RESEARCH ARTICLE

Performance Evaluation of Collaborative Filtering Recommender System on MovieLens Dataset

Muhammad Ahmad Hafeez[1,†], **Tahir Sher**[2,†], **Abdul Rehman**[3,*] **and Imran Ihsan**[1]

[1] Department of Creative Technologies, Air University, Islamabad 44000, Pakistan
[2] Department of Artificial Intelligence, Korea University, Seoul 02842, Republic of Korea
[3] Convergence Institute of Human Data Technology, Jeonju University, Jeonju 55069, Republic of Korea

Abstract

In today's technological landscape, recommender systems provide essential personalized suggestions by leveraging user preferences. This study evaluates User-Based (UBCF) and Model-Based Collaborative Filtering (MBCF) on the MovieLens 1M dataset, comparing performance on complete data versus partitions based on age and occupation. Using MAE and RMSE metrics, we assessed UBCF with Euclidean/Cosine similarity and MBCF with NMF/SVD. Results show MBCF with SVD achieved the best performance (MAE: 0.6909, RMSE: 0.8761), outperforming UBCF by approximately 5.2% in MAE and 5.4% in RMSE. This confirms model-based approaches, particularly SVD, excel with complete datasets, while demographic partitioning reduces accuracy due to data sparsity. Future work will explore hybrid models combining global and partition-based analysis with deep learning for enhanced personalization.

Keywords: recommender system, user-based & model-based collaborative filtering, cosine similarity, euclidean similarity, non-negative matrix factorization, singular value decomposition, mean absolute error, root mean squared error.

1 Introduction

Recommendations from others have been used by individuals for centuries, spanning from ancient times to modern-day, to aid in the process of making significant as well as trivial selections. Early in the history of computing, it was realized that computers could make suggestions. Bhatnagar. V a computer-based librarian [1], was a pioneer in the development of automatic recommender systems. Over the past 20 years, a lot of study has been done on how to automatically recommend items to individuals; several strategies have been put forth in this regard [2, 3]. The main purpose of a recommender system is to estimate item ratings and present a tailored list of recommended things to a specific user [4]. There are several techniques to this job, such as rating comparison, item or user qualities, and demographic relationships [5, 6]. User profiles [7, 8] and item characteristics are two popular tools used by

Submitted: 18 July 2025
Accepted: 12 August 2025
Published: 14 February 2026

Vol. 2, **No.** 2, 2026.
10.62762/TACS.2025.714333

***Corresponding author:**
✉ Abdul Rehman
a.rehman.jj@jj.ac.kr

† These authors contributed equally to this work

Citation
Hafeez, M. A., Sher, T., Rehman, A., & Ihsan, I. (2026). Performance Evaluation of Collaborative Filtering Recommender System on MovieLens Dataset. *ICCK Transactions on Advanced Computing and Systems*, 2(2), 137–157.

recommender algorithms. User profiles may include information such as gender, age, and occupation [9]. Movies, for example, may contain a genre, an actor list, a specific producer, or other relevant features that connect items together. Recommender Systems (RSs) are computer programmes that offer suggestions to end users based on their preferences or the preferences of other users who shared those preferences [10, 11]. Major utility of recommendation systems in numerous industries, like tourism, banking, etc is a result of their encouraging success in e-commerce, movies, music, books, and news suggestions.

1.1 Motivation.

Recommender systems assist users in making finest choices during online transactions, to increase the sales and to improve their web surfing and shopping experiences. Although nowadays, recommender systems are utilizing a variety of machine learning (ML) and deep learning methods (DL), there is always opportunity for improvement. In the case of the MovieLens dataset, for example, multiple types of recommender systems and algorithm combinations are employed to recommend a movie to the viewer based on his preferences. After downloading the MovieLens dataset, some preprocessing steps are required to check missing values, splitting of given genre column into more columns each for separate genre, extraction of movie years from movie title column and to get the descriptive statistics of the different features of the dataset. In this regard the following are few questions need to be answered:

- The Complexity of MovieLens data is increased by the aforementioned factors. So, what are the methods or techniques which can be used for pre-processing and cleaning data?
- What are the different types of recommender systems to be used to recommend the movie to the user accurately?
- What are the different algorithms which can be used in recommender systems?
- How can we apply different algorithms using different kinds of techniques on complete or partial dataset for getting the best results?

To answer above queries, we proposed few techniques for data pre-processing, algorithms for recommender system by using partial and complete MovieLens dataset

1.2 Problem Statement

Recommender Systems (RS) are software tools which give suggestions to customers about particular products. These recommendations are generally based on users' preferences, profiles, and past interactions between users and products. Although different types of recommender systems assist users in their decision-making process, researchers are still striving to design more effective recommendation engines using various algorithms, hybrid methods, and optimization techniques to enhance user satisfaction.

While numerous studies have evaluated Collaborative Filtering (CF) on complete datasets, far fewer have systematically investigated how partitioning datasets based on demographic attributes, such as age and occupation, affects system performance. These attributes were chosen because prior literature suggests they often correlate strongly with user preferences in domains such as movies and e-commerce. However, existing schemes frequently fail to account for the unique behavioral patterns within such subgroups. Moreover, when demographic partitions are used, smaller data subsets may lead to reduced accuracy, while ignoring them may overlook subgroup-specific recommendation patterns.

Our work addresses this gap by directly comparing the performance of User-Based Collaborative Filtering (UBCF) and Model-Based Collaborative Filtering (MBCF) on both complete and partitioned datasets from the MovieLens 1M dataset. By evaluating Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) across both scenarios, we aim to quantify the trade-offs between leveraging a larger, more diverse dataset versus tailoring recommendations to specific demographic segments. This dual evaluation provides a deeper understanding of whether demographic-based partitioning is beneficial and under what conditions it can degrade performance. In this study we tried to:

"Evaluate the Performance of Collaborative Filtering (CF) Recommender System on Complete and Partitioned Dataset using Root Mean Squared Error (RMSE) as well as Mean Absolute Error (MAE)"

1.3 Research Questions.

- RQ1: How can different algorithms be used to improve the performance of a recommendation system?
- RQ2: How does splitting the dataset into different partitions influence the performance of

CF Recommender System ?

- RQ3: How does UBCF and MBCF be compared in terms of MAE and RMSE using complete and partitioned MovieLens dataset?
- RQ4: What are potential areas for further improvement in recommender systems beyond collaborative filtering approaches?

The Introduction chapter will explore the origin, functions, and several different types of recommendation systems. This concept will be better understood if the different types of recommendation system and their examples are examined in depth. Experimental study has been conducted for the execution and evaluation of different types of Recommender Systems using Collaborative Filtering techniques. At the end of this chapter, we will also see how the whole study is ordered , as well as the contents covered in each of the succeeding sections.

This study introduces a comparative demographic-aware evaluation framework, in which collaborative filtering techniques based on user and model are rigorously tested not only on the full MovieLens 1M dataset, but also on strategically partitioned subsets based on age and occupation. The goal is to uncover how different algorithms handle varying levels of sparsity and subgroup diversity, thus providing operational insights for targeted recommendation strategies.

Main aim of recommendation system is to create an objective function which may be a mathematical model for end users and certain items, commodities or businesses [12]. The term "objective function" always refers to the function to be maximized or minimized in the specific optimization problem [13]. The objective function is shown in Figure 1.

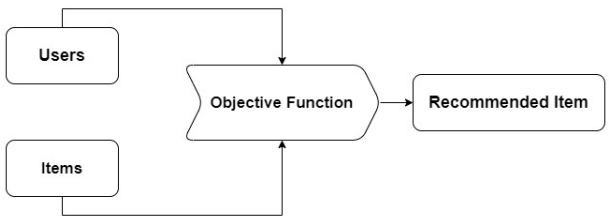

Figure 1. Objective function.

Recommender systems are often categorized into three main types [14, 15] which are further divided according to the requirements: Collaborative Filtering (CF), Content-Based Filtering (CBF), and Hybrid Recommendation Systems. In this article we will focus on Collaborative Filtering.

1.4 Collaborative Filtering Recommendation System.

The idea behind Collaborative Filtering is simple: Given a user's ratings, find all the users comparable to the active user who had similar views in the past, and then unknown products, about which active user has no ratings, will be predicted. Considering the preferences of neighbours, these unknown products are rated by neighbours but not by the active user [10]. The resemblance in the users' earlier rating is used to compute the similarity of two users [16]. Collaborative Filtering Recommendation System is depicted in Figure 2.

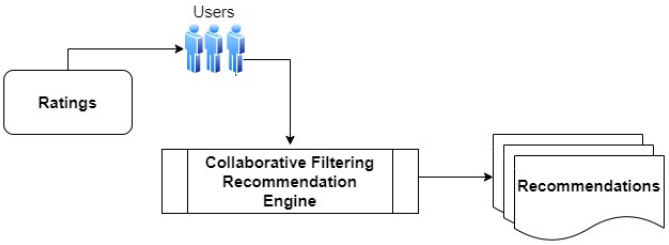

Figure 2. Collaborative filtering recommendation system.

Researchers have previously divided CF systems into following two categories [17] which are (i) Memory-based and (ii) Model-based Collaborative Filtering Recommendation System.

(i) Memory-based techniques calculate the similarity between users or items using historical user rating data [2]. It is further divided into the following two types:

- **User-Based Collaborative Filtering (UBCF):** User Based Collaborative Filtering recommender systems are based on the straightforward idea that if one person likes one thing, he/she will probably enjoy something similar, and if two users are similar, they will probably be interested in things that are similar to them. Unrated items are then offered to a connected user after the UBCF analyses the user commonalities [18, 19].
- **Item Based Collaborative Filtering (IBCF):** The concept is the similar as it was used in UBCF, but in IBCF starting from a specific movie or set of movies, we locate similar films based on user preferences. IBCF assesses product similarity and then advises the current user to go for the unrated comparable item.

(ii) The goal of a model-based algorithm is to condense a large database into a model. When the rating matrix grows to a huge size and there are so many people utilising the system, It faces a severe issues in recommending items. As a result of heavy resource consumption and decreased system performance, the system is unable to rapidly reply to user requests. Such issues are intended to be resolved via model-based approach. Several models, in this regard, have been developed. The following few models are elaborated here.

- **Matrix factorization-based CF:** In terms of matrix algebra, matrix factorization refers to methods for analyzing and utilizing rating matrices. Matrix factorization- based CF specifically tries to achieve two objectives. The initial objective is to minimize the rating matrix's dimensions. The second objective is to identify prospective qualities that may be included in the rating matrix as these traits would be used to make recommendations. Latent Semantic Analysis (LSA), Singular Value Decomposition, Latent Dirichlet Allocation (LDA), Principal Component Analysis (PCA) are few methods of matrix factorization [20].

- **Non-Negative Matrix Factorization:** Non-negative matrix factorization (NMF) is a matrix factorization technique that is often used in recommender systems. The goal of NMF is to factorize a given matrix into two non-negative matrices, typically representing users and items, respectively. This technique is used to discover latent features of the data that can be used to make recommendations to users [21]. In MBCF using NMF, the ratings matrix is decomposed into two matrices, namely the user matrix and the item matrix. The user matrix represents the latent factors that explain the user preferences, while the item matrix represents the latent factors that explain the item characteristics. The non-negativity constraint on the matrix elements ensures that the latent factors are non-negative and interpretable.

- **Clustering Model:** The basis for clustering CF [22] is the presumption that users who belong to the same group share a same interest and hence assess items similarly. As a result, users are divided up into clusters, which are collections of people with similar characteristics. Assume each user is represented by a rating vector$u_i = (r_{i1}, r_{i2}, \ldots, r_{im})$. The distance between two users serves as a measure of their dissimilarity. We have the options for measuring distance like Minkowski, Euclidian, Manhattan distance etc. [23].

- **Manhattan Distance.** The distance between two points measured along axes at right angles. In a plane with $p1$ at $(x1, y1)$ and $p2$ at $(x2, y2)$, it is $x1x2 + y1y2$. Higher dimensions may be derived from this with ease. When wires are only parallel to the X or Y axis in integrated circuits, Manhattan distance is frequently employed [24].

- **Minkowski Distance.** In order to measure the Minkowski distance, the expression for two variables a and b can be written as:

$$\left(\sum_{i=1}^{n} |a_i - b_i|^q\right)^{1/q} \tag{1}$$

It will represent the Manhattan distance when $q = 1$, and the Euclidean distance when $q = 2$. Although q can be any real value, it is typically set to a value between 1 and 2. In the above formula, when q is less than 1, it does not create a valid distance metric as the triangle inequality is not satisfied [25].

Two phases are involved in CF clustering: First, grouping people into clusters, with rating values present in every cluster. Every cluster produced by the k-mean technique, for instance, has a mean that is a rating vector similar to a user vector. second, the user who needs to be suggested is allotted a certain cluster and has ratings that are identical to the user of that cluster. Of course, the distance between a user and a cluster is used to determine how to assign a user an appropriate cluster. So how to divide users into clusters is the most crucial stage. Many clustering methods exist, including k-mean and k-centroid. The k-mean method is the most often used clustering algorithm [25–28].

This research article includes an introduction, literature review, methodology, experimental design and analysis, and conclusion. The introduction has provided an overview of the research topic, while the literature review analyzes existing research on the topic. The methodology chapter will describe how the research will be conducted, and the experimental design and analysis chapter will present the results. In the conclusion section we summarize the key findings, their implications and future scope in this research field.

2 Literature Review

A recommendation system is envisioned as a method to assist in decision making, particularly for users who have so far encountered a complicated environment for information processing. So many research related work has been done and is still in pipeline in this regard. The related work on recommendation systems [29] and user experience [30] that many researchers have utilized machine learning, deep learning algorithms by various combinations and strategies for enhancing the accuracy in predicting movies and other media related products e.g., news [31]. The subsequent part will delve into this matter. We will also talk about the issues that still need to be resolved. In this study, Ahuja et al. [32] developed a K-Means Clustering and KNN algorithm-based movie recommendation system. In the suggested approach, the RMSE value drops as the number of clusters does. The optimum RMSE value was found to be 1.081648. Katarya et al. [33] used k-means clustering and cuckoo search optimization method on MovieLens dataset. The suggested system, K-mean Cuckoo, results with 0.68. MAE. The research work of Alam et al. [34] consists of two main components. In the first part, performance analysis of BayesNet, Decision Table, Random Forest etc was done using Accuracy, RMSE, MAE, F Measure, etc. In the second part, they used the WEKA tool on ten different algorithms and found the best accuracy of 99.39 percent by using random forest and JRip classifiers. In this study Singh et al. [35] worked for both item and user-based similarity Collaborative Filtering. They looked for the most precise combinations of prediction techniques with lowest RMSE is 0.0762 for PC.

Zhang et al. [36] developed a novel collaborative filtering method called Weighted KM-Slope-VU for quick and scalable movie recommendations. They launched a customized movie recommendation site MovieWatch, to gather real- world user feedback. They used the virtual leader item opinion matrix instead of original complete rating matrix. Comparison of the accuracy of weighted KM-Slope-VU based on user profile and raw rating matrix on MovieLens 100K dataset was performed which shows the lowest RMSE value 0.9419 for user profile and 0.9433 for rating clusters. Zarzour et al. [37] used clustering and dimensionality reduction methods with both the Singular Value Decomposition (SVD) and k-means approach. Results showed that the k-means SVD based suggestion has the lowest RMSE value 0.618 and 0.621 for with 50 neighbors. Lund et al.[38] suggests a deep learning technique based on autoencoders. They compared this deep learning approach to k-nearest neighbor and matrix-factorization. They used the Netix dataset to test deep learning recommender system, and it produced a score of 0.782 RMSE, which is less than MudRecS on the Netflix dataset. Hans Byström [39] used the hetrec2011-movielens-2k dataset, in order to construct soft-max regression classification and k-means clustering for movie recommendation. For a high number of classes (k = 100), the softmax predictor demonstrated good processing speed. This approach performs better.

Gupta et al. [40] evaluated the two approaches Item-based collaborative filtering (IBCF) and User-based collaborative filtering (UBCF) using Movie-Lens benchmark dataset. On three different datasets, 10M, 1M and 100K, UBCF performs better with lowest RMSE values for 100K, 1M and 10M are 1.0906673, 1.0906673 and 1.0543233 respectively. Guan et al. [41] put forward a new matrix factorization method called Enhanced Singular Value Decomposition (ESVD). On the renowned Netflix and MovieLens data sets, when N D 20 percent, the ESVD for the MovieLens 100K dataset reach 0.9570, which lowers the RMSE by 0.0139 compared to the RSVD of 0.9709 and increases precision. Ayyaz et al. [42] employed threshold-based and k-nearest-neighbor methods on 1M dataset. The Root Mean Squared Error for Correlation, Log Likelihood, Euclidean and Cosine, is determined by picking 2 to 50 random number of user's clusters from. The best expected RMSE value obtained with Euclidean at cluster 50 is 1.05 When compared. Isma'il [43] developed a unique recommendation system that suggest which course a potential applicant should apply for and be provided by the institution. To train and evaluate the algorithm, 8,700 data points spanning three sessions were gathered from two separate universities. Naive Bayes, K-Nearest Neighbor, Linear Regression, Support Vector Machine, and Decision Tree Algorithm, were used and RMSE for the linear regression model is 2.64e-14. In order to improve the accuracy.

Sallam et al. [44] used Item based and singular value decomposition based (SVD) collaborative filtering methods on "Large Scale Arabic Book Reviews" (LABR). SVD based CF produced better recommendations than item-based CF; their respective RMSE values were 1.0187 and 1.1969. Additionally, they obtained MAE results of 0.8077 and 0.922, respectively. Ramni Harbir Singh explained the deep learning architecture recommendation system, which

uses Cosine Similarity (CS) and (CBF) Content-based filtering to forecast the outcome. The K-NN with cosine similarity was used and provided more accuracy than the others. Kuzelewska [45] built a system in Apache Mahout environment on 100K MovieLens dataset. It produced values of RMSE 1.05 using 50 clusters with Euclidean distance, RMSE 1.05 using 5 clusters with Euclidean distance and RMSE 1.19 using 02 cluster with cosine similarity. A new hybrid recommendation system AIMED was proposed by Hsu et al. [46] that user attributes like Demographic data, Activities, Experiences, Interests and Moods, which are the foundation of AIMED. With various hidden layer counts from 5 to 12, and achieved the maximum accuracy of 76.6. Ma [47] developed a multi-objective optimization (MO) and Kernel density estimation in UBCF are used on Netflix data set. It is observed that MO-KUCF produced the least MAE mean average error 0.70 with 40 number of items. A mathematical model of the intelligent recommendation system based on association rules was created by Pandya et al [48]. Five separate subsets, Q1 through Q5, are created at random from the test dataset. The five subsets are run using the collaborative filtering, the classic content-based and the traditional association rule recommendation system. Lowest MAE is that of the intelligent recommendation system, which is maintained at about 0.73, with the highest accuracy.

Agrawal [49] proposed a Hybrid approach using genetic algorithm and Support Vector Machine as a classifier. It shows highest accuracy 90 percent as compared to collaborative and content-based filtering. Ponnam et al. [50] used IBCF on 1M dataset with modified cosine similarity. The MAE evaluation of the suggested strategy produced a score of 0.938 with 20 neighbors. Shah et al. [51] used IBCF and developed a book recommendation system using goodbooks10k dataset. Similarity measures based on cosine distance have been used to compare literature. In terms of MAE, the experimental findings attained 0.72 value. Detecting Hepatitis early is critical to save lives. Different traditional medical test like Abdominal Ultrasound, Liver Biopsy, and LFT are used for diagnosis. However, with advancements in AI and machine learning algorithms, medically intelligent systems can be developed for early prediction of fatal liver diseases. In a study, M. Ahmad Hafeez [52] analyzed various ML algorithms and found that Support Vector Machine had the highest accuracy of 91.84% in predicting deadly liver diseases. Early prediction of this deadly liver disease will help in recommending the appropriate treatment in order to save precious human lives. Liu et al. [53] used Deep Learning methods on the good- books 10k dataset. RMSE was used to assess the suggested method. In terms of RMSE, it received 1.1426. A web-based movie recommendation system was proposed by [54] that used K-Nearest Neighbor (KNN), Alternating Least Squares (ALS), SVD, and Restricted Boltzmann Machines. SVD produced more accurate suggestions than others. In terms of RMSE and MAE, SVD obtained 0.9002 and 0.6925, respectively. Raghuwanshi et al. [55] used Accelerated Singular Value Decomposition on Yahoo Movie, Film Trust and 100K MovieLens. The suggested method was assessed using RMSE and MAE. The results of the experiments demonstrated that the suggested ASVD performed better than previous SVD models. Researchers found that different recommendation systems are using various machine learning algorithms to enhance accuracy in predicting movies and other products. Spe- cific examples of research work includes K-Means Clustering, KNN algorithm-based movie recommendation systems, cuckoo search optimization-based systems, a novel collaborative filtering method called Weighted KM-Slope-VU and many more. Different evaluation approaches, such as item-based, user-based and hybrid collaborative filtering were used in order to evaluate various recommender systems. So, in this study we will evaluate the performance of Col- laborative filtering Recommender systems using User-Based and Model-Based algorithms on complete and partitioned Movielens.

While modern deep learning-based recommender models such as NCF, AutoRec, LightGCN, and BERT4Rec offer enhanced capabilities, our study focuses on classical collaborative filtering methods to ensure interpretability, computational simplicity, and clear insights into the effect of demographic-based dataset partitioning - serving as a foundational benchmark for future integration of advanced approaches.

3 Proposed Methodology

MovieLens dataset, which comprises one million ratings, will be used in this research to examine two different types of collaborative filtering approaches. These two methods are user-based collaborative filtering with cosine similarity function, Euclidean Distance Function and Model-based collaborative filtering using Non-negative Matrix Factorization and

Singular Value Decomposition (SVD). Before making any modifications to the data, some preprocessing steps must be done. The preprocessed dataset will then be divided into different portions. RMSE and MAE will be used to assess the proposed mechanism's accuracy. Methodology that will be followed in this research work is shown in Figure 3.

3.1 Dataset

The MovieLens datasets [56] are frequently employed in academia, business, and research. They are downloaded thou- sands of times annually, which is a reflection of their usage in software, online and traditional courses, and programming books from the popular press. These datasets are the result of members' engagement on the active research platform MovieLens, which has hosted several experiments since its introduction in 1997. Different sizes are present in the MovieLens databases. Each dataset stands alone; the smaller dataset is not a portion of the larger one [9]. The 1M dataset, which was published in 2003, is the one we shall utilize for this thesis. The 1M MovieLens dataset will be used for both training and testing. There are three files that make up the MovieLens-1M dataset: users.dat, ratings.dat, and movies.dat.

3.1.1 Rating file description.

The file "ratings.dat" has all the ratings in the following format:

- **UserIDs** range from 1 through 6040, every user has minimum of 20 ratings.
- **MovieIDs** range from 1 through 3952.
- **Ratings:** The ratings are provided on a scale of 1 to 5, with 5 being the highest rating (whole-star ratings only).
- **Timestamp** field represents the time at which the rating was given, and it is measured in seconds.

3.1.2 Users file description.

This file contains information about 6,040 users who rated movies in the dataset. Each row in the file represents a single user, and there are five fields separated by double colons "::". The fields are as follows:

- **UserID:** A unique identifier for each user, ranging from 1 to 6040.
- **Gender:** M" for male and a "F" for female are used to indicate gender
- **Age:** The following ranges are specified for the age: 1:"Under 18", 18:"18-24", 25:"25-34",35:"35-44", 45:"45- 49", 50: "50-55", 56: "56+".
- **Occupation :** It is specified as per the following: "0:"other or not specified" 1:"academic/educator", 2:"artist", 3:"clerical/admin", 4:"college/grad student", 5:"customer service", 6:"doctor /health care", 7:"executive/managerial", 8:"farmer", 9:"homemaker", 10:"K-12 student", 11:"lawyer", 12:"programmer", 13:"retired", 14:"sales/marketing", 15:"scientist", 16:"self-employed", 17:"technician/engineer", 18:"tradesman/craftsman", 19:"unemployed", 20:"writer"”
- **Zip-code: :** A string representing the zip code of the user's location, which is not necessarily their current location.

3.1.3 Movies File Description.

This file contains information about 3,706 movies that were rated in the dataset. Each row in the file represents a single movie, and there are three fields separated by double colons ("::").

- **Movie ID:** A unique identifier for each movie, ranging from 1 to 3952. Note that there are only 3,706 movies in the movies.dat file, which means that some movie IDs in the ratings.dat file may not be present in this file.
- **Title** the same as those supplied by IMDB and year of release are also included.
- **Genres** are from the following which are pipe- separated: Fantasy, Action, Western, Children's, Comedy, Documentary, FilmNoir, Musical, Drama, Mystery, Crime, Romance, Sci-Fi, Animation, Thriller, War, Adventure, Horror.

In this study, "age" and "occupation" were selected as partitioning criteria due to their strong correlation with media consumption patterns, as documented in prior recommender systems research. Both attributes are well-represented in the MovieLens 1M dataset, ensuring sufficient sample sizes in each partition to avoid excessive sparsity. Alternative demographic or contextual attributes in the dataset were either absent, too sparse, or not directly relevant to the collaborative filtering approaches under investigation and thus were excluded to maintain experimental validity.

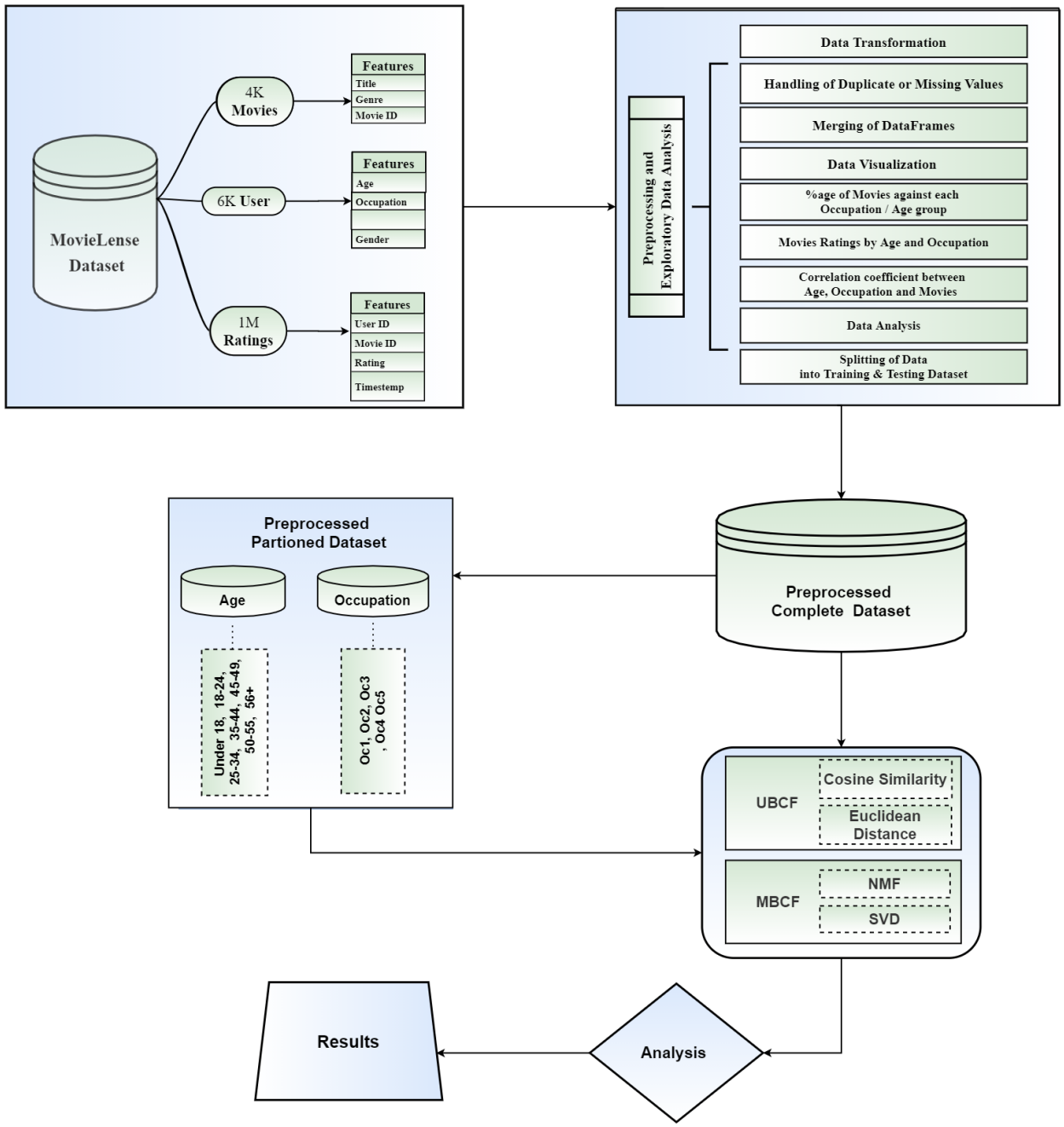

Figure 3. Collaborative filtering recommender system using partitioned and complete 1M movieLens dataset.

3.2 Preprocessing and Exploratory Data Analysis (EDA)

3.2.1 Data cleaning and preparation.

Before performing exploratory analysis on the 1M Movielens dataset, performing some data cleaning and preparation steps are of ultimate importance.

- **Duplicates and Missing Values:** The dataset was checked if there may be any duplicate rows which was removed accordingly. The dataset's missing values were also examined by researchers and were handled correctly.
- **Merge dataframes:** Data in 1M MovieLens dataset is spread across multiple data frames. To perform any analysis or modeling that involves both user and movie information, we need to merge the 'ratings.dat' file with either the 'users.dat' file or the 'movies.dat' file. We can do this by using the common keys user_id and movie_id. Similarly, we can merge the ratings.dat file with the movies.dat file using the common key movie_id. These data cleaning and preparation steps will help us to have a clean and structured dataset that can be easily used for exploratory

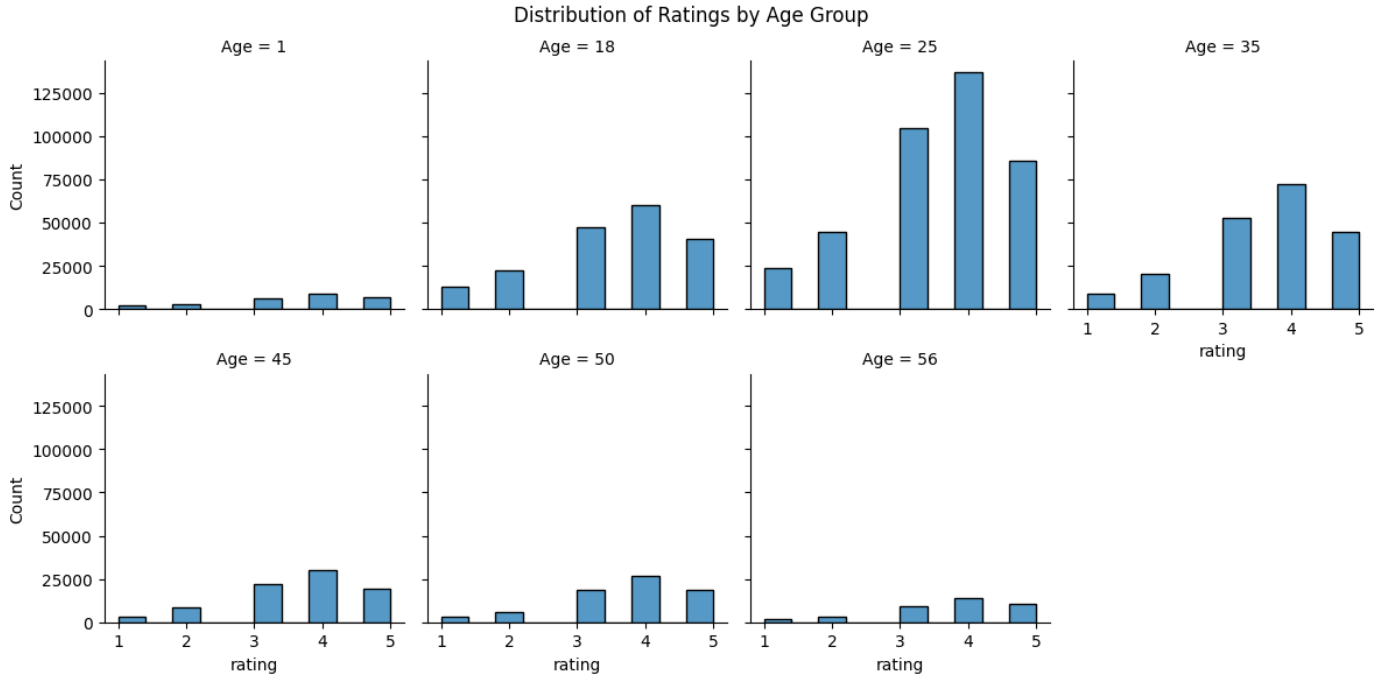

Figure 4. Distribution of ratings by Age group.

analysis.

3.2.2 *Data Visualization*

Visualizing the data is an important step in exploratory data analysis. Here are some ways researchers used to visualize the 1M Movielens dataset:

- **Distribution of Movie Ratings:** Using Matplot lib, a popular data visualization library for Python, histogram is drawn to visualize the distribution of movie ratings. This histogram gives insights into the characteristics of the dataset that how the ratings are distributed among the movies. It also gives a better understanding of the ratings data in the dataset, which can be useful for making data-driven decisions and predictions.
- **Distribution of ratings by age group:** Bar plot is drawn to visualize the top most age group based on the number of ratings. New DataFrame will be created by merging the users DataFrame and the ratings DataFrame on the 'userid' column as shown in Figure 4.

  Figure 4 Facet Grid plot shows the distribution of movie ratings by age group which will be useful for identifying any age-related patterns or trends in movie preferences and may help with targeted marketing or promotional efforts.
- **Top 10 most popular movies:** Bar plot is used to visualize the top 10 most popular movies based on the number of ratings. It groups the merged DataFrame by movie title and counts the number of occurrences of each title. It sorts the resulting DataFrame by the count of occurrences in descending order and selects the top 10 movies.

  Figure 5 identifies the most popular movies in the dataset by counting the number of occurrences of each movie title and selecting the top 10 movies based on this count. The resulting Data Frame is used for gaining insights into the most popular movies in the dataset and their relative popularity compared to other movies.
- **5 most active occupations:**We will draw a bar plot to visualize the top 5 most active occupations based on the number of movies watched.

  The bar graph in Figure 6 identifies top 5 most active occupations in watching movies in the dataset. This will give insights into the movie-watching habits of different occupations and identifying potential target audiences for marketing or promotional purposes.

3.2.3 *Data Analysis*

Performing data analysis provides us with valuable insights into the data. Here are some analyses which has been performed on the dataset: Histogram of distribution of movie ratings plotted in data visualization step gives us the information that the majority of the ratings in the dataset are between

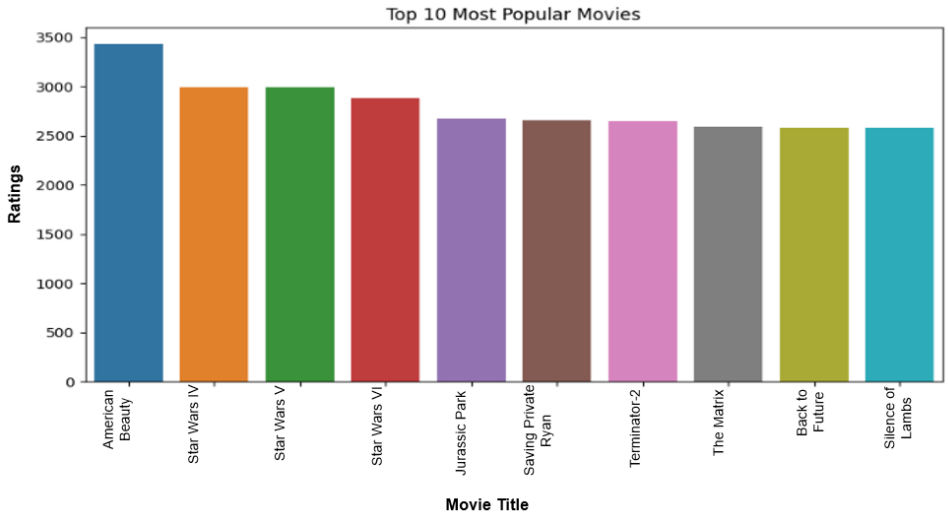

Figure 5. Top 10 most popular movies.

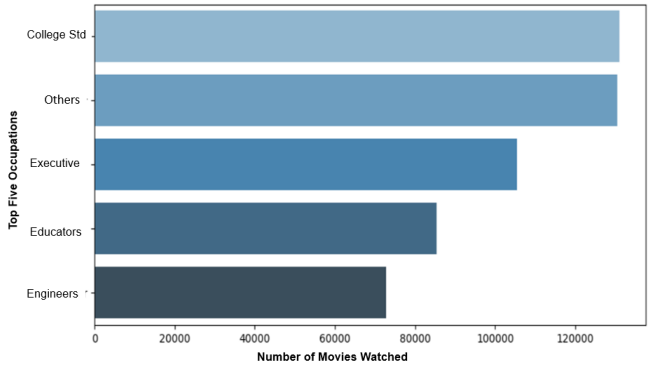

Figure 6. Top five most active occupations.

3 and 4, with the most frequent rating being 4. The distribution is slightly left-skewed, indicating that users tend to rate movies higher than lower. Overall, the distribution of movie ratings in the 1M Movielens dataset is fairly normal with a mean rating of approximately 3.5. We calculated the number of ratings for each movie and sorting them in descending orde, r to determine the most popular movies. Finding the average rating for each movie and sorting them in descending order would find the highest rated movies. For calculating the average rating by occupation. Average rating for each occupation was calculated and plotted over time to see how the average rating changes over the occupation. Top-rated movies by genre could be found by calculating the average rating for each movie in each genre and selecting the top-rated movie for each genre.

To determine if there are any correlations between movie ratings and age groups, we calculated the average rating for each age group which showed that the highest average ratings came from users in the age group of 18-24 years old. Similarly, to see correlations between movie ratings and user occupations, we calculated the average rating for each occupation and plotted them. The result showed that the highest average ratings came from users in the occupation of "retired", followed by "homemaker" and "doctor/healthcare". The lowest average ratings come from users in the occupation of "student", "sales/marketing", and "programmer". However, it is important to note that there is not a significant difference in the average ratings between different occupations. Therefore, it can be concluded that there is no strong correlation between movie ratings and user occupations in the 1M Movielens dataset These analyses provided us with insights into the data and helped to make decisions about the dataset.

3.2.4 Partitioned Dataset

Partitioning the 1M MovieLens dataset on the basis of age, occupation, gender, and other demographic factors can be useful for analyzing and provides insights into how different groups of users interact with movies differently and help to inform targeted marketing and recommendation strategies. In our case, we specifically use the partitioning based on age and occupation. The results obtained from these can be generalize for other demographic factors. In this study we will use age and occupation portion of the 1M dataset which have already been described under the heading of dataset.

3.3 User-Based Collaborative Filtering(UBCF)

UBCF is a popular recommendation technique that relies on finding similar users and recommending items that similar users have liked. UBCF has already

been explained in detail in chapter 1. Different similarity measures can be used in UBCF and we applied the following two in this regards.

3.3.1 UBCF with Cosine Similarity Function

UBCF with cosine similarity is a simple and effective recommendation technique that computes a matrix of user-item ratings, where each row corresponds to a user and each column corresponds to an item. The cells of the matrix represent the ratings that users have given to items. Cosine similarity between every pair of users based on their rating profiles will be calculated. The dot product is crucial for judging similarity. The ratio between product of the magnitude of two vectors and dot product of x and y and are used to measure the similarity between the given two vectors.

$$Cos\theta = \frac{\sum_i^n (g_i)(h_i)}{\sqrt{\sum_i^n (g_i)^2} \sqrt{\sum_i^n (h_i)^2}} \quad (2)$$

If the two vectors are same or identical then it will be equal to 1 and 0 if they are orthogonal. UBCF with cosine similarity for predicting ratings can be an effective technique for filling in missing ratings in a dataset.

3.3.2 UBCF with Euclidean Distance Function

UBCF (User-Based Collaborative Filtering) using Euclidean Distance Function for rating purposes is used in recommendation systems to recommend items to a user based on the preferences of other similar users. In this method, the similarity between two users is calculated using the Euclidean distance between their rating vectors. The following is equation of Euclidean similarity:

$$Euclidean\ dis(a,b) = \sqrt{\sum_{i=1}^{m} (a_i - b_j)^2} \quad (3)$$

and

$$Euclidean\ sim\ (a,b) = \frac{1}{\frac{1}{(1+Euclidean\ dis\ (a,b))}} \quad (4)$$

where a and b denotes two points. the value of Euclidean similarity will be between 1 and 0. After calculating the similarity between all pairs of users, it identifies the most similar users to a target user. Then items are recommended to the target user that are highly rated by most similar users but not rated by the target user. Overall, UBCF using Euclidean Distance Function for rating is an effective approach to make personalized recommendations in recommendation systems.

3.4 Model-Based Collaborative Filtering

MBCF is an approach used in recommendation systems that uses machine learning models to make personalized recommendations to users based on their past behaviors or preferences. MBCF uses statistical or machine learning models to learn the underlying patterns or relationships in the rating data and to make predictions for new user-item pairs.

3.4.1 MBCF with Non-Negative Matrix Factorization (NMF)

Non-negative Matrix Factorization is a variant of matrix factorization that imposes the constraint that the factor matrices contain only non-negative values. In Model-Based Collaborative Filtering (MBCF) using NMF, the ratings matrix is decomposed into two matrices, namely the user matrix W and the item matrix H. The user matrix represents the latent factors that explain the user preferences, while the item matrix represents the latent factors that explain the item characteristics. The non-negativity constraint on the matrix elements ensures that the latent factors are non-negative and interpretable. NMF uses items, users, and their ratings for these items. $R[i,j]$ represents the rating of user i for item j, and then it will be factorized into two matrices, W and H, such that: $R = W \cdot H$. Here, W represents the user matrix, and H represents the item matrix. The number of latent factors is chosen by the user as a parameter.

The factorization is performed using an optimization algorithm that minimizes the reconstruction error between the original ratings matrix R and the approximated ratings matrix R', which is given by:

$$R' = W \cdot H \quad (5)$$

The optimization algorithm used in MBCF using NMF is typically based on gradient descent, and it updates the matrix elements of W and H iteratively until convergence. The non-negativity constraint on the matrix elements ensures that the latent factors are non-negative and interpretable.

Once the user matrix and item matrix are computed, it finds the latent factors that represent the user's preferences in the user matrix, and then identifies the items that have high values for these latent factors in the item matrix. Then it will recommend these items

to the user. Overall, MBCF using NMF is an effective approach to making personalized recommendations in recommendation systems.

3.4.2 MBCF with Singular Value Decomposition (SVD)

SVD is a powerful matrix factorization technique that decomposes a matrix into three different matrices: U, W, and V^T. SVD has significant geometrical and theoretical insights regarding linear transformations, making it useful in many fields, including data analysis and machine learning. The SVD of an $m \times n$ matrix M can be represented as:

$$M = U \cdot W \cdot V^T \tag{6}$$

where U is an $m \times n$ matrix of the eigenvectors of the product of M and its transpose (MM^T), V^T is a transpose of an $n \times n$ matrix containing the orthonormal eigenvectors of the transpose of M (M^TM), and W is an $n \times n$ diagonal matrix containing the singular values.

To calculate U, we first have to find the eigenvectors of MM^T. Eigenvectors are vectors that remain in the same direction after a linear transformation. In other words, when we multiply a matrix by its eigenvector, the resulting vector is a scaled version of the original eigenvector. We can find the eigenvectors of MM^T by solving the characteristic equation:

$$\det(MM^T - \lambda I) = 0 \tag{7}$$

where λ is the eigenvalue, I is the identity matrix, and det() is the determinant function. The resulting eigenvectors of MM^T are the columns of U.

To calculate V^T, we need to find the eigenvectors of M^TM. Similarly, we can find the eigenvectors of M^TM by solving the characteristic equation:

$$\det(M^TM - \lambda I) = 0 \tag{8}$$

The resulting eigenvectors of M^TM are the columns of V. However, since V is an orthonormal matrix, we need to normalize its columns to have unit length. To calculate the singular values, we need to take the square root of the eigenvalues of MM^T or M^TM, which are the same. The singular values are arranged in decreasing order along the diagonal of W. The SVD provides significant insights into the structure of a matrix and can be used for various applications, such as dimensionality reduction, recommendation systems, and image compression. For the SVD implementation, the number of latent factors was set 50, with a learning rate of 0.005 and regularization parameter of 0.02, as these values provided a good balance between convergence speed and accuracy during preliminary trials. For NMF, we used 50 components with a maximum of 200 iterations, which ensured stable convergence without overfitting. These parameters were chosen following commonly reported configurations in the literature and validated through small-scale grid search experiments on a subset of the MovieLens dataset.

Additionally, All experiments were implemented in Python using the Surprise and Scikit-learn libraries. The experiments were conducted on a system with Intel Core i7 (12th Gen) processor, 32GB RAM, and NVIDIA RTX 3060 GPU.

- Training UBCF with Cosine and Euclidean similarity required 45 seconds and 50 seconds, respectively, for the complete 1M dataset.
- Training MBCF with NMF required 2.5 minutes, whereas SVD required 3 minutes due to iterative matrix factorization.

These details highlight the computational efficiency of user-based methods and the slightly higher cost of model-based methods in exchange for improved accuracy.

3.5 Evaluation Metric

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are commonly used evaluation metrics in recommendation systems to measure the performance predictions made by the different machine learning techniques and models.

3.5.1 Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in the predicted ratings, where the error is the absolute difference between the predicted rating and the actual rating. The equation for MAE is:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{9}$$

where y_i represents the predicted value, x_i shows the actual values, and n is the total number of data points.

3.5.2 Root Mean Square Error (RMSE)

RMSE measures the average magnitude of the squared errors in the predicted ratings, The equation for RMSE

can be written as:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y'_i - y_i)^2}{n}} \quad (10)$$

where y'_i represents the predicted values, y_i represents the observed values, and n is the number of observations. Both MAE and RMSE have their advantages and disadvantages. MAE is more interpretable since it directly measures the average absolute error, while RMSE penalizes large errors more heavily due to the squared term. However, RMSE is more sensitive to outliers since it squares the errors. In general, both metrics should be used together to provide a more comprehensive evaluation of the model's performance.

The study design provided an in-depth description of 1M Movielens dataset, data preprocessing and exploratory data analysis which includes handling of duplicate and missing values, merging of dataframes, data visualization and detailed analysis of the data. Moreover, the models utilized in the research, the essential functions and their associated processes were extensively discussed. In the next chapter, User-base Collaborative filtering (UBCF)with cosine and Euclidean distance function and Model-base Collaborative filtering (MBCF) with Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization on complete and partitioned dataset will be applied. Performance on the basis of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), will be analyzed and comparative results will be drawn.

4 Experimental Design, Analysis And Results

The objective of this study is to evaluate the performance of Model-based and user-based collaborative filtering recommender system on entire and partitioned datasets of the whole. Researchers used user-based collaborative filtering with cosine and Euclidean distances function. For model-based collaborative filtering we applied Non-Negative Matrix factorization and Singular Value Decompositions (SVD) models on 1M movielense dataset. The following methodology was adopted to evaluate the recommendation algorithms. To evaluate the performance of the recommendation algorithm, we split the dataset into two parts: the training set and the testing set. We then applied the recommender algorithms to the training set and used it to predict the ratings for the testing set. The predicted ratings were compared with the actual ratings in the testing set, and MAE and RMSE were calculated to determine the level of error between the predicted and actual ratings.

Statistical significance: To verify whether these performance differences are statistically meaningful, paired t-tests were conducted for MAE and RMSE results across the algorithms. The difference in MAE between MBCF with SVD and MBCF with NMF was statistically significant ($p < 0.01$), as was the difference in RMSE ($p < 0.05$). Comparisons between SVD and both UBCF methods yielded p-values < 0.01 for both MAE and RMSE, confirming that SVD's superior performance is highly unlikely to be due to random variation.

For most age groups, SVD's MAE improvement over NMF and both UBCF variants was statistically significant ($p < 0.05$), with several cases reaching $p < 0.01$. However, in smaller partitions (e.g. Age group 46–56), the differences were not statistically significant ($p > 0.05$), likely due to the reduced sample size and lower statistical power.Statistical significance: For three of the five occupations, SVD's advantage over other methods in both MAE and RMSE was significant at $p < 0.05$. In two cases (Artist and Writer), while the trend favoured SVD, the differences did not reach significance, suggesting that the smaller number of ratings within those groups limited the ability to detect differences with high confidence.

4.1 Evaluating User Based Collaborative Filtering with Cosine Function

4.1.1 Partitions Based on Age

We divided the ratings by different age groups. While shifting the ratio of the training and testing dataset, we calculated the root mean squared error (RMSE) and mean average error MAE of user-based collaborative filtering UBCF with cosine similarity function for complete and partitioned datasets.

MAEs and RMSEs of UBCF with Cosine function for different age groups are given below. This group shows relatively lower MAE/RMSE compared to older groups due to higher rating density and more homogeneous preferences, enabling the similarity metric to perform better.

According to the results presented in Table 1 and Figure 7 the MAE and RMSE values for different age groups and a total of 1 million samples. MAE and RMSE values generally decrease as age increases, indicating more accurate predictions for older age

Table 1. MAE and RMSE of UBCF with Cosine on Complete 1M Ratings and Age-based Partition Dataset.

Sr No.	Data	MAE	RMSE
1	1M	0.7476	0.9397
2	Under 18 yrs	0.8771	1.103
3	18-24 yrs	0.7692	0.9669
4	25-34 yrs	0.7402	0.9338
5	35-44 yrs	0.7247	0.9161
6	45-49 yrs	0.7226	0.9168
7	50-55 yrs	0.7161	0.9148
8	56 yrs and above	0.7211	0.922

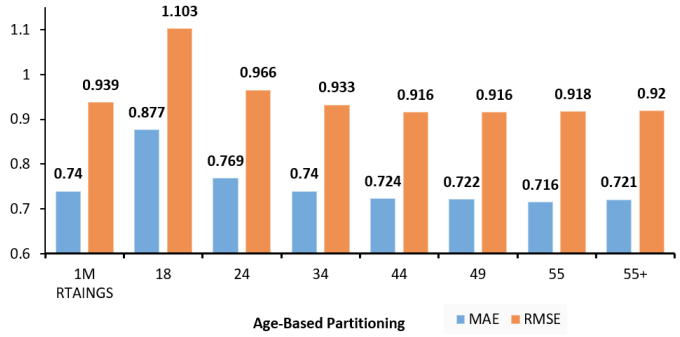

Figure 7. MAE and RMSE for User-Based Collaborative Filtering (Cosine Similarity) on the Age 18–35 partition of the MovieLens 1M dataset.

groups. The MAE and RMSE values are highest for the 18 & 24 age group and lowest for the 55 and above age group. The MAE and RMSE values for the total sample are generally lower than the values for any specific age group, suggesting better prediction accuracy when considering all age groups together.

4.1.2 Partitions Based on Occupation

The performance of user-based collaborative filtering with cosine similarity function was evaluated by partitioning the ratings based on Occupation. The level of errors MAE and RMSE was calculated by varying the percentage of the training dataset used in the evaluation process. MAEs and RMSEs of UBCF with Cosine function on different occupations are given below.

Table 2. MAE and RMSE of UBCF with Cosine on Complete 1M Ratings and Occupation-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.7476	0.9397
2	Other	0.7496	0.9459
3	College std	0.7775	0.9915
4	Academic	0.7778	0.9904
5	Engineer	0.7923	1.0093
6	Executive	0.9568	1.202

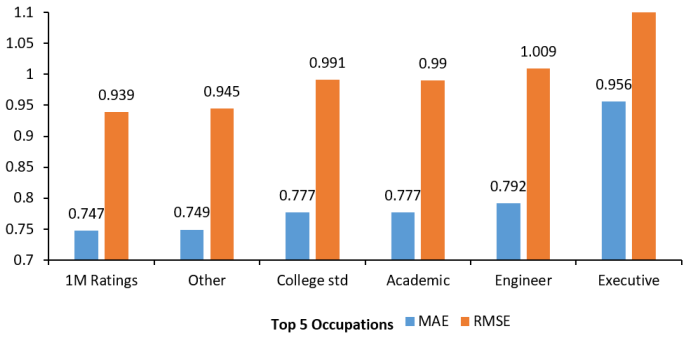

Figure 8. MAE and RMSE of user-based collaborative filtering (UBCF) using cosine similarity on the complete MovieLens 1M dataset and its occupation-based partitions.

According to the results presented in Table 2 and Figure 8 when comparing the MAE and RMSE values across different occupational categories, the analysis shows that the values are generally lower for the "Others" and College students categories and higher for rest of the occupational categories. This suggests that the model's predictions are more accurate for individuals with occupational backgrounds in the "Other". Overall analysis suggests that the model is generally accurate in predicting outcomes for a diverse range of occupational categories and a total of 1 million samples. However, the MAE and RMSE values for the entire dataset are generally lower than the values for any specific occupation.

4.2 Evaluating User-Based Collaborative Filtering with Euclidean Distance

4.2.1 Partitions Based on Age

Researchers calculated the root mean squared error RMSE and mean absolute error MAE of user-based collaborative filtering UBCF with Euclidean distance for complete and partitioned dataset. MAEs and RMSEs of UBCF with Euclidean distance for different age groups are given below.

Table 3. MAE and RMSE of UBCF with Euclidean Distance on Complete 1M Ratings and Age-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.7286	0.9236
2	Under 18	0.8458	1.0706
3	18-24	0.7642	0.9627
4	25-34	0.7287	0.9192
5	35-44	0.7153	0.9074
6	45-49	0.7041	0.8992
7	50-55	0.7193	0.9148
8	56 and above	0.7150	0.9205

According to the results presented in Table 3 and

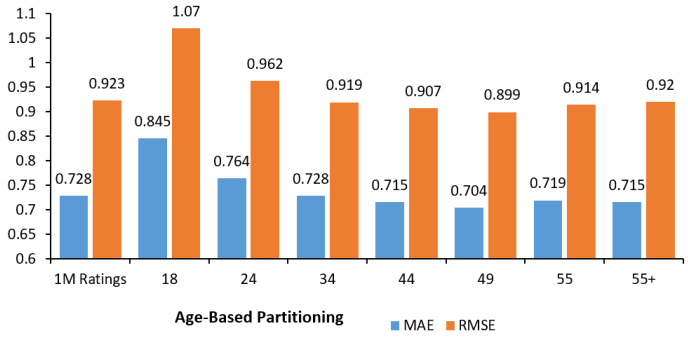

Figure 9. MAE and RMSE of UBCF with Euclidean Distance on Complete 1M Ratings and Age-based Partition.

Figure 9 when comparing the values for 1M and different age groups in the provided table, it is clear that the MAE and RMSE values are generally lower for the 1M group compared to the different age groups. This suggests that the predictions are more accurate for the 1M group compared to the various age groups. Specifically, the MAE value for the 1M group is 0.7286, which is lower than the corresponding values for all other age groups. This comparison suggests that the model performs better in predicting outcomes for the 1M group compared to the other age groups.

4.2.2 Partitions Based on Occupation

The performance of user-based collaborative filtering with cosine similarity function was evaluated by partitioning the ratings based on Occupation. The level of errors of MAE and RMSE was calculated by varying the percentage of the training dataset used in the evaluation process. MAEs and RMSEs of UBCF with Cosine function on different occupations are given below.

4.3 Evaluating Model-Based Collaborative Filtering With Non-Negative Matrix Factorization

4.3.1 Partitions Based on Age

Researchers calculated the mean average error MAE and root mean squared error RMSE of Model-based collaborative filtering MBCF with Non-Negative Matrix Factorization NMF for complete and partitioned dataset. MAEs and RMSEs of Model-based CF with Non-Negative Matrix Factorization for different age groups are given below.

According to the results presented in Table 4 and Figure 10, the error rate of MAEs and RMSEs for Model-based collaborative filtering MBCF with Non-Negative Matrix Factorization NMF is generally lower for the entire dataset as compared to the different age groups.

Table 4. MAE and RMSE of MBCF Using NMF on Complete 1M Ratings and Age-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.698	0.897
2	Under 18	0.7273	0.9201
3	18-24	0.7274	0.9208
4	25-34	0.7242	0.9173
5	35-44	0.7257	0.9192
6	45-49	0.7256	0.9184
7	50-55	0.7253	0.918
8	56 and above	0.7247	0.9183

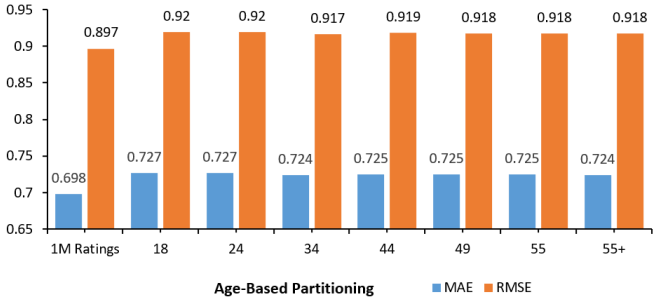

Figure 10. MAE and RMSE of MBCF Using NMF on Complete 1M Ratings and Age-based Partition.

4.3.2 Partitions Based on Occupation

MAE and RMSE of Model-based collaborative filtering with Non-Negative Matrix Factorization for complete and partitioned dataset on the basis of occupations are calculated and shown in the following Table 5.

Table 5. MAE and RMSE of MBCF Using NMF on Complete 1M Ratings and Occupation-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.698	0.897
2	Other	0.7127	0.9021
3	College std	0.726	0.9186
4	Academic	0.727	0.921
5	Engineer	0.7272	0.9202
6	Executive	0.7256	0.9173

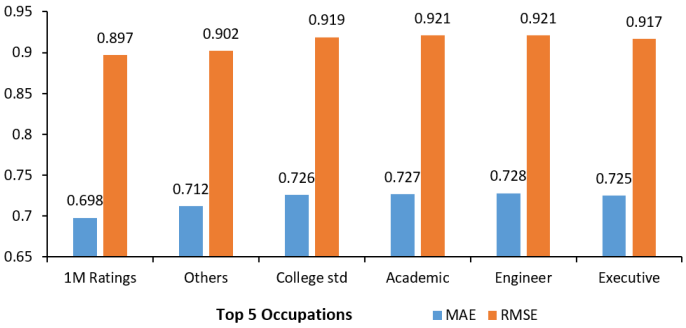

Figure 11. MAE and RMSE on Complete 1M Ratings and Occupation-Based Partition.

Results presented in Table 5 and Figure 11 show MAE and RMSE values for 1M complete and occupation based partitioned datasets. It clearly depicts that the predictions for 1M dataset has lowest error values for both MAE and RMSE.

4.4 Evaluating Model-Based Collaborative Filtering With Singular Value Decomposition

4.4.1 Partitions Based on Age

We divided the ratings by age categories and calculated the mean average error (MAE) and root mean squared error (RMSE) of Model-based collaborative filtering (MBCF) using SVD for complete and partitioned datasets, as shown in Table 6 and Figure 12.

Table 6. MAE and RMSE using MBCF with SVD on Complete 1M Ratings and Age-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.6909	0.8761
2	Under 18	0.6947	0.8839
3	18-24	0.6949	0.8837
4	25-34	0.6942	0.8838
5	35-44	0.6937	0.8841
6	45-49	0.6936	0.8834
7	50-55	0.6917	0.8803
8	56 and above	0.6937	0.8820

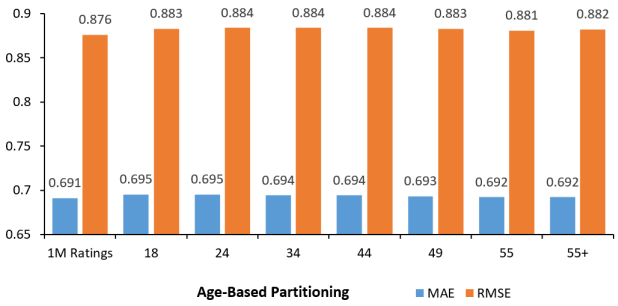

Figure 12. MAE and RMSE using MBCF with SVD on Complete 1M Ratings and Age-based Partition.

According to the results presented in Table 6 and Figure 12, in comparison to other age groups, the 1M dataset has lower MAE and RMSE values, indicating better performance in terms of prediction accuracy. The MAE and RMSE of 1M dataset are 0.6909 and 0.8761 respectively, which are the lowest among all the other age groups. This demonstrates that the MBCF using SVD algorithm performed better on the complete 1M dataset than on the age-based partitions.

4.4.2 Partitions Based on Occupation

We divided the ratings by occupation categories and calculated MAE and RMSE of MBCF model using SVD for complete and partitioned dataset.

Table 7. MAE and RMSE of MBCF Using SVD on Complete 1M Ratings and Occupation-based Partition.

Sr No.	Data	MAE	RMSE
1	1M	0.698	0.897
2	Other	0.7127	0.9021
3	College std	0.726	0.9186
4	Academic	0.727	0.921
5	Engineer	0.7272	0.9202
6	Executive	0.7256	0.9173

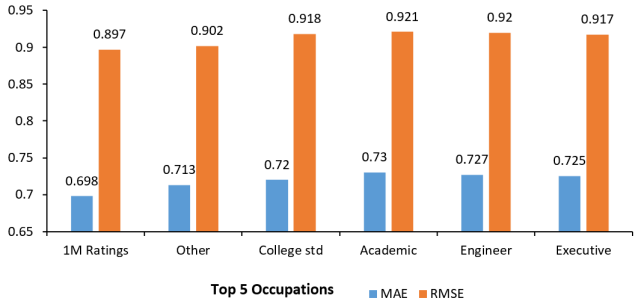

Figure 13. MAE and RMSE of MBCF Using SVD on Complete 1M Ratings and Occupation-based Partition.

According to the results presented in Table 7. and Figure 13 comparing the evaluation metrics of 1M with the rest of the partitions, we can see that the MAE and RMSE values of 1M are smallest than those of the other partitions. This suggests that the performance of this recommendation system on the 1M dataset is better than the other partitions.

In addition to reporting MAE and RMSE values, we conducted paired t-tests to evaluate whether the observed performance differences among the four algorithms were statistically significant. For the complete 1M Movielens dataset, the difference in MAE between MBCF with SVD and MBCF with NMF was found to be statistically significant ($p < 0.01$). Similarly, the difference in RMSE between these two methods was also significant ($p < 0.05$). Comparisons between SVD and both UBCF variants (Cosine and Euclidean similarity) yielded p-values below 0.01 for both MAE and RMSE, confirming that SVD's superior performance is unlikely to be due to random variation. These findings reinforce the robustness of our conclusion that MBCF with SVD offers the most accurate predictions among the tested approaches.

4.5 Comparative Analysis

In chapter four we have applied User based Collaborative Filtering with Cosine and Euclidean Similarities and Model Based Collaborative Filtering using NMF and SVD to calculate the MAE and

RMSE on 1M complete ratings as well as on age and occupation based partitions. All the four algorithms produced the least MAE and RMSE values on complete 1M ratings dataset as compared to different partitioned datasets. To uncover patterns and similarities in users' preferences, UBCF and MBCF are utilized for analyzing user behavior. These approaches rely on the assumption that users with similar past preferences will exhibit similar future preferences. However, when the dataset is divided into smaller subsets based on factors such as age, occupation, etc, it may be challenging to accurately identify patterns. In other words, when the dataset is partitioned, collaborative filtering techniques may not accurately identify patterns in user behavior. This is because the technique only considers user preferences within the partition that is being evaluated, ignoring preferences in other partitions. As a result of insufficient data, it becomes difficult to establish statistically significant correlations between user preferences, resulting in less accurate recommendations for those particular groups and recommendations for users in the evaluated partition will be less precise. On the other hand, using the entire dataset allows collaborative filtering techniques to analyze a larger and more diverse pool of data, making it easier to identify patterns and provide more accurate recommendations for the entire dataset. Therefore, recommendations based on the complete dataset are proven to be more accurate than those based on individual partitions defined by characteristics like age and occupation.

Based on the detailed analysis, it is inferred that User Based Collaborative Filtering or Model-Based Collaborative Filtering both produced more accurate recommendations on the complete dataset as compared to different partitions of the dataset. Additionally, researchers also compared the performance of aforementioned four different recommendation algorithms on complete 1M MovieLens dataset using MAE and RMSE values. As clearly illustrated in Figure 14, it was found that Model Based Collaborative Filtering (MBCF) with Singular Value Decomposition (SVD) produced the best results among all the algorithms tested. The results indicate that model-based methods consistently outperform user-based methods because they capture latent factors in the user-item matrix, which are not directly observable in sparse partitioned datasets. SVD, in particular, reduces noise and leverages dimensionality reduction to uncover hidden user preferences, leading to the lowest MAE and RMSE. Conversely, UBCF methods rely solely on direct rating overlaps, which become sparse in partitioned datasets, reducing accuracy.

Additionally, complete datasets provide a larger pool of interactions, allowing the algorithms to learn more reliable user-item similarities. Partitioning by age or occupation leads to smaller, sparse matrices, which diminishes the ability to find neighbors and extract robust latent features. This explains why all algorithms achieved their best performance on the complete 1M dataset, with SVD outperforming other techniques due to its superior capability to handle sparsity and capture global patterns.

5 Conclusion and Future Scope

Various e-commerce applications, including Amazon, Netflix and many more have made use of recommender systems. These systems fall under three categories: content-based filtering, collaborative filtering, and hybrid techniques. In this thesis we evaluated two types of collaborative filtering techniques, namely Model-base (MBCF) and user-based collaborative filtering (UBCF) with cosine, Euclidean distance and Matrix Factorization, SVD respectively. The evaluation was based on the MovieLens dataset, which contains one million ratings, and the performance was measured using MAE and RMSE on the complete dataset and different partitions of the dataset based on age and occupation. We applied two user-based and two model-based collaborative filtering recommender systems in two different ways. In the first step we partitioned the ratings on the basis of age and occupations. Furthermore, we applied the aforesaid algorithms on complete age (1-56 years) and smaller groups of ages. Similarly, the same algorithms were applied on all the occupations mentioned in the dataset as well as on five top most contributed occupations separately. In the second step, the same two techniques with four different algorithms were applied on the entire 1M ratings dataset. Collaborative filtering techniques, such as User-based Collaborative Filtering (UBCF) and Model-based Collaborative Filtering (MBCF), work by analyzing the behavior of users to identify patterns and similarities in their preferences. These techniques rely on the assumption that users who have similar preferences in the past are likely to have similar preferences in the future as well. However, when the dataset is partitioned based on factors such as age, occupation etc, it may result in smaller subsets of data, which may not be sufficient to identify patterns

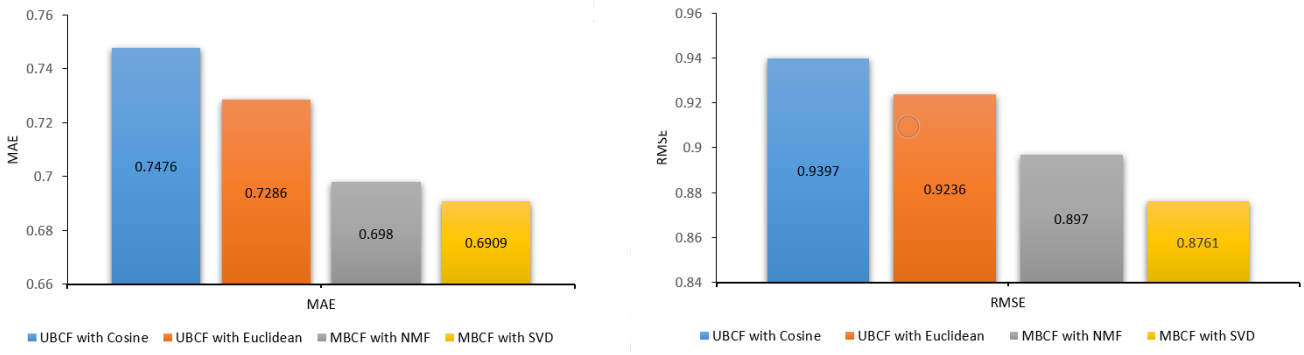

Figure 14. Comparing MAE and RMSE for UBCF and MBCF on Complete 1M Ratings.

accurately. These smaller subsets may not have enough data points to establish statistically significant correlations between the preferences of different users, resulting in less accurate recommendations for those groups. In other words, using partitioned dataset, the contribution of one part is maximized, while the contribution of other parts is ignored. This method is subtractive, as it only considers the preferences of users in the partition being evaluated and disregards the preferences of users in other partitions. This can result in less accurate recommendations, particularly for users in the partition being evaluated. In contrast, when the entire dataset is used, the collaborative filtering techniques have access to a larger and more diverse pool of data to analyze and identify patterns. This results in more accurate recommendations for the entire dataset as compared to individual partitions based on factors such as age and occupation. According to the results, both User-Based Collaborative Filtering (UBCF) and Model-Based Collaborative Filtering (MBCF) showed better performance on the complete 1M ratings dataset than on different partitions of the dataset. We also compared the MAE and RMSEs of user based collaborative filtering (UBCF) with Euclidean & cosine functions as well as Model-based Collaborative Filtering (MBCF) using Non-Negative Matrix Factorization and SVD on the complete 1M Movielens dataset. Resultantly, we found Model-Based Collaborative Filtering (MBCF) with Singular Value Decomposition (SVD) the best among all four technique that the MAE and RMSE are found lowest than MBCF with NMF and UBCF with Cosine and Euclidean Similarities.As a result, we can deduce that there is a need of an additive method in which we can maximize the contribution of each partition, while keeping the contribution of the other partitions. The statistical significance tests provide further validation of our findings. The differences in MAE and RMSE between the top-performing method (MBCF with SVD) and the other tested algorithms were consistently significant at the 95% confidence level, with p-values < 0.05 in all cases, and < 0.01 in most. This strengthens our claim that the performance advantage of SVD is not merely incidental but reflects a meaningful improvement over alternative collaborative filtering approaches.

This work addresses a gap left by many prior recommender system studies, which have rarely compared full dataset training against demographic-based partitions (e.g., age, occupation) using rigorous metrics such as MAE and RMSE. While earlier works often assume that subgroup segmentation improves personalization, they have not thoroughly tested whether such segmentation sacrifices overall accuracy. Our findings show that complete datasets often yield more reliable recommendations, than the partitioned dataset.

In future work, we aim to develop hybrid collaborative filtering models that effectively combine global dataset analysis with partition-based insights to maximize the contribution of each user subgroup while maintaining overall prediction accuracy. Specifically, additive or ensemble methods can be designed to integrate recommendations from multiple partitions, thereby overcoming the limitations of isolated partitions. Additionally, exploring deep learning approaches and context-aware recommendation techniques could further improve personalization by incorporating

additional user and item features. Finally, expanding the demographic factors beyond age and occupation, such as location or user behavior patterns, may provide richer models and more tailored recommendations.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Bhatnagar, V. (2016). *Collaborative filtering using data mining and analysis*. IGI Global.

[2] 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. [CrossRef]

[3] Fields, B., Rhodes, C., & d'Inverno, M. (2010). Using song social tags and topic models to describe and compare playlists. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*.

[4] Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications, 41*, 1–10. [CrossRef]

[5] Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). Recommender systems. *Physics Reports, 519*(1), 1–49. [CrossRef]

[6] Koren, Y., Rendle, S., & Bell, R. (2021). Advances in collaborative filtering. *Recommender systems handbook*, 91-142. [CrossRef]

[7] Rajaraman, A., & Ullman, J. D. (2011). *Mining of massive datasets*. Autoedicion.

[8] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182). [CrossRef]

[9] Yuan, T., Cheng, J., Zhang, X., Qiu, S., & Lu, H. (2014, June). Recommendation by mining multiple user behaviors with group sparsity. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 28, No. 1). [CrossRef]

[10] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46*, 109–132. [CrossRef]

[11] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.

[12] Rendle, S. (2010, December). Factorization machines. In *2010 IEEE International conference on data mining* (pp. 995-1000). IEEE. [CrossRef]

[13] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.

[14] Núñez-Valdéz, E. R., Lovelle, J. M. C., Martínez, O. S., García-Díaz, V., De Pablos, P. O., & Marín, C. E. M. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior, 28*(4), 1186–1193. [CrossRef]

[15] Bell, R. M., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 95-104). [CrossRef]

[16] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization* (pp. 291-324). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[17] Robillard, M., Walker, R., & Zimmermann, T. (2009). Recommendation systems for software engineering. *IEEE Software, 27*(4), 80–86. [CrossRef]

[18] Lokesh, A. (2019). A Comparative Study of Recommendation Systems. *Masters Theses & Specialist Projects*. Paper 3166.

[19] Desrosiers, C., & Karypis, G. (2010). A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, 107-144. [CrossRef]

[20] Aggarwal, C. C., & Aggarwal, C. C. (2016). Model-based collaborative filtering. In *Recommender Systems: The Textbook* (pp. 71–138). Springer. [CrossRef]

[21] Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics, 10*(2), 1273–1284. [CrossRef]

[22] Ungar, L., & Foster, D. P. (1998). A formal statistical approach to collaborative filtering. *CONALD'98*.

[23] Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the*

American Society for Information Science, 41(4), 288–297. [CrossRef]

[24] Black, P. E. (2006). Manhattan distance. In *Dictionary of Algorithms and Data Structures*. Retrieved from https://xlinux.nist.gov/dads/HTML/manhattanDistance.html (accessed on 31 December 2025).

[25] Casanovas, M., Torres-Martinez, A., & Merigo, J. M. (2016). Decision making in reinsurance with induced OWA operators and Minkowski distances. *Cybernetics and Systems, 47*(6), 460–477. [CrossRef]

[26] Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems (pp. 39-46). [CrossRef]

[27] Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2009). Scalable Collaborative Filtering Approaches for Large Recommender Systems. *The Journal of Machine Learning Research, 10*, 623-656.

[28] Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop* (Vol. 2007, pp. 5-8).

[29] Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 426-434). [CrossRef]

[30] Gunawardana, A., Shani, G., & Yogev, S. (2012). Evaluating recommender systems. In *Recommender systems handbook* (pp. 547-601). New York, NY: Springer US. [CrossRef]

[31] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53. [CrossRef]

[32] Ahuja, R., Solanki, A., & Nayyar, A. (2019). Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 263–268). IEEE. [CrossRef]

[33] Katarya, R., & Verma, O. P. (2016). A collaborative recommender system enhanced with particle swarm optimization technique. *Multimedia Tools and Applications, 75*(15), 9225-9239. [CrossRef]

[34] Alam, M. T., Ubaid, S., Sohail, S. S., Nadeem, M., Hussain, S., Siddiqui, J., & others. (2021). Comparative analysis of machine learning based filtering techniques using MovieLens dataset. *Procedia Computer Science, 194*, 210–217. [CrossRef]

[35] Singh, P. K., Pramanik, P. K. D., & Choudhury, P. (2019). A comparative study of different similarity metrics in highly sparse rating dataset. In *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2018, Volume 2* (pp. 45–60). Springer. [CrossRef]

[36] Zhang, J., Wang, Y., Yuan, Z., & Jin, Q. (2019). Personalized real-time movie recommendation system: Practical prototype and evaluation. *Tsinghua Science and Technology, 25*(2), 180–191. [CrossRef]

[37] Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M., & Jararweh, Y. (2018). A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In *2018 9th International Conference on Information and Communication Systems (ICICS)* (pp. 102–106). IEEE. [CrossRef]

[38] Lund, J., & Ng, Y. K. (2018). Movie recommendations using the deep learning approach. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)* (pp. 47–54). IEEE. [CrossRef]

[39] Byström, H. (2013). *Movie Recommendations from User Ratings*. Stanford University.

[40] Gupta, G., & Katarya, R. (2019). Recommendation analysis on item-based and user-based collaborative filtering. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1–4). IEEE. [CrossRef]

[41] Guan, X., Li, C.-T., & Guan, Y. (2017). Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems. *IEEE Access, 5*, 27668–27678. [CrossRef]

[42] Ayyaz, S., & Qamar, U. (2017). Improving collaborative filtering by selecting an effective user neighborhood for recommender systems. In *2017 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1244–1249). IEEE. [CrossRef]

[43] Isma'il, M., Haruna, U., Aliyu, G., Abdulmumin, I., & Adamu, S. (2020). An autonomous courses recommender system for undergraduate using machine learning techniques. In *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)* (pp. 1–6). IEEE. [CrossRef]

[44] Sallam, R. M., Hussein, M., & Mousa, H. M. (2020). An enhanced collaborative filtering-based approach for recommender systems. *International Journal of Computer Applications, 176*(41), 9–15.

[45] Kużelewska, U. (2014). Clustering algorithms in hybrid recommender system on movielens data. *Studies in logic, grammar and rhetoric, 37*(50), 125-139. [CrossRef]

[46] Hsu, S. H., Wen, M. H., Lin, H. C., Lee, C. C., & Lee, C. H. (2007, May). AIMED-A personalized TV recommendation system. In *European conference on interactive television* (pp. 166-174). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[47] Ma, T.-m., Wang, X., Zhou, F.-c., & Wang, S. (2023). Research on diversity and accuracy of the recommendation system based on multi-objective optimization. *Neural Computing and Applications, 35*(7), 5155–5163. [CrossRef]

[48] Pandya, S., Shah, J., Joshi, N., Ghayvat, H., Mukhopadhyay, S. C., & Yap, M. H. (2016, November). A novel hybrid based recommendation system based on clustering and association mining. In *2016 10th international conference on sensing technology (ICST)* (pp. 1-6). IEEE. [CrossRef]

[49] Agrawal, S., & Jain, P. (2017). An improved approach for movie recommendation system. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 336–342). IEEE. [CrossRef]

[50] Ponnam, L. T., Punyasamudram, S. D., Nallagulla, S. N., & Yellamati, S. (2016). Movie recommender system using item based collaborative filtering technique. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)* (pp. 1–5). IEEE. [CrossRef]

[51] Shah, K. (2019). Book recommendation system using item based collaborative filtering. *International Research Journal of Engineering and Technology, 6*(5), 5960–5965.

[52] Ricci, F., Rokach, L., & Shapira, B. (2021). Recommender systems: Techniques, applications, and challenges. *Recommender systems handbook*, 1-35. [CrossRef]

[53] Liu, A. S., & Gao, J. (2019). Book Recommendation Algorithm Based on Deep Learning. *International Journal of Science*, 152–156.

[54] Saraswat, M., Dubey, A., Naidu, S., Vashisht, R., & Singh, A. (2020). Web-Based Movie Recommender System. In *Ambient Communications and Computer Systems: RACCCS 2019* (pp. 291–301). Springer. [CrossRef]

[55] Raghuwanshi, S. K., & Pateriya, R. K. (2021). Accelerated singular value decomposition (asvd) using momentum based gradient descent optimization. *Journal of King Saud University-Computer and Information Sciences, 33*(4), 447–452. [CrossRef]

[56] Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS), 5*(4), 1–19. [CrossRef]

Muhammad Ahmad Hafeez earned his BS-Mathematics degree from B.Z University and M.S. degree in Data Science from Air University, Islamabad, Pakistan. He is currently pursuing his Ph.D degree in Artificial Intelligence from Air University, Islamabad, Pakistan. Mr. M Ahmad Hafeez has a strong academic and professional background in Mathematics, Data Science, with specific expertise in data analytics, natural language processing (NLP), and deep learning. His work focuses on bridging the gap between theoretical advancements in AI and practical, real-world applications. He has been performing instructional duties for the last 14 years. He has shown his teaching experties in the subjects of Calculus, Differential equations, Linear Algebra, Probability and Statistics. Currently he is performing his instructional duties in National University of Science and Technology (NUST) Pakistan. (Email: AhmadHafeez.edu@Gmail.com)

Tahir Sher pursuing his PhD in Artificial Intelligence from Korea University, Seoul Campus, Korea. He done his MS in Data Science from Air University and bachelor's degree in Mathematics from the International Islamic University, Islamabad and achieved gold-medal. With strong teaching experience, he has taught subjects like Calculus, Differential Equations, Linear Algebra, and Data Science methods to students from Pakistan and allied countries. As a research scholar in the Explainable AI Group at Air University, his interests span IoT, Social IoT, Machine Learning, NLP, Time Series, Mathematical Modeling, Federated Learning, and Computer Vision. He is dedicated to bridging computer science with human-centered research. (Email: 2025010294@korea.ac.kr)

Abdul Rehman is currently a Research Professor at the Human Data Convergence Research Institute, Jeonju University, South Korea. He previously worked as a Postdoctoral Research Associate at the Hyper-Connectivity Convergence Technology Research Center, Kyungpook National University (KNU), Daegu, South Korea, and served as an Assistant Professor at the University of Central Punjab, Lahore, Pakistan. He holds a bachelor's degree in Mathematics from the International Islamic University, Islamabad, Pakistan, and an Integrated Ph.D. in Computer Science and Engineering from KNU. His research interests include Deep Learning, Machine Learning, Data Science, and their applications in Computer Vision, Smart IoT (S-IoT), and Data Analysis. He also explores Complex Network Navigation, Mathematical Modeling, and Big Data Analytics. Dr. Rehman received the KINGS Scholarship for his Ph.D. studies and was honored with the "Outstanding Researcher" Award from the School of Computer Science and Engineering at KNU. He serves as a guest editor and editorial board member for several international journals and has contributed to numerous international conferences as a session chair and publication chair. More information about Dr. A. Rehman is available at https://sites.google.com/view/drrehman/home.

Imran Ihsan received a PhD degree in Knowledge Graphs and Explainable AI from the Capital University of Science and Technology, Islamabad, in 2020. He has over 25 years of academic (teaching, project supervision, and academic administration) and industry (software, web, graphics, animation, and game development and operational management) experience. Since 2013, he has been working as Professor with the Department of Creative Technologies, Faculty of Computing and AI, Air University, Islamabad. He has authored or coauthored research work in various conferences and journals. His research interests include knowledge engineering, semantic computing, natural language processing, and computational linguistics.