

MOVIE RECOMMENDER SYSTEM USING COLLABORATIVE FILTERING

Dr. G.N.BEENA BETHEL, Professor, MTech(Ph.d), Department of Computer Science and Engineering, beenabethel@gmail.com

D.Malles, BTech, Department of Computer Science and Engineering, malleshyadav6452@gmail.com

G.Sree Sai Raghavendar, BTech, Department of Computer Science and Engineering, raghavendar.gudipati@gmail.com

I.Goutham, BTech, Department of Computer Science and Engineering, inumulagoutham91@gmail.com

M.Shyam Nayak, BTech, Department of Computer Science and Engineering, shyamnayak906@gmail.com

K.Pavan Saradhi, BTech, Department of Computer Science and Engineering, kantipudi.saradhi@gmail.com

ABSTRACT: In this day in a contemporary era, there are limitless movies released every year, and users also inhabit varied choices of movies. the purpose of this project is to build movie recommendation engines for a better experience for the user and keep them engaged by recommending the movie of his/her choice. In this study, we have implemented and evaluated Content-based, Collaborative based filtering(Item-based) on the movie-lens-small-latest dataset. moreover, we build a hybrid model such as Collaborative based filtering with a Random forest model to increase the accuracy of movies recommended to the user, this study definitely helps the user to get a better experience and save a lot of time in choosing the great movie .Some people like genre-specific movies be it a thriller, romance, or sci-fi, while others focus on lead actors and directors. Recommendation System is a filtration program whose prime goal is to predict the “rating” or “preference” of a user towards a domain-specific item or item.

Keywords— *Movie recommender system, collaborative filtering.*

1. INTRODUCTION

Recommendation systems are predicting systems that radically recommend items to users or users to the items, and sometimes users to users too. Tech giants like YouTube, Amazon Prime, Netflix use similar methods to recommend video content according to their desired interest. As the internet contains huge loads of data, finding your content is very difficult and can be very time consuming, thus the recommendation plays an important role in minimizing our effort. These systems are getting more popular nowadays in various areas such as in books, videos, music, movies, and other social network sites where the recommendation is used to filter out the information. It is a tool that is using the user’s information to improve the suggestion result and give out the most preferred choice. User/Customer satisfaction is key for building the tool. It is beneficial for both customers and companies, as the more satisfied the customer is, the more likely he/she would want to use the system for their ease, which would ultimately make revenues for the companies. Recommendation system should always be improved as the user choice can differ from

2. LITERATURE REVIEW

Movie Recommender System Based on Percentage of View:

other users and if the user is not happy with the result, he/she might not use it again which is the case with our system. Although there are a lot of algorithms, collaborative filtering is the most popular one used by the companies as it involves user's interactions more. Collaborative filtering can predict better than content-based filtering because it analyses the user's browsing history and compares with other users and then suggests results [15]. Whereas, the content-based filtering takes the user's information as an input to find similar movies and recommends them in descending order (using cosine similarity). There's another method named context-based filtering where it extracts more information from the user like mood, release date, genre, etc., to give more efficient results. Our goal in this project was to keep our system very accurate compared to other recommendation techniques while making it as simple as possible. Content-based filtering has some drawbacks and a lack of accuracy and preciseness. So the proposed system is the collaborative filtering recommendation system using nearest neighbors.

With ever-increasing data on the internet, finding the desired content has become harder and that is why recommender systems' role is very important in business. As a specific example, media service providers, such as Netflix, can improve their service by recommending desirable content to each user. Most of the previous studies used explicit feedback of users, through likes and dislikes, to recommend items to their customers. However, in many cases, there is not much explicit feedback about items which cripples typical recommender systems to operate efficiently and provide accurate recommendation. In this paper, a percentage of view approach is proposed to find relevant movies for customers. To prove the effectiveness of the approach, first, it is shown that this feature can be a good indicator of users' like and dislike. Then the best approach is determined and used in a recommender system for Namava, a media service provider. Then the performance of this recommender system is compared to a random recommender system and the effectiveness of the approach is shown..

Fully content-based movie recommender system with feature extraction using neural network:

In recent years, movie industry is getting more and more prosperous. There are hundreds of movies released every year. However, it is difficult to notice the releasing of every movie, not to mention actually seeing it. Therefore, movie recommender system has become more and more popular as a research topic. Among a variety of movie recommender systems, content-based methods always ring a bell when it comes to recommending new movies. Content-based method uses the content of the movie as input so that it does not suffer from the "cold-start" problem. In this paper, we propose the Fully Content-based Movie Recommender System (FCMR) to recommend movies to users. The proposed method trains a neural network model, Word2Vec CROW, with content information



Fig.1: Content based filtering

(e.g., cast, crew, etc.) as the training data to obtain vector form features of each element, and then take advantage of the linear relationship of learned feature to calculate the similarity between each movie. In the end, the proposed FCMR recommends movies based on the similarity. The experiments are conducted on a massive real world dataset, and the intuition behind our proposed method has been proven by the experiment results.

2.3 Movies recommendation system using collaborative filtering and k-means:

The purpose of this research is to develop a movie recommender system using collaborative filtering technique and K-means. Collaborative filtering is the most successful algorithm in the recommender system's field. A recommender system is an intelligent system that can help a user to come across interesting items. This paper considers the users m (m is the number of users), points in n dimensional space (n is the number of items) and we present an approach based on user clustering to produce a recommendation for the active user by a new approach. We used k-means clustering algorithm to categorize users based on their interests. We evaluate the traditional collaborative filtering and our approach to compare them. Our results show the proposed algorithm is more accurate than the traditional existing one, besides it is less time consuming than the previous existing methods.

2.4 Determining Optimal Number of Neighbors in Item-based kNN Collaborative Filtering Algorithm for Learning Preferences of New Users:

Although the collaborative filtering (CF) is one of the efficient techniques to develop recommender systems, it suffers from a well-known problem called cold start which is a challenge to know the new user preferences. Ask To Rate technique is a simple way to solve this problem. In this technique, some items are shown to the new user, and ask her/him to rate them. Usually, Ask To Rate technique selects the items using kNN algorithm. However, determining k or number of the new user's neighbors in this algorithm is critical, because it

affects the accuracy of recommender system. In this paper, a CF based recommender system is improved by Ask To Rate technique to solve cold start problem. Consequently, k or number of the new user's neighbors is determined by an experimental evaluation. The experimental results on MovieLens dataset show that the highest accuracy of recommendations can be seen when the number of neighbors is set by a low value e.g. 10-15 neighbors.

2. Cosine similarity to determine similarity measure: Study case in online essay assessment:

Development of technology in educational field brings the easier ways through the variety of facilitation for learning process, sharing files, giving assignment and assessment. Automated Essay Scoring (AES) is one of the development systems for determining a score automatically from text document source to facilitate the correction and scoring by utilizing applications that run on the computer. AES process is used to help the lecturers to score efficiently and effectively. Besides it can reduce the subjectivity scoring problem. However, implementation of AES depends on many factors and cases, such as language and mechanism of scoring process especially for essay scoring. A number of methods implemented for weighting the terms from document and reaching the solutions for handling comparative level between documents answer and expert's document still defined. In this research, we implemented the weighting of Term Frequency - Inverse Document Frequency (TF-IDF) method and Cosine Similarity with the measuring degree concept of similarity terms in a document. Tests carried out on a number of Indonesian text-based documents that have gone through the stage of pre-processing for data extraction purposes. This process results is in a ranking of the document weight that have closeness match level with expert's document.

2.6 ORBIT: Hybrid movie recommendation engine:

Today, users will get so many movie recommendations websites, which suggest users best movies according to their interests. All

these websites have implemented one of the conventional content, context and collaborative recommendations algorithms. Alone, these algorithms are failed to recommend best and efficient recommendations to user. So, there is a need to evolve a unique algorithm which combines the features of conventional algorithm along with its new features. This paper describes the ORBIT, which is a movie recommendation engine, based on a unique Hybrid recommendation algorithm, satisfies a user by providing best and efficient books recommendations. Comparative case study of conventional recommendation algorithms to ORBIT's Hybrid movie recommendation algorithm has also been studied and presented in this paper. This case study is based on evaluating criteria of recommendation algorithm i.e. accuracy, precision, recall, F-measure etc. Results of this case study are represented in the form of tables and graphs to clearly specify the need of ORBIT.

3. IMPLEMENTATION

Content-Based Filtering is one of the existing system to which is the algorithm recommends products that are similar to the ones that a user has liked in the past as shown in fig2. Example- In this case if a user like Babhubali movie then it will recommend him the movies of same actor Director movies with same genre . It tries to find similar movies to that using the information available in the database such as the lead actors,the director etc.

Disadvantages:

Different products do not exposure to the user.

CONTENT-BASED FILTERING

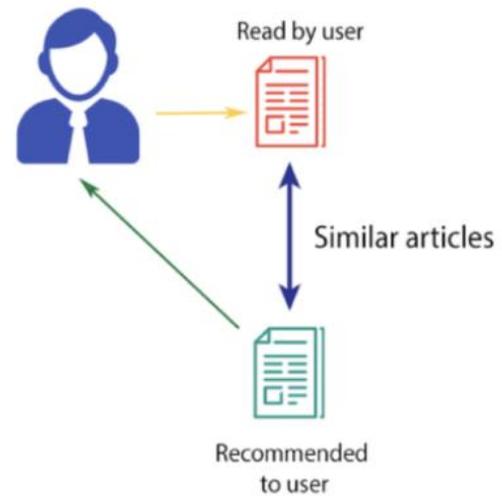


Fig.3: Existing system

To overcome the problems faced in Content-based filtering,we are using Collaborative Filtering . It is based on the combination of the user's behavior and comparing and contrasting that with other users' behavior in the database. The history of all users plays an important role in this algorithm. The main difference between content-based filtering and collaborative filtering that in the latter, the interaction of all users with the items influences the recommendation algorithm while for content-based filtering only the concerned user's data is taken into account.

They are 2 types of Collaborative Filtering .They are

- User-based Collaborative
- Item –based Collaborative

The concept in this case is to find similar movies instead of similar users and then recommending similar movies to that 'A' has had in his/her past preferences. For example, there are 2 movies 'A' and 'B' .and if most common users have rated 'A' and 'B' both similarly and it is highly probable that 'A' and 'B' are similar. If someone has watched and liked 'A' they should be

recommended 'B' and vice versa as shown in fig4.

Advantages:

There are usually a lot fewer items than people, therefore easier to maintain and compute the data. User can explore different products.

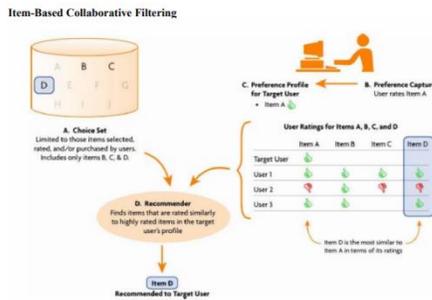


Fig.4: Item based collaborative filtering

MODULES:

Module-1(Data Collections and Data Preprocessing):

- In this module, collect the datasets (movie dataset and rating dataset.)
- Convert the raw_inputs to necessary data by removing null values from rating dataset.
- Join the 2 sets data by movieId which makes a required matrix.

Module-2:

- AdaBoosting increases Accuracy of Data.
- Perform Collaborative Filtering on matrix.
- Create a UI .

Module-3:

In this module use Random Forest algorithm to find the top 10 recommended movies to movie given by user.

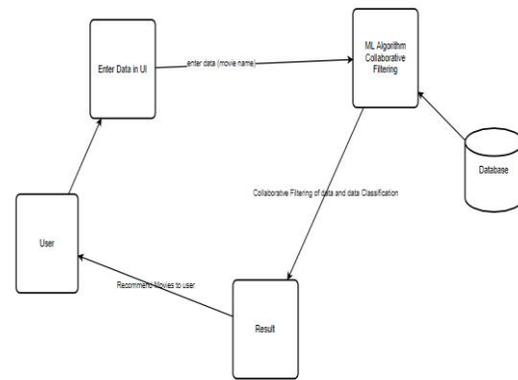


Fig.5: System architecture

5. ALGORITHM

RANDOM FOREST:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

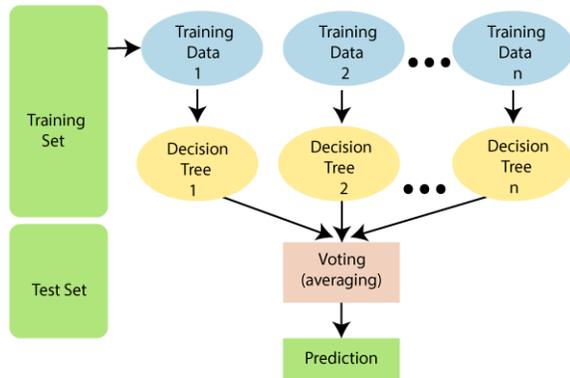


Fig.6: Random forest model

Uses:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Random forest algorithm working:

Random Forest works in two-phase

first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Applications of Random Forest:

There are mainly four sectors where Random forest mostly used:

Banking: Banking sector mostly uses this algorithm for the identification of loan risk.

Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.

Land Use: We can identify the areas of similar land use by this algorithm.

Marketing: Marketing trends can be identified using this algorithm.

Advantages of Random Forest:

1. Random Forest is capable of performing both Classification and Regression tasks.
2. It is capable of handling large datasets with high dimensionality.
3. It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest:

1. Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

6. CONCLUSION

Recommender systems expand the scope of tailored information retrieval on the Internet. It also helps to relieve the problem of information overload, which is a regular occurrence with information retrieval systems, by allowing users to access items and services that aren't easily available to other users on the system. We devise a technique that focuses on the user's specific preferences, and movies are recommended to him based on his prior evaluations. This method aids in the improvement of suggestion accuracy. In the realm of recommender systems, collaborative filtering is the most effective and widely used

algorithm. By proposing intriguing things, it assists customers in making better decisions. Despite being the best, this algorithm has low accuracy and a long running time. To address these issues, this research developed a user clustering-based recommendation strategy that uses the Euclidian distance to compute two users to cluster datasets. To create predictions, this approach combines clustering and neighbor votes. Techniques such as fuzzy c-means in the group phases of the first system may be developed in the future to give more effective segmentation.

7. FUTURE SCOPE

Some people are Fickle minded ,there movies preference is changes from time-to time.In such cases Iten based is more accurate than movie-based collaborative filtering because in Item based algorithm searches for similar movie ratings to recommend a movie to user by a given movie.

REFERENCES

1. C. S. M. Wu, D. Garg, and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 11-15, IEEE, 2018 Nov.
2. R. E. Nakhli, H. Moradi, and M. A. Sadeghi, "Movie Recommender System Based on Percentage of View," In 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEL), pp. 656-660, IEEE.
3. H. W. Chen, Y. L. Wu, M. K. Hor, and C. Y. Tang, "Fully contentbased movie recommender system with feature extraction using neural network," In2017 International Conference on Machine Learning and Cybernetics (ICMLC) , vol. 2, pp. 504-509, Jul, 2017, IEEE.
4. P. Phorasim and L. Yu, "Movies recommendation system using collaborative filtering and k-means," International Journal of Advanced Computer Research, vol. 7, no. 29, p.52, 2017.
5. M. Bahadorpour, B. S. Neysiani, and M. N. Shahraki, "Determining Optimal Number of Neighbors in Item-based kNN Collaborative Filtering Algorithm for Learning Preferences of New Users," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, no. 3, pp.163-167, 2017.
6. A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," In2016 4th International Conference on Cyber and IT Service Management, pp. 1-6, IEEE, Apr, 2016.
7. D. Pathak, S. Matharia, and C. N. S. Murthy, "ORBIT: Hybrid movie recommendation engine," In 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), pp. 19-24, IEEE, Mar, 2013.
8. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In Proceedings of the 10th international conference on World Wide Web, pp. 285-295, Apr, 2011.
9. R. Van Meteren, and M. Van Someren, "Using content-based filtering for recommendation," In Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, vol. 30, pp. 47-56, May 2000.
10. L. E. Molina and S. Bhulai, Recommendation System for Netflix, 2018.
11. P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems." In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA, pp. 157-164, 2011.
12. M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," J Inf Eng Appl, vol. 3, no. 10, 2013.

13. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

14. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," In Proceedings of the 1994 ACM conference on Computer supported cooperative work (1994, October) (pp. 175- 186).

15. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to Usenet news," Communications of the ACM, vol. 40, no. 3, pp. 77-87.

Journal of Engineering Sciences