How to Cite:
Kavi Priya, S., Manonmani, T., Dharshana, N., & Ragaanasuya, K. (2022). Movie recommendation system with hybrid collaborative and content-based filtering using convolutional neural network. *International Journal of Health Sciences*, *6*(S8), 5357–5372. Retrieved from https://sciencescholar.us/journal/index.php/ijhs/article/view/13454

Movie recommendation system with hybrid collaborative and content-based filtering using convolutional neural network

Kavi Priya S
Department of Computer Science and Engineering, Mepco Schlenk Engineering College (Autonomous), Sivakasi, India
Corresponding author email: urskavi@mepcoeng.ac.in

Manonmani T
Department of Computer Science and Engineering, Mepco Schlenk Engineering College (Autonomous), Sivakasi, India
Email: manonmani@mepcoeng.ac.in

Dharshana N
Department of Computer Science and Engineering, Mepco Schlenk Engineering College (Autonomous), Sivakasi, India
Email: dharshana_cs@mepcoeng.ac.in

Ragaanasuya K
Department of Computer Science and Engineering, Mepco Schlenk Engineering College (Autonomous), Sivakasi, India
Email: ragaanusuya_cs@mepcoeng.ac.in

***Abstract*---**With the advancements of big data, recommendation systems have become extremely useful in wide applications such as e-business service, social networks, e-commerce, e-learning etc. Typically, Movie Recommendation systems predict which movie a user likes based on the characteristics of earlier watched/liked movies or interests and their likelihoods. This recommendation system gathers information from users and offers top movie recommendations. Various researchers use collaborative and content-based filtering recommender systems. However, as the number of movies and users grows, neighbor selection becomes more difficult due to data scarcity. Thus, the proposed approach uses hybrid collaborative and content-based filtering. A crossover social recommender structure utilizing a significant CNN based RELU network is introduced. The experimental results on Movie Tweetings and Open Movie Database dataset shows that the accuracy of the proposed approach has improved compared to the existing techniques.

International Journal of Health Sciences ISSN 2550-6978 E-ISSN 2550-696X © 2022.
Manuscript submitted: 9 June 2022, Manuscript revised: 18 August 2022, Accepted for publication: 27 Sept 2022

***Keywords*---**movie recommendation system, open movie database, collaborative filtering, content-based filtering, convolutional neural network.

Introduction

In present world, the Internet has gained more significance in human lives. As the usage of Internet has grown, so has the volume of data. However, not all of the data available online is useful or yields consistent results for users. Data in such large volumes is frequently inconsistent, and without processing or storage, it will be wasted Reddy et al., (2019). In these kind of instances, users must repeat their exploration prior to final finding what they were searching for. To resolve this, a variety of Recommendation Systems (RS) were created to aid users in locating things of interest and selecting products from vast databases. These RS is used in a variety of online products and services, such as movie and music, social media, articles, and marketing Jhiang et al., (2020). There has also been a lot of research done both in industry and academia to develop movie recommendation algorithms.

Films can be effectively classified as thriller, horror, comedy, drama etc. One can arrange films in the order of year, language, director or by cast. Most web-based features give recommendations to the user based on their recently seen or history of movies. Movie Recommendation Systems assist us by looking through our favored films and furthermore decrease the difficulty of investing a ton of energy looking for positive films. These systems consider variety of factors, such as information about the movies itself, user profiles and their previous watched films, or simply program popularity Márcio Soares & Paula Viana. (2015). The essential necessity of a film suggestion framework is that it ought to be entirely solid and furnish the users with the proposal of films based on their interest. To design an efficient Recommendation System, the data should be thoroughly analyzed and the decide which parameters to use.

RS are extensively classified into two classifications: Collaborative filtering (CF) and Content-based filtering (CBF). Collaborative filtering is a technique that can sort out products that a user may like based on the responses of other users. Its requisite sufficient information from users to recommend products based on their interests Sang-Min et al., (2012). CF assists users to make decisions in light of the point of view of others. Recommendations in content-based filtering are based on the user's previous choices. Content-based filtering relies heavily on item description and a profile of the user's orientation. Content-based filtering algorithms attempt to recommend items based on the number of similarities Fernando et al., (2013). If two users are viewing a similar film while their rating for things is different, then in CBF, films are recommended by calculating the closeness among the contextual data. With the arrival of various online media stages like Quora, Facebook, and Twitter, individuals can share their everyday perspective on the web. The data on Twitter are called tweets which are of restricted character that keep users refreshed about their cherished subjects, individuals, and videos.

The sparsity issues arise due to a lack of user history rating data, and they are exacerbated by the dramatic growth of users and items. Furthermore, high-dimensional rating data may make it difficult to extract common interesting users through similarity computation, leading to poor recommendations. To address the aforementioned challenges, a hybrid model-based movie recommendation approach is proposed to address both high dimensionality and data sparsity. Thus, in this paper hybrid collaborative and content-based filtering method with CNN is proposed. Deep learning has been advanced to the fields of information retrieval and recommendation systems as a result of latest developments in the field of deep learning in different application domains such as computer vision and speech recognition Lecun et al., (2015). Deep learning has recently demonstrated its effectiveness in dealing with recommendation tasks. Deep learning techniques have gained traction in recommender systems due to their cutting-edge performance and high-quality recommendations. Deep learning provides a better understanding of user demands, item characteristics, and historical interactions between them than traditional recommendation models. Thus, the deep learning approach is considered for movie recommendation. The rest of the paper is organized as follows: Section 2 explores the work related to the movie recommendation system. Section 3 deals with the detailed description of the proposed system. Section 4 explains the experimental results on MovieTweetings and Open Movie Database. Finally, the paper is concluded with the future work.

Related Work

Currently collaborative filtering is broadly utilized in recommender systems. The author of Ouyang et al., (2014) use CF to recommend the movies and also calculated the recall value. Recommended system find the TopN movies displayed to user. Therefore, topN recommendations results in real-world applications. In this paper, a new method called Collaborative Noise Reduction Autoencoder (CDAE) for topN recommendations using the idea of noise reduction autoencoder (Wu et al., 2016).

Given the robust features of the input sample are used to produce robust ones Expression from two perspectives: From immutability Slight deviation from the sample and Minimum error decoder are used in Thus, in Sun et al., (2016) autoencoder unsupervised feature learning and deep architecture are used to learn the initial parameter to perform recommendation. YouTube is one of the most comprehensive and advanced industrial recommendation systems. The author of Covington et al., (2016) outlines the recommendation system and focuses on detail about deep ranking model. It also provides hands-on lessons and insights from the development, iteration, and maintenance of large-scale recommender systems that have a significant impact on users.

Maintaining feature interactions through feature transformations between different products is effective and interpretable, but generalizing it requires more feature engineering efforts. Due to less feature engineering, low-dimensional high-density embeddings learned from sparse features can generalize deep neural networks into invisible feature combinations Cheng et al., (2016). Enhancement in recommender frameworks has become one of the significant procedures for taking care of the issue of over-fitting. Variety has additionally been highlighted at a few unmistakable gatherings as all things considered a unique area or as an examination challenge, which effectively demonstrates its pertinence. Because at least some aspects of diversity are extremely subjective, the author of Kunaver, M., & Pozˇrl, T. (2017) conducted a survey believes that researchers would benefit from using expert knowledge from the field of psychology when developing new diversity measures.

In this paper Li X., & She J. (2017) a multimedia modality other than text is considered. Collaborative Variational Auto Encoder recommender systems is used. Experiments have shown that CVAE has more robust performance and far exceeds state-of-the-art recommendations. In Li S., & Fu Y. (2017) the recommended system assists users with personalized service support by learning past actions and predicting the current settings for a particular product. Artificial intelligence (AI), especially computing agencies and machine learning methods and algorithms, have been used to develop recommended systems for improving prediction accuracy and saving data.

Joint filtering and factor decomposition of Low rank are known to create recommendations in Okura et al., (2017). Information-based words commonly used in the search settings are excellent candidates' perspectives, The experimental results shows that there is a 23% increase in click-through rate (CTR) and a 10% increase in total duration compared to traditional embedding methods. Chen Y et al., (2018) proposes deep learning for learning feature expressions from side information. Learning feature expressions, on the other hand, guarantees a sufficient recommendation method.

Online entertainment is most of the current life. Individuals who are everywhere on the planet use online entertainment for irregular purposes. You will post an exhibition, score, as a result, occasional function, etc. in online entertainment. As you can do it, you do not have misty ideas from regularly. Online security is the main highlight to guarantee your own information. However, it is difficult to ensure the protection of an online interpersonal organization (OSN) as OSN follows or deviates from the strategy. Customers must provide the data specified by the Academic Institution in order to be able to use OSN Das et al., (2018). Zhang S et al (2019) in research deep learning method and their significance in recommendation systems. Wang K et al (2019), introduces a new method called Top recommended collaborative denoising autoencoder (CDAE) using the autoencoder. The author of Kiran et al., (2020) It uses cutting edge principle in collaborative filtering system to recommend the movie based on the user interest. In Tahmasebi et al., (2021) the content based filtering and collaborative filtering are combined to recommend the movie based on the user interest.

Proposed Work

Overall design

The system diagram is shown in Figure 1. Steps of proposed diagram are explained as follows:

Initially the movie Tweeting dataset and Open Movie Database datase is downloaded from Kaggle. OMDB API extract the features of the movie title, perform the content and collaborative filtering, and hybrid filtering to recommend the movie based on user's interest.

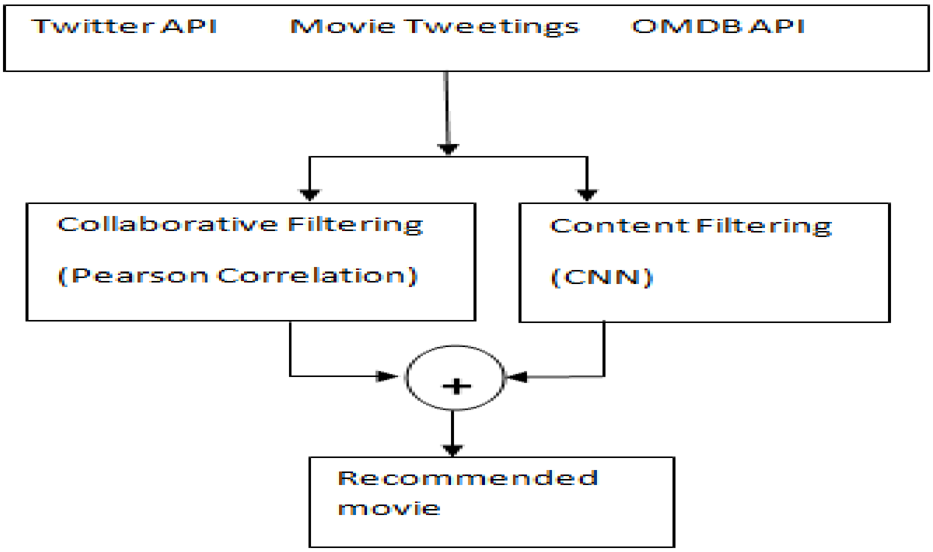

Figure 1. Flowchart of the proposed system

Data

The dataset was collected from the Movie Tweeting's and Open Movie Database repository. The data set contains the information such as users rating, movieId, userId and so on.

Data Preprocessing

Data Preprocessing is the method involved with eliminating the undesirable or pointless information from the information dataset. In this work, the invalid values for example, missing values and Nan values are replaced by 0. Absent and copy values were eliminated, such that the anomalies are removed. The data cleaning process is performed in title for removing comma, special character, and quotations.

Table 1
Preprocessing results

Before Preprocessing	After Preprocessing
Le fils de l'épicier(2007)	Le fils de lpicier (2007)
Sa-rang-ha-ni-gga-gwen-chan-a (2006)	Saranghaniggagwenchana (2006)
Tada, kimi wo aishiteru (2006)	Tada kimi wo aishiteru (2006)

Tada, kimi wo aishiteru (2006)	Tada kimi wo aishiteru (2006)

Recommendation Score

The proposed framework determines the score based on the interest of the user. Client rating conduct is a score given by the client to an item. This score esteem provides the client, and it is well known as the client rating esteem. Users will give top ratings to products they are interested in and satisfied with, and low ratings to products they are interested in but not satisfied with. As a result, regardless of whether the user gives a product a high or low rating, the user's behavior in rating the product fully indicates that the user is interested in this type of product Xiong et al., (2010).

Data Splitting

Separating the data into training and test datasets is an important part of evaluating a data mining model. In the proposed work, the dataset is typically splitted into 80% training dataset and 20% test dataset.

Collaborative filtering

Collaborative Filtering is the recommendation algorithm that is used to find the films based on the user interest but seen by other users. It is the popular recommendation algorithm. It finds a user's interest in some films based on the scores generated and the correlation calculated between the users.

$$t = \sum (ai-\bar{a})(ei-\bar{e}) \,/\, \mathrm{sqrt}\,(\sum (ai-\bar{a})^2 \sum (ei-\bar{e})^2) \qquad (1)$$

t = percentage of similarity
ai = values of a variable
ā = mean of a variable
ei = values of e variable
ē = mean of e variable

Algorithm of Colloborative filtering

Input: movieId, userId, title
Output: Recommend the movie based on user interest.

Step1: i $\leftarrow$ input
Step2: j $\leftarrow$user
    if j € i
        display j
Step3: Using Pearson correlation (1) calculates the similarity index between the user input and the group of users.
Step4: Find the weighted rating
    Weighted rating=similarity index * rating
Step5: Find the sum of similarity index and the sum of weighted rating of the individual movieId for all the movies.
Step6: Find the average recommendation score.

Average recommendation score=similarity index /weighted rating .
Step7: Sort the value in descending order.
Step8: Recommend the top 10 movies.

The following section discusses how the collaborative filtering works on the datasets. Table 2 is formed from the user input watched movies such as Species, Avatar, Taxi Driver, Se7en, Snatch. It describes the rating and Movie_id of each movies. Rating is given by the user for each movies.

Table 2
Dataset description

title	Rating	Movie_id
Species	8	114508
Avatar	9	499549
Taxi Driver	1	75314
Se7en	10	114369
Snatch	5	208092

Table 3 shows the results of user who have seen already watched the movies that the user Input have watched. It consists of the user_id, movie_id, rating, and title of each movie. For example, the following table shows the user whose id is three with their watched movie id and the rating given by them to the corresponding movies.

Table 3
Results of the movies watched by other users

user_id	movie_id	Rating	title
3	75314	1	Taxi Driver
3	114369	10	Se7en
3	208092	5	Snatch

Table 4 shows the results of grouping of user Id who are all having the same user Id. The table consists of values of user_id, movie_id, rating, and title of the movie. For example, the users whose id is 3 are grouped and formed in a single table.

Table 4
Grouping of userId

user_id	movie_id	rating	Title
1	114508	8	Species
2	499549	9	Avatar
3	75314	1	Taxi Driver
3	114369	10	Se7en
3	208092	5	Snatch
71586	114508	8	Species

Table 5 shows the result of similarity index value which is calculated using the Pearson Correlation between the user input and the group of user who have already watched the same movies that the user input has watched.

Table 5
Similarity Index of user_id

similarityIndex	user_id
-93.362792	21295
-47.024781	28245
-3.196721	3
-19.526734	333
-12.890508	1430

In Table 6 the weighted rating is calculated. Weighted rating is nothing but multiply the similarity index with respect to rating. For example, if the user_id has similarity index 0 then weighted rating will be 0.

Table 6
Weighted rating calculation

similarityIndex	user_id	movie_id	rating	weightedrating
0.000000	7097	22100	10	0.000000
0.000000	7097	31381	10	0.000000
0.000000	7097	41959	10	0.000000

The movie Id of the same movies are grouped and the sum of similarity index and weighted rating of all the individual grouped movies are calculated. Table 7 displays the sum_similarityIndex and sum_weightedRating values of the corresponding movie_ids. Majority of the values are zero.

Table 7
Similarity Index and Weighted rating calculation

movie_id	sum_similarityIndex	sum_weightedRating
12349	-24.5	-206.0
15864	0.0	0.0
18773	0.0	0.0
21749	-18.5	-168.0
22100	0.0	0.0
11474156	0.0	0.0
12027020	0.0	0.0

Afterwards weighted average recommendation Score is calculated. Weighted average recommendation score is equal to ratio of weighted rating and similarity index. Table 8 shows the weighted average recommendation score of the corresponding movie-id.

Table 8
Recommendation Scores calculation

movie_id	weighted average recommendation score
12349	8.408163
15864	NaN
18773	NaN
21749	9.081081
22100	NaN
11474156	NaN
12027020	NaN

In Table 9 Top movies are identified based on the weighted average recommendation score. According to the collaborative filtering the top recommended movie is circle and its rating is 8. The low recommended movie is The Third Man, and its rating is 10.

Table 9
Recommended movies using CF

Top Recommended Movie	user_id	movie_id	rating	title
1	39	3118452	8	Circle
2	47	3110958	8	Now You See Me 2
3	69	1790809	7	Pirates of the Caribbean
4	84	1477834	7	Aqua man
5	111	1477834	6	Aqua man 1
6	71688	1477834	7	Cool Hand Luke
7	71694	1255953	10	Incendies
8	71694	1790809	7	Green Book
9	71694	3110958	10	The Third Man

Content based filtering

Content based filtering is a recommendation algorithm. This algorithm finds the movies based on the user interest. It recommends movie to the single user.

$$\mathrm{Cos}(t, k) = t.\ k\ /\ ||t|| * ||k|| \tag{2}$$

t. k = product of t and k.
||t|| and ||k|| = length of t and k.
||t|| * ||k|| = cross product of t and k.

Algorithm

Input: genre,rating,movieId,userId.
Output: Recommend the movie based on user interest

Step1: Calculate genre count
Step2: Calculate rating density
Step3: Merge movie and rating dataset based on userId
Step4: From the output, group the userId, calculate the size and mean of

movie_id and rating.
Step5: Calculate the mean rating, sort the values in descending order.
Step6: Calculate the cosine similarity from equation (2).
Step7: Find movies < 1.5
Step8: Top movies are recommended

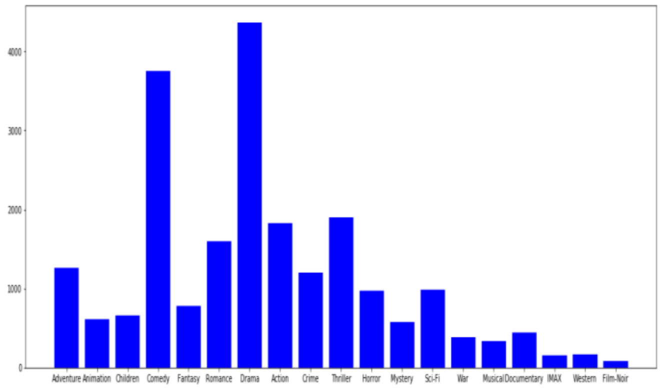

Figure2: Genre count

Figure 2 describes the bar chart which consists of genre on x axis such as adventure, animation, comedy, drama etc. and count on y axis. The highest genre count is drama which is highly available thus it is viewed by majority of the people and the lower genre count is Film Noir thus it is lowly viewed.

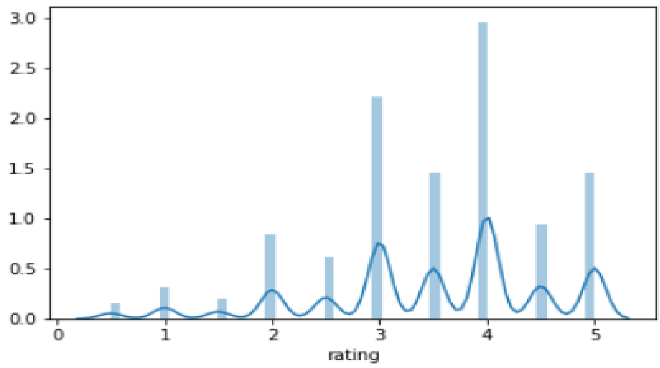

Figure3: Rating density

Figure 3 describe about the rating density using the distplot function. It gives the frequency of all the rating given in the dataset. The rating density differs on each movie and its genre. Table 10 shows the result of mean of userId and the mean rating is calculated and displayed. For each movieId the mean of userId who has watched that movie is calculated along with its ratings. For example, the mean of userId is 306.53 and rating is 3.92 for movieId 1

Table 10
Mean of rating and userId

movieId	userId (mean)	Rating (mean)
1	306.530233	3.920930
2	329.554545	3.431818
3	283.596154	3.259615

Figure 4 explores the mean rating that is arranged in descending order. Movies are plotted with respect to mean rating in the bar graph. The graph shows the Top 10 movies based on the mean rating. From the graph it shows that all the movieId are having 5 ratings.

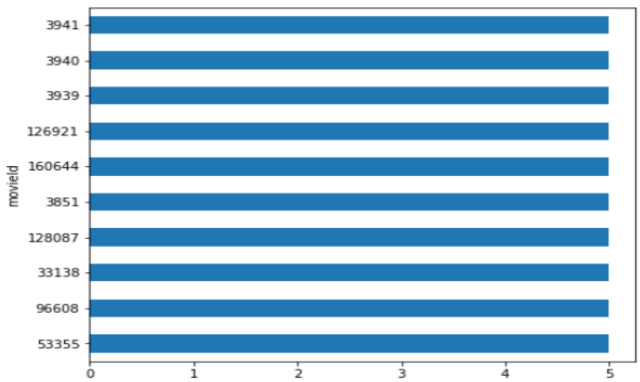

Figure4: Mean rating of top 10 movies

Figure 5 shows the movies which has mean rating value less than 1.5 From the graph it is shown that the movieId 1328 has lower rating about 0.7 and 1389 and 312 has higher rating.

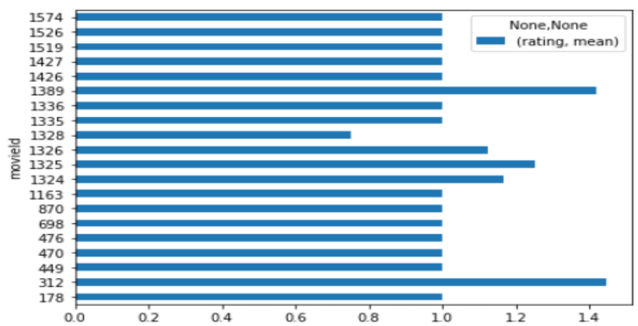

Figure5: Mean rating of movies less than 1.5

From the result Table 11 shows the Top 6 movies which are recommended using CBF. Based on the table the top movie recommended is Aqua man with the rating of 7 and the less recommended movie is Baran with the rating of 8.

Table 11
Movies recommended using CBF

user_id	movie_id	rating	title
84	1477834	7	Aqua man
38	7865	9	Touch of Evil
17	89065	8	All the King's Men
71688	1477834	6	Cool Hand Luke
1711	230211	5	Incendies
1017	100305	8	Baran

Hybrid Filtering

Hybrid filtering is a technique, and it is the combination of the content based filtering and collaborative based filtering. To perform hybrid filtering Convolutional Neural Network (CNN) based RELU neural network are used in the proposed work.

Convolutional Neural Network

In this proposed hybrid CNN network consists of several layers such as convolutional layer, pooling layer, and a fully connected layer. In the convolutional layer the input is convolved with the features and output is produced. In the pooling layer, the activation function RELU is performed. The input from the convolutional layer is given to the activation function and output is produced. In the fully connected layer, the output from the pooling layer is given to the sigmoid activation function and the final output is performed.

$$f'(q) = 0, \text{ for } q<0 \tag{3}$$
$$f'(q) = 1, \text{ for } q>=0 \tag{4}$$

Algorithm

Input: output of content and collaborative filtering. Output: recommend a movie based on user interest.
Step1: Get the movie recommended in the collaborative-based filtering. Step2: Get the movie recommended in the content based filtering. Step3: Combine the movie from content-based filtering and collaborative based filtering. Step4: using equation (3) and (4) to find the testing and training result. Step5: Finally, recommend the movie.

Table 12 shows the movies recommended using Hybrid Filtering related to Ironman. If the user likes/watches the Ironman movie then the hybrid recommendation system recommends the top most All the Kings men movie to the user and lowly recommend To Sir, With love movie to the user.

Table 12
Recommended movies using Hybrid filtering

Title	Vote Average	TMDb Id	Estimated Prediction
All the King's Men	6.3	25430	3.830741
Touch of Evil	7.6	1480	3.828555
Baran	7.6	43774	3.644015
Rebel Without a Cause	7.6	221	3.546547
Batman Begins	7.5	272	3.452842
Rififi	7.7	934	3.416841
To Sir, with Love	7.6	25934	3.379592

Experimental Results

To validate the recommender systems such as Collaborative Filtering, Content-Based Filtering, and Hybrid CNN-RELU based filtering performance metrics such as accuracy, recall, Root Mean Square Error (RMSE), and Mean Square Error (MAE) are calculated.

Accuracy

Classifier accuracy refers the ability of the classifier. It predicts correctly classified labels and shows how well a particular predictor can infer the value of a predicted attribute of new data.
AC= (TP+TN)/ (TP+TN+FP+FN) (5)

Recall

Recall is the number of correct results divided by the number of correct results and false results. In binary classification, recall is called sensitivity.
Recall=TP/ (TP+FN) (6)

RMSE

It is used to determine the error of a technique.
RMSE=sqrt($\sum_{k=0}^{n}$(a −â) ^{2}/n) (7)

MAE

It calculates the difference between the true values and the predicted values.
MAE = True values – Predicted values (8)

Table 13 displays the results of MAE, RMSE, accuracy, recall for Open Movie Database Dataset. For collaborative Filtering the values of MAE, RMSE are equal, Accuracy, and Recall 1% lower. For Content-Based filtering the values of MAE and Recall are equal and RMSE and accuracy are 2% and 1% are lower. For hybrid Collaborative Filtering and Content-Based Filtering the values of RMSE are equal and MAE, Accuracy and Recall are 1% lower. From the results it is shown that the hybrid filtering with CNN-RELU based method performs well when compared to other traditional filtering methods.

Table 13
Performance metrics results

Approach	MAE	RMSE	Accuracy	Recall
CF	0.66	0.86	0.62	0.72
CBF	0.64	0.84	0.61	0.73
CF+CBF	0.65	0.86	0.60	0.72
HF+RELU	0.66	0.86	0.61	0.73

Conclusion

Film recommender system plays a significant job in recognizing a bunch of movies for clients in light of client interest. Many move suggestion frameworks are available for clients; but those frameworks have the limit of not prescribing the movie efficiently to the current clients. This paper presented a movie recommender system based on hybrid collaborative filtering and content filtering using CNN-RELU. From the outcomes, the selection of boundaries of recommendation score calculations can influence the recommender system. Performance is assessed by utilizing different metrics such as execution time, RMSE of rating forecast, and MAE. Top 10 films are recommended using hybrid filtering. From these outcomes the hybrid filtering provides better results in recommending movies. In future, different datasets will be considered by employing Multi- Criteria Decision Making for movies recommendation.

References

1. Chen Y., & de Rijke M. (2018), A collective variational autoencoder for top-n recommendation with side information. In: Proceedings of the 3rd workshop on deep learning for recommender systems. ACM, pp 3–9. https://doi.org/10.48550/arXiv.1807.05730.
2. Cheng H.T., Koc L., Harmsen J., Shaked T., Chandra T., Aradhye H., Anderson G., Corrado G., Chai W., Ispir M. (2016), Wide & deep learning for recommender systems, *In: Proceedings of the 1st workshop on deep learning for recommender systems. ACM*, pp 7–10. https://doi.org/10.1145/2988450.2988454.
3. Covington P, Adams J, & Sargin E (2016), Deep neural networks for youtube recommendations. *In: Proceedings of the 10th ACM conference on recommender systems. ACM*, pp 191–198. : http://dx.doi.org/10.1145/2959100.2959190.
4. Das, N., Borra, S., Dey, N., & Borah S. (2018), Social networking in web-based movie recommendation system. *In: Dey N, Babo R, Ashour AS, Bhatnagar V, Bouhlel MS (eds) Social networks science: design, implementation, security, and challenges. Springer, Berlin*, pp 25–45. https://doi.org/10.1007/978-3-319-90059-9_2.
5. Fernando Ortega, Jose-Luis Sanchez, Jesus Bobadilla, & Abraham GutiRrez (2013), Improving Collaborative Filtering Based Recommender Systems

Results Using Pareto Dominance, *Information Sciences: an International Journal*, 239, 50-61. https://doi.org/10.1016/j.ins.2013.03.011.
6. Jhiang Zhang, Yufeng Wang, Zhiyuan Yuan, & Qun Jin, (2020), Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation, *Tsinghua Science and Technology,* Vol. 25, No. 2,. doi: 10.26599/TST.2018.9010118.
7. Kiran, R., Kumar, P., Bhasker, B. (2020), DNNRec: a novel deep learning-based hybrid recommender system, *Expert Syst Appl*, 144:113054. https://doi.org/10.1016/j.eswa.2019.113054
8. Kunaver, M., & Pozˇrl, T. (2017), Diversity in recommender systems— a survey. *Knowl Based Syst* 123:154–162. https://doi.org/10.1016/j.knosys.2017.02.009
9. LeCun, Y., Bengio, Y., & Hinton, G. (2015), Deep learning, *Nature*, vol. 521, no. 7553, pp. 436–444. https://doi.org/10.1038/nature14539.
10. Li S., & Fu Y. (2017), Robust representations for collaborative filtering. *Robust representation for data analytics. Springer, Berlin*, pp 123–146. https://doi.org/10.1007/978-3-319-60176-2_7.
11. Li X., & She J. (2017), Collaborative variational autoencoder for recommender systems. *In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. ACM*, pp 305–314. 10.1145/3097983.3098077.
12. Márcio Soares & Paula Viana. (2015), "Tuning Metadata for better movie content-based recommendation systems", *Multimedia Tools Appl,* 74, 7015-7036. https://doi.org/10.1007/s11042-014-1950-1.
13. Okura S., Tagami Y., Ono S., & Tajima A. (2017), Embedding-based news recommendation for millions of users, *In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, ACM*, pp 1933–1942. https://doi.org/10.1145/3097983.3098108.
14. Ouyang Y, Liu W, Rong W, & Xiong Z (2014), Autoencoder-based collaborative filtering. *In: International conference on neural information processing. Springer, Berlin*, pp 284–291. https://doi.org/10.1007/978-3-319-12643-2_35.
15. Sang-Min Choi, Sang-Ki Ko, & Yo-Sub Han, (2012), A movie recommendation algorithm based on genre correlations, Expert Systems with Applications, Vol. 39, 8079-8085. https://doi.org/10.1016/j.eswa.2012.01.132
16. SRS Reddy, Sravani Nalluri, Subramanyam Kunisetti, Ashok S., & Venkatesh B. (2019), Content-Based Movie Recommendation System using Genre Correlation, Smart Intelligent Computing and Applications, Springer. https://doi.org/10.1007/978-981-13-1927-3_42.
17. Sun Y., Mao H., Guo Q., & Yi Z. (2016). Learning a good representation with unsymmetrical auto-encoder, *Neural Comput Appl*, 27(5), 1361–1367. http://dx.doi.org/10.1007/s00521-015-1939-3.
18. Tahmasebi, H., Ravanmehr, R., & Mohamadrezaei, R. (2021), Social movie recommender system based on deep autoencoder network using Twitter data, *Neural Comput & Applic*, 33, 1607–1623. https://doi.org/10.1007/s00521-020-05085-1. https://doi.org/10.1007/s00521-020-05085-1.
19. Wang K., Xu L., Huang L., Wang C.D., & Lai J.H. (2019), SDDRS: stacked discriminative denoising auto-encoder based recommender system, *Cogn Syst Res,* 55,164–174. https://doi.org/10.1016/j.cogsys.2019.01.011

20. Wu, Y., DuBois, C., Zheng, A.X., & Ester M. (2016), Collaborative denoising auto-encoders for top-n recommender systems, *In: Proceedings of the ninth ACM international conference on web search and data mining. ACM*, 153–162. http://dx.doi.org/10.1145/2835776.2835837.
21. Xiong, N., Vasilakosb, A.V., Yang, L.T., Wang, C., Kannane, R.C., & Chang, Y. Pan. (2010), A novel self-tuning feedback controller for active queue management supporting TCP flows, Inf. Sci, 180(11), 2249–2263. https://doi.org/10.1016/j.ins.2009.12.001
22. Zhang, S., Yao L, Sun, A., & Tay, Y., (2019), Deep learning-based recommender system: a survey and new perspectives, *ACM Comput Surv (CSUR,)* 52(1), 1–38. https://doi.org/10.1145/3285029.