I.J. Information Technology and Computer Science, **2022, 4, 48-56**
Published Online August 2022 in MECS (http://www.mecs-press.org/)
DOI:10.5815/ijitcs.2022.04.05

Incorporating Preference Changes through Users' Input in Collaborative Filtering Movie Recommender System

Abba Almu*
Department of Computer Science, Usmanu Danfodiyo University, Sokoto, Nigeria
E-mail of Corresponding Author: almul2003@yahoo.com

Aliyu Ahmad
Department of Computer Science, Usmanu Danfodiyo University, Sokoto, Nigeria
E-mail: aliyuahmad@fugusau.edu.ng

Abubakar Roko
Department of Computer Science, Usmanu Danfodiyo University, Sokoto, Nigeria
E-mail: abroko@yahoo.com

Mansur Aliyu
Department of Computer Science, Sokoto State University, Sokoto, Nigeria
E-mail: mansuraliyu@gmail.com

Received: 26 January 2022; Revised: 10 March 2022; Accepted: 01 April 2022; Published: 08 August 2022

Abstract: The usefulness of Collaborative filtering recommender system is affected by its ability to capture users' preference changes on the recommended items during recommendation process. This makes it easy for the system to satisfy users' interest over time providing good and quality recommendations. The Existing system studied fails to solicit for user inputs on the recommended items and it is also unable to incorporate users' preference changes with time which lead to poor quality recommendations. In this work, an Enhanced Movie Recommender system that recommends movies to users is presented to improve the quality of recommendations. The system solicits for users' inputs to create a user profiles. It then incorporates a set of new features (such as age and genre) to be able to predict user's preference changes with time. This enabled it to recommend movies to the users based on users' new preferences. The experimental study conducted on Netflix and Movielens datasets demonstrated that, compared to the existing work, the proposed work improved the recommendation results to the users based on the values of Precision and RMSE obtained in this study which in turn returns good recommendations to the users.

Index Terms: Collaborative Filtering, Recommender System, Users' Input, Preference Changes, Recommended Items.

1. Introduction

In this era when the internet became an important part of human life, there is significant increase in the amount of information and services available on it which makes it difficult to find information at the right time. This led to information overload, where finding required information from an overwhelming set of choices have become a problem. Therefore, there is need for systems to help users discover items or products of interest from the large collection of data are required. Such systems are called Recommender systems [1]. Several recommender systems to suggest relevant items to users were developed [2-7] are classified into three major categories which include Collaborative Filtering (CF), Content-Based (CB) and Hybrid (HB) recommender systems.

Collaborative filtering systems are more widely researched because they can recommend any kind of item to users such as Books, Music and Movies. As a result, several Collaborative filtering Recommender Systems were developed [3, 8, 9]. Most researches on recommender systems [2- 7, 10, 11] have all focused mainly on ways of guessing or modeling users' preferences and algorithms for identifying items that a user is likely to evaluate positively. Despite the success achieved by these researches, the following problems were left unresolved [7] which include: the system fails to solicit for user inputs on the recommended items and it is unable to incorporate users' preference changes with time since what is like today may be different from what may be like tomorrow.

In this work, an Enhanced Movie Recommender system that recommends movies to users is presented to improve the quality of recommendations and also to address the stated problems. First, the system solicits for users' inputs to create a user profiles. Second, it then incorporates a set of new features (such as age and genre) to be able to predict user's preference changes with time. This enabled it to recommend movies to the users based on users new preferences. The experimental results indicated that compared to the existing work, the proposed work improves the recommendations relevancy and accuracy to the users based on the values of Precision and RMSE obtained which in turn returns good quality recommendations to the user.

2. Related Works

In this section, a review of some existing works, methods and algorithms proposed to provide relevant recommendations to users in CF recommender systems are presented.

Eyrun, Gaurangi and Nan [2] MovieGen Recommender system using machine learning Support Vector Machine (SVM) approach and cluster analysis based on hybrid recommendation approach. The system used SVM algorithm to predict movies preference for the user based on the information identified by incorporating the CF approach (i.e it predict the user choice based on the choice of other similar users) and employ content-based approach by taking into consideration the user choice not based on the user past ratings alone but rather based on the answers given to the question asked by the system. The K-means clustering algorithm was used to improve the recommendation by choosing cluster with maximum number of movies once genres and period have been predicted using the user preferences based on the personal information. However, the system asks series of questions from the users which take alot of time to predict users' choice.

Manoj *et al.* [3] presented CF Movie Recommender System (MOVREC) that recommends movies to users based on the information provided by the users. The user is given options to choose from some set of attributes which include Actor, Director, Genre, Year and Ratings and then predict the user's choices based on the choices of previous visited history of users. It then collects actions and feedbacks of users and store it in a customized database, which is used for generating new recommendation in the next user-system interaction. When any user visited the system, it will give the user some couple of options such as search a particular movie, see upcoming movies list or go to recommendation page to aid the recommendation process. However, the system solely depends on the input or value given by the users before it recommends movies, hence, this makes it difficult to recommend movies for unregistered users.

Shreya and Pooja [5] proposed an Improved Approach for Movie Recommendation System to improve the accuracy, quality and scalability of movie recommender system. The hybrid approach unifies the content-based filtering and collaborative filtering using Support Vector Machine (SVM) as a classifier and genetic algorithm to perform optimisation. It was experimentally implemented on three different Movielens datasets consisting of 1M, small and 10M datasets. The results of the study have shown that, the proposed approach improves accuracy, quality and scalability of the movie recommendation system than other approaches. However, the system makes recommendation based on genre correlation thereby making it difficult to satisfy the user's dynamic interests.

Muyeed *et al.* [6] proposed a machine learning based movies recommender system using clustering and machine learning approaches. The system uses clustering algorithm in order to find users with similar taste of movies and Machine learning approaches to guess what rating a particular user might give to a particular movie so that the information can be used to recommend movies to viewers. The two files of MovieLens dataset containing information about movies such as movie id, movie title and its list of genres while the other file contains information about the users ratings of a particular movie were used for the experiment. These two files have been pre-processed in order to build the model. The results indicated that, the system shows 95% accuracy on average in predicting rating from new user which can be used to analyze which movie should be recommended to new users. However, the system solely depend on user rating of a particular movie before the movie will be recommended to the network of users in a cluster, but the problem is that if a user decide not to rate a movie after watching then the system would not provide recommendations.

Meenu *et al.* [7] proposed a Movie Recommender System Using item-based Collaborative Filtering that uses cosine similarity and K-Nearest Neighbour. The KNN algorithm is used to find the distance between the target movies with every other movie in the dataset and then it ranks the top k nearest similar movies using cosine similarity to make prediction. First it extracts the dataset to gather information about the target movie and the user's rating, then it used collaborative filtering with the rating dataset so that it can be consumed by the KNN model, to remove the huge dataset handling problems. Then finally cosine similarity is used to find the distance between the target movie and other movies, which gives the top k nearest neighbors, and then displayed the required recommended list of movies with descending order of distance. To evaluate the performance of the proposed system with other existing systems in terms of quality, accuracy, precision, recall, time and computational efficiency while the system is said to performed better. However, it fails to consider user inputs on the recommendations provided and system did not consider user preference changes with time.

Harpreet, Maninder and Amritpal [9] proposed a hybrid online movie recommender system. The system provides movie recommendation based on Content-based and collaborative filtering and also used context for better recommendations. In content-based aspect it used user profile created at the beginning of the user interaction with system. The profile created contained information such as age, gender, taste etc. To make recommendation to the user, the system compares items that were rated by the user with the items that are not rated by the user to calculate similarities. Those items that are similar to rated items will be recommended to the user. In collaborative aspect they focused on relationship between user and items, similarities of items are determined by similarity of rating of those items by the user who have rated both of them. However, the system fails to recommend items without any ratings to the users.

Ponnam *et al.* [12] presented Movie Recommender System Using Item-Based CF technique to examine the user item rating matrix and identify the relationships among various items. The system uses the identified relationships in order to compute the recommendations for the user. It also uses the adjusted cosine similarity for calculating the similarities between the items and then utilised the similarity weights value calculated to predict rating of the movies and then recommend the most top N recommendations to the users. However, the system relies on user feedbacks provided to make recommendations to users without that, the system cannot give accurate recommendations.

Hafed *et al.* [13] proposed a new collaborative filtering recommendation algorithm that benefits from the capabilities provided by the k-means clustering algorithm and SVD technique. The k-means clustering algorithm clustered users in the same partition according to their preferences, and then the SVD utilised each cluster not only as a dimensionality reduction technique but also as a powerful mechanism to efficiently help in finding the most similar users. The proposed system is in two main phases: The first phase is called offline model creation. In this phase, the model of recommendation is created by clustering the users' ratings by following their preferences, reducing the dimensions of data and then calculating the similarities. The k-means algorithm and SVD technique are both used in this phase to cluster similar users and reduce the dimensionality respectively. The second phase is the online model utilization whereby the created model is used in producing accurate recommendations for a given active user. The experimental result indicated that, the system provides good prediction accuracy. However, new user/movie suffers in the system, since it recommends movies to cluster of similar users based on rating from users of same cluster.

This review gives brief description of some important related works on movie recommender systems researches. Despite the success of these studies several limitations have been identified. These include: the existing system fails to solicit for user inputs on the recommended items and it is unable to incorporate users' preference changes with time thereby affecting quality of recommendations to target users. The review also revealed that, none of the investigated existing systems addressed the identified problems completely and thus necessitated further research. This forms the basis and motivation for this research work.

3. Proposed Method

In this section, an Enhanced Movie Recommender system is presented. The general scheme for the study framework is depicted in Fig. 1.

3.1. Problem Formulation

After conducting an intensive review of related works which led to the research gab identification on the existing systems. Several methods, systems, techniques, strengths and weaknesses of the approaches were described. The research problems derived from previous works and its scope were extensively identified as follows: the system fails to solicit for user inputs on the recommended movies and it is also unable to incorporate users' preference changes with time during recommendation processes.

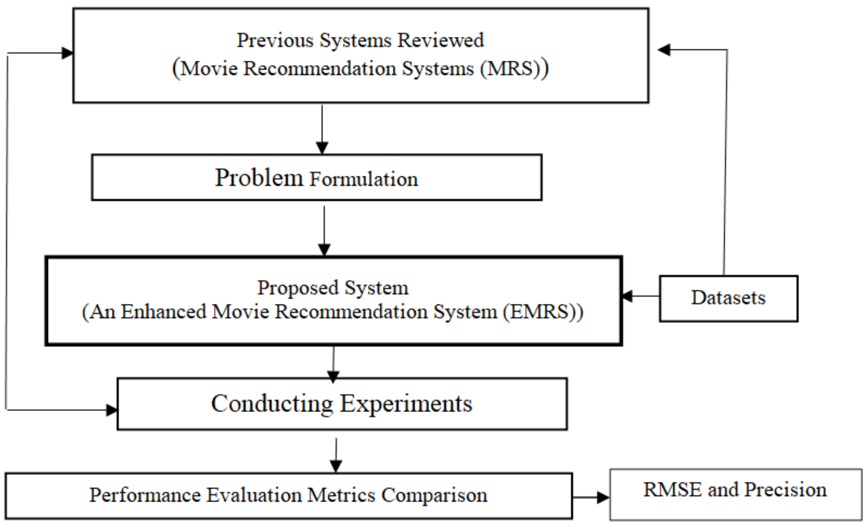


Fig.1. Scheme of the Research Framework

3.2. An Enhanced Movie Recommender System (EMRS)

EMRS is a hybrid Movie Recommender System that is aimed at predicting current user preference changes based on the class (genre) of movies. The proposed system possessed the following additional features that are not in the existing system under study: Incorporating User Inputs on the recommendations and User Preference Changes Incorporation.

A. Data Preprocessing

The datasets are huge and include a wide range of different users and ratings. These movie datasets are rated by only a few users, hence the need to preprocess the data in order to get accurate data with more similar users. In this regard, filtering of the datasets is done based on popular movies in the data frame. In this case, only movies that were rated by more than 20 users is considered to reduce the sparsity of the data [7, 14, 15, 16]. For every user in each of these datasets, the numbers range from 1-5 for each genre was calculated, representing their appreciation of that genre. The numbers were calculated by taking the users' mean rating for the genre.

B. Similarity Measurement

There have been many functions studied for finding similarities, methods like Pearson correlation, Jaccard similarity, Cosine similarity, and more. But, Pearson correlation coefficient was used in this research to measure the similarities between users as presented in Equation (1). Then, the model is trained, and the K-nearest neighbors for each user were chosen from among the similar users. In this case, for every user U_i in the system, K-nearest neighbors are calculated as follows:

$K_{i1}, K_{i2}, \ldots K_{ik}$ are calculated
where:
$0 < i < n$,
Kik is the kth nearest neighbours of user *Ui*
n is the total number of users
k are the values to be chosen for the experimental study while trying different values of k.

This Pearson Correlation coefficient measures how highly correlated two variables are, and this is measured from -1 to 1. Whereby, Pearson Correlation Coefficient of +1 indicates that the objects are perfectly correlated while those of -1 means objects are not correlated. In simple words, Pearson's correlation coefficient calculates the effect of change in one variable when the other variable changes.

To determine which are the most relevant neighbours to select and generate reliable prediction and recommendations, the K-nearest neighbours (K-NN) algorithm was employed which allows selecting only the K best neighbours with the highest correlation value.

$$r = \frac{\sum_{i\in I}(x_i-\overline{x_i})(y_i-\overline{y_i})}{\sqrt{\sum_{i\varepsilon I}(x_i-\bar{x})^2}\sqrt{\sum_{i\varepsilon I}(y_i-\bar{y})^2}} \tag{1}$$

Where, r is the correlation coefficient, x_i is the x-variable in a sample, $\bar{x}_i$ is the mean of the values of the x-variable, y_i is values of the y-variable in a sample and $\bar{y}_i$ is the mean of the values of the y-variable.

C. Prediction

After calculating similarity between items and the K nearest neighbors were found for each item in order to provide recommendations to users based on preferences. Then the prediction a users' rating for a movie and all other users who had watch and rated that movie would be used. The users' predicted rating for a movie will be estimated using Equation (2) based on the values of the"K nearest neighbors". Here, K is a number which is chosen by validation of the recommendations.

$$P_{um} = \bar{r}_u + \frac{\sum_{i=1}^{k}(rim-\bar{r}i) \text{ x } \mathrm{r}}{\sum_{a=1}^{n}|sim(i,j)|} \tag{2}$$

Where P_{um} is the predicted rating on movie *m* for user *u* and $\bar{r}_u$ is the user *u* average rating.

D. Incorporating User Inputs and Preference

Here current user's preference is giving due consideration. This was achieved by soliciting user inputs, with which the user profile is generated, where the system frequently focuses on two types of information: (i) a model of the user's preference (ii) a record of the user's interaction with the system. Then, consider a class of movie. A variety of candidate items are compared with items earlier rated by the user and the best-matching items are recommended as shown in Table 1 and 2 respectively which consists of a list of movies rated by user based on his preference.

Table 1. User Movie Preferences and Ratings

	Title	Rating
0	Toy Story	4.9
1	Final Destination	4.9
2	Mission Impossible	4.0
3	Father of the Bride Part II	3.0
4	Exorcist, The	4.8
5	Waiting to Exhale	3.9
6	Avengers, The	4.5
7	Maximum Ride	5.0

Table 2. User Current Movie Preference Details

	MovieId	Title	Genres	Year	Rating
0	1	Toy Story	[Adventure, Animation, Children, Comedy, Fantasy]	1995	4.9
1	4	Waiting to Exhale	[Comedy, Drama, Romance]	1995	3.9
2	5	Father of the Bride Part II	[Comedy]	1995	3.0
3	1997	Exorcist, The	[Horror, Mystery]	1973	4.8
4	2153	Avengers, The	[Action, Adventure]	1998	4.5
5	89745	Avengers, The	[Action, Adventure, Sci-Fi, IMAX]	2012	4.5
6	3409	Final Destination	[Drama, Thriller]	2000	4.9
7	164226	Maximum Ride	[Action, Adventure, Comedy, Fantasy, Sci-Fi, T...]	2016	5.0

Table 2 shows user's current movie preference details, that is the list of movies to be watched based on ratings given to those movies that the user likes and prefer most. This will enable the system to recommend some movies to the user that fold on similar class with his preference.

3.3. Enhanced Movie Recommender Algorithm

Input: Movielens Dataset (UserId, MovieId and Ratings), user inputs
Output: N Recommended Movies to User

Step 1: Read all information CSV as data frame within user and item variable ratings.
Step 2: Split dataframe ratings into training sets and test sets by 80:20
Step 3: Create a pivot matrix that tells which user rating belong to which movie in the dataframe.
Step 4: Apply the Pearson correlation coefficient metric to the pivot matrix to calculate the similarity matrix between users.
Step 5: Apply SVD and KNN by using userId and movieId two parameters input to estimate the movie ratings for TopN similar users, using the rating test data frame.
Step 6: Solicit for user input, to create new user profile
Step 7: Use the new created user profile to get his movie preference
Step 8: then make movie recommendations to user.

4. Experiments

This section describes the datasets, experimental environment and evaluation metrics used to assess the effectiveness of the proposed EMRS as well as the presentation and the description of the results obtained. It is also aimed to evaluate the relevancy and accuracy of the recommendations for both the existing and the enhanced movie recommender systems.

4.1. Description of Dataset

The two well known datasets are used to evaluate the performance of the proposed EMRS system compared to MRS system. These datasets consist of MovieLens-100K and MovieLens-1M. These are free datasets of movie ratings gathered from movielens.org. It contains user ratings, movie metadata and user metadata that were collected over period of time. Movielens-100K consists of 100,000 movie rating from 1000 users on 1700 movies while Movielens-1M contains movie rating from 6000 users on 4000 movies [17]. The Movielens datasets have been chosen to be suitable for this research because of the following: It is publicly available for download and consists of unnamed ratings of movies gathered by the GroupLens Research. These MovieLens datasets were used in different researches [18, 19] and turn to be efficient on the research outcome.

4.2. Experimental Environment

The computer system used for the experimental study possessed the following specifications: Windows 8, 64-bit operating system, Intel (R) Core ((TM) i3-3120M CPU @ 2.5GHz) and the memory size of 4.0GB. The tools used for the experimental programs are Python platform 3.9.2, Anaconda 3, Jupyter notebook and spider, where libraries like "surprise" a Python Scikit, Numpy, Pandas, Matplotlib and sklearn are imported to implement the required functionality for the EMRS.

4.3. Evaluation Metrics

The performance of the proposed EMRS is evaluated using the following performance metrics:

- **Precision:** Precision measures the percentage of interesting items suggested to the users with respect to the total number of suggested items. Precision represents the probability that a selected item is relevant. It is computed as shown in Equation (3).

$$\text{Precision} = \left(\frac{\text{\#Correctly recommended items}}{\text{\#Total recommended items}}\right) \tag{3}$$

In this research Precision was used to determined the relevance of the recommendations, because it is typically interested only in binary values, that is, either the item was selected (1) or not (0).

- **Root Mean Square Error (RMSE):** is a frequently used measure of the differences between values predicted by a model and the values observed. RMSE is more widely used in computing the prediction accuracy of the recommender system [20]. The RMSE between the predicted and actual ratings is described in Equation (4).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2} \tag{4}$$

Where, n is the number of corresponding ratings-prediction pairs, y is the actual rating and $\hat{y}$ is the predicted rating. Therefore, RMSE is used in this research to evaluate the accuracy of recommendations.

5. Results

This section presents the results of the experiments of the proposed EMRS compared to MRS based on two datasets namely MovieLens-100K and MovieLens-1M in terms of precision and RMSE metrics at different value of K-Nearest Neighbour (K). The results in Fig. 2 and Fig. 3 show that, EMSR works better and shows better results than the regular MRS approach. The Precision of EMRS is 49.8% which is higher than that of MRS 48.2%. The accuracy of EMRS is improved for the RMSE value of 0.071 obtained which is lesser than that of MRS when the value of K=10 and also at different value of K as shown in Figure 2 and Figure 3 when tested on 100K MovieLens Dataset. Also, on the 1M MovieLens Datasets EMRS outperformed MRS as shown in Fig. 4 and Fig. 5 with small decrease in Precision and increase in RMSE for both EMRS and MRS. This implies that, the proposed EMRS can provide more relevant and accurate recommendations even of the user preference changes.

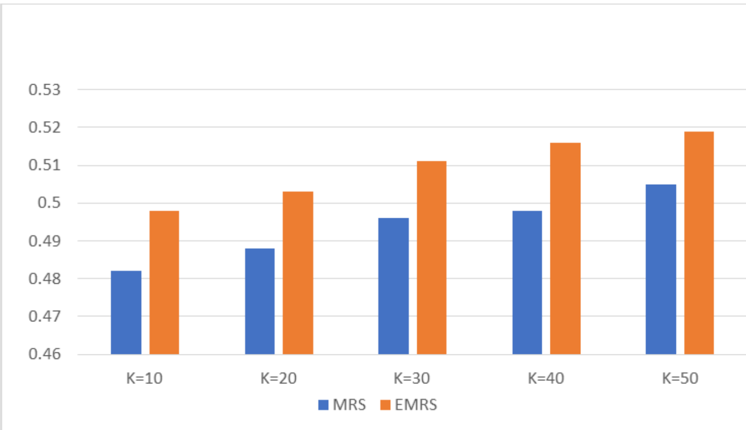


Fig.2. Precision of MRS and EMRS on MovieLens-100K

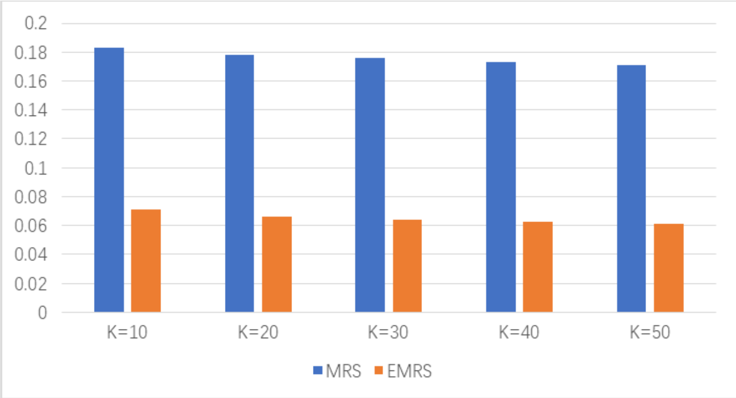


Fig.3. RMSE of MRS and EMRS on MovieLens-100K

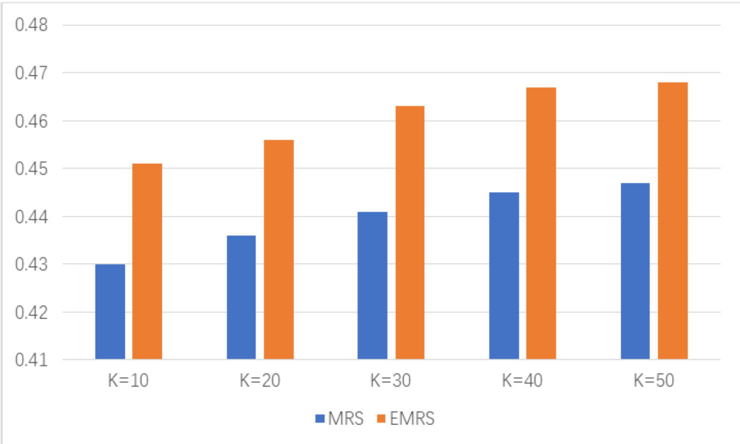


Fig.4. Precision of MRS and EMRS on MovieLens-1M

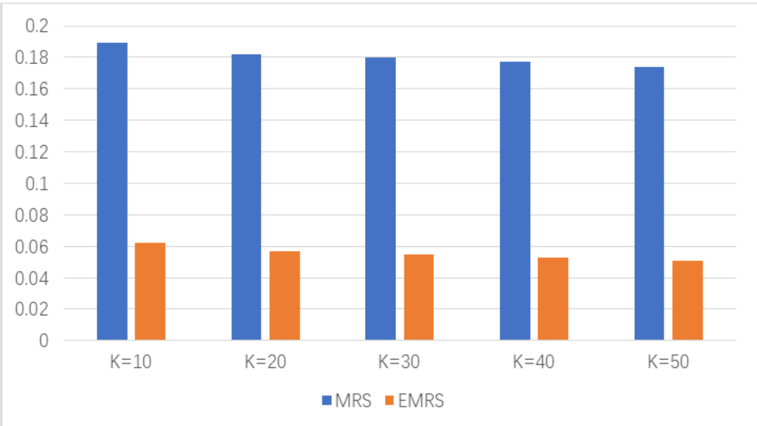


Fig.5. RMSE of MRS and EMRS on MovieLens-1M

Fig. 2 and 4 demonstrated the precision of the proposed EMRS as compared to the existing system (MRS). The figures show that EMRS has a higher precision values compared to the MRS. The reason for this performance is because; the EMRS takes users input on the recommendations to users thereby enhancing the relevance of the movies to be recommended. Fig. 3 and 5 demonstrated the RMSE of the proposed EMRS as compared to the MRS. The figures show that EMRS has a lower RSME values MRS. The reason for this performance is because the EMRS incorporates user preference changes with time since the users movies of interest changes from time to time. Therefore, the proposed EMRS not only enhanced the accuracy of the recommendations but also provided interested movies to users.

6. Conclusions

The existing MRS is unable to solicit for user inputs on the recommended movies and also fails to incorporate users' preference changes with time thereby affecting quality and relevancy of the recommendations to the active users.

In this research work, preference changes through users' input are incorporated in the existing Collaborative Filtering Movie Recommender System to enhance its operations. The enhanced system called EMRS is developed to assist users by providing relevant movie recommendations based on users most recent preferences. This will go a long way in helping users to deal with information overload by getting better recommendations at a right time. The EMRS is using simple hybrid of content-based filtering and collaborative filtering. The advantage of this hybrid system is its ability to combine the content-based filtering approach and collaborative filtering approach, to enables movie predictions to be improved. This movie recommendation engine is developed using KNN to provide relevant recommendations. The results of this study EMRS shows Precision 49.8% and RMSE value of 0.071 as against that of MRS with precision of 48.2% and RMSE value of 0.202. This implies that, the enhanced system utilises KNN to provide better result in terms of giving good and accurate recommendations. It also improves the recommendations of relevance movies based on the values of precision obtained in this study.

However, further extension of this study could be investigated such as considering other approaches of KNN using different datasets, as well as investigating the usage of KNN in movie recommender systems from other aspects such as economic and social perspectives.

References

[1] Rajani C., and Philippe L. A personalized Recommender System from Probabilistic Relational Model and Users' Preferences. *18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014, Procedia Computer Science, 4(11):469-473, 2014.*

[2] Eyrun E., Gaurangi T., and Nan L. A Movie Recommendation System (MovieGEN). *Unpublished. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.4954&rep=rep1&type=pdf, 2008.*

[3] Manoj K., Yadav D., Ankur S., and Vijay K. A Movie Recommender System: MOVREC. *International Journal of Computer Applications, 124(3):7-16, 2015.*

[4] Rupali H., Ajinkya G., Kevin S., Jeet G., and Vrushal K. Moviemender- A Movie Recommender System, *International Journal of Engineering Sciences & Research Technology, 5(11)*:469-473, 2016.

[5] Shreya Agrawal and Pooja Jain. An Improved Approach for Movie Recommendation System. *International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017) 3(17), 336-342, 2017.*

[6] Muyeed A., Mir Tahsin I., and Raiyan K. Movie Recommendation System Using Clustering and Pattern Recognition Network. *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC 2018). 143 –147, 2018.*

[7] Meenu G., Aditya T., Vishal G., Dhruv P., and Singh R. Movie Recommender System Using Collaborative Filtering. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), 415-420, 2020.*

[8] Michael D. E., John T. R. and Joseph A. K.. Collaborative Filtering Recommender Systems. *Foundation and Trends in Human-Computer interaction, 4(2), 81-173,* 2010.

[9] Sanjeev D., Kulvinder S., and Naveen K.. Collaborative Filterng Based Product Recommendation System for Online Social Networks. *International Journal of Computer Science, Engineering and Information Technology. 2(4):126-131, 2017.*

[10] Harpreet K. V, Maninder S. and Amritpal S. Analysis and Design of Hybrid Online Movie Recommender System. *International Journal of Innovations in Engineering and Technology (IJIET) 5(2), 159-162, 2015.*

[11] Prerana K., and Shabnam P. Effective Hybrid Recommender Approach using Improved K-means And Similarity. *International Journal of Computer Trends and Technology (IJCTT), 36(3):20-31, 2016.*

[12] Ponnam L., Deepak P., Nallagulla S., & Yellamati S. Movie recommender system using Item-Based Collaborative Filtering Technique. *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS) 1:55-60, 2016.*

[13] Hafed Z., Ziad A., Mahmoud A., and Yaser J. A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques. *2018 9th International Conference on Information and Communication Systems (ICICS), 2(18):102-106, 2018.*

[14] Badrul S., George K., Joseph K., and John R. Item-based Collaborative Filtering Recommendation Algorithms. *WWW '01: Proceedings of the 10th international conference on World Wide Web. 285-295, 2001.*

[15] Mohammed E,, Badr H and Abdelkader G. Building Recommendation Systems Using The Algorithms KNN and SVD. *International Journal of Recent Contributions from Engineering, Science & IT (iJES), 09(1):71-80, 2021.*

[16] Almu A. and Bello Z.. An Experimental Study on the Accuracy and Efficiency of some Similarity Measures for Collaborative Filtering Recommender System. *International Journal of Computer Engineering in Research Trends, 8(2), 33-3,* 2021.

[17] GroupLens Research (2019). Movielens, retrieved from https://grouplens.org/datasets/movielens/

[18] Roko, A., Almu, A., Mohammed, A., and Saidu, I. An Enhanced Data Sparsity Reduction Method for Effective Collaborative Filtering Recommendations. *International Journal of Education and Management Engineering(IJEME),* 10(1), 27-42, 2020.

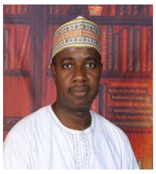
[19] Harper, F., M., and Konstan, J., A. The MovieLens Datasets: History and Context.. *ACM Transactions on Interactive Intelligent Systems (TiiS*), 5(4), 1-19, 2015.

[20] Mingang C. and Pan L. Performance Evaluation of Recommender Systems. *International Journal of Performability Engineering, 13(8):1246-1256, 2017.*

Authors' Profiles

Abba Almu received his PhD in Computer Science from Usmanu Danfodiyo University, Sokoto. Masters Degree in Computing: Information Engineering from Robert Gordon University, Aberdeen United Kingdom. He also received his B.Sc. (Hons) in Computer Science from Usmanu Danfodiyo University, Sokoto. He is currently a Senior Lecturer at the Computer Science Unit of Usmanu Danfodiyo University, Sokoto Nigeria. His research interests include information retrieval, recommender systems and machine learning.

Aliyu Ahmad received his B.Sc. (Hons) in Computer Science from Gombe State University. He is currently a Masters Student at the Computer Science Unit, Usmanu Danfodiyo University, Sokoto, Nigeria. His research interest includes information systems and recommender systems.



Abubakar Roko is a Reader at the Computer Science Unit, Faculty of Science, Usmanu Danfodiyo Univesity Sokoto, Nigeria. He received his PhD from Univeristy Putra Malaysia in 2016, specializing in the field of XML Retrieval. Currently, his research interest is in the area of text data management and analysis, focusing in particular on query processing in Information retrieval, Recommender Systems, and Sentiment Analysis.

Mansur Aliyu is a Senior Lecturer at the Department of Computer Science, Sokoto State University, Nigeria. He received his PhD from International Islamic University Malaysia (IIUM). His research interests include HCI, UX, User Evaluation and Cybersecurity.

How to cite this paper: Abba Almu, Aliyu Ahmad, Abubakar Roko, Mansur Aliyu, "Incorporating Preference Changes through Users' Input in Collaborative Filtering Movie Recommender System", International Journal of Information Technology and Computer Science(IJITCS), Vol.14, No.4, pp.48-56, 2022. DOI:10.5815/ijitcs.2022.04.05