INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 14-08-2023

Accepted: 02-09-2023

Published: 03-10-2023

Citation: Bohra S, Gaikwad A, Singh G (2023) Hybrid Machine Learning Based Recommendation Algorithm for Multiple Movie Dataset. Indian Journal of Science and Technology 16(37): 3121-3128. https://doi.org/10.17485/IJST/v16i37.2065

* **Corresponding author**.

snehab30@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Hybrid Machine Learning Based Recommendation Algorithm for Multiple Movie Dataset

Sneha Bohra[1]*, Amit Gaikwad[1], Ghanapriya Singh[2]

1 Department of Computer Science and Engineering, School of Engineering & Technology, G H Raisoni University, Amravati, 444701, Maharashtra, India
2 Department of Electronics & Communication Engg, National Institute of Technology Kurukshetra136119, Haryana, India

Abstract

Objective: The objective of this study is to design a machine learning based hybrid recommendation algorithm using Matrix Factorization and SVD to provide top – n movie recommendations. **Methods:** The proposed work is an integration of four well-known mechanisms namely Model-based Collaborative Matrix Factorization, KNN- based Clustering, SVD and Popularity based module to predict top-n recommendations. This work is implemented on movie-based datasets Movielens and tmdb-5000. The dataset is divided in 80: 20 for training and testing and we have used Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) performance metrics for evaluation of efficiency of proposed algorithm. **Findings:** The RMSE and MAE for the proposed hybrid model is 0.58 and 0.44 respectively**. Novelty/Applications:** The novelty of the work lies in two major aspects, firstly the linear ensemble of individual modules using popularity based, KNN based ,collabortiave MF and SVD and secondly the feedback evaluating mechanism which computes the relevancy of each recommendation generated. The proposed hybrid scheme focuses on user preferences and generates novel recommendations**.**

Keywords: kNearest Algorithm (KNN); Clustering; Matrix Factorization (MF); Singular Value Decomposition (SVD); Collaborative Filtering (CF)

1 Introduction

Recommender Systems (RS) are efficient web services that assist people to manage information overload and provide personalized information. With the vast information available online, individuals might need support that cater to their particular needs and provide them with selected best items. Typically, recommendations are calculated using historical user online behaviour data. The past user-item relationship help in prediction of user's future interactions, assuming that the user's behaviour does not change over time, but in reality, this assumption may be imprecise. The ignorance of the changing user-item interactions for recommendation generation result in user dissatisfaction. Hybrid Recommendation Algorithms are therefore implemented to achieve promising results with the combinations of various individual mechanisms. Literature Review discussed below describes several state-of the art recommendation algorithms.

CNN based recommendation strategy was implemented on Movielens dataset. It combined the content based and user and item based collaborative algorithms to generate recommendations. It implemented the cosine similarity approach to evaluate the user –item similarities. The system was evaluated and it was found that it considered a specific range of similarity metric for similarities. On the contrary, a mixed range of metric could provide remarkable results[1]. Another strategy handled the dynamically changing user preferences by implementation of collaborative matric factorization which considered the temporal features of movie. This strategy was tested on Movielens dataset and achieved remarkable values for RMSE and MAE as 0.91 and 0.7 respectively[2]. A LSTM and CNN based multimodal recommendation algorithm was presented which considered genre classification and combined the weighted average in ensemble modelling. The work was implemented on IMDB dataset and evaluation showed F1-score as 0.65. The major disadvantage of the work was the unbalanced dataset with very few genre details[3].

A deep learning based model was presented which used the Convolutional Neural Network (CNN) for feature extraction and identification of the user item relationship. The deep learning based model was provided the refined recommendation to the users. The work was implemented on Movielens Dataset and the observed value of RMSE was 0.98[4]. Another strategy presented the use of sentiment analysis for recommendation of movie on tmdb dataset. The work implemented cosine similarity metric and a combined approach of Naïve Bayes (NB) and Support Vector Machine (SVM). The strategy was evaluated on Precision, Recall with remarkable values of 0.98 and 0.99[5]. Another study suggested a recommender system approach on Movielens dataset that combined user demographic and location data with a graph-based model based on the similarity of user ratings. The strategy extracted new features using auto encoder feature extraction. It tried to resolve cold-start issue. The observed values for Precision and Recall were 0.77 and 0.79[6].

Another work presented a combined approach of domain-specific and item based recommendation algorithm and considered Amazon Product Dataset. The work clustered the item on the basis of its category. The suggested work showcased 12% rise in the sale of the products with average precision and recall values as 0.7, 0.6 respectively and noticeable accuracy of 89.6%. The drawback of the work is the non-consideration of the short-term user preferences[7]. A user –item based collaborative filtering algorithm was implemented on Amazon Product dataset. This work took into consideration combined user and item based feature to analyses the user interests and generate recommendations. The algorithm was evaluated on Precision and Recall metric with observed value as 0.7 and 0.68. The recommendation algorithm took comparatively less time for recommendation generation[8]. A combined strategy utilizing the content based and collaborative user and item based filtering was implemented on Movielens dataset. The combination was implemented as stacks to achieve remarkable results. The observed values of Precision and Recall are 0.78 and 0.82 respectively[9].

A combined strategy using DB-Scan and DNN was implemented on Movielens dataset. It considered the demographic data and user rating to generate recommendation. It also implemented friend link algorithm to compute similarity among users. The observed RMSE and MAE values are 1.03 and 0.76 respectively[10]. The hybrid movie recommendation system and optimization technique proposed was based on user-collaborative filtering algorithms and weighted classification. The scoring matrix is then transformed into a low-dimensional, dense item category preference matrix based on item category preference, multiple cluster centers were obtained, the distance between each cluster center and the target user was calculated, and the target user was categorized into the closest cluster. Finally recommendation list was presented to user[11].

A KNN based hybrid strategy used user liking for item as an input and implemented a hybrid approach to get the necessary information from the given data repositories. The method discovered similarities among users and active users. The categorization process took into account the nearest neighbors determined using the KNN methodology. The algorithm took into account user preferences as well as various socio-demographic factors. The observed Precision values were 0.85, 0.22 for Recall, and 0.35 for F1 measure[12]. Another work suggested a hybrid recommender system that made sequential recommendations using markov chain similarity models. To identify comparable users and items, the system used cosine similarity and content-based filtering. The system used deep learning approaches to deal with data sparsity, but did not take context awareness or product features into account[13]. Another technique employed object descriptions in free text. Bidirectional encoder representations and hybrid recommendation models were combined in the method. The process Analyzes product descriptions using NLP, but only takes into account reviews that are a predetermined length[14].

With the help of the MF framework, the suggested hybrid Bayesian stacked auto-denoising encoder (HBSADE) model was able to assess contextual feedback and uncover users' hidden preferences. The deep learning based methodology collected the user-item interactions and user interest to provide individualized suggestions. The approach resolved data sparsity and analyzed user interest from reviews. Item reviews and ratings are not taken into account[15]. Another system combined the outputs of CBF and CF using the gradient descent algorithm. The 9.12% increase was noted in F1 measure performance but it comes at a considerable computational expense[16].

The hybrid deep learning approach technique made use of the collaborative filtering technique. In order to extract contextual information, the model used the input in raw format. This information was then added to the Neural Collaborative Filtering approach for additional processing. Deep learning techniques are used in the process to investigate the metadata and use it to generate customized recommendations. The system's reliance on offline datasets is its lone downside [17]. In order to estimate the user rating on an item, the hybrid approach based on Bayesian inference turned implicit information such as clicks, purchase history, reviews, likes, etc. into a numerical property that is comparable to any explicit feedback provided by users. The chance of ratings was predicted using a Bayesian technique using the observed input. The value of recall was observed to be improved by 3% but at the cost of Resource consumption by some datasets [18].

Another hybrid framework that employed NLP to create consistent tags for movie information. This framework also used a three-layer auto-encoder to cut down on repetition in tag names and produce compact tags. The resulting tags were then sent to SVD for improvement of performance. The RMSE and MAE decreased by 3.93% and 4.34%, respectively. The system experienced higher processing time [19]. A hybrid architecture built on dual auto-encoders and deep learning learnt the hidden features of both users and items. It used matrix factorization to uncover the hidden characteristics. The system resolves cold start issue. The hybrid model should have large training set [20].

A mixed-objective optimization-focused hybrid recommendation model solved and rigorously tested using the multi-objective evolutionary algorithm (MaOEA). 94% of the systems recommendations are diversified, and 45% are innovative. Unbalanced optimization results harm the system [21]. A hybrid model called DST-HRS was based on deep semantics that leveraged item definition semantics, the textual detail and incorporated subject descriptions with probabilistic matrix factorization (PMF). The sparseness problem is solved by the system, although computation and temporal complexity are slow. The fact that it needs additional information is a drawback [22].

The previous research on hybrid recommendations has mainly focused on mining implicit relationships between users and items and solving the sparsity of the rating matrix. However, not enough consideration has been given to user feedback on recommendations. The previous work of collaborative filtering serves as the foundation for this work, and in order to improve the performance of recommendation, a close attention is paid to the user feedback on recommendation. The main aim of the proposed method is to consider the changing user preference shift and provide time-aware recommendations.

The paper is arranged into 4 sections. Section 1 presents an introduction and review of previous works done. Section 2 provides the details about the dataset and Proposed methodology. Section 3 discusses the obtained results and compares it with the existing literature. Section 4 proposes the Conclusion.

2 Methodology

A recommendation system is a machine learning technique that utilizes data to predict user's response for an item. These systems take into account the user preferences, previous user-item interactions and features to generate recommendations. User-Item interactions include clicks, likes, and purchases. Recommender System can be broadly categorized into Content-filtering, Collaborative –Filtering and hybrid systems. Content filtering methods use the user and item features to compute the user- item similarities whereas Collaborative filtering uses explicitly available user-item preferences to suggest recommendations. Hybrid system can be combination of one or more types of recommender system generally implemented to overcome the drawbacks of the individual types.

The proposed framework is a hybrid module which combines results of four modules namely KNN based Clustering module, Collaborative Matrix Factorization based, Single Value Decomposition based and popularity based module. The recommendations provided by these modules are sorted and reordered to recommendations. The recommendation evaluating function then demands the user for the feedback on the recommendations provided so that it can take necessary action for re-generation of recommendations if required.

The Movielens 25 M (ml-small) and The Movie Dataset (tmdb) dataset are used for validation of the proposed approach. The ml-small dataset describes 5-star rating containing approximately 25 million ratings for more than 60 thousand movie titles. These data was generated by more than 1 lakh users in year 2019.It contains Movies.csv which has 62423 unique movie title and has features like numeric id for each movie, the movie title and the genres of the movie. Movie Genres is a list that describes the movie. The dataset also has other files Ratings.csv and Tags.csv.Ratings.csv contains the rating provided by the user for the particular movie title and Tags.csv has tags that gives metadata about movies.

Another dataset tmdb-5000 dataset has many features like genre, id, title, popularity to describe movies. The suggested system was assessed using the mentioned dataset in a 60:20:20 ratio for training, testing, and validation.

Figure 1 shows the basic idea of the proposed methodology which is an integrated module for recommendation. Step 1 is computation of dataset into a user- movie vector which can then be utilized directly by user as input. The feature engineering step will then process the pre- computed input vector so that it may be utilized by the model. The popularity based module uses the

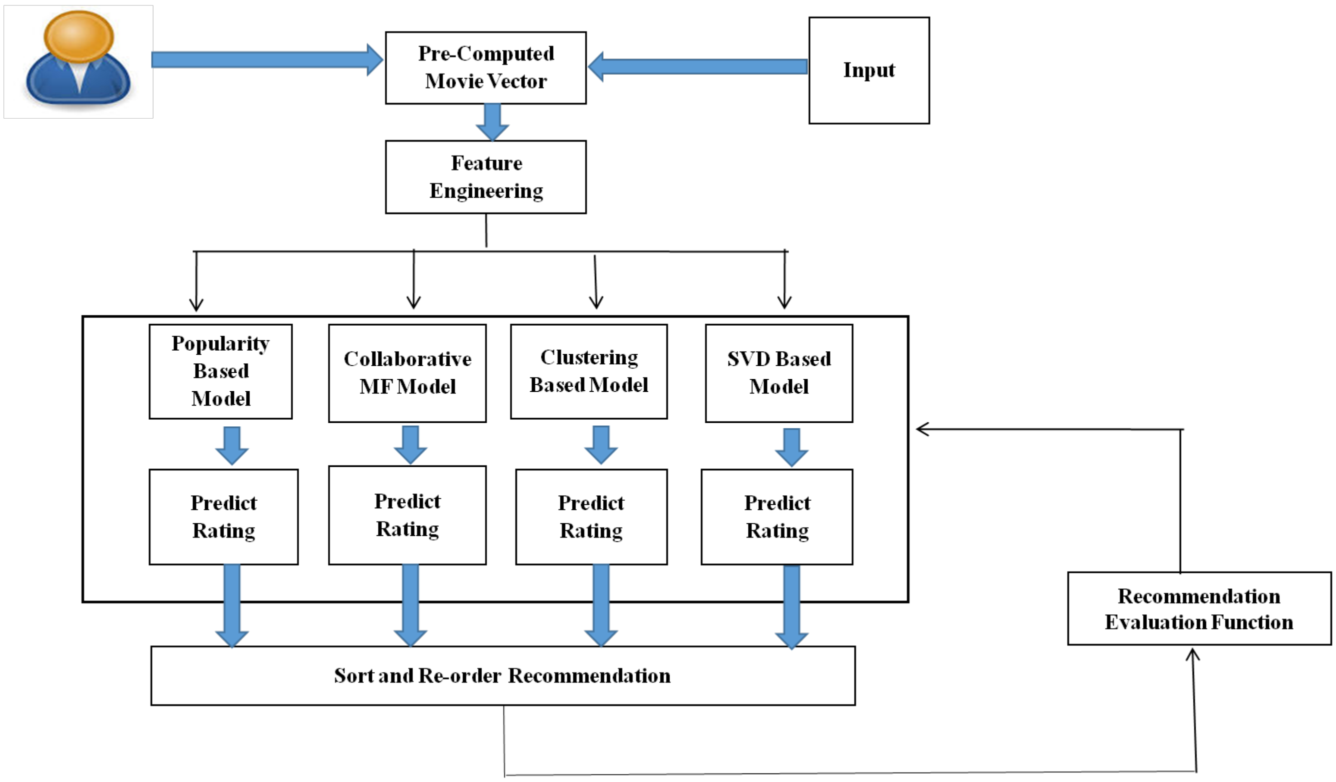

Fig 1. Proposed Hybrid Framework

weighted movie ratings to suggest the top n recommendations. The KNN clustering based module uses the movie-rating matrix and applies clustering algorithm to form clusters of the movies. This module then generates intermediate recommendations from the relevant cluster. The collaborative matrix factorization module takes into consideration the movie –movie similarities to provide recommendations. The last module is based on SVD for generating the recommendations. The recommendation from all these modules are then combined, sorted and re-ordered and presented to the user. Further the user must provide feedback on the presented recommendations which is evaluated by the recommendation evaluation function and if the evaluation parameter has value less than the threshold value then the recommendation are fed to the model for regeneration and the process is continuous till the user is satisfied with the presented time –aware recommendations.

2.1 Data Preprocessing

The preprocessing step includes the data wrangling tasks like removal of irrelevant features from dataset and conversion of categorical features into ordinal feature. Also to enhance the predication accuracy of the proposed model, the user who haven't rated for at least 5 movies and movie titles with no rating are removed.

2.2 Data Visualization

The data visualization of tag.csv infer that the data is free from null values and has 17269 unique tag values. Movies.csv has unique 27278 movie titles and is describes using 19 movie genres. Figure 2 shows the genre distribution over the ml-small dataset and Figure 3 gives idea about the distribution of ratings.. The data analysis of the movie-rating data shows that the average user rating is 477 and the average number of movies that are rated by the user are 97.

2.3 Building Individual Model

The user-movie input vector is generated by the merging of necessary feature from movie and rating data frame. For the input to the popularity module, the movie data frame from ml-small dataset and movie data frame from tmdb dataset are merged on common feature movie-id. The sample input vector is shown in Table 1.

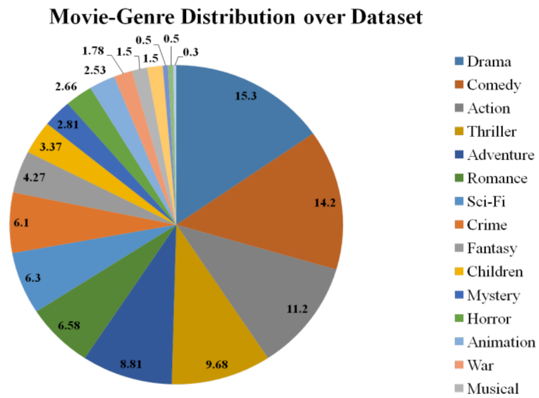

Fig 2. Distribution of Genres across the Movie Dataset

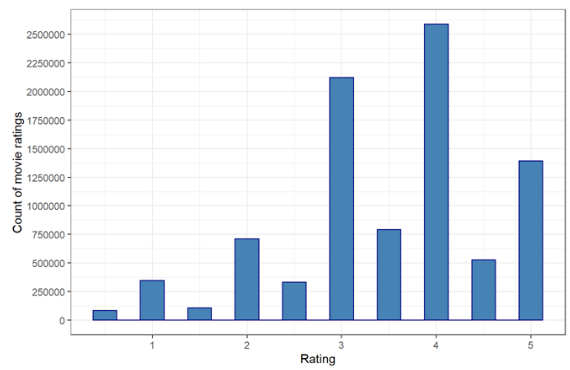

Fig 3. Histogram of Ratings

Table 1. Sample Movie-User Input Vector

userId	movieId	rating	timestamp	genres
1	1	4.0	964982703	Adventure|Animation|Children|Comedy|Fantasy
1	3	4.0	964981247	Comedy|Romance
1	6	4.0	964982224	Action|Crime|Thriller
1	47	5.0	964983815	Mystery|Thriller
1	50	5.0	964982931	Crime|Mystery|Thriller

The popularity model works on the data frame which contains the combined data from both the mentioned datasets. This data frame describes the user activity on the basis of features like number of votes for a movie, total rating per movie and average rating of the movie. The popularity model combines two vital features of the input vector to generate the weighted movie-rating which is computed using following equation:

$$w_{rm} = \frac{(avg\ r_m * v_m) + (avg\ v_m * min(v))}{v_m + min(v)} \tag{1}$$

Where,

w_{rm}= weighted movie-rating for movie m,

$avg\ r_m$= average rating for movie m,

$avg\ v_m$= average vote for movie m,

v_m= total votes for the movie m and

min (v) = minimum number of votes

Figure 4 shows the result presented by the popularity model based on the genres having highest value for weighted movie-rating.

movie_id	original_title	popularity
211672	Minions	875.581305
157336	Interstellar	724.247784
293660	Dead pool	514.569956
118340	Guardians of the	481.098624
76341	Mad Max	434.278564
135397	Jurassic World	418.708552
22	Pirates of the	271.972889
119450	Dawn of the Planet	243.791743
131631	The Hunger Games:	206.227151

Fig 4. Popular Movies Recommended By Popularity Model

The knn-clustering model works on movie –rating vector to form the cluster of the user with similar ratings for similar movie. This module uses the mean rating of the movie when evaluating the rating of the user. The similarity metric used for the formation of cluster is the pearson_baseline which take into consideration the pairs of movies using the mean rating of the movie. The collaborative matrix factorization module considers the movie –rating as input vector and performs item – item collaborative filtering. It considers the explicit movie ratings to predict the rating (hidden latent factors). It predicts the ratings for unrated movie on the basis of mapping of the similar movies. The SVD module takes into consideration the implicit ratings of the movie. Implicit ratings describe the user's interaction with the movie without considering the actual rating value.

To enhance the performance and overcome the drawbacks of the individual methods, and combination of predicted ratings is considered. These are then rearranged and presented to the user. The user must provide feedback to the presented recommendations as relevant or irrelevant. The recommendation evaluation function will evaluate the feedback given for the recommendation and decide if there is need for recomputation of the recommendation depending upon the threshold value of evaluating factor. The evaluation function is as described in equation 2. If the value of the evaluating factor (e) is less than 0.70 then it means that the user does not find the recommendations to be relevant and hence it needs to be sent back to the proposed model and recomputed.

$$e = Relevant\ recommendations/Total\ number\ of\ recommendations \tag{2}$$

The proposed hybrid model is analyzed for two basic scenario of movie recommendations.

Scenario 1: The user-movie vector has enough interactions and

Scenario 2: The user-movie vector does not have enough interactions.

In scenario 1, the user has rated number of movies, the predicted ratings of each movie is quite high since the highly rated movies are present in test set. Whereas in the scenario 2, since the user has not rated much of the movies, the collaborative movie-movie based filtering and popularity module comes in consideration. Based on similar genre preferences and popular movies, the recommendation is computed.

3 Results and Discussion

The proposed methodology is coded using Python and the dataset is stored and processed using Pandas data frame. The Surprise Library has been used coding for the MF, SVD module and evaluating the predictions. Decision Support Accuracy Metrics and Statistical Accuracy Metrics are used for evaluation of the recommendation system's accuracy. For evaluation of the ratio of predicted ratings to verified ratings, statistical metrics like Mean Average Error (MAE), Root Mean Square Error (RMSE) are used. Another category of metrics are the Decision Support Metric Precision and Recall which rates the best recommendations out of all recommendations. Recall is defined as the proportion of relevant recommendations among all recommendations, whereas precision is the fraction of the positive recommendations to the total number of relevant recommendations [7].

The performance of the proposed hybrid model depends on the performance of the individual models. The popularity module provides the recommendation on the basis of the popularity within the genre and the weighted rating calculated for the movie title. The Clustering module uses the KNN algorithm. In order to enhance the performance of this module, various

KNN based algorithms were evaluated on different similarity metrics and the number of clusters. According to the experiments conducted, it was analyzed that the KNN-Baseline algorithm with person- baseline similarity metric out performed for movie-movie similarity on movie datasets. Also, with the increase in the number of clusters, the performance of module seems to improving. The performance of variations of KNN is as shown in Table 2 below.

Table 2. Analysis of Algorithms Implemented in Surprise Library

Algorithm	Statistical metrics		Decision support metrics	
	RMSE	MAE	Precision	Recall
KNN basic	0.941	0.783	0.812	0.424
KNN Baseline	0.901	0.672	0.803	0.453
KNN Baseline-Pearson	**0.8752**	**0.682**	**0.855**	**0.523**
MF	0.921	0.699	0.769	0.394
Collaborative Clustering	0.962	0.734	0.792	0.384
SVD	0.892	0.698	0.818	0.412

The Matrix Factorization module uses the item-item based collaborative filtering which presents better results when the user –rating matrix is sparse. The proposed recommendation model should predict ratings and in order to achieve this we implement the SVD method with latent factors. Users with similar liking towards movie are mapped together in the lower-dimensional representation of movies in SVD. To map movies and users in the same lateral space, it computes the hidden latent factors. To reduce the RMSE for test data, we use regularization and gradient descent algorithm to optimize the SVD model. The model is implemented using GridSearchCV Algorithm of scikit library.

Finally, the recommendation list generated is filtered by the filtering mechanism and arranged in sorted order before being presented to the user. The user then provides the degree of relevancy to each recommendations. The evaluation factor of relevancy decides the recomputation of the generated recommendations. In case of lower degree of relevancy, the hybrid model regenerates the recommendation list to be provided to the user.

Table 3. Comparative Analysis of Performance of Proposed Algorithm

Algorithm	Statistical metrics	
	RMSE	MAE
KNN-genome [19]	0.63	0.58
HCRDa [20]	0.61	0.60
Proposed Hybrid	**0.58**	**0.44**

A linear ensemble of the recommendations from the aforementioned methods is done in order to enhance the test accuracy results and balance the drawbacks of each method. The popularity model provide the popular genres, KNNBaseline provides movie recommendations on the basis of item- item similarity, and CMF and SVD embedded the implicit ratings in addition to the latent factor model. These models are combined to improve the RMSE, MAE, and recommendation evaluations. The integration of the methods into single model improves the quality and diversity of the recommendations. A weighted linear combination of the top methods gave better testing accuracy of ratings and even outperforms the best individual models. Table 3 provides the comparative analysis of performance of proposed methods to the algorithms studied in the rich literature implemented on similar datasets.

4 Conclusion

The popularity module are based on user and movie similarities and generate decent recommendations. Although this module resolves the cold start problem for the new user but lacks in providing the personalization in recommendations. For large datasets, collaborative filtering and latent factor methods like MF and SVD are found to be more efficient to record and utilize the user preferences. The linear combination of the models improved the accuracy of the recommendations. The final list of recommendations generated by the hybrid model is a balanced list for a user is produced which take into consideration features from each model. This balanced list of recommendation is again verified for the relevancy from the user. The hybrid model extracts the best features for the individual modules and use them to provide personalized recommendations for the user. The model is evaluated on the basis of statistical metrics like RMSE and MAE and have shown remarkable performance. The proposed hybrid model can be extended with the computation of individual list of recommendations as feature vector and neural networks can be trained with weighted input to generate combined recommendations. Also, the weighted recommendations can also be passed as input to the deep learning model. The utilization of features other than genres and users' demographic features from the dataset can be a possible future research.

Acknowledgement

The authors are thankful to the G H Raisoni University Amravati and NIT Kurukshetra for providing research support for completely this work smoothly.

References

1) Gopal KS, Patro, Mishra SBK, Panda RK, Kumar, Hoang V, et al. A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems. *IEEE Access*. 2020;8(2020):90978–90991. Available from: https://ieeexplore.ieee.org/document/9091509.
2) Yang Y, Jang HJ, Kim B. A Hybrid Recommender System for Sequential Recommendation: Combining Similarity Models With Markov Chains. *IEEE Access*. 2020;8(2020):190136–190146. Available from: https://ieeexplore.ieee.org/document/9207880.
3) Islek I, Oguducu SG. A Hybrid Recommendation System Based on Bidirectional Encoder Representations. *ECML PKDD 2020 Workshops*. 2020;p. 225–236. Available from: https://link.springer.com/chapter/10.1007/978-3-030-65965-3_14.
4) Subramaniyaswamy NS, Viloria A, Senthilselvan VVN. A deep learning-based hybrid model for recommendation generation and ranking. *Neural Computing and Applications*. 2020;4:10719–10736. Available from: https://link.springer.com/article/10.1007/s00521-020-04844-4.
5) Li H, Han D. A Novel Time-Aware Hybrid Recommendation Scheme Combining User Feedback and Collaborative Filtering. *Mobile Information Systems*. 2020;2020:1–16. Available from: https://ieeexplore.ieee.org/document/9239872.
6) Mashhour I, Bellavista P, Corradi A, Foschini L, Montanari R, Beerocal J, et al. A Pre-Filtering Approach for Incorporating Contextual Information Into Deep Learning Based Recommender Systems. *IEEE Access*. 2020;8(2020):40485–40498. Available from: https://ieeexplore.ieee.org/document/9004579.
7) Ngaffo AN, Ayeb WE, Choukair Z. A Bayesian Inference Based Hybrid Recommender System. *IEEE Access*. 2020;8(2020):101682–101701. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9104694.
8) Duong TN, Vuong TA, Nguyen DM, Dang QH. Utilizing an Autoencoder-Generated Item Representation in Hybrid Recommendation System. *IEEE Access*. 2020;8(2020):75094–75104. Available from: https://ieeexplore.ieee.org/document/9075162.
9) Dong B, Zhu Y, Li L, Wu X. Hybrid Collaborative Recommendation via Dual-Autoencoder. *IEEE Access*. 2020;8(2020):46030–46040. Available from: https://ieeexplore.ieee.org/document/9027867.
10) Cai X, Hu Z, Zhao P, Zhang W, Chen J. A hybrid recommendation system with many-objective evolutionary algorithm. *Expert Systems with Applications*. 2020;159(2020):113648. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0957417420304723?via%3Dihub.
11) Khan Z, Iltaf N, Afzal H, Abbas H. DST-HRS: A topic driven hybrid recommender system based on deep semantics. *Computer Communications*. 2020;156(2020):183–191. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0140366419304062.
12) Khant P, Tidke B. Multimodal Approach to Recommend Movie Genres Based on Multi Datasets. *Indian Journal Of Science And Technology*. 2023;16(30):2304–2310. Available from: https://doi.org/10.17485/IJST/v16i30.1238.
13) Bohra S, Bartere MM. Hybrid Recommendation Algorithm for Personalization of Customer Experience. *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*. 2022. Available from: https://ieeexplore.ieee.org/document/9824545.
14) Bohra S, Bartere M. Implementing a Hybrid Recommendation System to Personalize Customer Experience in E-Commerce Domain. *ECS Transactions*. 2022;107(1). Available from: https://iopscience.iop.org/article/10.1149/10701.9211ecst/pdf.
15) Mu Y, Wu Y. Multimodal Movie Recommendation System Using Deep Learning. *Mathematics*. 2023;11(4). Available from: https://doi.org/10.3390/math11040895.
16) Pavitha N, Pungliya V, Raut A, Bhonsle R, Purohit A, Patel A, et al. Movie recommendation and sentiment analysis using machine learning. *Global Transitions Proceedings*. 2022;3(1):279–284. Available from: https://doi.org/10.1016/j.gltp.2022.03.012.
17) He H, Shang Z, Wu M, Zhang Y. Movie Recommendation System Based on Traditional Recommendation Algorithm and CNN Model. *Highlights in Science, Engineering and Technology*. 2023;34(2022):255–261. Available from: https://doi.org/10.54097/hset.v34i.5481.
18) Behera G, Nain N. Collaborative Filtering with Temporal Features for Movie Recommendation System. *Procedia Computer Science*. 2023;218:1366–1373. Available from: https://doi.org/10.1016/j.procs.2023.01.115.
19) Darban ZZ, Valipour MH. GHRS: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*. 2022;200:116850. Available from: https://doi.org/10.1016/j.eswa.2022.116850.
20) Salmani S, Kulkarni S. Hybrid Movie Recommendation System Using Machine Learning. *2021 International Conference on Communication information and Computing Technology (ICCICT)*. 2021. Available from: https://doi.org/10.1109/ICCICT50803.2021.9510058.
21) Farashah MV, Etebarian A, Azmi R, Dastjerdi RE. A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data*. 2021;8(1). Available from: https://doi.org/10.1186/s40537-021-00422-0.
22) Yuan Z, Lee JH, Zhang S. Optimization of the Hybrid Movie Recommendation System Based on Weighted Classification and User Collaborative Filtering Algorithm. 2021. Available from: https://doi.org/10.1155/2021/4476560.