DOI: 10.21917/ijsc.2025.0511

MOVIE RECOMMENDATION USING MODIFIED NEURAL COLLABORATIVE FILTERING AND FP-GROWTH ALGORITHM

R. Bhavani[1] and K.G. Yamuna[2]

[1]*Department of Computer Science and Engineering, Government College of Engineering, Srirangam, India*
[2]*L&T Technology Services, Chennai, India*

Abstract

Recommendation system is a process of suggesting more likely items to the users based on their preferences and interest. Applications of recommendation system are seen in almost many areas like e-commerce, social media and multimedia platform. In recent days, the hybrid collaborative filtering techniques are used for the recommendation to improve the suggestions for users. Nowadays, collaborative filtering using neural networks is used in recommendation process. This paper aims to develop a modified neural collaborative filtering model to recommend movies to users. Movies are rated by the users in the scale of 1 to 5. Users prefer to watch movies not only based on the ratings but also like other factors like genre, cast, crew, etc. In this paper genre is also considered along with the ratings to train the neural model. The user-movie interaction matrix used by many collaborative filtering techniques suffers from sparsity. To overcome this problem, the modified neural collaborative filtering model is used to find the ratings of the movies not watched by the user. Finally a recommendation module using FP-Growth algorithm is developed to suggest users with movies considering ratings for the movie and genre of the movie. The loss function, mean absolute error is used to analyze the performance of the modified neural collaborative filtering model. The testing loss of trained model is found to be 0.1017. The recommendation module is evaluated using Normalized Discounted Cumulative Gain (NDCG) metric. The experimental results show that the proposed recommendation system performs better compared to Nearest neighbor and correlation based recommendation systems.

Keywords:

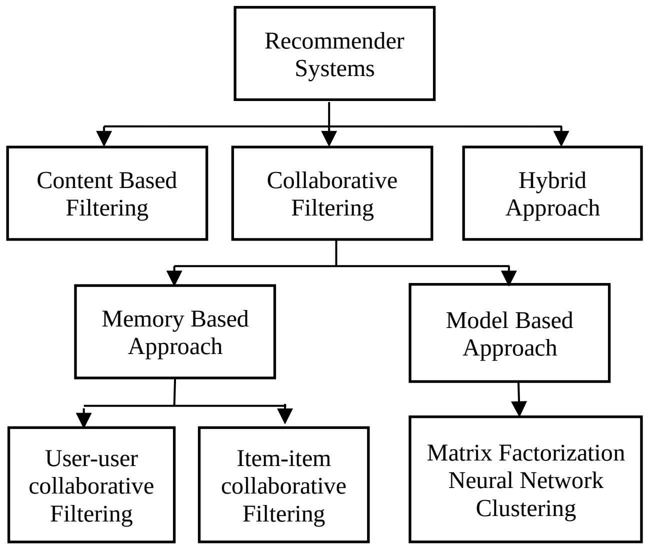
Collaborative Filtering, Neural Network, Recommendation System, Neural Collaborative Filtering, Movie Recommendation

1. INTRODUCTION

In the past, people depended on their peers in making decisions and utilized their recommendations in purchasing items, watching movies, listening to music, etc. Over the past two decades, the growth of digital information, especially in the web is exponential and this leads to the problem of information overload. Nowadays, online applications are available for teaching, conducting examinations, listening to music, watching movies and playing games, booking tickets and many more. These online applications capture data about preferences of the people and pool it in data repositories. Hence digital information is growing day by day.

As there are more applications available to do a single job, it limits the human capability to review the specifications, and it is hard to choose between numerous alternatives available in the online stores. The evolution and development of information technology provides us many advanced technologies like Computer Vision, Deep Learning, Edge Computing, Internet of Things (IoT), etc. Humans use these technologies in almost all part of their work to satisfy their needs. The major areas where these technologies are used are image classification, speech recognition, malware filtering, online trading, item recommendation and natural language processing.

In general recommendation systems are software tools and techniques used to forecast the interested entities to users [1]. Product recommendations include recommending books, music, purchase items, movies, etc. These systems emerged in mid 90's and with their developments, they are used everywhere for searching different types of interesting products and items. Movies are one of the popular entertainment media to people in the digital era. Due to the development of many streaming services like Netflix, Amazon prime, YouTube, etc., movies are watched by people in many genres like comedy, drama, documentary, comedy. Watching movies in the digital age is easier with the features in the smart phones. The availability of movies is very abundant and it takes more time to sort out the movies that the user prefers. Thus, there is a need to develop a very useful system that recommends movies to users that can search movie libraries automatically based on their previous watching history.



2. FIG.1. RECOMMENDER SYSTEM CLASSIFICATION

As shown in Fig.1, the recommender systems can be categorized into two broader types: content-based recommendation system and collaborative filtering methods [2]. The earlier approach compares the features of the new items with the features of previously preferred items and recommends the most similar items to users. This type of filtering approach suffers from a cold start problem. It is difficult to recommend items to a new user. Also the system suffers when new item is added the system. The interest of the user is unknown, and the preference of an item is also unknown.

Collaborative filtering works with the concept of collaborations. It is classified into memory-based approach and model based approach. Both the approahces starts by creating the user-item interaction matrix. In this matrix, rows contain user information and columns contain movie information. Each cell of the matrix is filled with the user rating on the particular item. According to the authors of [3], this interaction matrix is highly sparse and is a significant factor of considered in recommending items to users. Collaborative filtering can be either user-user interaction model or item-item interaction model. In user-user model, the similarity of different users to make predictions about what a particular user may like is done. Item–item approach works by finding the similarity of items to provide suggestions. The best performance is experienced with hybrid approach which uses both the types of filtering approaches.

The remaining part of this paper is organized in the following manner. Section 2 discucess the works in literature related to this study. Dataset used for experimentation and the problem under study is explained in section 3. The proposed methodology used is described in section 4 and section 5 elaborates the experimental results of the proposed method. Conclusion is given in section 6 of this paper.

3. RELATED WORK

A movie recommendation system that hybridzes K-means clustering with genetic algorithms was proposed [4]. The population space was reduced using Principal Component Analysis (PCA). This dimensionality reduction helps in reducing the computation complexity of hybrid movie recommendation model. The central issues in collaborative filtering such as, cold-start and sparsity was addressed in [5]. A hybrid methodology utilizing matrix factorization and nearest-neighbour in film recommendation systems was proposed. Metrics such as RMSE and MAE was used to examine the performance of the system. The system yielded the smallest RMSE value of 0.788649 for k=30. User satisfaction was also based on recommending a assorted set of items, as the system provided a larger set of items from which recommendations could be given in many ways.

Clustering based method was proposed for recommendation of movies, books and jokes [6]. In this system users could opt to change the levels of diversity of items in their recommendation lists. First, items were clustered related to the rating of items. Then metrics like z-diversity and standard deviation were used to provide the diversity levels to the users. User preferences study based on rating data about movies was carried out in [7]. First a bipartite hyper-network was built to map users' interests on movies using the rating data. Data-scaling was controlled using factors such as shrinking factor and stretching factor. The bipartite hyper-network were analyzed for close cliques by considering first-order h-neighbors to understand users' explicit preferences. It was observed from two-member cliques constructed using rating distribution of movies that many users had same type of interests on movies.

A K-means clustering based movie recommender system was developed in [8]. User were represented in n dimensional space. Movies were categorized using K-means clustering algorithm based on their viewing interests. From the clustered data, other users were compared with active users using Pearson correlation coefficient. Next nearest neighbor algorithm was used to provide top-30 recommendations. The developed collaborative filtering model was less time consuming and provided more accurate recommendations compared to other state-of-art methods. FCMR, a content-based movie recommender system was proposed in [9]. A neural network model was trained with features such as cast and crew. Initially, vector form of features were obtained for each movie. Similarity between each movie were calculated and finally movies with high similarity score were recommended to the user. A massive real-world dataset were used for experimentation and better results were obtained. Singular Value Decomposition (SVD) based movie recommendations was developed in [10]. The algorithm first constructs user-item matrix which was sparse with maximum number of zero values in it. To remove the sparsity, SVD was applied on the user-item interaction matrix to obtain decomposed matrix and finally recommendations were made. However, this system suffered from cold start problem. Data mining algorithms-based movie recommendation was followed [11]. The cast and crew information of the movies were extracted first. From this the movies with average ratings greater than seven was filtered out. Next, Apriori algorithm is applied to the crew column and frequent items are obtained. Items which satisfy the support and confidence were recommended to the users.

A hybrid graph databased-based model which also combines content-based and item-item CF was proposed [12]. This system used features such as genres and ratings of the movies. These features are stored in Neo4j graph database. This method evaluated the impact of using movie's closed caption as syntactic feature. The algorithm started with query movie as the starting node and next the syntactic feature of the query movie was searched for. Next, matching nodes were got by querying the database with the pre-defined Cypher query. While traversing, the similarities of the movies were calculated using cosine similarity measure and the overall similarity was calculated for making recommendations.

A deep learning model for recommendation system was developed using Restricted boltzman machine [13]. The proposed model had only two layers viz. the input layer and the hidden layer. The input layer takes the value either 0 or 1. If the user likes the item, then it is represented as 1. If the item is not preferred by the then it is represented as 0. Hidden layer has latent factors that describe the profiles of the items. The model was trained with profiles that match the user's preferences. Finally, the model predicts the ratings of the items that was not seen by the target users.

From all the works reviewed in the literature it is observed that recommendations were provided using content-based filtering, collaborative filtering, clustering, classification, singular value decomposition, association analysis, neural network and deep learning models. It is also noted that the performance of hybrid models is better compared to many state-of-art methods. So, this paper aims to provide a hybrid collaborative filtering using neural networks and FP-Growth algorithm. The performance of hybrid collaborative filtering model proposed in this paper is compared with the most widely used methods like nearest neighbor approach and correlation-based approach.

4. DATASET DESCRIPTION AND PROBLEM DEFINITION

In this paper the movie ratings dataset released by MovieLens(100k) [18] is used for experimentation. The dataset consists of three files namely users, movies and rating. The rating file has four headers namely userID, movieID, rating and timestamp. The ratings are made on a range of one to five and all users has given ratings for at least twenty movies. Fig.2 shows the first five records in the rating data.

userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.0	964981247
1	6	4.0	964982224
1	47	5.0	964983815
1	50	5.0	964982931

Fig.2: Sample records from rating data

The next step in collaborative filtering is to construct the user-movie interaction matrix. Fig.3 shows the first 7 users and their ratings for first 10 movies in the user-movie interaction matrix obtained from rating data.

movie_id user_id	1	2	3	4	5	6	7	8	9	10
1	4.0	NaN	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	NaN	4.0	5.0	3.0	5.0	4.0	4.0	3.0	NaN	3.0
7	4.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Fig.3. Sample User-Movie Interaction Matrix

From Fig.3, it is observed that the user-movie interaction matrix is sparse. NaN in the user-movie interaction matrix represents that the movie is not rated by the user. Not many users have rated the movies. So the problem here is to guess the rating the movies that are not already watched by the users and understand their preferences. Many methods have been found in the literature to solve this problem. Latent factor models are used to describe the user interests of movies and find the patterns of the users in rating the movies. Matrix factorization is an approach to generate latent features by multiplying two different kinds of entities [14]. Neural network modeling that matrix multiplication in matrix factorization was proposed in [15]. These previous works considers only the latent vector representation of users and movies information. This paper aims to consider the genre of the movie along with the user and movie latent representation and provides a modified neural collaborative filtering approach to rate the movies. The movies in the dataset fall into the combination of 18 genres namely “Drama, Comedy, Thriller, Action, Romance, Adventure, Crime, Sci-Fi, Horror, Fantasy, Children, Animation, Mystery, Documentary, War, Musical, Western, IMAX, Film-Noir”. The neural collaborative filtering model is also fed with the vector representing the combination of genre as input. Thus, the main objective of the paper is to develop a modified neural collaborative model that accepts user_id, movie_id and genre vector as input and predicts the rating of the movie that is not watched by the user. Finally, FP-Growth algorithm is applied to find the most interesting movies for recommendation.

5. MATERIALS AND METHODS

This section explains the methodology proposed in the paper. Fig.4 depicts the working of the proposed model. It has two modules as mentioned below.

- Use modified neural collaborative filtering to find the rating of the unobserved entries.
- Recommend movie to the user based on their interestingness using FP-Growth algorithm.

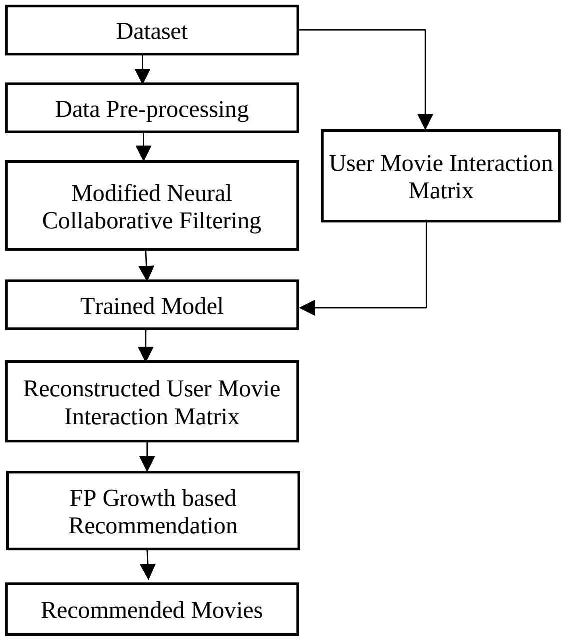


Fig.4. Flow Diagram of the proposed method

The proposed system takes movie data file and rating data file as input. The rating data file is merged with movie data file based on movie_id. Next the genre attribute is encoded using one hot encoding. The proposed modified neural collaborative filtering model is trained for various epoch and batchsize and finally the optimal model is chosen. The trained model is used to find the ratings of unobserved user-movie interaction and a new user-movie interaction matrix is created. FP-Growth algorithm uses this constructed matrix to recommend movies to the user based on their interestingness.

5.1 ARCHITECTURE OF PROPOSED MODIFIED NEURAL COLLABORATIVE FILTERING MODEL

Neural Collaborative filtering uses two tracks to input user and movie data. These two tracks are concatenated to combine their embedding vectors [15]. In the proposed approach, in addition to the latent representation on user and movie, the genre is also concatenated. Fig.5, depicts the architecture of the proposed model.

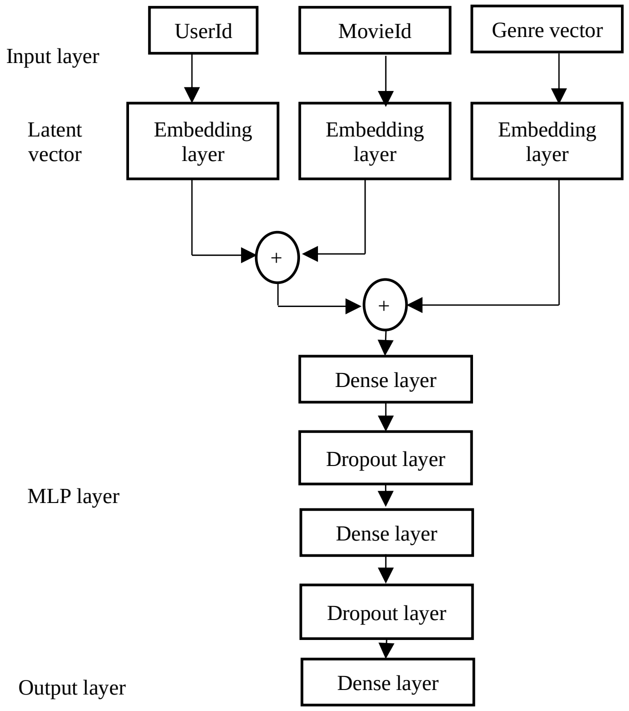


Fig.5. Proposed Modified Neural Collaborative Filtering Model

The input to the proposed modified neural collaborative filtering model is the information containing user_id, movie_id and genre vector. The user_id and movie_id are converted to its latent features using the embedding layer. This is done by peforming one hot encoding of user/movie information and then applying dot product of the encoded data with the weigths in the embedding layers. Next, the encoded user input and the movie input is concatenated. The genre vector is processed in the dense layer and then concatenated to the encoded user/movie input. This input is processed by the multilayer perceptron. So, hidden layers are added to the concatenated vector and the interaction between user, movie, genre and the ratings features are learned by the multilayer perceptron. Finally the output layer has one neuron to provide the rating of the user for the given movie.

Activation function plays a important part in presenting non-linearity into neural networks and is essential for learning complex patterns from the dataset [18]. The activation function used in all the layers of multilayer perceptron is ReLU activation function. The simplicity of ReLU activation function not only speeds up training but also helps mitigate the vanishing gradient problem and is given in (1).

$$RELU(v) = \begin{cases} 0 & if \quad v < 0 \\ v & if \quad v \geq 0 \end{cases} \tag{1}$$

Dropout layer is introduced between each dense layer of the multilayer percerptron in order to avoid overfitting of the model. Dropout is a regularization technique where in some neurons in the layer are randomly ignored during training. The dropout probability of 0.2 is followed in this proposed model. Optimizers are applied in neural networks to modify the model parameters in order to minimize loss. Adam optimizer is applied in the training phase of neural network as it stabilizes the training process and helps neural networks converge to optimal solutions efficiently. Fig.6, shows the sample portion of user-movie interaction matrix after processing by the modified neural collaborative filtering model.

	1	2	3	4	5	6	7	8	9	10
user_id										
1	4.0	4.0	4.0	3.5	3.5	4.0	4.0	3.5	4.0	4.5
2	4.0	3.5	4.0	3.5	3.5	4.0	3.0	3.0	3.5	4.0
3	4.0	3.5	1.5	1.5	3.0	2.0	2.5	2.5	3.0	3.0
4	4.0	4.0	3.5	2.0	3.0	4.0	2.5	2.0	2.0	3.5
5	4.0	3.0	3.0	1.0	2.0	4.5	2.5	1.0	1.5	2.5
6	4.5	4.0	5.0	3.0	5.0	4.0	4.0	3.0	4.0	3.0
7	4.5	2.5	2.0	1.5	1.5	4.0	2.5	2.0	2.0	3.0

Fig.6. Sample of Processed User-Movie Interaction Matrix

5.2 RECOMMENDATION MODULE USING FP-GROWTH ALGORITHM

User preferences are mainly analyzed in designing a recommendation system, because it converts data about users and their likings into predictions of their next likings [7]. General form of user preference analysis is to prepare a questionnaire, circulate among the users and retrieve their answer. Based on the response, users with similar preferences are clustered and recommendations were made. In data mining, user preferences are analyzed using association rule mining. There are many methods used to derive association rules from a dataset. Frequent Pattern (FP)-growth algorithm is a tree-based algorithm used in representing the dataset in the form of FP-tree and arriving at rules based on the interestingness measure. FP- growth algorithm is efficient in processing dense data [19].

The steps in FP-growth algorithm for constructing the FP-tree and finding frequent patterns of movies watched is given Fig.7. The algorithm starts by scanning the user-movie interaction matrix to find the frequency of occurrence of the movies in it. The frequency of occurrence of the movies in the dataset is termed as support count. Movies are arranged in descending order of their support count. The movies that have support count less than the minimum support threshold are considered as less important and hence they are removed in the next step. Minimum support threshold of 20% is used in experimentation.

The FP-growth algorithms continue by building the FP-tree. Initially, the root of the FP-tree is represented by null and all other nodes have two fields namely, movieID and count. The next step is to scan the user-movie interaction matrix again and examine the movies watched by the users. The movie with the maximum support count is represented in the top level and the one with least count occurs in the lowest level. That is, the branch of the tree is constructed with movies in descending order of support count. If any movie in the records is already present in another branch, then this record would share a common prefix to the root by incrementing its count along the path.

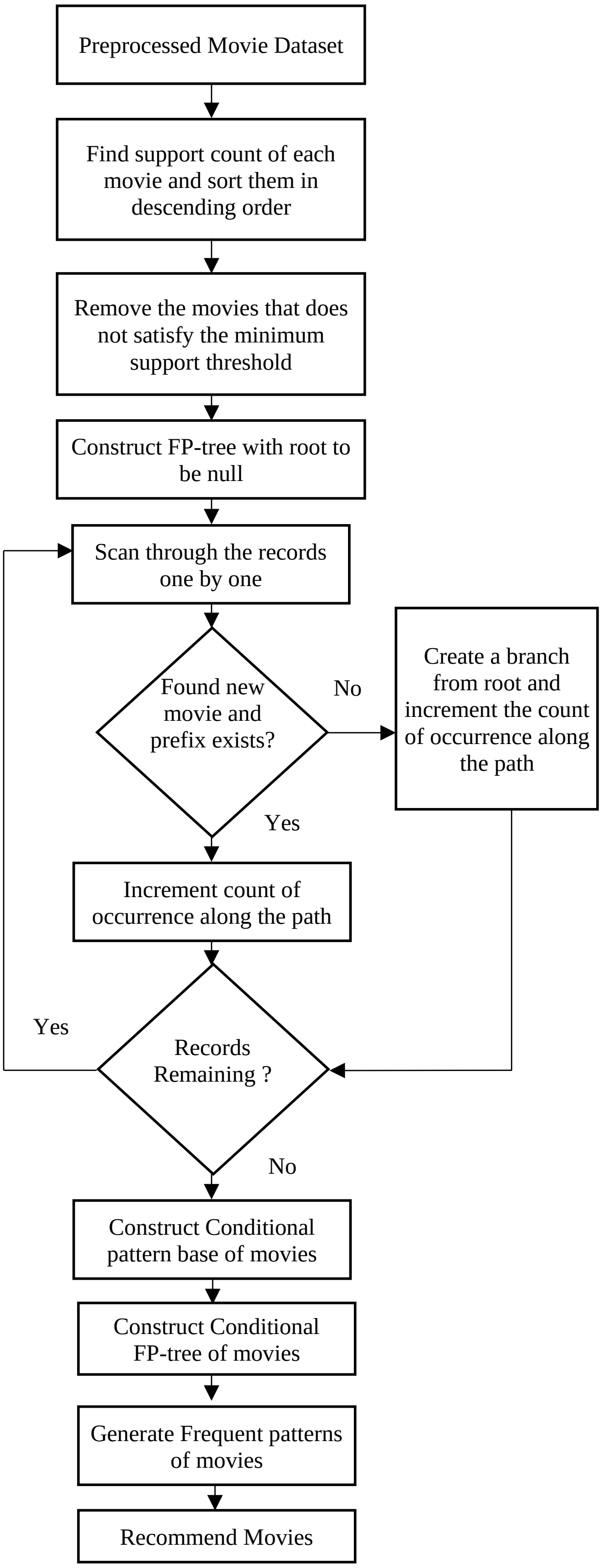


Fig.7. Flowchart of FP-growth algorithm

The algorithm is repeated until all the records in the user-movie interaction matrix are processed. Next step is to construct the conditional pattern base for each movie. To do this, the constructed FP-tree is mined by examining the nodes along with the links to it. The leaf node represents the movies with frequency pattern length 1. Start from this and traverse through the FP-Tree. While traversing, the movies that appear in every paths of the conditional pattern base of those movies are retained. Calculate the support count of the movie by totaling the support counts along the paths in the conditional pattern base. Next step is to construct Conditional FP Tree for the movies by counting the movieset along the path and holding only the movieset whose support count value is larger than or equal to the minimum support threshold.

The recommendation process starts once the conditional FP-tree is developed. This is done using association rule mining from the frequent patterns generated. The interesting relationship and the associations among the movies in the frequent patterns are analyzed based on support, confidence and lift measure of the association rule.

Support of an association rule ($Mi \rightarrow Mj$) is the percentage of records that has $Mi \cup Mj$. Support is given by equation (2).

$$Support(Mi \rightarrow Mj) = \frac{frq(Mi,Mj)}{N} \quad (2)$$

where $frq(Mi,Mj)$ is the frequency of occurrence of the set *Mi* and *Mj* and *N* is the total number of records present in the dataset. Support of 30% is followed in the experiments.

Confidence shows how often the rule is found to be true. Confidence of the association rule ($Mi \rightarrow Mj$) is the percentage of records containing *Mi* that also contain *Mj* and is given by Eq.(3).

$$Confidence(Mi \rightarrow Mj) = \frac{frq(Mi,Mj)}{frq(Mi)} \quad (3)$$

where $frq(Mi,Mj)$ is the frequency of occurrence of the set *Mi* and *Mj* and $frq(Mi)$ is the frequency of occurrence Mi in the dataset. Confidence of 80% is followed in the experiments.

Lift is the correlation measure to identify the interesting association rules. It is given by Eq.(4).

$$Lift(Mi \rightarrow Mj) = \frac{Support(Mi,Mj)}{Support(Mi)\cdot Support(Mj)} \quad (4)$$

If the resulting value is less than one, the occurrence of *Mi* is negatively correlated with the occurrence of *Mj*. If the resulting value is greater than one, the occurrences of *Mi* and *Mj* are positively correlated. If the resulting value is one, then *Mi* and *Mj* are independent and there is no correlation between them. So, rules having lift measure greater than one are considered in the experiments.

After filtering the association rules based on support, confidence and lift measure, 239 rules were obtained for the dataset considered in this study. Table 1 shows the first 10 association rules among the 239 rules obtained after filtering and sorted by lift column. From the association rules arrived, the recommendation is done as follows.

- If the user is new to the system, then recommend movieset with one element. By doing this cold start problem could be resolved.
- Else, the consequent of the association rules having the movies already rated by the user as antecedent is recommended.

It is observed that, movieset with one element contains most popular movies in all genres. Thus recommending it to new users helps in analyzing their preferences and recommending further movies based on their interestingness.

Table.1. Sample Association Rules Generated

Rule (Mi→Mj)	Support	Confidence	Lift
(2571,5952)→(7153,4993)	0.306878	0.878788	2.407115
(7153,4999)→(2571,4139)	0.306878	0.84058	2.407115
(4993,2571,5952)→(7153)	0.306878	0.958678	2.368498
(4993,2571)→(7153,5952)	0.306878	0.859259	2.353623
(7153,5952)→(4993,2571)	0.306878	0.84058	2.353623
(7153,4993,2571)→(5952)	0.306878	0.97479	2.346946
(5952)→(7153,4993)	0.354497	0.853503	2.337857
(7153,4993)→(5952)	0.354497	0.971014	2.337857
(7153,2571)→(4993, 5952)	0.306878	0.892308	2.326154
(4993,5952)→(7153)	0.354497	0.924138	2.283164

6. RESULTS AND DISCUSSION

Implementations were done in Google Claob using python libraries. The performance of the proposed modified neural collaborative filtering model and the recommendation module using FP-growth algorithm are discussed in the following subsection.

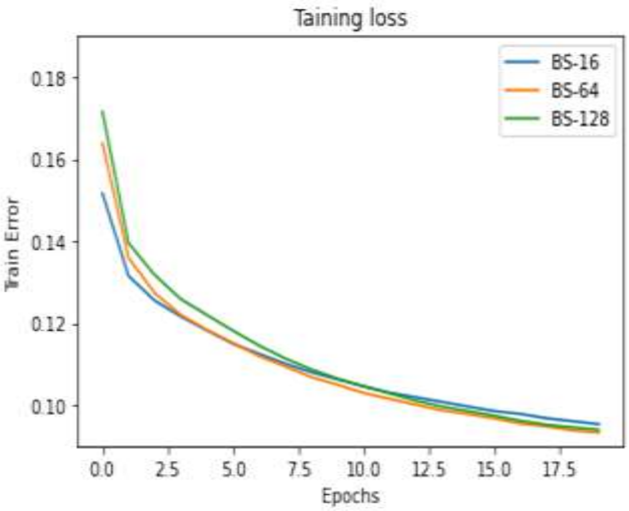


Fig.7. Training loss in terms of Mean Absolute Error

6.1 PERFORMANCE OF PROPOSED MODIFIED NEURAL COLLABORATIVE FILTERING MODEL

The proposed neural network model was trained for various epochs and batchsize. Mean absolute error [16] is the widely used loss function to measure the performance of the neural network model. The Fig.7 shows the graph of loss encountered while training the modified neural collaborative filtering model. As training loss for batchsize 32 and 64 is more similar, the graph of batchsize 32 is not shown in figure 7 for better vizualization. The Table.2 provides the testing loss for various epochs and batch size. It is observed that the proposed model performed well when batch size was 64 and number of epochs was 20 yielding a loss of 0.1017.

Table.2. Testing Loss in terms of MAE

Epochs	Batch size	Test Loss
10	16	0.1357
	32	0.1211
	64	0.1210
	128	0.1189
15	16	0.1311
	32	0.1278
	64	0.1142
	128	0.1110
20	16	0.1210
	32	0.1235
	64	**0.1017**
	128	0.1157

6.2 PERFORMANCE OF RECOMMENDATION MODULE USING FP-GROWTH ALGORITHM

The performance of the recommendation module is calculated using Normalized Discounted Cumulative Gain (NDCG) [17]. NDCG is a measures the quality of ranking. It is computed by comparing the significance of the items given by the recommendation to the significance of the item that an ideal recommendation would return. It is given by Eq.(5).

$$NDCG@K = \frac{DCG@K}{IDCG@K} = \frac{\sum_{j=1}^{k(actualorder)} \frac{Gains}{\log_2(j+1)}}{\sum_{j=1}^{k(idealorder)} \frac{Gains}{\log_2(j+1)}} \tag{5}$$

where, DCG is the discounted cumulative gain, IDCG is the ideal discounted cumulative gain and K is the number relevance attributes considered for calculating gain.

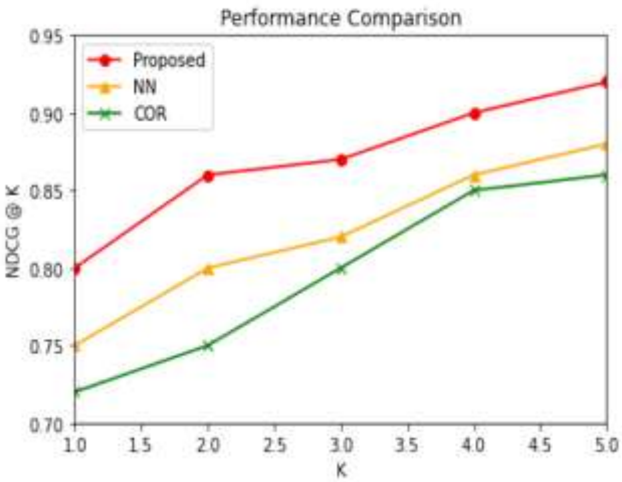


Fig.8. Evaluation of Top 5 movie recommendation

The Fig.8 shows the performance of recommendation lists with ranking position K ranging from 1 to 5. The proposed recommendation model is compared with nearest neighbor recommendation (NN) and correlation based recommendation (COR). It is said that the system performs well when NDCG is 1.0. From Fig.8 it is observed that the proposed approach yields 4.8 percentage (in average) and 7.4 percentage (in average) relative improvements in NDCG@5 compared to the NN approach and COR approach respectively.

7. CONCLUSION

A modified neural collaborative filtering-based movie recommendation system is proposed in this paper. The system uses ratings and genres to train the modified neural collaborative filtering model. The trained model is used to guess the ratings of the movies not watched by the user and hence, sparsity in user-movie interaction matrix is reduced. Next FP-growth algorithm is used for recommending movies to the user with the interesting scores like support, confidence and lift measure of the association rules. The results show that, in an average, the proposed recommendation model yields 4.8 and 7.4 percentage relative improvements of NDCG@5 compared to the nearest neighbor approach and correlation-based approach respectively. In future many other factors like cast, crew, director, etc and influential factors like user demographics could be added to build the deep neural network to perform recommendations. Mapreduce programming could be used for the same approach to handle larger datasets. Also, shifts in user interests could be considered for further improvement.

REFERENCES

[1] A. Joseph Konstan and John Riedl, “Recommender Systems: from Algorithms to user Experience”, *User Modeling and user-Adapted Interaction*, Vol. 22, pp. 101-123, 2012.

[2] Shahab Saquib Sohail, Jamshed Siddiqui and Rashid Ali, “Classifications of Recommender Systems: A Review”, *Journal of Engineering Science and Technology Review*, Vol. 10, No. 4, pp.132-153, 2017.

[3] R. Anaya Antonio, Manuel Luque and Tomas Garcia-Saiz, “Recommender System in Collaborative Learning Environment using an Influence Diagram”, *Expert Systems with Applications*, Vol. 40, No. 18, pp. 7193-7202, 2013.

[4] Zan Wang, Xue Yu, Nan Feng and Zhenhua Wang, “An Improved Collaborative Movie Recommendation System using Computational Intelligence”, *Journal of Visual Languages and Computing*, Vol. 25, No. 6, pp. 667-675, 2014.

[5] Mirza Ilhami and Suharjito, “Film Recommendation Systems using Matrix Factorization and Collaborative Filtering”, *Proceedings of International Conference on Information Technology Systems and Innovation*, pp. 1-6, 2014.

[6] Tevfik Aytekin and Mahmut Ozge Karakaya, “Clustering-based Diversity Improvement in Top-N Recommendation”, *Journal of Intelligent Information Systems*, Vol. 42, No. 1, pp. 1-18, 2014.

[7] YingSi Zhao1 and BoShen, “Empirical Study of user Preferences based on Rating Data of Movies”, *Plos One*, Vol. 11, No. 1, 2016.

[8] Phongsavanh Phorasim and Lasheng Yu, “Movies Recommendation System using Collaborative Filtering and k-Means”, *International Journal of Advanced Computer Research*, Vol. 7, No. 29, pp. 52-59, 2017.

[9] Hung-Wei Chen, Yi-Leh Wu, Maw-Kae Hor and Cheng-Yuan Tang, “Fully Content-based Movie Recommender System with Feature Extraction using Neural Network”, *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 504-509, 2017.

[10] Asoke Nath, Adityam Ghosh and Arion Mitra, “Building a Movie Recommendation System using SVD Algorithm”, *International Journal of Computer Sciences and Engineering*, Vol. 6, No. 11, pp. 727-729, 2018.

[11] B.R. Ranjitha, “Recommendation System for Movie Cast and Crew using Data mining Algorithm”, *International Journal of Engineering and Advanced Technology*, Vol. 9, No. 4, pp. 1495-1497, 2020.

[12] Putra Pandu Adikara, Yuita Arum Sari, Sigit Adinugroho and Budi Darma Setiawan, “Movie Recommender Systems using Hybrid Model based on Graphs with Co-Rated, Genre and Closed Caption Features”, *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, Vol. 7, No. 1, pp. 31-42, 2021.

[13] Lakshmi Palaniappan and K. Selvaraj, “Profile and Rating Similarity Analysis for Recommendation Systems using Deep Learning”, *Computer Systems Science and Engineering*, Vol. 41, No. 3, pp. 903-917, 2022.

[14] Yehuda Koren, Robert Bell and Chris Volinsky, “Matrix Factorization Techniques for Recommender Systems”, *Computer*, Vol. 42, No. 8, pp. 30-37, 2009.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu and Tat-Seng Chua, “Neural Collaborative Filtering”, *Proceedings of International Conference on World Wide Web*, pp. 173-182, 2017.

[16] Jun Qi, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma and Chin-Hui Lee, “On Mean Absolute Error for Deep Neural Network based Vector-to-Vector Regression”, *IEEE Signal Processing Letters*, Vol. 27, pp. 1485-1489, 2020.

[17] Kalervo Jarvelin and Jaana Kekalainen, “Cumulated Gain-based Evaluation of IR Techniques”, *ACM Transactions on Information Systems*, Vol. 20, No. 4, pp. 422-446, 2002.

[18] A. James Anderson, “*An Introduction to Neural Network*”, The MIT Press, 1995.

[19] Lan Vu and Gita Alaghband, “Efficient Algorithms for Mining Frequent Patterns from Sparse and Dense Databases”, *Journal of Intelligent Systems*, Vol. 24, No. 2, pp. 181-197, 2015.