WORLD ORGANIZATION OF APPLIED SCIENCES JOURNALS

International Journal of Information Technology and Applied Sciences
ISSN (2709-2208)
Int. J. Inf. Tec. App. Sci. 3, No.3 (July-2021)
https://doi.org/10.52502/ijitas.v3i3.128

A Hybrid Movie Recommender System and Rating Prediction Model

Muhammad Sanwal[1], Cafer ÇALIŞKAN[2]

[1] Electrical and Computer Engineering, Antalya, Turkey
[2] Electrical and Computer Engineering, Antalya, Turkey

Abstract: In the current era, a rapid increase in data volume produces redundant information on the internet. This predicts the appropriate items for users a great challenge in information systems. As a result, recommender systems have emerged in this decade to resolve such problems. Various e-commerce platforms such as Amazon and Netflix prefer using some decent systems to recommend their items to users. In literature, multiple methods such as matrix factorization and collaborative filtering exist and have been implemented for a long time, however recent studies show that some other approaches, especially using artificial neural networks, have promising improvements in this area of research.

In this research, we propose a new hybrid recommender system that results in better performance. In the proposed system, the users are divided into two main categories, namely average users, and non-average users. Then, various machine learning and deep learning methods are applied within these categories to achieve better results. Some methods such as decision trees, support vector regression, and random forest are applied to the average users. On the other side, matrix factorization, collaborative filtering, and some deep learning methods are implemented for non-average users. This approach achieves better compared to the traditional methods.

Keywords: Recommender systems, matrix factorization, collaborative filtering, hybrid systems, decision tree method, support vector regression, random forest method.

1. INTRODUCTION

In the current era of modern technology, the amount of information is increasing rapidly. Available information on the internet is not relevant to the users' needs and preferences [1]. Most of the users spend their precious time navigating towards useful information. Therefore, recommender systems are getting popular especially with the rapid growth of e-commerce. Such systems provide the best solutions for this problem. E-commerce platforms such as Netflix, Amazon Prime have millions of users with millions of items to offer [2]. As a result, it is a great challenge for these companies to recommend items to the users according to their preferences. In this sense, a recommender system is one of the modern tools to solve this sort of problem in the current era.

Recommender systems are categorized into three types, namely content-based filtering, collaborative filtering, and knowledge-based filtering [3]. Generally, items are recommended on a similarity basis either on a user profile or an item profile. These approaches find the similarities among users or items, then make suggestions to specific users according to their profiles. They mostly rely on explicit feedback, meaning that users provide explicit input regarding their interests in various products into the system. For example, Netflix benefits from the star rating system in which users submit their evaluations after watching movies.

In the current decade, hybrid recommender systems are emerging as successful solutions as hybrid systems achieve better results compared to the conventional methods[4]. These methods have overcome the issues due to the weaknesses of recommendation techniques by replacing them with the strength of other techniques, so their performances depend upon the integration of their components.

Mostly in the application, databases have large numbers of items, and this makes it very difficult for any user to view or rate all the items. As a result, every user visits or rates just a limited number of items in the related database which results in sparse user-item matrices for recommender systems. This also makes it very challenging to recommend the desired items to a specific user. Moreover, another main challenge in this field is fetching the appropriate features for the items themselves.

Furthermore, for a new user in the system, it is difficult to recommend personalized items as this user's priorities are not known. This leads us to the cold start problem which is still one of the major issues in recommender systems.

The main objective of this paper is to recommend items to the users according to their interests with a new hybrid approach. As the user-item matrix is sparse, it is very difficult to recommend items to the users. The sparsity of the matrix leads to a decrease in the efficiency of recommender systems as it does not contain enough ratings which should help in the recommendation. In this research, we combine different methods such as decision trees, support vector regression, matrix factorization, and artificial neural networks and obtain better results.

The paper is organized as follows. Section 2 includes a brief literature review, then section 3 presents the preliminaries of the existing methods in the literature. Section 4 includes a brief description of the data set that is used in this study and the proposed hybrid approach. After that section 5 discusses the results belonging to the proposed approach and finally, section 6 has a comparison of the results of the proposed approach with the previously applied methods.

2. LITERATURE REVIEW

In the related literature, multiple techniques have been studied and applied for recommendation purposes. The methods such as content-based filtering and collaborative filtering are the most effective ones in such systems. In what follows, we discuss these techniques in detail.

The content-based approach depends on the content viewed or rated by the user. For better recommendations, it requires a user profile that holds information about the user's activity and preferences through the items. It is generated by keyword analysis, previously seen, and rated items. In general, it involves the latest activity of the user, and the system considers the positively and negatively rated items for each user. Then the recommendation is made according to the user's captured preference from the activities. The recommended items are generally similar to the positively rated items or high rated items overall in the database [5].

The collaborative filtering approach uses multiple filtering techniques for recommendation purposes. Some well-known collaborative filtering techniques are item-based filtering, user-based filtering, neighborhood-based filtering, and model-based filtering techniques. In a user-based approach, the user performs a major role in the recommendation process. The user's behavior determines which items to be recommended. For instance, if a user likes an item from a particular category, then more items will be recommended from the same category to this user. Moreover, in this approach, items from the same set of users can be recommended to specific users. For example, if a user likes an item called A, then items all liked by a group of people who have also liked item A will be recommended to this user. This approach determines the common behaviors of the users according to their liked items, and then, later items are recommended to a user according to his behavior/neighborhood [6]. Mostly, these so-called neighborhood models are used in this approach. Such user-oriented systems estimate the ratings of an item, based on the similarity of the items or users.

Pu and Hong Wu [7] propose a collaborative filtering algorithm for personalized recommendations of an item to the users. In this study, after the slope one algorithm is used to fill the gaps of the user-item matrix to reduce the sparsity, they implement collaborative filtering for the recommendation of items to the users. The slope one algorithm utilizes information such as ratings of a specific item from other users and all items rated by the same user. Given two items I_i and I_j the algorithm considers the average deviation of item I_i concerning item I_j. The second step involves calculation of predictions. Furthermore, for the measurement of user rating similarity Pearson's correlation is used to measure the correlation between two vectors of rating. Moreover, for recommendation purposes, a weighted average of neighbor's ratings is calculated according to the target user.

The item-based approach uses the data and history of items rated or viewed by the user for a recommendation. It is obvious that the preferences of a user remain similar or slightly change over time so a similar approach as user-based is applied here. Items with similar ratings or content are recommended to the user. The same neighborhood method is applied to the items as it recommends a similar item according to the user's preference [8].

Yifan et al. [9] propose a model that does not consider any direct inputs from the users regarding their preferences. In this work, the data set was treated as positive and negative indications of an item with varying confidence levels about an item towards a specific user. In this work, for a given user u, they introduce a set of binary variables that indicate the user preference p_{ui} to any specific item i. For example, if the user u has purchased an item i then the user u likes the item and if the user u does not purchase the item i then the user u doesn't prefer the item i . Moreover, in this study confidence levels vary according to the behavior of the user. In the beginning, if the user likes the very first item, then the confidence level is much lower compared to a user who likes the series of similar items. For instance, a user may watch a TV show because s/he is staying on the same channel after watching a previous show. It does not mean that s/he likes the current show very much. This model has different confidence levels for different users and items. But confidence level grows as the model has a stronger indication that the user likes the specific type of items. To measure the confidence level, a new variable is introduced, and the value of the confidence level increases with more user preferences about items.

The model-based approaches use some machine and deep learning algorithms to learn new ratings by analyzing the previously given ratings. These methods are very fast in computation, and they are likely to result in more accurate predictions. Once the model is trained, it can make predictions very quickly on the new data entries in the database. Examples of these techniques are included in [10].

Ajesh et. al. [11] propose a system that uses clustering and random forest algorithms for recommendation purposes. This recommendation system is designed on user ratings and evaluated by computing accuracy and mean square error. Moreover, the users are clustered based on user ratings for each movie.

Mohammad et al. [12] propose a collaborative filtering method that implies matrix factorization and deep neural networks under its framework. This study mainly proposes a recommender system in which a deep neural network replaces the inner product of the matrix factorization so it can learn the non-linearities of the system. A single user can rate multiple items in the personalized recommender system. For this reason, unique users are formulated against unique items and transformed into a multidimensional space. So, user and item embeddings are used to represent the latent features of the users and movies which determine the strength of the relationship for each user.

Xiangnan [13] proposes a neural network approach for collaborative filtering. In the proposed approach, the inner product of matrix factorization is replaced with a neural network architecture that can learn an arbitrary function from the data. Moreover, multiple layers are introduced in this research. The first layer maps the embeddings of users and items to learn more features about both entities to find a powerful relationship and then neural collaborative filtering layers are used to find the most promising relationships between the entities and prediction.

3. Preliminaries

In this section, we will introduce the background of user-items interaction in multiple studies using techniques such as collaborative filtering, matrix factorization. We also apply random forest and artificial neural network methods.

A. Collaborative Filtering

For U users and I items, the profiles of the users are represented in an U x I user-item matrix X. If the entry X_{ui} has the value r in matrix X, it indicates that user u rated item i by r, where r is a rating from a scale of pre-determined interval (mostly 1-10). If $X_{ui} = \emptyset$, it means that the rating for the movie is unknown for a specific user.

This methodology personalizes the score for each user to each given item. Firstly, it calculates the average rating of all items rated by a user.

The average rating of a user will help to calculate the deviation between the user's new rating from his/her average rating for other items. The average rating avg_u of a user u is calculated as follows:

We first compute that

$$avg_u = \frac{1}{|\mu_u|} \cdot \sum_{u \in \Omega} X_{ui} \qquad (1)$$

where $u = (1, \dots, U)$, $i = (1, \dots, I)$, μ_u is the total items rated by user u and, Ω is the set of all users.

Afterward, this method calculates the average deviation of an item for all users. The deviation of a user is the difference between his/her average rating and a given rating to an item i. Firstly, the deviation of a single user u for an item i is calculated as follows:

$$dev_{ui} = X_{ui} - avg_u \qquad (2)$$

Now, it calculates the average deviation of all users for an item i as follows:

$$avg_d\,(i) = \frac{1}{|\Omega_u|} \cdot \sum_{u \in \Omega} X_{ui} - avg_u \qquad (3)$$

where $|\Omega_u|$ is the number of users who rated item i. The above calculated average deviation provides the predicted deviation of an item. The concept of deviation is introduced to remove the biases from user’s ratings. For example, user A may have a different approach in the interpretation of good/bad ratings compared to user B. For instance, user A prefer using rating 5 for liked-items and 3 for unliked-items. Whereas user B prefers 4 for liked-items and 1 or 2 for unliked-items. The calculation of deviation will minimize the biases in the user’s ratings. Moreover, a limited number of users are included to calculate the average deviation. These limited users are neighbors between 25-50 in numbers that have similar ratings as user u.

Furthermore, this method adds the average rating of a specific user equation (1) and the average deviation of all users to predict the unknown rating for a specific item for a particular user in equation (3) as follows:

$$pred_{ui} = avg_u + avg_d\,(u) \qquad (4)$$

Furthermore, the weights are introduced in equation (4) to improve accuracy. The weight is higher if the deviation between two users is minimum otherwise smaller. For instance, if the deviation of user A for an item i is similar to the user B rating, then the weight of the user A rating in computing the predicted rating of user B will be larger otherwise smaller. The weights are included in the above eq (4) as follows:

$$pred_{ui} = avg_u + \frac{\sum_{u\in\Omega} w_{u_{1\to U}}\{\, avg_d\ (i)\}}{\sum_{u\in\Omega} |w_{u_{1\to U}}|} \tag{5}$$

Where $w_{u_{1\to U}}$ are the calculated weights between user u and all other users for a specific item i. The weights are calculated using the Pearson Correlation Coefficient in the above equation. Finally, Mean Squared Error is calculated between the actual rating and the predicted rating.

B. Matrix Factorization

The matrix factorization is a highly successful model that contains implicit feedback. The information is not directly given by the user towards the item but it can be derived by analyzing the behavior of the user towards the items by utilizing the user-item interaction matrix. This technique is useful for the estimation or prediction of user ratings towards any specific item. The rating matrix R is the approximation of the product between two matrices $P\ and\ Q$, where $\boldsymbol{P} \in \mathbb{R}^{u\,\mathrm{x}\,k}$ is the user latent matrix and $\boldsymbol{Q} \in \mathbb{R}^{i\,\mathrm{x}\,k}$ is the item latent matrix, where k is the total number of latent factors and $k \ll u, i$.

Accordingly, each item i is associated with a vector $q_i \in \mathbb{R}^{i\,\mathrm{x}\,k}$, and each user u is associated with a vector $p_u \in \mathbb{R}^{u\,\mathrm{x}\,k}$. For a given item i the element of q_i measures that how much an item possesses those factors, positive or negative, and for given user u the element of p_u measures how much user u has an interest in the specific item. The resulting dot product $q_i \cdot p_u$ predict the interaction between the user and an item as follows

$$pred_{interaction} = q_i^T p_u \tag{6}$$

Where $pred_{ui}$ is the approximation of predicted interaction.

In the previous research, the proposed systems relied on imputation to fulfill the missing values to make the matrix dense but modern studies such as [14] and [15] prefer to model the observed ratings directly. This study minimizes the regularized squared error on the given known ratings while learning the factor vectors of $q_i\ and\ p_u$ as follows:

$$\min_{q^*,p^*} \sum_{(u,i)\in k}^{n} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|)^2 + \lambda(\|p_u\|)^2 \tag{7}$$

where the term $\lambda(\|q_i\|)^2 + \lambda(\|p_u\|)^2$ is a regularization term, it is added to avoid overfitting of decomposed matrix P and Q to the original matrix R and k is the set of (u, i) pair for which rating is known. The main goal of the model is to predict future ratings based on previously known ratings. Furthermore, the study adds bias to the above equation as some items may be perceived better than the other items in the matrix. For instance, a movie can be biased on genre, production company, director, or the user can be biased while rating the items. Now, the prediction rating equation is updated as follows:

$$pred_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{8}$$

where μ denote the overall average rating of the user, $b_i\ and\ b_u$ denotes the biases of item i and user u respectively. Now, the squared error function will be updated as follows:

$$\min_{\mathrm{m}^{q},p^*,b^*} \sum_{(u,i)\in k}^{n} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2\ ) \tag{9}$$

4. The Proposed Approach

In this section, we propose a new approach in which the users are categorized into two types, namely average and non-average users. The concept of defining a user as average originates from the overall comparison of the user's original ratings (over 1-5 scale) with the corresponding IMDb ratings (over 1-10 scale) for the same items. In what follows whenever a comparison between these two ratings is needed, the IMDb rating value is halved, so it is also considered to be over 1-5 scale similar to the user's original ratings. We denote by $r_{A\to M}$ and $r_{IMDb\to M}$ the rating of user A for the movie M and IMDb rating for the same movie M, respectively.

Average/non-average users are defined according to the following criteria.

- Let's assume that user A has rated some movies M_i, and for each movie, the difference between the user's rating and the corresponding IMDb rating, i.e. $\delta_i = |r_{A\to M_i} - r_{IMDb\to M_i}|$ is computed.
- A user is said to be average with parameters C and δ_0 if, for at least C% of the movies rated by this user, it satisfies that $\delta_i \le \delta_0$.
- All other users who are not satisfying the above criterion are said to be non-average.

In this research, it is assumed that $C = 80$ and so it is easier to control the number of average users by just depending on the value of δ_0.

A. Data Set

The dataset used for this research is MovieLens [16]. It has been collected over multiple times of duration and contains 25 million entries of ratings of movies rated by users. It includes over 62,000 movies and 162,000 users. Users have been selected randomly and every user has rated at least 20 movies. Each user and movie are represented by userId and movieId. Furthermore, we have combined the IMDb dataset [17] for the categorizing of users into average and non-average users. In this research, a subset of this dataset is used in which it has a size of one million ratings and around 7000 users [18].

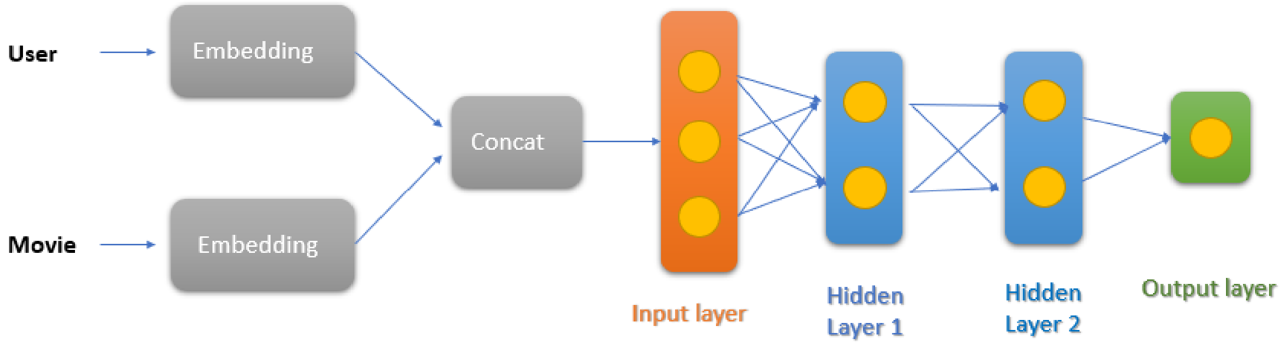

Figure 1. Illustration of the proposed Artificial Neural Network.

B. Methods

In this research, various methods are applied to the dataset. Throughout the article, we assume that 70% of the dataset is used for training the model and 30% of the data is applied for testing purposes. Moreover, Mean Absolute Error and Mean Squared Error is calculated for the accuracy of the models.

We firstly, apply the random forest method and SVR on average and non-average users. A total of 300 estimators are used for the model evaluation. Then we continue with the implementation of collaborative filtering on the dataset. This method is the same one mentioned earlier. Moreover, we just applied this technique only to non-average users, and consequently, we achieved better results. The matrix factorization method described preliminary section is also applied to non-average users.

Artificial neural networks have not been used widely in the field of recommendation systems. For the implementation of matrix factorization using a neural network, this study adopts multi-layer representation to model a user-item interaction. In this method, we have replaced the inner product of Matrix Factorization with neural network architecture. In the first layer, user and item embeddings are concatenated and given to the first layer as an input feature. As this is a regression problem the output layer only consists of one layer. In the beginning, a dense layer of 600 neurons with batch normalization and *ReLU* activation function. Afterward, a dropout of 0.3 and 200 neurons are used. The proposed model is represented in Fig. 01.

5. RESULTS

In this section, we will discuss about the results of the applied methods in this research.

The following tables show the results of the random forest algorithm on both average and non-average people with the difference between actual and IMDb ratings.

TABLE I. RESULTS OF RANDOM FOREST WITH AVERAGE PEOPLE

δ_0	**Results of Random Forest with Average people, with $C = 80$.**		
	No. of Avg People	***MSE***	***MAE***
1.00	2519	0.66	0.62
0.95	1575	0.62	0.60
0.90	1461	0.60	0.59
0.85	1019	0.54	0.56
0.80	879	0.53	0.55
0.75	612	0.51	0.54

TABLE II. RESULTS OF RANDOM FOREST WITH AVERAGE PEOPLE

δ_0	**Results of Random Forest with Non-Average people, with $C = 80$.**		
	No. of Non-avg People	***MSE***	***MAE***
1.00	4229	1.40	0.92
0.95	5173	1.29	0.89
0.90	5287	1.28	0.88
0.85	5729	1.26	0.87
0.80	5869	1.25	0.87
0.75	6136	1.23	0.86

The following tables show the results of the SVR algorithm on both average and non-average people with the difference between actual and IMDb ratings.

TABLE III. RESULTS OF SVR WITH AVERAGE PEOPLE

$\boldsymbol{\delta_0}$	**Results of SVR with Average people, with $\boldsymbol{C = 80}$.**		
	No. of avg People	***MSE***	***MAE***
1.00	2519	0.69	0.68
0.95	1575	0.69	0.67
0.90	1461	0.67	0.66
0.85	1019	0.63	0.64
0.80	879	0.60	0.62
0.75	612	0.59	0.60

TABLE IV. RESULTS OF SVR WITH NON-AVERAGE PEOPLE

$\boldsymbol{\delta_0}$	**Results of SVR with Non-Average people, with $\boldsymbol{C = 80}$.**		
	No. of avg People	***MSE***	***MAE***
1.00	4229	1.38	0.94
0.95	5173	1.29	0.89
0.90	5287	1.27	0.89
0.85	5729	1.18	0.85
0.80	5869	1.18	0.85
0.75	6136	1.17	0.84

For the collaborative filtering approach, the non-average people dataset has been used for the evaluation of this method. The approximate train and test dataset for the non-average people with $\delta_0 \geq 0.85$ is 80% of the original dataset. After training the model, we achieved Mean Squared Error and Mean Absolute Error of 0.66 and 0.61, respectively. Moreover, we achieved Mean Squared Error and Mean absolute error of 0.91 and 0.73 respectively after testing the model.

For the Matrix Factorization approach, the non-average people dataset has been used for the evaluation of this method. The approximate train and test dataset for the non-average people with $\delta_0 \geq 0.85$ is 80% of the original dataset. Keras embedding layers are implemented to learn the embeddings for both users and movies. Moreover, user and items embeddings $(k = 10)$ are used for learning features, epoch size of 15 is considered and the learning rate of 0.07 is used after parameter tuning. After training the model, we achieved Mean Squared Error and Mean Absolute Error of 0.53 and 0.53 respectively. Moreover, we achieved Mean Squared Error and Mean absolute error of 0.78 and 0.53 respectively after testing the model. Figure 2 illustrates the results of the above-mentioned model.

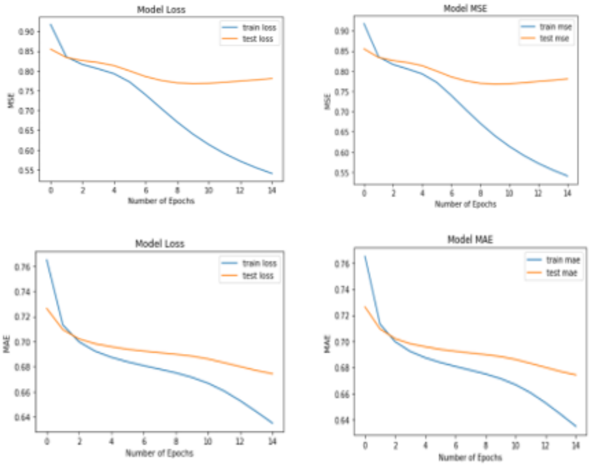

Figure 2. Results of Matrix Factorization.

For the Neural Network approach, the non-average people dataset has been used for the evaluation of this method. The approximate train and test dataset for the non-average people with $\delta_0 \geq 0.85$ is 80% of the original dataset. Keras embedding layers are implemented to learn the embeddings for both users and movies. Moreover, user and items embeddings $(k = 15)$ are used for learning features, epoch size of 15 is considered and the learning rate of 0.07 is used after parameter tuning. After training the model, we achieved Mean Squared Error and Mean Absolute Error of 0.68 and 0.62 respectively. Moreover, we achieved Mean Squared Error and Mean absolute error of 0.80 and 0.67 respectively after testing the model.

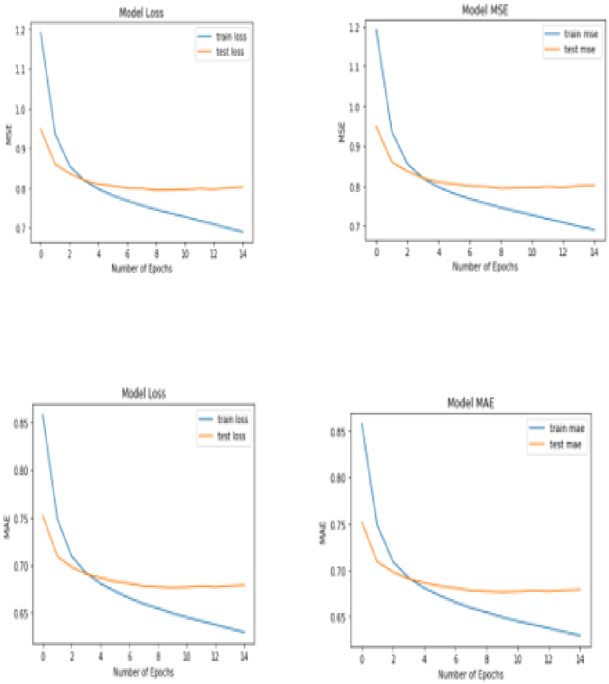

Figure 3. Results of the Artificial Neural Network.

TABLE V. COMPARISON OF THE PROPOSED METHOD WITH SIMILAR METHODS IN THE LITERATURE.

Collaborative Filtering Techniques		*Matrix Factorization*		*Artificial Neural Network*		*References*
MAE	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	
0.84	-	-	-	-	-	[21]
-	-	-	-	-	0.81	[22]
-	-	-	-	-	0.77	[23]
0.78	-	0.74	-	0.69	-	[12]
0.81	-	-	-	-	-	[27]
0.84	-	-	-	-	-	[28]
0.69	-	-	-	-	-	[24]
0.73	0.91	0.67	0.78	0.67	0.80	(Proposed Model)

6. Discussion and Future Work

In this chapter, we will discuss the results of our proposed method and other similar approaches in the literature. The proposed method is categorized into average and non-average users. To decide whether the user is average or non-average, the dataset was tested on multiple values to achieve the optimized results. The criterion to determine the average and non-average users are determined in such a way that at least 10% of the total users should belong to the average users. The specific values for the criterion are described in section 4. The major purpose of dividing the total users into two different categories is to recommend better items from a similar set of people.

The following figure illustrates the workflow of the algorithm proposed in this research. In the beginning, the model will decide either the user is average or non-average. After taking a decision it will run the appropriate method for a specific user. For example, if the new user is average then it will evaluate Random Forest and if the new user is non-average, then it will evaluate different methods such as Matrix Factorization, Collaborative Filtering (user-based), and Deep Learning method.

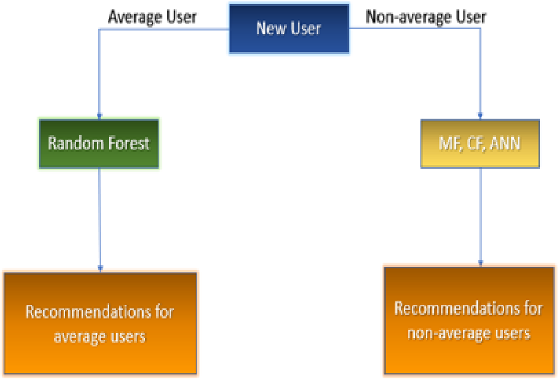
Average User
New User
Non-average User
Random Forest
MF, CF, ANN
Recommendations for average users
Recommendations for non-average users

Figure 4 Proposed Model.

In our research, for average users, Random Forest produced the best results and for non-average users, the Matrix Factorization method produced the best results. Although the results of Matrix Factorization and Artificial Neural Network are almost similar. We believe that better results can be achieved by tuning the different parameters.

The experiments in our proposed research suggest that for a new user the possibility of being an average user is at least 10% and the non-average user is 90% approximately. For the average user, the MSE value in the case of Random Forest is around 0.50. The MAE value for the non-average user is 0.67 in the case of ANNs and MF.

The following table illustrates the results of our method and other similar approaches. Our results shown in the table are based on non-average people.

For comparing the results, MSE and MAE are calculated and compared as an evaluation matrix. It is clear from the above table that our model produced better results than the previous approaches. Moreover, our results on average people are much better than the above-shown results. Furthermore, our results are also even better on non-average people than the compared models. The overall performance of the proposed method has improved significantly.

In the future, we will focus on the implementation of some link prediction methods such as [25] and [26] to better understand the correlation between users and their behaviors. Link prediction techniques such as meta-paths can achieve better results if we categorize the users as proposed in our research. Autoencoder and Decoder have also shown promising improvement in this field of research. Moreover, sparsity and cold start problems are still difficult to resolve even in this decade. Our goal is to handle the sparsity and cold start problems suitably. Furthermore, we can learn different methods for the vector representation of the features in the user-item matrix which can be helpful to improve the results.

REFERENCES

[1] G. Gediminas A and Alexander T 2005, "Toward the next generation of recommender: a survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. June 2005, pp. 734-749, 2005.

[2] J. James B Charles P Bing L Domonkos T and Padhraic S 2007, "The Netflix Prize", KDD Cup and Workshop," www.netflixprize. com, 2007.

[3] J Sarika Anjali G Parveen T and Sourabh C 2015, "Trends, problems, and solutions of a recommender system," International Conference on Computing, Communication & Automation, Noida, pp. 955-958, 2015.

[4] P Maria and Giuseppe M. L. Sarnè 2010, "A Neural Network Hybrid Recommender System," Frontiers in Artificial Intelligence and Applications., pp. 180-187, 2010.

[5] Asanov 2011, "Algorithms, and Methods in Recommender Systems," in Berlin Institute of Technology, Berlin, German, 2011.

[6] Z. Zhi-Dan Z and Ming S 2010, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop, 2010," Third International Conference on Knowledge Discovery and Data Mining, Phuket, 2010, pp. 478-481, 2010.

[7] W Pu and HongWu Y, "A Personalized Recommendation Algorithm Combining Slope One Scheme and User-Based Collaborative Filtering," International Conference on Industrial and Information Systems, Haikou, 2009, pp. 152-154, 2009.

[8] L. Lalita S Anju G 2013, "A Survey of Recommendation System: Research Challenges," International Journal of Engineering Trends and Technology (IJETT), pp. 2231-5381, May 2013.

[9] Y. Yifan H Yehdu K and Chris V 2008, "Collaborative Filtering for Implicit Feedback Datasets," Eighth IEEE International Conference on Data Mining, Pisa, 2008,, pp. 263-272, 2008.

[10] K. Uuszula 2013, "Advantages of information granulation in clustering algorithms Agents and artificial intelligence," Springer, NY (2013), pp. 131-145, 2013.

[11] A. A Ajesh Jayashree N and P S jijin 2016, "A random forest approach for rating based recommender systems.," 2016, pp. 1293-1297, 2016.

[12] M. Mohammad A and Shyla. A 2019, "A Recommender System Based on Deep Neural Network and Matrix Factorization for Collaborative Filtering," International Conference on Electrical, Computer and Communication Engineering (ECCE) Cox'sBazar, Bangladesh, 2019, pp. 1-5, 2019.

[13] L. Xiangnan H 2017, "Neural Collaborative Filtering," Proceedings of the 26th International Conference on World Wide Web (WWW '17, p. 173–182., 2017.

[14] S. Funk 2006, "Netflix Update: Try This at Home,," https://sifter.org/~simon/journal/20061211.html, 2006.

[15] Y. Yehuda K 2008, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model,," Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, ACM Press, 2008, pp. 426-434, 2008.

[16] MovieLens, "https://grouplens.org/datasets/movielens/," 2020. [Online].

[17] IMDb movies extensive dataset, "https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset," [Online].

[18] sanwal-1852, "Thesis-DataSets," https://github.com/sanwal-1852/Thesis-DataSets, 2020.

[19] W. Pu and HongWu Y 2009, "A Personalized Recommendation Algorithm Combining Slope One Scheme and User-Based Collaborative Filtering," International Conference on Industrial and Information Systems, Haikou, 2009, pp. 152-154, 2009.

[20] Y. Yehuda K Robert B and Chris V 2009, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37,, 2019.

[21] Y Wei 2015, "User's Interests-Based Movie Recommendation in Heterogeneous Network," International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI), pp. 74-77, Beijing, 2015.

[22] S Mojdesh, "Movie Recommender Systems: Implementation and Performance Evaluation".

[23] Z Libo 2018, "A Recommendation Model Based on Deep Neural Network," IEEE Access, vol. 6, 2018.

[24] S. Dushyant Bahl, " A novel hybrid approach towards movie recommender systems," Journal of Statistics and Management Systems, pp. 1049-1058, 2020.

[25] C. Chuan S Binbin H Wayne Z and Philip Y 2019, "Heterogeneous Information Network Embedding for Recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 2, pp. 357-370, 2019.

[26] B. Hu Chuan S Wayne Z and Philip Y 2018, "Leveraging Meta-path-based Context for Top- N Recommendation with A Neural Co-Attention Mode," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), 2018.

[27] Y. Xiaokun W 2017, "A New Similarity Computation Method in Collaborative Filtering Based Recommendation System," IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-5, 2017.

[28] P. Xiao 2013, ""Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie Recommendations," Fourth International Conference on Intelligent Systems Design and Engineering Applications, Zhangjiajie, 2013, pp. 349-352, 2013.