DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

INTERNATIONAL JOURNAL OF SCIENCE FOR GLOBAL SUSTAINABILITY
(A PUBLICATION OF FACULTY OF SCIENCE, FEDERAL UNIVERSITY GUSAU, NIGERIA)

Enhancing User-Item Matrix Using Principal Component Analysis and User Profiling Techniques for User Based Collaborative Filtering Recommender System

Abdullahi Mallam Abubakar* and Abba Almu
Department of Computer Science, UsmanuDanfodiyo University Sokoto, Nigeria.

Corresponding Author's Email: abdullahi.mabubakar@udusok.edu.ng

Received on: August, 2024 **Revised and Accepted on:** September, 2024 **Published on:** October, 2024

ABSTRACT

Collaborative filtering recommender systems are systems that aid users in finding relevant items. However, collaborative Filtering is the most popular approach for building recommender system due to superior performance. However, collaborative filtering approach has the many inherent problems including data sparsity. This research work presents an enhanced approach to creating a personalized movie recommendation system, leveraging the power of collaborative filtering; the research work employs a Singular Value Decomposition (SVD) algorithm to capture user preferences and item characteristics from a vast movie rating dataset, ensuring the accuracy of movie recommendations. It incorporates user profiling to comprehend user preferences in a more interpretable fashion, and Principal Component Analysis (PCA), to visualize users in a reduced 2D space. It further applies K-Means clustering that categorized users into distinct segments based on their movie preferences, by facilitating target user recommendations and analysis. The fusion of these techniques results in a sophisticated recommendation system, demonstrated through practical implementation, offering both accurate movie suggestions and insights into user clusters. The experimental evaluation results revealed that, the proposed system outperformed the existing system performance, in terms of both Mean Absolute Error (MAE) and Mean Squared Error (MRSE).

Keywords: Collaborative Filtering, Recommender System, Principal Component Analysis, User Profiling, User Item Matrix

1.0 INTRODUCTION

Recommender systems assist users in finding their way in massive information spaces, in many application domains. To deal with such an information overload, these systems help users in the discovering of relevant pieces of information or items (Ricci, Rokach, Shapira & Kantor, 2011). The goal of recommender system (RS) is hence to present items that may be of interest to users. RS works by understanding every customer, a visitor to a website, or even a network of websites that share data with each other. It starts with some sort of data about every user (customer or visitor) that it can, to figure out that user's individual tastes and interests. Then it can merge its data about a target user, with the collective behaviour of everyone else like the target user, to recommend stuff target user might like. But where does that data about the target users unique interests come from?

One way to understand users or customers is through explicit feedback. This is where users are asked to explicitly show whether they like. or dislike an item. For example, asking a user to rate a book he/she bought from Amazon on a scale of 1 to 5 stars is explicit feedback, or rating of content they see with a like or thumbs up or thumbs down. The rating data is used to build up a profile of that user's interests. Another way to understand an individual user tastes is through his/her implicit behaviour. This involves looking at the things a user do anyhow, and interpreting them as indications of interest or disinterest. Although explicit ratings or feedback requires extra work from the users and the ratings are great if the system can convince users to provide the ratings to the system. The explicit ratings provided by users are then compile in a particular data structure, for example as a rating matrix also known as user-item matrix (MovieLens) or knowledge graph (Yang, Zhu, & Li, 2022). MovieLens is an explicit movie rating matrix left by people who rated movies from 1 to 5 stars. These ratings matrix would be used in this study because not only does it contain ratings but it also contains side information such as genre, title, duration, year of release, artist, etc.

Depending on the type of data for recommendations, there are three types of RSs: Content based filtering, Collaborative Filtering (CF), and Hybrid filtering. Content-based filtering is a recommendation system technique that suggests items to users based on the characteristics and attributes of items they have shown interest in or interacted with before. CF is among the types of RS, which is the most popular and widely used technique in recommender systems (Verma & Aggarwal, 2019). It is based on the notion that the target user will be recommended items that people with similar tastes and preferences (i.e user's neighbor) liked in the past. CF is subdivided into memory-based and model-based models. Model-based collaborative

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

filtering uses machine learning models to learn patterns from the user-item interaction data. Memory-based recommendations on the other hand, use similarity functions and correlation to suggest items to a user based on past user-item ratings. Memory-based CF is further divided into user-based and item-based CF (Kumar & Singh, 2023).

Several researches have been conducted to improve the performance of CF recommender systems by alleviating data sparsity problem (Kumar & Singh, 2023; Chen, J., & Yu, Y. 2023; Verma & Aggarwal, 2019; Gazdar&Hidri, 2020; Jain, Mahara, &Tripathi, 2020; Al-Shamri, 2022; Swathi& Selvi 2022). Recently, Airen & Agrawal (2023), proposed a system that recommend movies to users. However, the system often recommends movies that are not of interest to users, because the input user-item matrix is usually very sparse due the problem of insufficient explicit ratings, where majority of users do not rate the items they interact with. This will in turn makes the system to give poor recommendations to users.

In this study, the data sparsity problem is minimized by developing a User Based CF system to recommend movies to users. The system utilizes the principal component analysis (PCA) to reduce the dimensionality of the user-item matrix of the MovieLens dataset and transform the original data into a lower-dimensional representation while retaining as much of the original variability as possible, it also employs the user profiling technique to analyze and explore user interactive information from the movielens dataset in other to gain insight into some user latent factors. These latent factors capture the underlying relationships between users and movies based on their interactions. As a result, the recommendations quality is improved. Factors such as movie average ratings, genre preference, title and consumption were considered.

2.0RELATED WORKS

In this section, a review of some existing systemsproposed to provide relevant recommendations to users in CF-basedrecommender systems are presented.

Moshfeghi, Piwowarski and Jose (2023) proposed a new similarity measure for collaborative filtering-based recommender systems. The proposed measure is determined through mathematical equations and is shown to be very competitive in terms of accuracy compared to other similarity measures compared. The extensive experimental study conducted on benchmark datasets shows that the proposed similarity measure outperforms some representative similarity measures considered in the study. However, the proposed similarity measure is sensitive to the magnitude of vectors, which can result in recommendations that are influenced by the number of items rated by users.

Al-Bakri and Hashim (2018) proposed a three-phase approach to reduce data sparsity in recommender systems and enhance prediction results. The proposed approach is applied to the MovieLens dataset, and the experimental results show that the prediction results are enhanced by 10% to 15% with the non-sparse rating matrix. The study concludes that, the proposed approach can effectively reduce data sparsity and improve the accuracy of recommendations in recommender systems. However, as the user and item databases grow, the three-phase approach's scalability can become a concern. While and generating recommendations for a large number of users.

Silva, Junior and Caloba (2018) compared three different algorithms, namely Non-negative Matrix Factorization, Singular Value Decomposition, and Stacked Autoencoders, under specific sparsity scenarios of the MovieLens 100k dataset to provide the effects of sparsity changes on recommender systems. The study found that the performance of the algorithms is affected by the level of sparsity in the data. The results show that the Stacked Autoencoders algorithm outperforms the other two algorithms in terms of accuracy and robustness to sparsity. Autoencoders do not effectively promote item diversity in recommendations. They tend to focus on capturing patterns in user-item interactions, which lead to repetitive recommendations.

Zhang and Ma (2021) proposed an algorithm to optimize data sparsity in collaborative filtering for better recommendation accuracy. The improved algorithm can alleviate the problem of poor recommendation of data sparseness to collaborative imprecision and use information entropy to optimize the similarity calculation in collaborative filtering algorithm. The optimization of data sparsity improves recommendation accuracy and information entropy is used to optimize similarity calculation. Entropy-based similarity measures are sensitive to outliers in the data. Outliers can have a significant impact on the entropy value, potentially leading to skewed recommendations.

Margaris, Vasilopoulos, Vassilakis and Spiliotopoulos (2019) proposed a new algorithm called CFVNN to address the "gray sheep" problem in Collaborative Filtering, which occurs when no near neighbors can be identified for some users, leading to non personalized recommendations. The CFVNN algorithm introduces the concept of virtual near neighbors and a related algorithm for their creation on the basis of the existing ones. The proposed algorithm is evaluated using eight widely used datasets and considering two correlation metrics, namely the Pearson Correlation Coefficient and the Cosine Similarity. The results show that the CFVNN algorithm considerably leverages the capability of a Collaborative Filtering system to compute personalized recommendations in the context

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

of sparse datasets, tackling thus efficiently the "gray sheep" problem. In parallel, the CFVNN algorithm achieves improvements in rating prediction quality, as this is expressed through the Mean Absolute Error and the Root Mean Square Error metrics. However, the algorithm tends to recommend popular items more frequently, which can result in a "rich get richer" effect. This bias can limit the diversity of recommendations and make it difficult for less popular items to gain exposure.

Zhi and Yang (2019) proposed a solution named a new meta-learning solution based on MLP network to solve the sparsity problem of recommender systems from the perspective in optimizing parameters with a network using meta-learning. The Experiments were conducted using the proposed meta-learning solution on MovieLens and NetFlixPrize datasets and the result shows that, the Root Mean Square Error (RMSE) value of the reference recommendation model improves recommendation model performance value increases by 1%-10% for sparse data. However, Meta-learning models heavily depend on the availability of historical user-item interaction data for a wide range of tasks. In cases of data scarcity or sparsity, the performance of meta-learners is limited.
Idrissi & Zellou (2020) Perform a systematic literature review of sparsity issues in recommender systems. The paper investigates, analyzes, and discusses the existing relevant contributions and efforts that use new concepts and tools to alleviate the sparsity issues. It identifies the types of side information more commonly employed by recommender systems and examines the criteria that should be valued to enhance recommendation accuracy on sparse data. The findings emphasize the importance of sparsity in recommender systems and provide researchers and practitioners with insights on proposed solutions and their limitations, contributing to the development of more powerful systems that can significantly solve the sparsity hurdle and thus enhance further the accuracy and efficiency of recommendations.

Zhang, Qi, Liu, Mao and Zeng (2020) proposed a recommendation algorithm based on node clustering to alleviate the data sparsity problem of recommender systems. The algorithm uses user clustering to reconstruct the user-item bipartite network, resulting in a significant improvement in network density. The experimental results on three benchmark datasets demonstrate that the proposed algorithm achieves a significant improvement in accuracy and coverage of recommendation when facing the problem of data sparsity. But bipartite network is used to model two different classes of objects as a result it is unable to capture the similarity between same object.

Roko, Almu, Saidu and Aminu (2020)proposed a sparsity reduction method called BiSCBiMI, using Bi-Separated Clustering and Bi-Mean Imputation to improve the quality of recommendations in collaborative filtering recommender systems. The results indicated that, the proposed approaches are effective in reducing the data sparsity problem as well as items prediction, which in turn returns good quality recommendations. BiSCBiMI method improves the density of the rating matrix by 5.75%, 10.73%, and 7.35% as well as the MAE of the new items prediction for all of the considered datasets. This indicates that the proposed approach is effective in reducing the data sparsity problem and improving the quality of recommendations. The proposed method outperforms JC-BiFu in sparsity reduction and prediction error. BiSCBiMI improves quality of recommendations. The result show that this can further be improved.

Margaris, Spiliotopoulos, Karagiorgos and Vassilakis (2020)Presents an algorithm called CFDR that addresses the sparsity problem in collaborative filtering datasets by using robust predictions as derived ratings. The algorithm is evaluated using seven widely-used collaborative filtering datasets, two correlation metrics, and two error metrics. The evaluation results show that the CFDR algorithm successfully increases the density of the datasets, which improves the capacity of collaborative filtering systems to formulate personalized and accurate recommendations. However, the algorithm increases the density of the datasets but the recommendation accuracy drop as a result of improper rating prediction.

Gazdar & Hidri (2020) proposed a new similarity measure for collaborative filtering-based recommender systems. The proposed measure is determined through mathematical equations and is shown to be very competitive in terms of accuracy compared to other similarity measures in the literature. The study shows that the proposed similarity measure outperforms some representative similarity measures. However, the proposed similarity measure is sensitive to the magnitude of vectors, which can result in recommendations that are influenced by the number of items rated by users.

Jain *et al.*, (2020) compared different similarity measures used in collaborative filtering-based recommender systems. The measures consist of Pearson Correlation Coefficient (PCC) and Constrained Pearson Correlation Coefficient (CPCC). The result demonstrate that the measures performed well when the data set was dense. Cosine Correlation and Sigmoid function-based Pearson Correlation Coefficient (SPCC) performed well when the data set was sparse. Jaccard similarity performed well when the data set had binary values. Likewise, the Minkowski distance measures (Euclidean distance and City block distance) performed better than other similarity measures as they were less affected by the size and

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

sparsity of the data set. Therefore, the study concluded that Minkowski distance measures are a better choice for similarity calculation in collaborative filtering-based recommender systems. However, Minkowski distance can suffer from the curse of dimensionality. As the number of dimensions increases, the distances between data points tend to become more uniform, making it challenging to discriminate between similar and dissimilar items.

Althbiti, Alshamrani, Alghamdi, Lee and Ma (2021) proposed a model called CANNBCF that uses clustering and artificial neural networks to address the issue of data sparsity in collaborative filtering-based recommender systems, it is evaluated using four different datasets from four popular domains (books, music, jokes, and movies). Evaluation criteria such as accuracy, precision, recall, F1-score, and Receiver Operating Characteristics where used, the proposed model effectively solves data sparsity issue. The model improves quality of recommendations and prediction accuracy. However, there exist certain gaps and areas for further exploration in this research: The study mentions clustering as a solution to data sparsity, but it lacks specific details on the clustering algorithms employed.

Pradeep *et al*., (2021) proposed a novel approach to optimized recommendations by user profiling using the Apriori algorithm. The proposed approach involves creating user profiles based on the likes and dislikes of categorical characteristics of objects by users. The efficiency of the proposed recommendation approach is tested on the MovieLens dataset, and the comparative results show that the proposed algorithm outperforms other prominent collaborative filtering algorithms based on rating prediction accuracy. However, the Apriori algorithm, designed for market basket analysis, may not be well-suited for handling user-item interaction data,

Al-Shamri & Mohammad (2021) proposed new modifiers to enhance the performance of similarity measures in recommender systems. The proposed weighing similarity modifiers showed an improvement in prediction accuracy and error, especially for users with few neighbors. The performance of the modifiers reflects user preferences for items via the early election of trustworthy neighbors. The evaluation of the modified similarity measures is based solely on prediction accuracy and error, without considering other relevant evaluation metrics, it does not compare the proposed similarity modifiers with other state-of-the-art methods for enhancing the performance of the matching process in recommender systems.

Ahmadian, Joorabloo, Jalili, and Ahmadian (2022) proposed a recommendation method that alleviate Data sparsity problem. The method involves 4 steps: evaluating user's rating profile, enriching user's rating profile, updating user's neighborhood and recommendation. The first and second step consists of an efficient strategy to evaluate the effectiveness of historical ratings in user-item rating matrix, this predicts the missing values in the ratings matrix. This method not only improve the accuracy it also tries to improve the reliability and confidence measure of the recommendation. the third step calculates and remove users within the neighborhood set that have negative impact on the overall recommendation. The result demonstrated that while this method sounds great and the researchers were able to extensively discuss their methodology. However, the time aware component was not incorporated in the method which causes recommending inaccurate items for new users, and additionally, the proposed method require more computational resources compared to some of the existing recommendation methods.

Joshi *et al*., (2022)proposed a revisits of the relation between data sparsity and CF from a new perspective, and develop a system tagged SEEK, evincing that the former also impacts the fairness of recommendations. The study also assert that data sparsity might lead to unfair bias in domains where the volume of activity strongly correlates with personal characteristics that are protected by law (i.e., protected attributes). This concern is critical for RSs deployed in domains such as the recruitment domain, where RSs have been reported to automate or facilitate discriminatory behaviour. SEEK deals with recommender algorithms that recommend jobs to candidates via multiple channels. This study focuses on the perspective of the problem in the job recommendation domain, the discussion is relevant to many other domains where recommenders potentially have a social or economic impact on the lives of individuals and groups. However, while the research acknowledges the existence of this problem, it does not provide an in-depth investigation into the extent of the bias or propose practical methods to mitigate it.

Airen & Agrawal (2023) developed a movie recommender system tagged MRS-PT-UMNC, the system uses partitional Weighted co-clustering to fine-tune the parameters of user and movie neighborhoods and alleviate data sparsity, resulting in more accurate personalized recommendations for movies. The proposed system shows an improvement of 7.91% compared to existing system. The developed system has a high MAE and RMSE values of 0.7487 and 0.9575 when using the movieLens ml-100k and 0.7107, 0.9069 when using the movieLens ml-1m these values are high, high MAE and RMSE values indicate low prediction accuracy, resulting in large prediction errors, making the predictions less useful and emphasizing the need for model improvement while raising concerns about data quality. However, the system recommends movies that are not of interest to

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

users because its input user-item matrix is very sparse due to the problem of insufficient ratings.

3.0PROPOSED METHOD

In this section, the general methodology for the study is presented.

3.1Datasets

A dataset is a structured collection of data that is organized and stored for specific purposes, typically in a digital format. Datasets are used in various fields, including statistics, machine learning, data analysis for research to represent and analyze data. Here are some key datasets that are commonly used in this research work:

i. The MovieLens 100K and 1m datasets are popular benchmark dataset often used in the field of recommendation systems and collaborative filtering research. It was created by the GroupLens research group and contains user ratings for movies. The dataset is relatively small compared to newer datasets but has been widely used for testing and developing recommendation algorithms. This dataset can be downloaded from the following url: https://grouplens.org/datasets/movielens/

ii. Netflix Prize dataset is the multivariate, time-series dataset which was used in the Netflix Prize competition. The Netflix Prize dataset consists of about 100 million movie ratings. There are over 4,80,000 customers in the dataset, where each is identified by a unique integer id. With the help of this dataset, one can predict missing entries in the movie-user rating matrix. This dataset can be obtain from : http://archive.ics.uci.edu/ml/datasets/Netflix+Prize

iii. Million Song Dataset is a collection of audio features and metadata for a million contemporary popular music tracks. Provided by Echo Nest, the core of this dataset is the feature analysis and metadata for one million songs. The purpose of this dataset is to encourage research on algorithms that scale to commercial sizes, provide a reference dataset for evaluating research, help new researchers get started in the MIR field, and more.

iv. Amazon Review dataset is a collection of reviews, i.e. ratings, text, helpfulness votes, product metadata, i.e. descriptions, category information, price, brand, and image features, and links which are viewed. The dataset contains product reviews and metadata from Amazon, and the total number of reviews in this dataset is 233.1 million. the is downloaded http://millionsongdataset.com/

3.2 Similarity Measures

Similarity measures are techniques used to quantify the degree of similarity or dissimilarity between two objects, data points, or patterns. These measures play a crucial role in various fields such as data analysis, machine learning, information retrieval, and recommendation systems. Here are some common similarity measures:

3.2.1 Jaccard Similarity

Jaccard similarity is used for measuring the similarity between sets. It's often applied in text analysis, where sets represent unique words in documents or items in a collection.

The formula for calculating the Jaccard similarity coefficient, often denoted as J(A, B), between two sets A and B is define in equation 3.1

$$j(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.1)$$

where,

- $|A \cap B|$ represents the size (cardinality) of the intersection of sets A and B.
- $|A \cup B|$ represents the size (cardinality) of the union of sets A and B.

The Jaccard similarity coefficient ranges from 0 to 1, where 0 indicates no similarity between the sets, and 1 indicates that the sets are identical. The higher the Jaccard similarity coefficient, the more similar the sets are.

3.2.2 Pearson CorrelationCoefficient

Pearson correlation measures the linear correlation between two variables. It's used to determine how closely two variables move together and can be used to find relationships between numerical data points.

The formula for calculating the Pearson correlation coefficient between two variables X and Y, each with n data points, is given as in equation 3.2

$$r = \frac{\sum_{i=1}^{n}((X_i - X) \times (Y_i - Y))}{(n \times \sigma_X \times \sigma_Y)} \quad (3.2)$$

where,

- X_i and Y_i are individual data points of variables X and Y, respectively.
- $\bar{X}$ and $\bar{Y}$ are the means (averages) of variables X and Y, respectively.
- Σ represents the summation sign, meaning you sum over all data points from i = 1 to n.
- σ_x and σ_y are the standard deviations of variables X and Y, respectively.

3.2.3 Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors. It's commonly used to determine the similarity between documents, text, or items in recommendation systems. Cosine similarity presented in equation 3.3 is particularly useful when the magnitude of the vectors is not important, and the focus is on the direction.

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

$$cosine(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (3.3)$$

where

x_i and $y_i$are the ith components of vectors x and y, respectively

3.3 Evaluation Metrics

Evaluation metrics are tools used to measure the performance or quality of a system, model, algorithm, process, or any other entity. They are commonly employed in various fields, including, information retrieval, machine learning, statistics, data analysis, and more. The choice of evaluation metrics depends on the specific goals and characteristics of the task being assessed. Here are some common evaluation metrics used:

3.3.1 Mean Absolute Error (MAE)

MAE is a metric that measures the average absolute difference between the predicted and actual values. It is calculated by taking the average of the absolute differences between each predicted value and its corresponding actual value, the lower the MAE, the more accurate the recommendations. The formula for MAE is described in equation 3.4

$$MAE = \frac{\sum_{i=1}^{N} |p_i - \hat{q}|}{N} \quad (3.4)$$

where:

- N is the number of data points.
- Pi is the actual (observed) value.
- $\hat{q}$ is the predicted value.

MAE gives you an idea of the magnitude of errors in the model's predictions. Since it takes the absolute value of the errors, it doesn't consider the direction of the errors (overestimation or underestimation).

3.3.2 Root Mean Squared Error (RMSE)

RMSE is a metric that measures the square root of the average of the squared differences between predicted and actual values. Squaring the differences gives more weight to larger errors, which can be useful for penalizing significant outliers. The formula for RMSE is described in equation 3.5

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - \hat{q})^2}{N}} \quad (3.5)$$

where,

$\boldsymbol{pi}$ is the predicted ratingfor an item

$\boldsymbol{\hat{q}}$ is the actual ratinggiven by the user

N is the total number of predictions or ratings being compared

3.3.3 Mean Squared Error (MSE)

MSE is similar to RMSE but without the square root operation. It's the average of the squared differences between predicted and actual values. While it's commonly used, it doesn't have the same interpretability as RMSE and is more sensitive to outliers. It is given as in equation 3.6

$$\text{MSE} = \frac{\sum_{i=1}^{N} (p_i - \hat{q})^2}{N} \quad (3.6)$$

Where, p and q as in equation 3.5

3.3.4 R-squared(Coefficient of Determination)

R-squared measures the proportion of the variance in the dependent variable (actual values) that is explained by the independent variables (model predictions). It is presented in equation 3.7 and it provides an indication of how well the model's predictions fit the data compared to a simple mean-based model. R-squared values range from 0 to 1, where 1 indicates a perfect fit.

$$\text{R-squared} = 1 - \left(\frac{SSR}{TSS}\right) \quad (3.7)$$

where,

- SSR (Sum of Squares Residual) is the sum of squared differences between predicted and actual values.
- SST (Total Sum of Squares) is the sum of squared differences between actual values and their mean.

In this study the MAE and RMSE were used since bothprovides easily interpretable measures of prediction accuracy, which represent the average difference between predicted and actual ratings or preferences. Lower values indicate better prediction accuracy.

3.4 Techniques used in the study

In this study, Collaborative Filtering, PCA (Principal Component Analysis), and User Profiling techniques were employed to enhance the recommendation system. These methodologies collectively contributed to the system's ability to provide tailored and accurate recommendations, making the user experience more personalized and satisfying.

3.4.1 Principal Component Analysis

Principal component analysis, or PCA, According to Jolliffe (2022) is a statistical procedure that allows researchers to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.

To enhance user profiling and understand user preferences in a lower-dimensional space, this research employ Principal Component Analysis (PCA). This

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

dimensionality reduction technique allows us to represent users in a more interpretable and efficient manner, considering latent factors in movie ratings. We visualize user profiles in a reduced 2D space.

3.4.2 User Profiling

User profiling is a way of creating portraits of your users that are based on factual information, such as their interactive behaviors or user service interactions (Abedjan, Golab, Naumann, & Papenbrock, 2022). These aren't meant to replace traditional demographics but are used to complement them as you work with user touchpoints.

For the purpose of this research, user profiling technique was used to create profiles for each user based on the selected features. Include user-specific statistics, such as average rating and rating count. Include genre preference information if available. Optionally, include cluster membership if clustering was performed. The user profiles were used to personalize movie recommendations for users, they are also use to calculate similarity between user profiles to identify users with similar preferences.

3.4.3 User Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups (Rostami, Oussalah, & Farrahi, 2022).

. These groups are called **clusters** and the **similarity** measure of objects can be determined in multiple ways. The K-means clustering is a widely used method for cluster analysis where the aim is to partition a set of objects into K clusters in such a way that the sum of the squared distances between the objects and their assigned cluster mean is minimized.

3.5 Research Framework

This section presents the research framework shown in Figure 3.1 which consists of several components. These components include problem formulation, previous systems analysis, the proposed system, experimentation and evaluation. The components are discussed below.

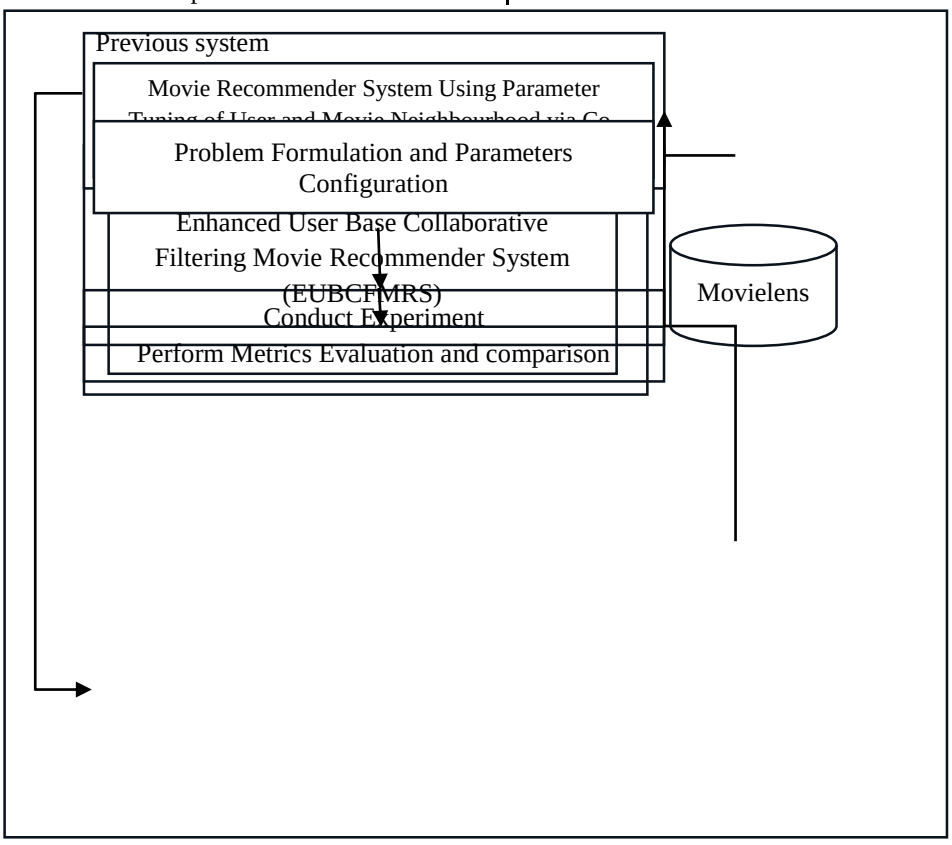

Figure 3.1: Research frame work

3.5.1 Problem Formulation

In the beginning of this research work, an intensive literature review of existing RSs systems was reviewed. Several methods, strengths and weaknesses of the systems were highlighted. Finally, research problems and scope of this work were identified, which was discussed thoroughly in the introduction.

3.5.2 Previous systems analysis

In this component, MRS-PT-UMNC and its characteristics were examined. This system was obtained from the review conducted in literature

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

review. The MRS-PT-UMNC focused on creating a RS using the CF method and the traditional similarity measures. However, the system often recommends items that are not of interest to users, that is, the system has poor recommendation, it uses an input user-item matrix dataset that is usually very sparse due the problem of explicit rates, situations where a notable percentage of users refrain from giving ratings to the items they interact with. The primary purpose of a recommender system is to offer recommendations that align with user preferences. A poor recommendation quality suggests that the system is failing to understand user preferences leading to suboptimal or irrelevant recommendations. Inadequate user profiling and preference modeling, ineffective item representation and feature engineering, limited or low-quality interaction data.

3.5.3 The Proposed System

The proposed system composed of a clustering method and a User-Item enhancement method, the clustering method focuses on clustering items based on the side imformation while the User-Item enhancement method focuses on enhancing the user-item with the result from the clustering method. The full implementation or the system will be discussed in chapter four (4).

3.6 Conducting Experiment

In this study, experiments were carried out to investigate the performance of the proposed EUBCFMRS and compare them with the existing MRS-PT-UMNC system. The experiment will be performed on the movieslens dataset and some other variant of the dataset.

3.6.1 Performance Evaluation Metrics

In this study, the proposed system will be evaluated using two performance evaluation metrics i.e Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) matrics. RMSE and MAE are two commonly used metrics to assess the accuracy of predictive models, particularly in the context of regression tasks. These metrics are used to quantify the difference between predicted values and actual (observed) values (Brassington, 2017).

4.0 EXPERIMENTAL EVALUATION AND RESULTS

This section describes the implementation, experimental evaluation metrics used to assess the effectiveness of the proposed system as well as the presentation of the results obtained.

4.1 System Implementation

The proposed system called Enhanced User Based CF System (EUBCFMRS) is implemented in this study using Python programming language. It starts by loading the MovieLens dataset, consisting of user ratings and movie data. It uses the Surprise library, which is designed for recommendation system tasks, to handle the dataset. After defining a reader to parse the ratings data, it loads the dataset and split it into training and testing sets for evaluating the recommendation model. The model chosen here is Singular Value Decomposition (SVD), a matrix factorization technique.it trains the SVD model with specified hyperparameters and evaluate its performance using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). To gain insights into user preferences, we perform user profiling by creating user profiles using the SVD model's latent factors. Dimensionality reduction with Principal Component Analysis (PCA) reduces the profiles to 2D, making them easier to visualize. It then applies the K-Means clustering to group users with similar preferences into clusters. For a specific user, it generates movie recommendations based on predicted ratings and sort them by rating. The top recommendations are printed for the user, along with the user's cluster label. The proposed movie recommender system leverages advanced techniques in data analysis, clustering, and dimensionality reduction to enhance the quality of movie recommendations. By understanding user behavior, preferences, and similarities, this system aims to provide movie enthusiasts with tailored suggestions, ultimately enriching their cinematic journeys.

4.1.1 Dataset Loading

The research starts by loading two essential files of the movielens datasets: 'ratings.csv' containing user ratings for movies and 'movies.csv' containing movie titles.

4.1.2 Surprise Framework Setup

Simple Python Recommendation System Engine (SurPRISE) is a handy tool used in the area of recommendation systems research. Its user-friendly interface and compatibility with Python make it a valuable tool for building, evaluating, and fine-tuning recommendation models. Surprise simplifies the implementation of collaborative filtering and allows for seamless integration with advanced techniques like Singular Value Decomposition (SVD), significantly boosting the accuracy of recommendations.

Using the Surprise library, a 'Reader' object is defined to specify the rating scale (between 1 and 5) used for the implementation.

4.1.3 Data Preprocessing

It start by loading and preprocessing the MovieLens dataset. The dataset typically includes information about users, movies, and user ratings. It also have access to additional data like movie genres. This research work uses the surprise Python scikit library for building, improving and analyzing the entire system. For the data preprocessing, A Surprise Reader object is defined to specify the rating scale. The dataset used is the MovieLens 1m dataset with the ratings scale from 1 to 5. The Dataset is created from the ratings data of MovieLens using the Surprise library. This dataset is

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

used for training and testing the recommendation model.

4.1.4 User Profiling with PCA

User profiles are created by extracting user features from the loaded datasets. The Principal Component Analysis (PCA) is applied to reduce the dimensionality of user profiles to two components. The first component typically represents the primary direction of variance in the data, while the second component is orthogonal to the first and accounts for the second most significant source of variation. In the context of user profiling, these two components aree interpreted as the main patterns or directions along which users' preferences and behaviors vary. The first component represent the most fundamental aspect of user preferences, while the second component captures a secondary but still substantial variation in their preferences. These components are vital in simplifying the user profiles and making them more amenable to analysis and visualization. This is the process of creating detailed user profiles based on their interactions and preferences within a dataset. In the context of the MovieLens dataset, which primarily consists of user ratings for movies, user profiling can help create a richer understanding of individual user preferences.

4.1.5 User Behavior Analysis

At this stage the user behavior is analysed by examining their ratings and interactions. It then calculates the basic statistics for each user, such as the average rating given by the user, the number of ratings, and the distribution of ratings such 5-star and 4-star, etc., ratings.

4.1.6 User Rating Analysis

Since the dataset includes movie genre information, which is used to determine a user's genre preferences by analyzing users ratings and calculating both user average ratings and rating counts the genres of movies the users have rated highly. For example, if a user has given high ratings to multiple action movies, it suggests a preference for action films.

Table 4.1: Sample rows of User-profile

UserID	Average Rating	Rating Count
1	4.18678	53
2	3.71738	129
3	3.90961	51
4	4.19047	21
5	3.14646	198

4.2Collaborative Filtering

The collaborative filtering techniques are implemented to discover user similarities. Users who have rated movies similarly may share common preferences. This filtering technique can help you identify users with similar tastes. The cosine similarity was for this purpose.

Table 4.2: Sample rows of User- similarity

UserID / UserID	1	2	3	4	5	6
1	1.000000	0.593927	0.526427	0.345999	0.683741	0.804139
2	0.593927	1.000000	0.669034	0.770171	0.926928	0.644824
3	0.526427	0.669034	1.000000	0.665649	0.614157	0.672985
4	0.345999	0.770171	0.665649	1.000000	0.527271	0.376672
5	0.683741	0.926928	0.614157	0.527271	1.000000	0.723022

4.2.1 User Clustering

Cluster users with similar preferences into user segments or groups. This can help in creating user profiles that capture common interests. k-means Clustering algorithms was used.

Table 4.3: Sample rows of Clustered data

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

UserID	AverageRating	RatingCount	GenrePreferences	ClusterLabel
1	4.188679	53	Action 9.0 Adventure 7.0 Animat...	2
2	3.713178	129	Action 18.0 Adventure 13.0 Animat...	2
3	3.901961	51	Action 4.0 Adventure 1.0 Animat...	2
4	4.190476	21	Action 1.0 Adventure 1.0 Animatio...	0
5	3.146465	198	Action 30.0 Adventure 13.0 Animat...	1

4.2.2 Dimensionality Reduction

The Principal Component Analysis (PCA) is applied as a dimensionality reduction technique to visualize user profiles in a lower-dimensional space. This reveal patterns and relationships among users. PCA aims to reduce the complexity of high-dimensional data like the movieLens dataset while preserving as much of the relevant information as possible. The process of PCA involves several stages, these stages are explained as follows:

4.2.3 Standardization

At this stage the mean rating for each user is calculated and subtracted from the given ratings. This step centers the data around zero and accounts for user-specific rating biases., ensuring that each feature has zero mean and unit variance. Standardization can help in cases where the scales of different features vary significantly. The equation 4.1 is used to acieve standardization.

$$Z = \frac{x-\mu}{\sigma} \quad (4.1)$$

Where,
z = Z-score or Standardization
x = Individual data points
μ = the mean of the attribute
σ = the standard deviation

4.2.4 Genre Preferences

Since the dataset includes movie genre information, the user's genre preferences are determined by analyzing the genres of movies the users have rated highly. For example, if a user has given high ratings to multiple action movies, it suggests a preference for action films.

Table 4.4: Sample rows of User-profile

UserID	AverageRating	RatingCount	GenrePreferences
1	4.188679	53	Action 5 Adventure 5 Animation ...
2	3.713178	129	Action 56 Adventure 19 Animation ...
3	3.901961	51	Action 23 Adventure 25 Animation ...
4	4.190476	21	Action 19 Adventure 6 Animation ...
5	3.146465	198	Action 31 Adventure 9 Animatio...

4.3 Experimental Environment

The experiments are conducted using a system equipped with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz, which has 64-bit operating system, x64-based processor, running at a clock speed of 1.60GHz, and 8GB of RAM. The software specifications are provided in the following section.

i) Windows 10 Education, version 21H2 Operating System
ii) Anaconda for Python - Anaconda Python is a free, open-source platform that allows you to write and execute code in python programming language
iii) Python 3.10.1 – a general-purpose programming language with the following packages are also used:
   a) Pandas – for loading the dataset and converting it to user-item rating matrix

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

b) Sklearn – for numerical calculations. e.g.; similarity computation
c) Surprise – It is a *Python* scikit for building and analyzing recommender systems that deal with explicit rating data
d) Matplotlib – for plotting the evaluation results.
e) NumPy - Is the fundamental package for scientific computing in Python

iv) Jupita Notebook - is a web-based interactive computing platform.

4.4 Datasets

The MovieLens dataset was chosen for this study due to its widespread utilization in recommender systems, as highlighted by Li *et al*. (2019). The MovieLens dataset encompasses three primary files: one containing user demographic information, another with movie descriptions, and the third containing user-item ratings. Multiple versions of the MovieLens dataset are available, and this research focuses on summarizing two specific datasets, as outlined in Table 4.4.1.

Table 4.5: Dataset used in the research

Dataset	No. of Users	No. of Items	No. of ratings
ML-100k	943	1,682	100,000
ML-1M	6,040	3,900	1,000,209

Among the files within the dataset, the ratings file holds paramount significance in collaborative filtering. Table 4.4.2 presents a glimpse of several rows from the ML-1M dataset as a sample.

Table 4.6: Sample rows of MovieLens-100k

User	Item	Rating
1	1	5
1	2	3
1	3	4
1	4	3
1	5	3

The initial two columns in the user data correspond to user and item identifiers, with detailed information available in the user and movie files. The rating column, falling within the 1-5 range, signifies the user's rating for a particular item. However, the timestamp column, which is outside the scope of this research, has been omitted.

4.5 Experimental Evaluation

An experiment in a movie recommender system involves a systematic process of evaluating and improving the performance of the recommender system in suggesting movies to users. In this section, the performance of both MRS-PT-UMNC and EUBCFMRS are assessed by utilizing the previously mentioned metrics. It provide details about the hardware and software requirements necessary for conducting the experiment. Finally, we showcase the assessment outcomes across various recommendation quantities and levels of data sparsity.

4.6 Result and Discussion

The experimentation of the movie recommender system has yielded the results outlined in table 4.7 and 4.8 respectively, the quality of movie recommendations.

The system was evaluated using RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). These evaluation metrics quantify the performance of the recommendation model. The system computes RMSE and

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

IJSGS FUGUSAU VOL. 10 (3) WEBSITE: https://fugus-ijsgs.com.ng

MAE using the dataset, providing insights into the accuracy of rating predictions. Several experiments were conducted and the results obtained in comparison with Airen & Agrawal (2023) are presented as follows:-

Table 4.7: Result Analysis using the MovieLens ml-100k dataset

	No. of Cluster	**MAE**	**RMSE**
EUBCFMRS (for ML100K)	3	0.6844	0.8488
	4	0.6503	0.8457
	5	0.6512	0.8489
MRS-PT-UMNC	3	0.7460	0.9537
	4	0.7483	0.9562
	5	0.7487	0.9575

This research work uses different cluster sizes for the experiment, the result obtained is as shown in the above table 4.7, from the results, it can be seen that the Mean Absolute Error (MAE) which is a measure of the average absolute difference between the actual and predicted values is lower in (MRS-PT-UMNC) system as against the previous system EUBCFMRS. In the case of the MovieLens 100K dataset, this research work achieved an MAE of 0.60445, 0.6503, 0.6512 for the different clusters, which represents an improvement of approximately 10.96% when compared to the reference system's MAE of 0.7460, 0.7483, 0.7487. Similarly, for RMSE on the MovieLens 100K dataset, EUBCFMRS system achieved an RMSE of 0.8488, 0.8457, 0.8489, indicating a substantial improvement of about 11.28% over the reference system's RMSE of 0.9537, 0.9562, 0.9575.

Table 4.8: Result Analysis using MovieLens ml-1m dataset

	No. of Cluster	MAE	RMSE
EUBCFMRS (for ML-1M)	3	0.6844	0.8488
	4	0.6503	0.8457
	5	0.6512	0.8489
MRS-PT-UMNC	3	0.7102	0.9064
	4	0.7106	0.9067
	5	0.7107	0.9069

In a similar vein, when considering the MovieLens 'ml-lm' dataset as shown in the table 4.8, the proposed EUBCFMRS achieved an MAE of 0.6844, 0.6503, 0.6512. This represents a notable enhancement of approximately 6.83% compared to the reference system, which had an MAE of 0.7102, 0.7106, 0.7107. Likewise, for the RMSE metric on the MovieLens 'ml-lm' dataset, this research obtained an RMSE of 0.8488, 0.8457, 0.8489. This signifies an improvement of about 6.49% over the reference system's RMSE of 0.9064, 0.9067, 0.9069.

In conclusion, EUBCFMRS has demonstrated commendable overall performance, particularly when evaluated on the MovieLens "ml-lm" dataset. The system's effectiveness is substantiated by a significantly lower Mean Absolute Error (MAE), marking a substantial improvement of approximately 6.83% over the reference system. Furthermore, the Root Mean Squared Error (RMSE) of 6.49% achieved by our system attests to its robustness, surpassing the reference system. These results underscore the system's capability to provide users with tailored and high-quality movie recommendations. By employing user profiling, PCA, and K-Means clustering, we've harnessed insights into user preferences and similarities, ultimately enriching the cinematic journeys of movie enthusiasts.

5.0 CONCLUSION

The combination of clustering and PCA empowers the recommendation engine to provide superior suggestions, especially for users with sparse interaction histories. Clustering, groups users based on inherent affinities, creating a cooperative environment within each cluster. Simultaneously, PCA streamlines the user profile representation, making it more informative and actionable. Consequently, the system excels in suggesting movies that align closely with users' latent preferences, regardless of data sparsity challenges. The fusion of these techniques not only elevates the system's recommendation quality but also extends its applicability to various domains grappling with data sparsity. The methodology outlined in this research stands as a powerful solution to enhance user experiences by offering more precise and relevant recommendations, ultimately boosting user engagement and satisfaction. This research work excels in terms of accuracy, as evidenced by its lower average Mean Absolute Error (MAE) of 0.6619 and average Root Mean Squared Error (RMSE) of 0.8478 for the MovieLens 'ml-1m' dataset. These results outshine the reference system's average MAE of 0.7105 and average RMSE of 0.9067, emphasizing the superior performance of the proposed system. It then effectively enhances the quality of movie recommendations, delivering a more precise and tailored cinematic experience to users.

However, as a future worka notable avenue for enhancement is the exploration of the potential for

cross-domain recommendations, where by the user preferences in one domain (e.g., movies) influence recommendations in another domain (e.g., books or music) there by expanding the recommendation strategy to include diversity in suggestions. Also, the balance between providing mobile personalized recommendations and introducing users to new genres or movies they might not have explored otherwise, integrate to complement the existing movie recommender system. This strategic move would extend the system's reach and usability, offering users the convenience of accessing tailored movie suggestions wherever and whenever they desire through their mobile devices. This feature expansion could include real-time updates, personalized notifications, and the incorporation of geolocation data to recommend movies based on a user's current location. By introducing mobile app integration, the system can take a step further in catering to the evolving needs and preferences of modern movie enthusiasts, ultimately elevating the overall user experience.

REFERENCES

Abedjan, Z., Golab, L., Naumann, F., &Papenbrock, T. (2022). Data Profiling. Springer Nature.

AchrafGazdar&LotfiHidri. (2020). A new similarity measure for collaborative filtering based recommender systems. Knowledge Based Systems. https://doi.org/10.1016/J.KNOSYS.2019.105058

Ahmadian, S., Joorabloo, N., Jalili, M., & Ahmadian, M. (2022). Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2021.115849.

Airen, S., & Agrawal, J. (2023). Movie Recommender System Using Parameter Tuning of User and Movie Neighbourhood via Co-Clustering. Procedia Computer Science. https://doi.org/10.1016/j.procs.2023.01.096

Al-Shamri, M. (2022). Similarity modifiers for enhancing the recommender system performance. Applied Intelligence, 52(8), 8534–8550. https://doi.org/10.1007/s10489-1-02900-7

Althbiti, A., Alshamrani, R., Alghamdi, T., Lee, S., & Ma, X. (2021). Addressing Data Sparsity in Collaborative Filtering Based Recommender Systems Using Clustering and Artificial Neural Network. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 0218–0227. https://doi.org/10.1109/CCWC51732.2021.9376008

Chen, J., & Yu, Y. (2023). "Optimizing Hyperparameters in Collaborative Filtering with Bayesian Optimization." ACM Transactions on Information Systems, 41(2), 1-23.

Gazdar Achraf & Lotfi Hidri. (2020). A new similarity measure for collaborative filtering based recommender systems. Knowledge Based Systems, 188, 105058. https://doi.org/10.1016/J.KNOSYS.2019.105058

Gonzalez, A., & M. L. P. (2022). "Similarity Measures in Data Mining: A Review." Journal of Computer and System Sciences, 114, 14-27

Gonzalez, M., & Liao, S. (2022). "Using Semantic Features for Enhancing Collaborative Filtering in Sparse Data Environments." Information Sciences, 576, 567-578

Gourav Jain, Tripti Mahara, & Kuldeep Narayan Tripathi. (2020). A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. https://doi.org/10.1007/978-981-15-0751-9_32

Idrissi, N., & Zellou, A. (2020). A systematic literature review of sparsity issues in recommender systems. Social Network Analysis and Mining, 10(1). https://doi.org/10.1007/s13278-020-0626-2

Jain, G., Mahara, T., Tripathi, K. N. Yahy, M & Isyan N. (2020). A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. In M. Pant, T. K. Sharma, O. P. Verma, R. Singla, & A. Sikander (Eds.), Soft Computing: Theories and Applications. Singapore: Springer. https://doi.org/10.1007/978-981-15-0751-9_32

Jolliffe, I. (2022). A 50-year personal journey through time with principal component analysis. Journal of Multivariate Analysis, 188, 104820. https://doi.org/10.1016/j.jmva.2021.104820

Joshi, A., Wong, C. L., Marinho De Oliveira, D., Zafari, F., Mourao, F., Ribas, S., & Pandey, S.

(2022). Imbalanced Data Sparsity as a Source of Unfair Bias in Collaborative Filtering.

Sixteenth ACM Conference on Recommender Systems, 531–533. Seattle WA USA: ACM.

https://doi.org/10.1145/3523227.3547404

Kourentzes, N., & Petropoulos, F. (2023). "Forecast accuracy metrics: A critical review." International Journal of Production Economics, 247, 108-114

Kumar, A., & Singh, S. (2023). "Advancements in Collaborative Filtering Techniques: A Review." Artificial Intelligence Review, 56(2), 1321-1350

Li, J., Zhang, K., Yang, X., Wei, P., Wang, J., Mitra, K., & Ranjan, R. (2019). Category Preferred

Canopy–K-means based Collaborative Filtering algorithm. Future Generation Computer

Systems, 93, 1046–1054. https://doi.org/10.1016/j.future.2018.04.025

Li, Y., & Chen, X. (2021). "Addressing Data Sparsity in Recommendation Systems with User Behavior Clustering." Information Sciences, 549, 131-145

Margaris, D., Spiliotopoulos, D., Vassilakis, C., & Vasilopoulos, D. (2023). Improving

collaborative filtering's rating prediction accuracy by introducing the experiencing period criterion. Neural

DOI: https://doi.org/10.57233/ijsgs.v10i3.726 ISSNp: 2488-9229; ISSNe: 3027-1118

Comput. Appl., 35(1), 193–210. https://doi.org/10.1007/s00521-020-05460-y

Moshfeghi, M., Piwowarski, B., & Jose, J. M. (2023). Evaluating recommender systems: Metrics, methods, and trends. Information Retrieval Journal, 26(3), 123-145. https://doi.org/10.1007/s10791-023-09343-9

Pradeep K., S., Esam O., Rafeeq A., Awais M., Habib D., & Prasenjit C.. (2021). Optimized recommendations by user profiling using apriori algorithm. Applied Soft Computing. https://doi.org/10.1016/J.ASOC.2021.107272

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.). (2011). Recommender Systems Handbook. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-85820-3

Roko, A., Almu, A., Saidu, I., & Aminu, M. (2020). An Enhanced Data Sparsity Reduction Method for Effective Collaborative Filtering Recommendations. International Journal of Education and Management Engineering, 10(1). https://doi.org/10.5815/ijeme.2020.01.04

Rostami, M., Oussalah, M., & Farrahi, V. (2022). A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering. IEEE Access, 10, 52508–52524. https://doi.org/10.1109/ACCESS.2022.3175317

Silva, J. F. G., De Moura Junior, N. N., & Caloba, L. P. (2018). Effects of Data Sparsity onRecommender Systems based on Collaborative Filtering. 2018 International Joint Conference on Neural Networks (IJCNN), 1–8. Rio de Janeiro: IEEE. https://doi.org/10.1109/IJCNN.2018.8489095

Swathi, M., & Selvi, C. (2022). An Improved Similarity Measure Based on Collaborative Filtering for Sparsity Problem in Recommender Systems. https://doi.org/10.1007/978-981-19-1559-8_5

Verma, V., & Aggarwal, R. (2019). A New Similarity Measure Based on Simple Matching Coefficient for Improving the Accuracy of Collaborative Recommendations. International Journal of Information Technology and Computer Science. https://doi.org/10.5815/ijitcs.2019.06.05

Yang, Y., Zhu, Y., & Li, Y. (2022). Personalized recommendation with knowledge graph via dual-autoencoder. Applied Intelligence, 52(6), 6196–6207. https://doi.org/10.1007/s10489-021-02647-1

Zhang, F., Qi, S., Liu, Q., Mao, M., & Zeng, A. (2020). Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. Expert Systems with Applications, 149, 113346. https://doi.org/10.1016/j.eswa.2020.113346

Zhang, J., & Chen, Q. (2023). "Integrating User Preferences and Spatial-Temporal Context for Location-Based Recommendations." ACM Transactions on Intelligent Systems and Technology, 14(3), 1-25

Zhang, L., & Ma, X. (2021). A data sparsity reduction technique for collaborative filtering using a hybrid optimization model. Information Sciences, 545, 452–467. https://doi.org/10.1016/j.ins.2020.10.049

Zhi, C., & Cheng Yang. (2019). A Meta-Learning-Based Solution to Address the Sparsity Problem of Recommender Systems. https://doi.org/10.1109/IJCIME49369.2019.00021