Enhanced Movie Recommendation Engine using Content Filtering, Collaborative Filtering and Popularity Filtering

Sagar Bala[1]

[1]Computer Science Engineering(Pursuing),Department of Computer Science ,SRM institute of Technology, Kattankalathur, Tamil Nadu, INDIA

Abstract - *A movie recommendation engine is used to recommend movies by giving the user the authority to watch what they desire. In this hectic life, People don't get that much time to watch the movies of their choice, movie recommendation engines consider all movies especially the niche movies, and gives the list of all preferred movies as manually searching the preferred movies will take more time. Thus, MRE(Movie Recommendation Engine) can be used for enhancing the user experience. MRE used in this study makes use of Popularity filtering, Content filtering, and Collaborative filtering using three different datasets. MRE gives users a more reliable, real-time, and user-friendly. However, the accuracy of the MRE can be improved using more precise data.*

Key Words: MRE(Movie Recommendation Engine), niche movies, Popularity filtering, Content filtering, Collaborative filtering, reliable, user friendly.

1. INTRODUCTION

In this generation, the internet has become an important part of our life. It gives us a massive amount of information and choices. Most of the information available is inconsistent and improper thus, will not give satisfactory result to the user, and information gets wasted. To get the desired result users to have to waste a lot of time and effort. To save time and effort recommendation systems are build using machine learning approaches. A recommendation system is developed to selectively choose information according to the user's needs, importance, and previous experiences. It helps to give relevant information to the users. Recommendation systems use a data filtering method to make the information more authentic and useful. Recommended systems have been build in various like for books, clothing, music, grocery, etc.

This MRE (Movie Recommendation Engine) is used for recommending movies according to the user's needs and requirements. It is a more improvised model as it uses three filtering method Content filtering, Collaborative filtering, and popularity filtering in datasets. It uses cosine similarity as a metric. The MRE system reduces time and manual effort to search and sort the movies.

The organization of the paper is as follows. Section 1 describes the Introduction of the model. Section 2 is about the Literature survey. Section 3 describes the methodologies and implementation. Section 4 gives the results and section 5 gives the conclusion and future enhancement.

2. LITERATURE SURVEY

This section discusses the recent researches done on Movie recommendation engines that have used Machine learning algorithms.

Using a hybrid approach of collaborative filtering and content filtering to improve the performance of movie recommendation, Lekakos, G. and Caravelas, P, 2008[1].Using attribute correlation for Item-based collaborative filtering case study for a movie recommendation, Pirasteh, P., Jung, J.J. and Hwang, D.,2014[2].Improved movie recommendation using genre for collaborative filtering, Lee,2007[3].Content boost collaborative filtering movie recommendation using local and global similarity and missing data prediction, Özbal,2011[4]. Using an artificial immune system and collaborative filtering in movie recommendation, Liao,2014[5].Movie recommendation system through group-level sentiment analysis in microblogs, Li,2016[6]. Using an artificial immune system in the movie recommendation engine, Chen, 2008[7].Movirec recommendation system using collaborative filtering based on user information, Manoj Kumar, 2015[8].

Content filtering and Collaborative filtering methods are two of the important methods used in this study.

Content filtering is used in web browsing, Bellinson, 2004[9]. Live broadcasts in TV terminals using real-time content filtering, Jin, 2008[10]. Collaborative filtering is used in restricted Boltzmann machines, Salakhutdinov, 2007[11].Sparse 3D transform-domain collaborative filtering for image denoising. Dabov, 2007[12].

This study uses the three approaches using Content filtering, Collaborative filtering, and Popularity filtering to make a better-improvised model.

3. METHODS AND IMPLEMENTATION

3.1 Data Information

In this study 3 datasets were used namely TMDB_5000_credits,TMDB_5000_movies for content and popularity based filtering .The features of TMDB_5000 datsets after combining both dataset mainly used for reccomendation were title, plot, cast, director, Keyword, vote_average, vote_count and Movie_metadata dataset for collaborative filtering.The features used were movieId,title,genres.

3. 2 Collaborative Filtering

Collaborative filtering takes more information from the user into consideration. It gives automatic prediction based on searching a larger group of people finding a group of people who like the particular thing. It is based on reactions by similar users. It has a wide range of applications in the area where there is a large amount of user data and interaction. Automatic recommendation for users using the information preference of many other users. Suppose if user 1 likes one set of movies and user 2 another set of movies respectively. For target user, the recommendation will be a set of movies common among the two user1 and user 2.Fig.1.

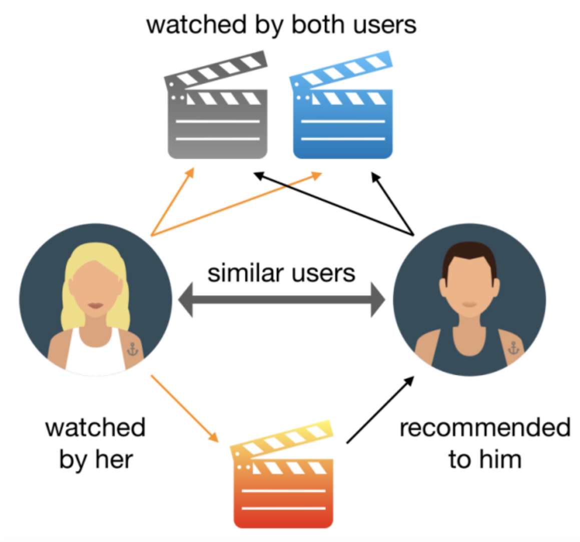

Fig.1. Collaborative Filtering Diagram

3.3 Content Filtering

Content filtering is the continuation of a collaborative filtering method. Content Filtering recommendation is a user-specific classification based on user preferences. It tries to recommend the items which user present or past feedback. It works on the item based on the comparison between the content of items and user profiles. It has a wide range of applications in the field where there is user interaction. Using the similarity in other items the most similar items are recommended.Fig.2.

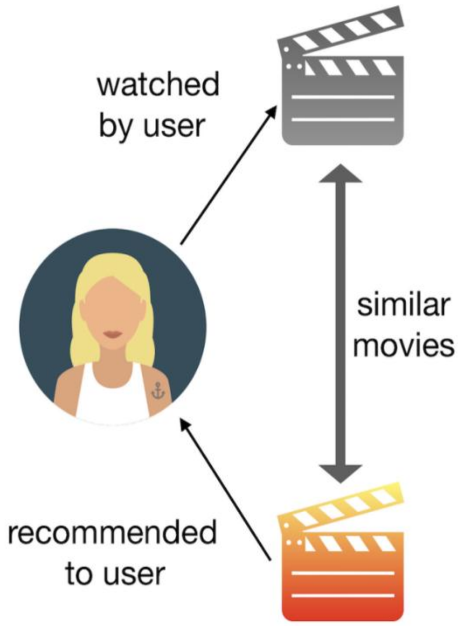

Fig.2. Content Filtering Diagram

3. 4 Popularity filtering

All registered user of IMDB can cast their vote for any movie. The registered user's votes are taken into consideration to calculate a single rating for a movie. To calculate the popularity of the movie IMDB created a formula weighted rating method. The more the popularity of the movie is the higher will be the recommendation to the user and the lower the popularity is the lower will be the recommendation for the user.IMDB weighted average is updated numerous times a day. Eq.1.

$$W = R\frac{v}{v+m} + C\frac{m}{v+m}$$

where:

W = Weighted Rating
R = average for the movie as a number from 0 to 10 (mean) = (Rating)
v = number of votes for the movie = (votes)
m = minimum votes required to be listed in the Top 250 (currently 1300)
C = the mean vote across the whole report (currently 6.7)

Eq.1. IMDB Weighted Rating formula

Cosine similarity is used for calculating the similarity between the user and content .E.q.2.

3.5 Cosine Similarity

The definition of similarity between two vectors **A** and **B** is, in fact, the ratio between their dot product and the product of their magnitudes. If the similarity is number bound between 0 and 1. The vectors will be equal to 1,f the two vectors are identical and 0, if the two vectors are orthogonal.E.q.2.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

E.q.2. Cosine Similarity equation

The implementation of the MRE system in given Fig.3.

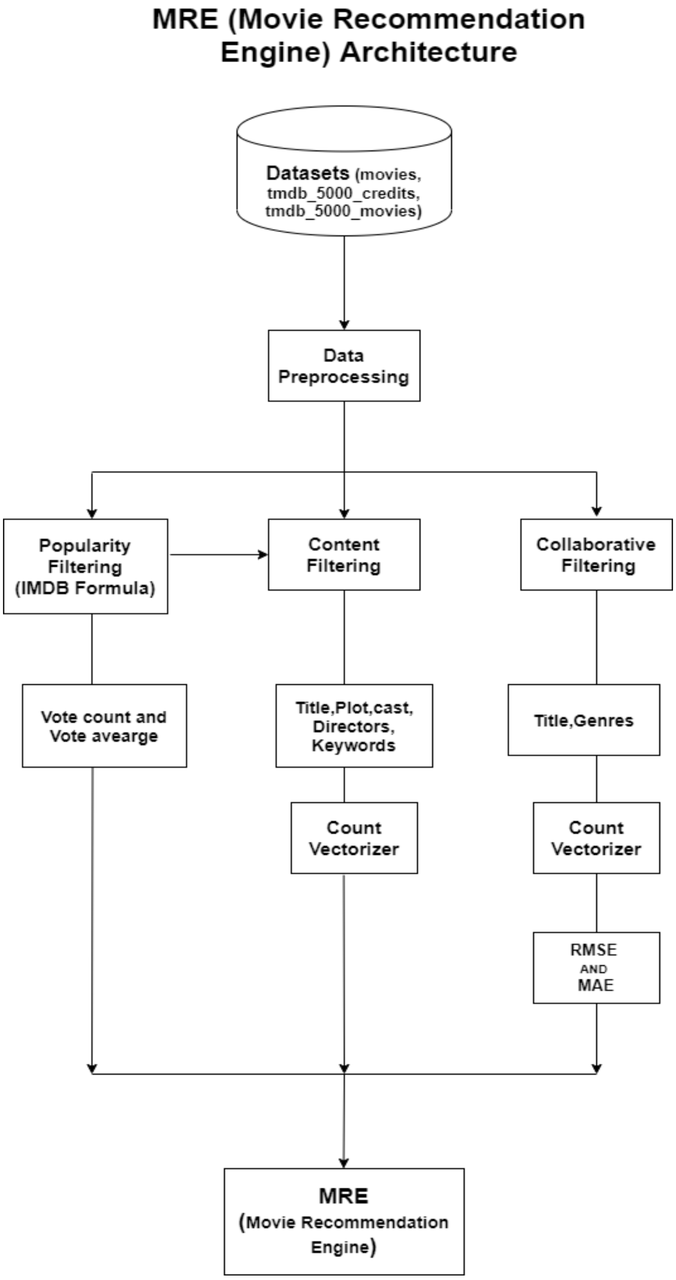

Fig.3. MRE System Architecture

There are two performance metrics RMSE and MAE were used in this study for Collaborative filtering .

3.6 Root Mean Squared Error(RMSE)

RMSE is measure of spread of regression line data by comparing the predicted value and the observed value ,E.q.3.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

E.q.3. Root Mean Squared Error (RMSE) equation

3.6 Mean Absolute Error(MAE)

MAE is used to measure average error between the predicted value and the observed value for a set of prediction ,E.q.4.

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

E.q.4. Mean Absolute Error(MAE) equation

4. Results

In this study, the datasets used are TMDB_5000_credits, TMDB_5000_movies, and movie_metadata, which contains over 5000 data. The recommended system uses three approaches namely collaborative filtering, content filtering, and popularity filtering.

The output for top 3 movies using content filtering with title ,cast ,director and keywords as features are given below.Fig.4.

0	Avatar
1	Pirates of the Caribbean: At World's End
2	Spectre

Fig.4. Content Filtering using title, cast, director and keywords

Using content filtering taking Plot as features and taking "Pulp Fiction" movie as example .The top 10 movies whose plot is similar to "Pulp Fiction" movie .Fig.5.

```
get_recommendations('Pulp Fiction', cosine_sim2)

830             Kill Bill: Vol. 2
574                      S.W.A.T.
684             The Hateful Eight
1156               A Time to Kill
2124             Lakeview Terrace
2822                 Jackie Brown
527                       Be Cool
1253                Kiss of Death
2723       The Caveman's Valentine
4300               Reservoir Dogs
Name: title, dtype: object
```

Fig.5. Content Filtering using Plot

The output for top 10 movies using collaborative filtering and taking features as Title ,genres.Taking "Spectre" as movie example the top 10 movies whose title and genre are similar are given below . Fig.6.

```
cosine_sim = cosine_similarity(count_matrix)
movie_user_likes = 'Spectre'
```

```
print(get_title_from_index(movie[0]))
i=i+1
if i>10:
 break

John Carter
Robin Hood
White House Down
Rise of the Guardians
The Lucky Ones
Men in Black 3
Hugo
Charlie and the Chocolate Factory
Bee Movie
Harry Potter and the Half-Blood Prince
The Lion King
```

Fig.6. Collaborative Filtering using Title,genres

Finally, the output for top 5 most popular movies using popularity filtering taking vote count and vote average as features are given below.Fig.7.

```
pop= df2.sort_values('popularity', ascending=False)
pop['title'].head(5)

546                    Minions
95                Interstellar
788                   Deadpool
94     Guardians of the Galaxy
127         Mad Max: Fury Road
Name: title, dtype: object
```

Fig.7. Popularity filtering using vote count and vote average

The collaborative filtering takes titles and genres to feature into consideration. The content filtering method takes plot, cast, keywords, title, and directors and popularity filtering consider vote count respectively. The result shows recommendations based on plot, characters, directors, genres, titles, votes, and keywords. This MRE changes the dynamic of the recommendation engine as it gives more options to select.

FOLD/ ERRORS	FOLD 1	FOLD 2	FOLD 3	FOLD 4	FOLD 5	Mean
RMSE	0.8687	0.8830	0.8834	0.8659	0.8667	0.8735
MAE	0.6702	0.6764	0.6768	0.6666	0.6669	0.6714

To reduce sparsity and scalability Single value decomposition (SVD) is used.RMSE and MAE are calculated for collaborative filtering in 5 folds.RMSE value is 0.87 and the MAE value is 0.67 states the model is more than good.

5. CONCLUSIONS

This MRE recommended system aims to build a more improvised model for recommendations using content filtering, popularity filtering, and collaborative filtering. The MRE model reduces the drawbacks of all the three methods by considering more number of features and makes it more reliable and efficient by giving the desired result. With this massive amount of information available these days, it becomes important to be selective with kind of entertainment. Thus, MRE gives the user the relevant recommendation of movies in which they are interested. It also gives the user more options to select making it more user-friendly and real-time. Taking more number of features into consideration enhances the classification.

6. Future scope of research

The recommendation can further be improved by including the association filtering method or incorporating more features for filtering like awards, release date, language, etc. It can also be improved by using different and accurate data or different clustering methods in machine learning. This study in the future can be applied to different systems to improve their recommended systems like Books, groceries, clothes, etc. Using a neural network and different classification algorithms can be used to give a more improvised model.

REFERENCES

[1] Lekakos, George, and Petros Caravelas. "A hybrid approach for movie recommendation." Multimedia tools and applications 36.1-2 (2008): 55-70.
[2] Pirasteh, Parivash, Jason J. Jung, and Dosam Hwang. "Item-based collaborative filtering with attribute correlation: a case study on movie recommendation." Asian conference on intelligent information and database systems. Springer, Cham, 2014.
[3] Lee, Jae-Sik, and Seog-Du Park. "Performance improvement of a movie recommendation system using genre-wise collaborative filtering." Journal of Intelligence and Information Systems 13.4 (2007): 65-78.
[4] Özbal, Gözde, Hılal Karaman, and Ferda N. Alpaslan. "A content-boosted collaborative filtering approach for movie recommendation based on local and global similarity and missing data prediction." The Computer Journal 54.9 (2011): 1535-1546.
[5] Liao, Jhen-Fu, et al. "Application of artificial immune systems combines collaborative filtering in movie recommendation system." Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2014.
[6] Li, Hui, et al. "An intelligent movie recommendation system through group-level sentiment analysis in microblogs." Neurocomputing 210 (2016): 164-173.
[7] Chen, Qi, and Uwe Aickelin. "Movie recommendation systems using an artificial immune system." arXiv preprint arXiv:0801.4287 (2008).
[8] Kumar, Manoj, et al. "A movie recommender system: Movrec." International Journal of Computer Applications 124.3 (2015).
[9] Bellinson, Craig, et al. "Content filtering for web browsing." U.S. Patent Application No. 10/183,657.
[10] Jin, Sung Ho, Jun Ho Cho, and Yong Man Ro. "Real-time content filtering for live broadcasts in TV terminals." Multimedia Tools and Applications 36.3 (2008): 285-301.
[11] Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." Proceedings of the 24th international conference on Machine learning. 2007.
[12] Dabov, Kostadin, et al. "Image denoising by sparse 3-D transform-domain collaborative filtering." IEEE Transactions on image processing 16.8 (2007): 2080-2095.