performance, the authors explore the application of Matrix Factorization, providing a comprehensive overview in this paper. They introduce the gradient descent technique employed in Matrix Factorization and elaborate on the procedures involved in Basic Matrix Factorization (MF), Regularized MF, and Biases MF. The study includes experiments that analyze and compare the performance of collaborative recommendation algorithms based on these three types of Matrix Factorization. By examining parameters such as the number of latent factors (K), the number of iterations (K), and the regularization coefficient (λ), the authors aim to optimize algorithm accuracy. The findings of Li *et al.*[13] contribute valuable insights to the field of collaborative filtering and Matrix Factorization, providing guidance for researchers and practitioners seeking to enhance the performance and accuracy of recommendation systems.

CHAPTER 3

METHODOLOGY

CHAPTER 3
METHODOLOGY

The methodology section outlines the research approach and experimental setup employed in this study to evaluate the effectiveness of neural collaborative filtering, multilayer perceptron, and matrix factorization techniques for recommender systems using the MovieLens 20M dataset. This section provides a detailed description of the data collection and pre-processing steps, the implementation of the selected techniques, the evaluation metrics utilized, and the experimental design. By following a systematic and rigorous methodology, this study aims to ensure the reliability and validity of the findings, facilitating a comprehensive and comparative analysis of the techniques under investigation.

3.1 Data Collection and Pre-processing

In this study, the first step involved in the methodology is the collection and pre-processing of the MovieLens 20M dataset to ensure its suitability for evaluating the effectiveness of neural collaborative filtering, multilayer perceptron, and matrix factorization techniques for recommender systems. This initial step focuses on curating and preparing the dataset, ensuring its quality and relevance to accurately represent user preferences and interactions. By carefully processing the dataset, the study aims to establish a reliable data source for subsequent evaluations and analyses of the recommendation techniques.

3.1.1 Data Collection

The dataset used in this study is the MovieLens dataset, which is a collection of movie ratings obtained from the MovieLens website. The dataset is maintained by GroupLens, a research group at the University of Minnesota. It consists of multiple versions, including "25m", "latest-small", "100k", "1m", and "20m".

The "25m" dataset is the latest stable version and is recommended for research purposes. The "latest-small" dataset is a smaller subset of the latest version that is regularly updated by GroupLens.

The "100k" dataset is the oldest version and includes demographic data in addition to movie and rating data.

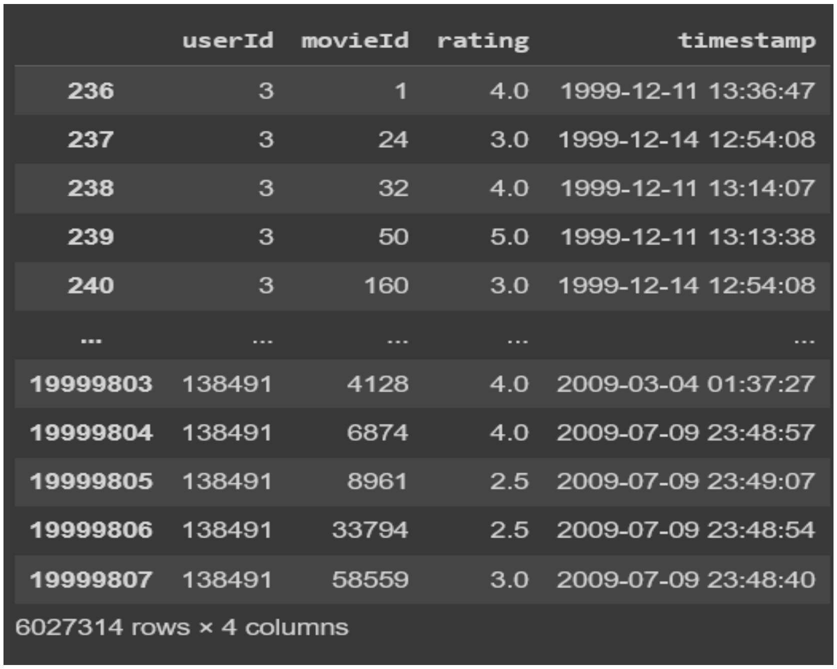
The "1m" dataset is the largest MovieLens dataset, containing demographic information along with movie and rating data. Lastly, the "20m" dataset is widely used in academic papers, alongside the "1m" dataset.

This study uses the “20m” dataset. It has 20000263 ratings and 465564 tag applications spread among 27278 films. 138493 users between January 9, 1995, and March 31, 2015, produced these statistics. On October 17, 2016, this dataset was produced. Users who wanted to be included were chosen at random. All the individuals who were chosen had rated at least 20 films. There is no demographic data provided. Each user is represented by an id and no other data is provided.

3.1.2 Data Pre-processing

In this study, a subsampling technique is applied to the dataset to manage memory usage effectively. Specifically, only data from 30% of the users will be selected for analysis. The user selection process is performed randomly, ensuring a representative sample.

After applying this subsampling approach, the dataset is reduced to 6,027,314 rows of data, representing user-authored movie reviews. These reviews are contributed by 41,547 users, providing a substantial amount of data for analysis. The resulting dataframe captures the user, movie, and corresponding review information for further examination and evaluation.



	userId	movieId	rating	timestamp
236	3	1	4.0	1999-12-11 13:36:47
237	3	24	3.0	1999-12-14 12:54:08
238	3	32	4.0	1999-12-11 13:14:07
239	3	50	5.0	1999-12-11 13:13:38
240	3	160	3.0	1999-12-14 12:54:08
...	...	...	...	...
19999803	138491	4128	4.0	2009-03-04 01:37:27
19999804	138491	6874	4.0	2009-07-09 23:48:57
19999805	138491	8961	2.5	2009-07-09 23:49:07
19999806	138491	33794	2.5	2009-07-09 23:48:54
19999807	138491	58559	3.0	2009-07-09 23:48:40

6027314 rows × 4 columns

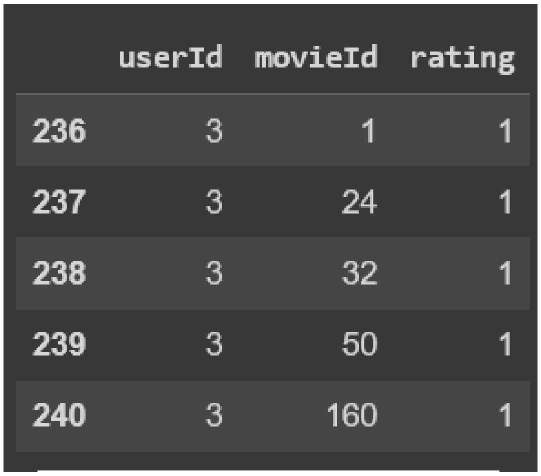
Figure 3.1 Dataset view of Movielens dataset

The timestamp column in the dataset provides valuable information about the date and time when each review was submitted, along with the corresponding rating. To ensure a reliable evaluation of the recommender system, the study adopts the leave-one-out approach for the train-test split strategy, leveraging the timestamp column.

Under this approach, the most recent review for each user is selected as the test set, while the remaining reviews are utilized as training data. This strategy avoids potential biases introduced by using a random split, which may lead to unfair evaluations. By considering the temporal aspect of the data, the train-test split technique helps prevent data leakage and look-ahead bias, enabling the trained model to demonstrate better real-world performance.

The adoption of the leave-one-out approach and careful consideration of the timestamp column contribute to the development and testing of the recommendation engine, ensuring the reliability and transferability of the trained model's performance.

This explicit data is then converted into implicit data. If a user has rated a movie, then it is believed that user has interacted with the item, hence given the value “1”, i.e. (positive class). Rest of the movies for that particular user are given the value “0”, i.e., member of the negative class, that the user has not interacted with.



	userId	movieId	rating
236	3	1	1
237	3	24	1
238	3	32	1
239	3	50	1
240	3	160	1

Figure 3.2 Implicit Dataset

3.2 Matrix Factorization

Matrix factorization is a technique employed in collaborative filtering-based recommendation systems to break down a user-item interaction matrix into lower-dimensional representations. It assumes that latent factors or features underlie the observed user-item interactions.

The user-item interaction matrix represents the historical behaviour or preferences of users towards items. It is typically a sparse matrix, with users as rows, items as columns, and entries indicating user-item interactions or ratings. Since most entries are missing, matrix factorization aims to fill these gaps by learning latent factors.

Users and items are represented as vectors in a latent space through matrix factorization. By decomposing the user-item interaction matrix into these latent representations, it becomes possible to capture the underlying characteristics or factors influencing user preferences. Depending on the domain, these factors can correspond to genres, topics, or styles.

The factorization process involves finding two matrices: the user matrix (U) and the item matrix (V). The user matrix contains the latent factors associated with each user, while the item matrix contains the latent factors associated with each item. The product of these matrices, U * V^T, approximates the user-item interaction matrix.

Optimization techniques such as gradient descent or alternating least squares are typically employed in the factorization process. The objective is to minimize the discrepancy between the approximated user-item interaction matrix and the observed interactions. This is achieved by adjusting the values in the user and item matrices to enhance the approximation.

Once the factorization is learned, the model can generate recommendations by predicting the missing entries in the user-item interaction matrix. These predicted ratings or interactions are used to rank items and offer personalized recommendations to users.

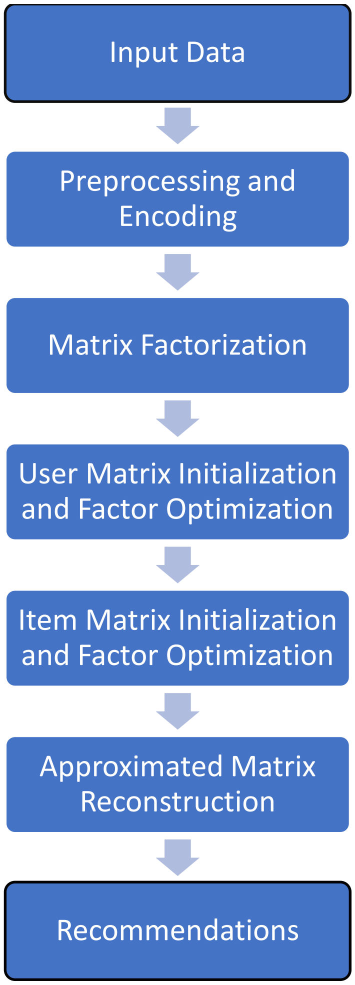

Input Data
Preprocessing and Encoding
Matrix Factorization
User Matrix Initialization and Factor Optimization
Item Matrix Initialization and Factor Optimization
Approximated Matrix Reconstruction
Recommendations

Figure 3.3 Matrix Factorization Framework

- Singular Value Decomposition (SVD)[15]: Matrix factorization in recommender systems is often based on Singular Value Decomposition (SVD). SVD is a mathematical technique that decomposes a matrix into three separate matrices: U, Σ, and V^T. U represents the left singular vectors, Σ is a diagonal matrix containing the singular values, and V^T represents the right singular vectors. SVD can be used to approximate the user-item interaction matrix by selecting a subset of the largest singular values and their corresponding singular vectors.
- Latent Factors: The idea behind MF is to represent users and items as vectors in a latent space of lower dimensionality. The latent factors capture the underlying characteristics or features that influence user preferences and item properties. By learning these latent representations, MF can model the relationships between users and items based on their shared latent factors.
- Sparsity and Missing Entries: The user-item interaction matrix in recommender systems is typically sparse, meaning most of the entries are missing because not all users interact with all items. MF addresses this sparsity issue by filling in the missing entries in the matrix using the learned latent factors. By reconstructing the matrix from the lower-dimensional latent representations, MF can estimate the missing values and make recommendations for the unseen user-item pairs.
- Regularization: Regularization techniques are often employed in MF to prevent overfitting and improve generalization. Regularization terms, such as L1 or L2 regularization, are added to the objective function to encourage small values for the latent factors. This regularization helps in controlling the complexity of the model and mitigates the impact of noise and outliers in the training data.
- Bias Terms: In addition to the latent factors, MF models may incorporate bias terms to account for user and item biases. User bias captures the inherent tendencies of users to rate items higher or lower, while item bias represents the inherent quality or popularity of items. These bias terms help improve the accuracy of predictions by accounting for the baseline preferences and biases of users and items.
- Optimization: MF models are typically trained using optimization algorithms like stochastic gradient descent (SGD)[20] or alternating least squares (ALS)[16]. These algorithms iteratively update the parameters (latent factors, bias terms) to minimize the objective

function. Gradient-based methods like SGD update the parameters by computing gradients of the loss function with respect to the model parameters, while ALS alternates between optimizing the user and item factors.

- Matrix factorization has been widely used in recommender systems due to its simplicity, interpretability, and ability to handle sparse and missing data. However, it is worth noting that MF assumes a linear relationship between users and items and may not capture complex non-linear patterns in the data. To address this limitation, hybrid models combining MF with other techniques like content-based filtering or neural networks have been developed to leverage the strengths of multiple approaches.

3.3 Multilayer Perceptron

A multilayer perceptron (MLP) is an artificial neural network (ANN)[21], [22] architecture consisting of interconnected nodes or neurons organized in multiple layers. It operates in a feedforward manner, where information flows from the input layer to the output layer without loops or cycles.

At its core, an MLP comprises individual neurons or units. Each neuron receives input signals, performs computations on them, and generates an output. The output of a neuron is determined by applying an activation function to the weighted sum of its inputs.

The MLP typically comprises three types of layers:

- Input Layer: The input layer receives the initial input data or features. Each neuron in the input layer represents a specific feature, and the values in these neurons propagate information to the subsequent layers.
- Hidden Layers: Hidden layers exist between the input and output layers. They process the inputs received from the previous layer. An MLP can have one or more hidden layers, and each hidden layer contains multiple neurons. Neurons in the hidden layers compute a weighted sum of inputs, followed by the activation function, to produce their outputs.
- Output Layer: The output layer generates the final output of the MLP. It receives inputs from the neurons in the last hidden layer and computes the final predictions or outputs based

on the specific task. The number of neurons in the output layer depends on the nature of the task. For instance, binary classification tasks typically involve a single neuron, while multi-class classification tasks involve multiple neurons, each representing a distinct class.

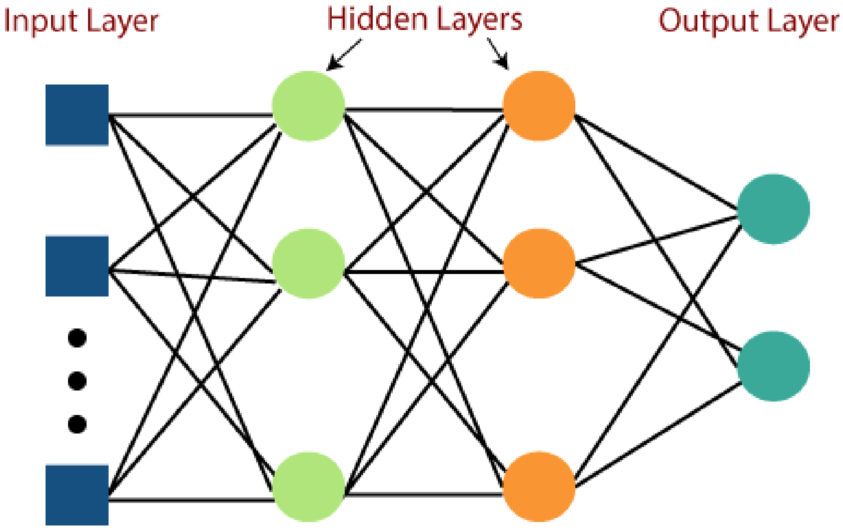


Figure 3.4 Multilayer Perceptron [26]

Connections between neurons in adjacent layers are associated with weights, which dictate the strength or significance of the connections. During training, the MLP learns the optimal values of these weights by iteratively adjusting them using a designated optimization algorithm (e.g., backpropagation).

The activation function applied to the weighted sum of inputs within each neuron introduces non-linearity to the MLP, enabling it to learn intricate patterns and relationships in the data. Common activation functions utilized in MLPs include sigmoid, tanh, and ReLU (Rectified Linear Unit)[23].

MLPs are potent models capable of approximating any continuous function given a sufficient number of hidden neurons. They have found wide-ranging applications, including pattern recognition, regression analysis, and classification tasks such as image recognition, natural language processing, and recommendation systems.

Multilayer perceptrons (MLPs) have indeed been utilized in recommendation systems to model user-item interactions and provide personalized recommendations. Here is a general outline of how MLPs can be applied in recommendation systems:

- Data Representation: The user-item interaction data is typically represented as a sparse matrix, with users as rows, items as columns, and the entries representing user-item interactions (e.g., ratings, clicks, purchases). This matrix serves as the input to the MLP-based recommendation model.
- Input Encoding: The data needs to be encoded in a suitable format to be fed into the MLP. One common approach is one-hot encoding, where each user and item is represented by a binary vector with a value of 1 in the corresponding ID position and 0 elsewhere. Alternatively, embeddings can be used to represent users and items as continuous low-dimensional vectors, capturing underlying preferences and similarities.
- MLP Architecture: The architecture of an MLP for recommendation systems typically consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the encoded user and item information. The hidden layers apply non-linear transformations to the inputs, enabling the model to learn complex patterns and interactions. The output layer produces the final recommendation scores or predictions.
- Training: The MLP is trained using an optimization algorithm, such as stochastic gradient descent (SGD), to minimize the prediction error between the model's outputs and the actual user-item interactions in the training data. The choice of loss function depends on the specific recommendation task, such as mean squared error (MSE) for rating prediction or binary cross-entropy for binary preference prediction.
- Recommendation Generation: Once trained, the MLP can generate recommendations. Given a user, the MLP takes the user's encoded representation as input and produces recommendation scores or predictions for all items. The items with the highest scores are recommended to the user. The number of recommendations and any additional criteria (e.g., diversity, novelty) can be incorporated based on the recommendation system's requirements.
- Evaluation and Iteration: The performance of the MLP-based recommendation system is evaluated using appropriate metrics like precision, recall, or mean average precision. Based

on the evaluation results, the model can be fine-tuned, hyperparameters optimized, and the training process iterated to improve recommendation quality.

MLPs in recommendation systems can benefit from additional techniques, such as incorporating side information (e.g., item attributes, user demographics) as input features or applying regularization methods (e.g., dropout) to prevent overfitting. Hybrid models can also be developed by combining MLPs with other recommendation techniques like matrix factorization or content-based filtering to leverage their strengths.

3.4 Neural Collaborative Filtering

NCF stands for Neural Collaborative Filtering, which is a recommendation algorithm that combines neural networks with collaborative filtering techniques. It was proposed as an extension of traditional collaborative filtering methods to leverage the power of deep learning models in capturing complex user-item interactions and generating accurate recommendations.

Collaborative filtering (CF) is a popular approach in recommender systems that relies on user-item interaction data to make recommendations. It assumes that users with similar preferences in the past will have similar preferences in the future. CF can be categorized into two types: memory-based CF, which uses similarity measures between users or items, and model-based CF, which learns a model from the data.

NCF, on the other hand, introduces neural networks into the collaborative filtering framework. Instead of relying solely on similarity measures or predefined models, NCF utilizes multilayer perceptrons (MLPs) or other neural network architectures to learn non-linear patterns and capture high-order relationships in the user-item interaction data.

The key idea behind NCF is to represent users and items as dense, low-dimensional vectors in a latent space. These representations, often referred to as embeddings, are learned through the neural network model. By feeding the user and item embeddings into the network, NCF can make predictions or recommendations for unseen user-item pairs.

One common architecture used in NCF is the concatenation of user and item embeddings followed by a sequence of fully connected layers with non-linear activation functions. The network is trained using optimization techniques such as stochastic gradient descent (SGD) to minimize the difference between the predicted and actual user-item interactions in the training data.

NCF offers several advantages over traditional collaborative filtering methods. It can capture more complex patterns and interactions in the data, handle sparsity and cold-start problems, and provide personalized recommendations. NCF has been applied successfully in various domains, including e-commerce, movie recommendations, and news article recommendations.

However, it is worth noting that NCF requires a large amount of training data and computational resources due to the complexity of neural network models. Additionally, hyperparameter tuning and regularization techniques are crucial for achieving optimal performance and preventing overfitting.

To encapsulate the key points, NCF is a recommendation algorithm that merges neural networks with collaborative filtering techniques. This fusion allows NCF to adeptly learn intricate user-item interactions and deliver precise recommendations. By harnessing the strengths of neural networks and collaborative filtering, NCF excels at capturing complex patterns and preferences, enabling it to surpass conventional collaborative filtering methods. The outcome is a recommendation system that offers highly personalized and accurate recommendations to users.

3.4.1 The Framework for Neural Collaborative Filtering

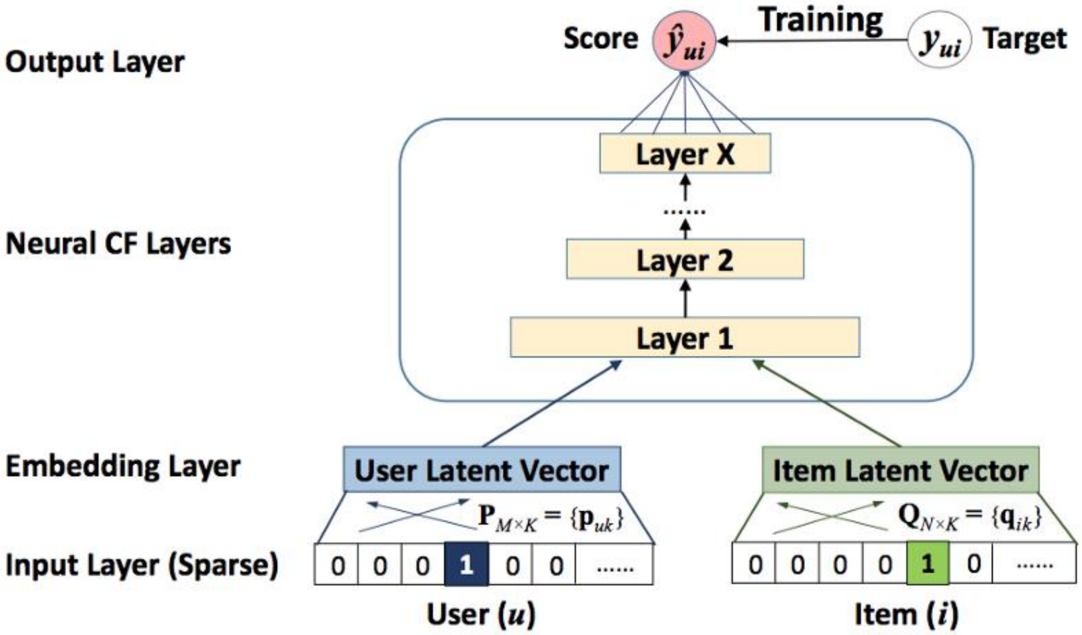
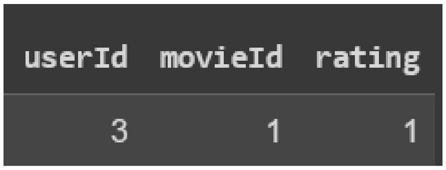


Figure 3.5 NCF Framework [27]

- The input layer of the neural network is responsible for binarizing a sparse vector to identify user and item interactions. A value of 1 indicates that user (u) has interacted with item (i), facilitating person and object identification.
- The embedding layer, which is fully connected, transforms the sparse representation into a dense vector. This layer utilizes user/item embeddings to generate latent vectors that capture underlying user/item characteristics.
- The neural collaborative filtering (CF) layers employ a multi-layered neural architecture to transfer the latent vectors into prediction scores. This architecture enables the network to learn complex patterns and relationships within the user-item interactions.
- In the final output layer, the network minimizes the pointwise loss or pairwise loss to generate the projected score. This process optimizes the model's performance by reducing the discrepancy between predicted scores and actual interactions.

3.4.2 Training sample for NCF



userId	movieId	rating
3	1	1

Figure 3.6 Training Sample

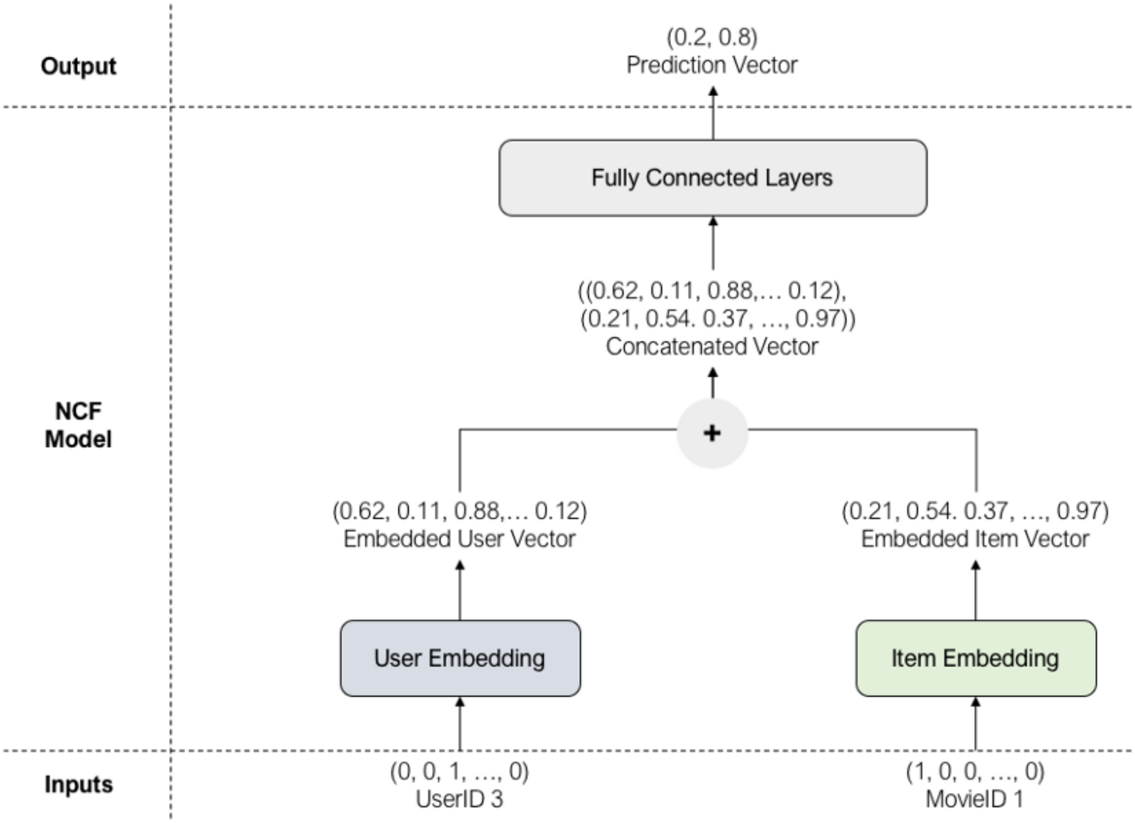


Figure 3.7 NCF working on the training sample [28]

The input to the model consists of a one-hot encoded vector representing the person (userId = 3) and the item (movieId = 1). The genuine label for this input is 1, indicating that the user has interacted with the movie (a positive sample, such as giving it a rating).

To obtain compact and dense user and item vectors, the user input vector and item input vector are separately passed through their corresponding embedding layers. These embedding layers transform the one-hot encoded vectors into lower-dimensional dense representations.

The embedded user and item vectors are then concatenated together and fed through multiple fully connected layers. These layers apply various computations and transformations to learn complex patterns and relationships within the user-item interactions. Ultimately, this process outputs a prediction vector by mapping the concatenated embeddings.

To determine the most likely class, a Sigmoid activation function is applied at the output layer. In the given example, since the predicted score of 0.8 is greater than 0.2, the model predicts the positive class (1) as the most likely outcome.

CHAPTER 4 RESULT AND DISCUSSION

CHAPTER 4

RESULT AND DISCUSSION

The assessment of recommender systems requires the consideration of appropriate evaluation measures based on the test results. Unlike conventional machine learning projects, where metrics like Accuracy and RMSE[24] are commonly used for classification and regression problems, such measures are inadequate for evaluating recommender systems. Instead, specialized evaluation metrics tailored to the unique characteristics of recommendation tasks are utilized to assess the performance of these systems.

4.1 Model Evaluation

The evaluation technique used in this study is Hit Ratio@10[25].

It is important to note that the requirement for successful recommendations does not necessitate user interaction with every suggestion in the list. Instead, the presence of user interaction with at least one item in the list indicates the success of the recommendations.

To replicate this assessment methodology, the following steps are conducted to generate a list of the top 10 items for each user:

- Randomly select 99 items that the user has not interacted with.
- Combine these 99 items with the test item (the last item the user interacted with). This results in a list of 100 items.
- Apply the model to predict the probabilities for these 100 items.
- Rank the items based on the predicted probabilities.
- Select the top 10 items from the ranked list. If the test item is among the top 10, it is considered a hit.

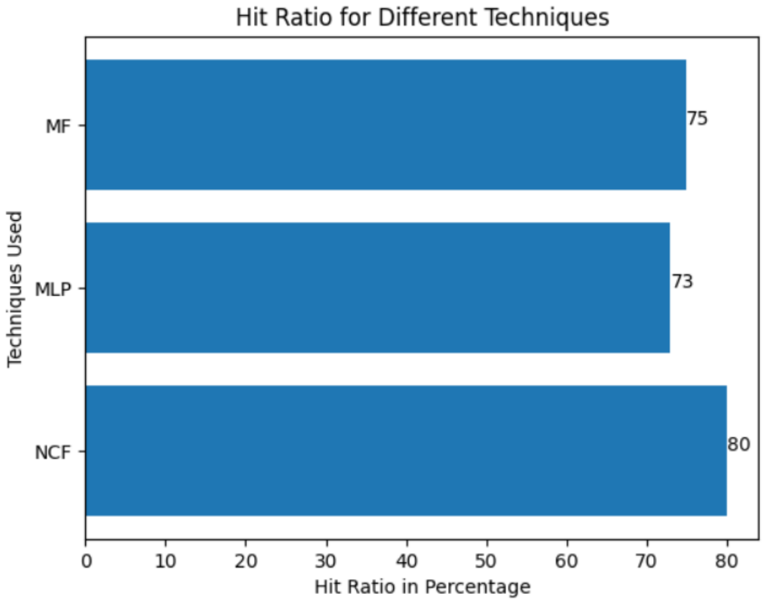
Repeat the procedure for each user and calculate the average number of hits. The average number of hits determine the Hit Ratio.

4.2 Result

Table 4.1 Hit Ratio of Techniques Used

S. No.	Technique	Hit Ratio@10
1	Matrix Factorization	0.75
2	Multilayer Perceptron	0.73
3	Neural Collaborative Filtering	0.80

The evaluation of recommender system techniques, including Neural Collaborative Filtering (NCF), Multilayer Perceptron (MLP), and Matrix Factorization (MF), based on hit ratio yields interesting findings. NCF demonstrates the highest hit ratio of 80%, indicating its effectiveness in accurately recommending relevant items to users. Following closely, MF achieves a hit ratio of 75%, showcasing its ability to generate personalized recommendations. However, MLP obtains a slightly lower hit ratio of 73%, suggesting potential limitations in capturing user preferences effectively. These results underscore the varying performance of the evaluated techniques and emphasize the importance of selecting the appropriate method for achieving accurate recommendations in different scenarios.



Graph 4.1 Hit Ratio of Techniques Used

If the hit ratio is 0.80, it implies that 80% of the top 10 recommended items are relevant to the user. Essentially, when the recommender system suggests 10 items to a user, it is expected that, on average, 8 out of those 10 items align with the user's interests or preferences based on their past behaviour or choices.

In the leave-one-out evaluation approach used in this study, a hit ratio of 80% for NCF implies that, on average, the item from the test set will be recommended to the user 8 out of 10 times. For MF, the hit ratio of 75% means that the item will appear in the recommendations approximately 7.5 times out of 10 on average. MLP achieved a slightly lower hit ratio of 73%, indicating that the item will be suggested around 7.3 times out of 10.

A higher hit ratio indicates better performance in terms of recommending relevant items. It signifies that a larger proportion of the top recommendations are in line with the user's preferences.

4.3 Discussion

For making a recommender system, there can be various techniques used. In this study the techniques that were used give a hit ratio of 80% for NCF, 75% for MF and 73% for MLP. The analysis of recommendation performance reveals interesting findings for the evaluated techniques: neural collaborative filtering (NCF), matrix factorization (MF), and multilayer perceptron (MLP). NCF achieved the highest performance, accurately recommending items that align with user preferences. MF closely followed, generating personalized recommendations tailored to individual users. MLP achieved slightly lower performance, suggesting potential limitations in capturing user preferences effectively. These results highlight the varying capabilities of the techniques and emphasize the significance of selecting the appropriate approach based on the specific requirements of the recommender system.

CHAPTER 5 CONCLUSION AND FUTUREWORK

CHAPTER 5
CONCLUSION AND FUTUREWORK

5.1 Conclusion

In the present work carried out, the effectiveness of the techniques Neural Collaborative Filtering (NCF), Multilayer Perceptron (M LP) and Matrix Factorization (MF) is evaluated.

Neural Collaborative Filtering gives the best Hit Ratio of 80% in comparison of 75% for Matrix Factorization and 73% for Multilayer Perceptron.

For larger and much complicated dataset, it can be safely concluded that NCF stand superior to Matrix Factorization and Multilayer Perceptron.

5.2 Future Work

In this study 30% of the dataset was used to manage memory. A stronger system that has better processing power can include all of the data and give better results.

This study does not include contextual information such as user demographics, temporal information (time decay: preferences of the user may change over time, and the relevance of historical interactions may diminish over time).

Making a hybrid model that includes NCF, MLP or MF with some other techniques may give better results.

REFERENCES

[1] C. Aggarwal, *Recommender Systems*. 2016. doi: 10.1007/978-3-319-29659-3.

[2] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, “Content-Based Movie Recommendation System Using Genre Correlation,” Mar. 2018.

[3] C. Zisopoulos, S. Karagiannidis, G. Demirtsoglou, and S. Antaris, “Content-Based Recommendation Systems,” Mar. 2008.

[4] K. Ma, “Content-based Recommender System for Movie Website,” 2016. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-188494

[5] A. Girsang, B. Al Faruq, H. Herlianto, and S. Simbolon, “Collaborative Recommendation System in Users of Anime Films,” *J. Phys. Conf. Ser.*, vol. 1566, p. 12057, Mar. 2020, doi: 10.1088/1742-6596/1566/1/012057.

[6] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, Berlin, Heidelberg: Springer-Verlag, 2007, pp. 291–324.

[7] S. Ildere, “Collaborative Filtering System,” Mar. 2019, pp. 624–628. doi: 10.1109/UBMK.2019.8907171.

[8] Y. Qingmin and C. Xingyu, “Research on Collaborative Filtering Recommendation Algorithm,” Mar. 2022, pp. 741–743. doi: 10.1109/ICPECA53709.2022.9719134.

[9] P. Mishra and S. Mohanty, “Collaborative Filtering Techniques,” 2021, pp. 181–220. doi: 10.1201/9780367631888-8.

[10] R. Vrushal, “Moviemender- A Movie Recommender System,” Mar. 2016, doi: 10.5281/zenodo.167478.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural Collaborative Filtering.” arXiv, 2017. doi: 10.48550/ARXIV.1708.05031.

[12] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, “Multilayer perceptron and neural networks,” *WSEAS Trans. Circuits Syst.*, vol. 8, 2009.

[13] C. Li and C. Yang, “The research based on the Matrix Factorization recommendation algorithms,” in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 691–698. doi: 10.1109/IMCEC.2016.7867298.

[14] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” *Computer (Long. Beach. Calif).*, vol. 42, no. 8, pp. 30–37, 2009, doi: 10.1109/MC.2009.263.

[15] S. Brunton and J. Kutz, “Singular Value Decomposition (SVD),” 2019, pp. 3–46. doi: 10.1017/9781108380690.002.

[16] G. Takács and D. Tikk, “Alternating least squares for personalized ranking,” 2012, doi: 10.1145/2365952.2365972.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An Open Architecture for Collaborative Filtering of Netnews,” in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 1994, pp. 175–186. doi: 10.1145/192844.192905.

[18] P. Singh, P. Dutta Pramanik, A. Dey, and P. Choudhury, “Recommender Systems: An Overview, Research Trends, and Future Directions,” *Int. J. Bus. Syst. Res.*, vol. 15, pp. 14–52, Mar. 2021.

[19] D. D. Sanandaj and S. H. Alizadeh, “A hybrid recommender system using multi layer perceptron neural network,” in *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*, 2018, pp. 7–13. doi: 10.1109/RIOS.2018.8406624.

[20] N. Ketkar, “Stochastic Gradient Descent,” 2017, pp. 111–130. doi: 10.1007/978-1-4842-2766-4_8.

[21] E. Grossi and M. Buscema, “Introduction to artificial neural networks,” *Eur. J. Gastroenterol. Hepatol.*, vol. 19, pp. 1046–1054, 2008, doi: 10.1097/MEG.0b013e3282f198a0.

[22] M. Mishra and M. Srivastava, “A view of Artificial Neural Network,” in *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, 2014, pp. 1–3. doi: 10.1109/ICAETR.2014.7012785.

[23] T. Szandala, “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks,” *CoRR*, vol. abs/2010.09458, 2020, [Online]. Available: https://arxiv.org/abs/2010.09458

[24] A. Botchkarev, “A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms,” *Interdiscip. J. Information, Knowledge, Manag.*, vol. 14, pp. 45–76, 2019, doi: 10.28945/4184.

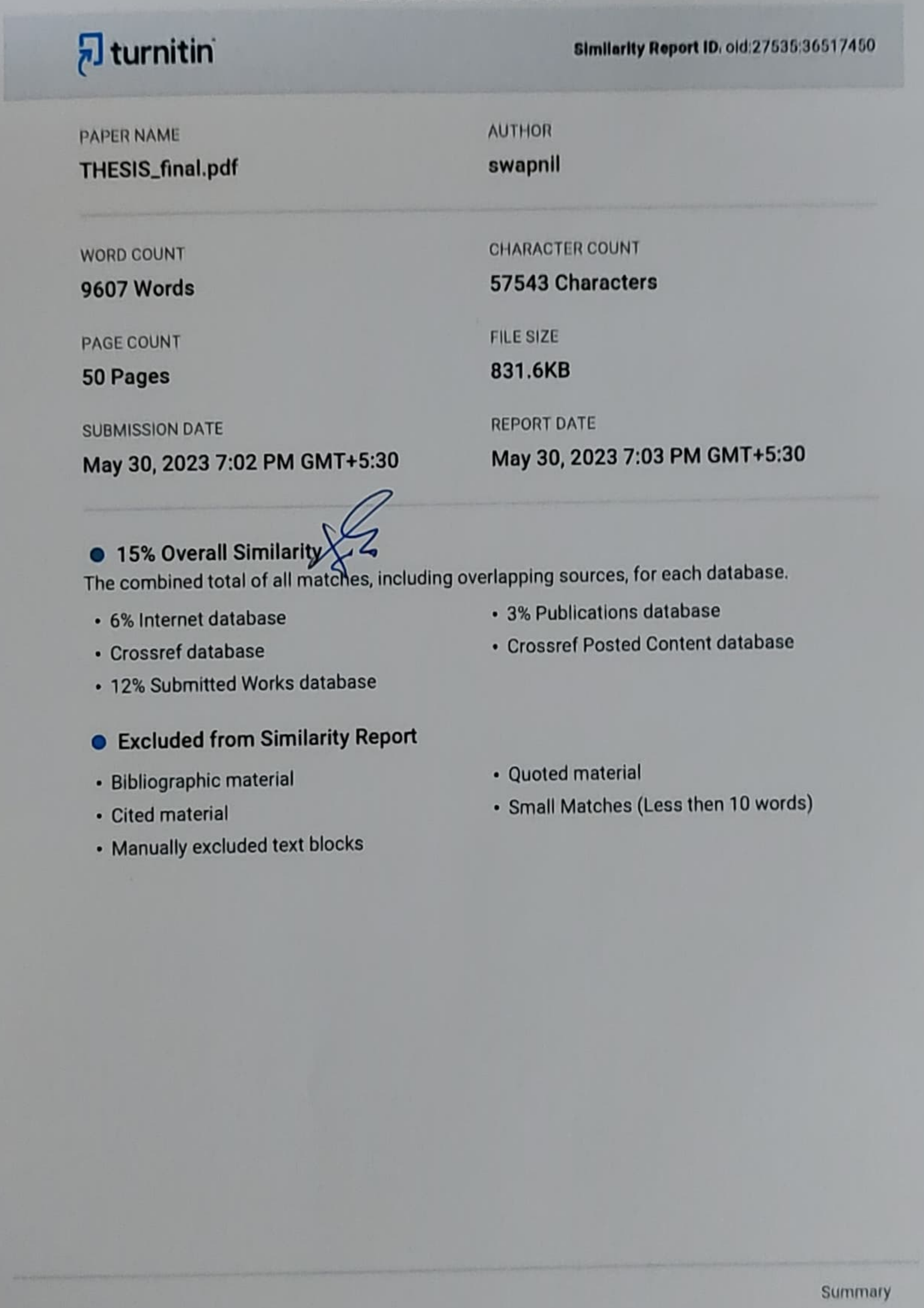
[25] A. Alsini, D. Huynh, and A. Datta, “Hit ratio: An Evaluation Metric for Hashtag Recommendation.” 2020.

[26] https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow

[27] https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401

[28] https://towardsdatascience.com/deep-learning-based-recommender-systems

PLAG REPORT



turnitin

Similarity Report ID: oid:27535:36517450

PAPER NAME

THESIS_final.pdf

AUTHOR

swapnil

WORD COUNT

9607 Words

CHARACTER COUNT

57543 Characters

PAGE COUNT

50 Pages

FILE SIZE

831.6KB

SUBMISSION DATE

May 30, 2023 7:02 PM GMT+5:30

REPORT DATE

May 30, 2023 7:03 PM GMT+5:30

● **15% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- Crossref database
- 12% Submitted Works database
- 3% Publications database
- Crossref Posted Content database

● **Excluded from Similarity Report**

- Bibliographic material
- Cited material
- Manually excluded text blocks
- Quoted material
- Small Matches (Less then 10 words)

Summary