EAI Endorsed Transactions
on Creative Technologies
Research Article EAI.EU

Movie Recommender System Using Machine Learning

Sonika Malik*

Asst. Prof., Dept. of IT, Maharaja Surajmal Institute of Technology, New Delhi, India

Abstract

In this research, we propose a movie recommender system that can recommend movies to both new and existing customers. It searches movie databases for all of the relevant data, such as popularity and beauty that is required for a recommendation. We apply both content-based and collaborative filtering and evaluate their advantages and disadvantages. To build a system that delivers more exact movie recommendations, we employ hybrid filtering, which is a combination of the outcomes of these two processes. The recommendation engines are also used for business purposes and to make strategies for organizations. Due to the growing demands of customers and user's recommendation systems plays a huge role. These recommender systems also help us to utilize our time in the busy world by giving us more relevant searches. These systems are generally used with the movie's websites or with many commercial applications and are of great use. This type of recommendation system can be also used for precise results. It will make movies suggestions more relevant as per the need of the users.

Keywords: Content based filtering, collaborative filtering, singular value decomposition, cosine similarity

Received on 20 September 2022, accepted on 26 September 2022, published on 11 October 2022

doi: 10.4108/eetct.v9i3.2712

*Corresponding author. Email: sonika.malik@msit.in

1. Introduction

Our main goal is to create an improved recommender system that provides precise recommendations to the customer. Recommendation engines are generally a mixture of both content-based filtering and collaborative filtering.

The Collaborative Filtering technique is based on the user's previous queries and experiences [1][2]. We'll anticipate what other users with similar experiences would watch based on that, and then provide the user those recommendations. In collaborative filtering, we will make the customers or users' recommendations and suggestions based on past experiences and behaviours. It works on the principle, for instance, you, have selected an item in the cart or you have purchased it from the website. The same types of recommendations it will show on the basis of your selections of items in the cart.

Content-based filtering techniques make recommendations based on movie characteristics such as genre, director, actor, plot, and so on [3]. We can improve the recommendation by putting more emphasis on a specific attribute. We'll also include a popularity and rating element in this. With Cosine similarity and Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer, a content-based filtering technique is applied. The content-based filtering technique gives us recommendations that are based on the user's interests. If the user has searched any of the item's past history. This filter will do it works in its way by suggesting to us the same type of content which we are surfing earlier. This tactic is totally dependent on the taste of the customers. Nowadays, we basically use the mixture of both the filtration techniques and called it a hybrid filtration. So, the mixture of both of these types of filtration will result in the new tactic that is the hybrid model. This model will give us a more precise approach to filtration techniques.

2. Literature Review

In [4] authors suggested a reasonable method of adopting data mining to create a recommendation list. Their method was established on pairs of items which was faster than normal ARM. On average the recommendations scored 88.94 %. In [5] author used collaborative filtering is focused primarily upon the premise that users who have bought a particular product will have similar needs to other users who

also bought the product. Based on customer past purchases, consumer browsing habits, and user segments, author analyse and evaluate three variants of a CF-based recommender system. In [6] procedure used by the authors was collaborative filtering and the similarity measured used was the Pearson correlation coefficient. The dataset was taken from Movie-Lens-100k and the ratings above 2.5 was taken into consideration for recommendations. In [7] movie recommendation system based on a modified user similarity metric and opinion mining has been presented. The primary goal of this paper is to identify the different types of movie opinions (positive, negative, or neutral), as well as to recommend a top-k recommendation list to users. And this system will get the ratings on the basis of particular ratings and reviews. This system will also recommend users depending on patterns and their similarities. Finally, the suggested movie recommendation system was validated using multiple evaluation criteria, and the proposed system outperformed existing systems. In [8] author examine E-commerce big data, concentrated on the K-means clustering algorithm. Geographic location and the customer's unique identification number are used as clustering restrictions in this study. The challenge of mining such data is difficult. One of the most significant mining tasks is clustering related objects or data, which is extremely beneficial for classification and modelling. The K-means clustering method is a prominent partition-based clustering method that produces high-quality results. In [9] movie recommendation system was devised and implemented. In the world of movies, there are several genres, cultures, and languages to pick from. Users can be recommended a set of movies based on their interests or the popularity of the films. In Hollywood, 600 films are released on average per year, according to a poll. Recommendation algorithms are critical for streaming movie services like Netflix. In helping customers discover new movies to watch. So far, a substantial amount of work has been done in this area. However, there is always an opportunity for improvement. In [10] authors executed a movie recommendation using collaborative filtering. This system is created using Apache Mahout and evaluates the ratings to give movie recommendations. The system displayed the raw output from the collaborative filtering technique. The system recommends 10 movies to users and returns the closest neighbours which have the most similar taste preferences as the user.

3. Methodology

In the discipline of machine learning, classification algorithms that use several ways of organizing and classify information.

1. Collaborative Filtering
2. Content Based Filtering

Collaborative Filtering

Collaborative filtering is based on the fact that products and people's interests have a relationship. Many recommendation systems employ collaborative filtering to uncover these connections and make an appropriate recommendation of a product that the consumer would enjoy or be interested in. The relation between user-based and item-based filtering is shown in Fig. 1.

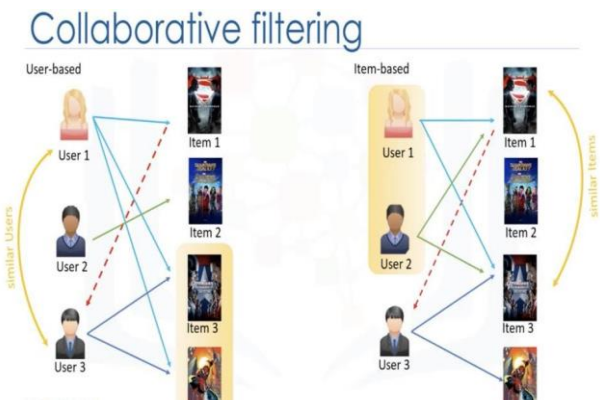

Figure 1. User based & Item based CF

The Singular Value Decomposition (SVD) is a linear algebra method that has been widely applied in machine learning as a dimensionality reduction mechanism. The SVD approach is a matrix factorization technique that decreases the number of features in a dataset by minimizing the space dimension from N to K (in which K<N). The SVD is a collaborative filtering algorithm that is utilized in the recommender system. It is organized as a matrix, with each row representing a user and each column representing an object. The ratings that users give to items are the elements of this matrix.

```
svd = SVD()
cross_validate(svd, data, measures=['RMSE', 'MAE'])
```

Figure 2. Computing the RMSE and MAE.

In Fig. 2, singular value decomposition is used in computing the root mean square error and mean absolute error.

Content Based Filtering

This filtering is done based on the product's description or certain data. Based on the context or description of the products, the algorithm determines their resemblance. The

user's previous purchases are considered when recommending related products.

1. Content Based Filtering using Cosine Similarity

Cosine Similarity is a metric that measures how similar two or more vectors are. The cosine of the angle between vectors is the cosine similarity. The vectors are usually non-zero and belong to an inner product space. The divide in between the Euclidean norms or vectors having magnitude or simply between the vectors describes the cosine similarity mathematically. Fig. 3 shows the relation between Collaborative filtering and Content Filtering.

$$\text{Similarity} = (A.B) / (||A||. ||B||) \quad (1)$$

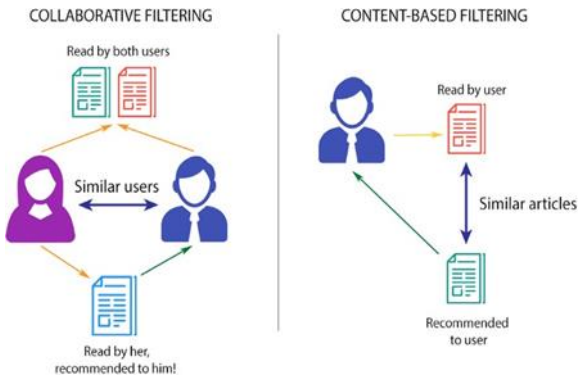

Figure 3. CF & Content Based Filtering

Many libraries such as scikit-learn, matplotlib has cosine similarities inbuilt which is of great use. In Fig. 4, count vectorizer matrix is used to calculate the occurrence of word in description of movie, after that we have compute the cosine similarity between the different movies.

```
count = CountVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0, stop_words='english')
count_matrix = count.fit_transform(smd['description'])

cosine_simr = cosine_similarity(count_matrix, count_matrix)
```

Figure 4. Content Based Filtering using cosine similarity

Content Based Filtering using TF-IDF

Term Frequency Inverse Document Frequency is a commonly used algorithm to convert the text into a more logical illustration. This is fit for the prediction algorithm. Term frequency is the number of words repeating or occurring in the document. Inverse term frequency is the informative part which is contained in the document. This gives the whole meaning to the documentations.

```
tf = TfidfVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0, stop_words='english')
tfidf_matrix = tf.fit_transform(smd['description'])
```

Figure 5. Computing the TF-IDF Vectorizer

In Fig. 5, TF-IDF vectorizer matrix is computed on the description of the movies provide in the database.

4. Result & Discussion

After learning and analysing the above methods we have tried to implement them. The implementations have been done on the given datasets. The dataset which we have taken are from the movies lens website. It has a huge database of the movies. We have considered the TMBD ratings for our implementations. We have taken in consideration genres, cast, crew, reviews and ratings. We have applied different types of filtration techniques like collaborative filtration and content based filtration.

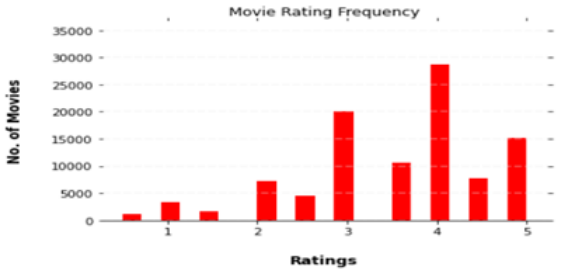

Figure 6. Movie Rating Frequency graph

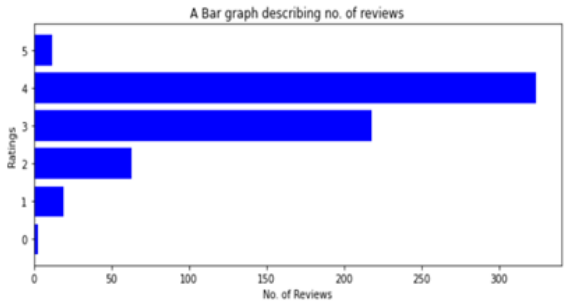

Figure 7. A bar graph describing no. of reviews

In Fig. 6, graph represents the no. of ratings given to the total number of movies in the database. In Fig. 7, graph is used to study the ratings and number of reviews that a particular movie is getting. This is basically a bar graph describing the no. of reviews that specific rating can have.

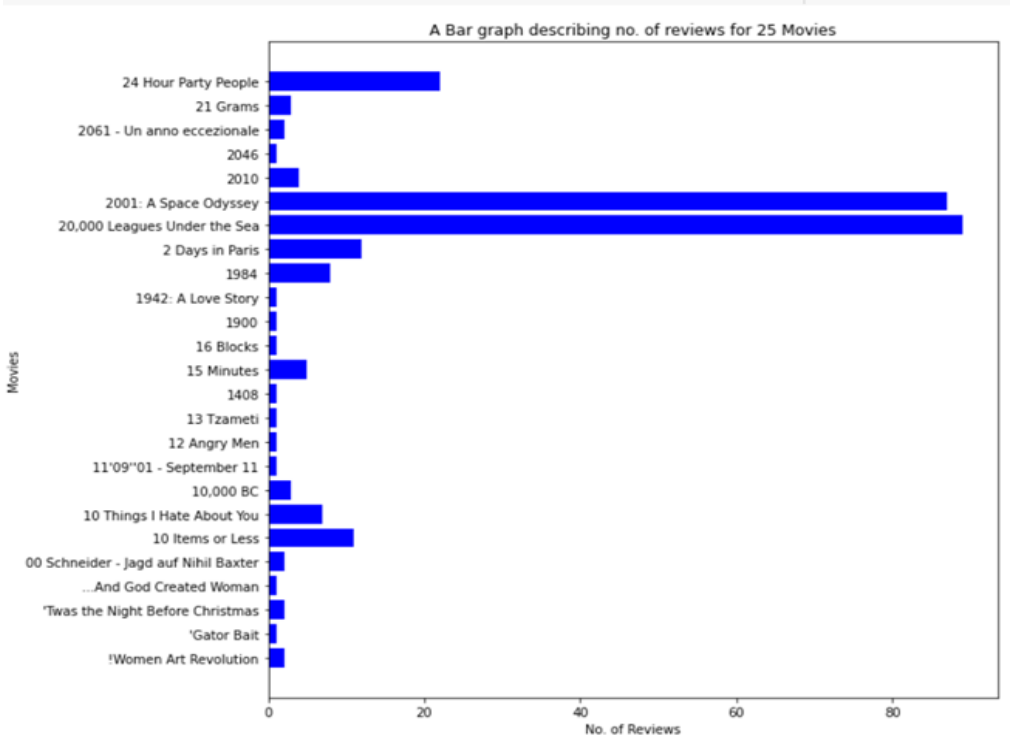

Figure 8. Graph describing no. of reviews for 25 movies

In Fig. 8, the graph is explaining the number of reviews of first 25 movies in the dataset. The given bar graph can be used for analysing and studying the extent for the particular movies. This graph gives us a clear picture of the number of reviews corresponding to a particular movie. Fig. 9 shows the ratings given by the user ID 1 to the different movies. In Fig. 10, we get an estimated prediction of 2.584 for movie ID 202 using collaborative filtering. This recommender has one great feature that it does not work on the basis of genre or what the people have watched. It purely works on basis of ratings or what the customers or users have rated for the specific product.

```
ratings[ratings['userId'] == 1]
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
5	1	1263	2.0	1260759151
6	1	1287	2.0	1260759187
7	1	1293	2.0	1260759148
8	1	1339	3.5	1260759125
9	1	1343	2.0	1260759131
10	1	1371	2.5	1260759135

Figure 9. Ratings given by userId =1 for different movieId.

```
svd.predict(1, 202)
```

```
Prediction(uid=1, iid=202, r_ui=None, est=2.5847099361698715
```

Figure 10. Computing similarity prediction using svd for user ID=1

In Fig. 11, we get the different movies on the basis of input movie provide 'The Godfather' using content based filtering with cosine similarity.

```
[ ] get_old_recommendations('The Godfather')
```

973	The Godfather: Part II
8387	The Family
9076	The Maid's Room
4196	Johnny Dangerously
9017	Manson Family Vacation
5728	Spider Baby
3509	Made

Figure 11. Content based filtering using cosine similarity

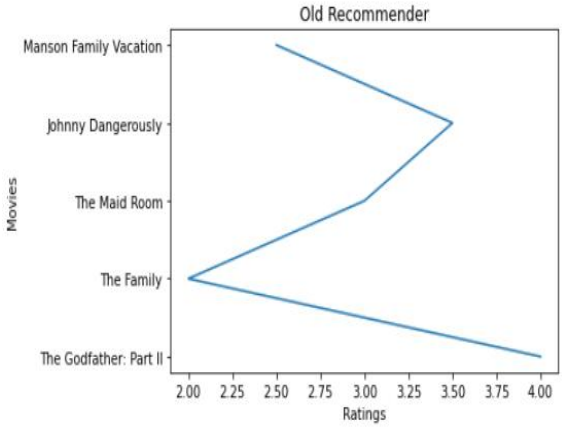

Figure 12. Old recommendation graph using cosine similarity

The above graph as shown in Fig. 12, provide the analytical understanding of the movies w.r.t. ratings with the help of traditional method i.e. cosine similarity, that recommends the movies of less ratings.

In content based filtering using TF-IDF vectorizer, the English words are eliminated which are not required for our recommendations. Elimination of these types of words will make our recommendations more precise than the traditional method of cosine similarity. In Figure 13 the output of content based filtering using TF-IDF vectorizer is more precise that the traditional cosine similarity method. With the ease of this method we can successfully get the suggestion of the users need.

```
get_recommendations('The Godfather')
```

```
973              The Godfather: Part II
8387                         The Family
3509                               Made
4196               Johnny Dangerously
29                      Shanghai Triad
5667                               Fury
2412                    American Movie
```

Figure 13. Content based filtering using TF-IDF vectorizer

Following is the graph as shown in Figure 14 explaining the movies with respect to ratings. It is a more precise as compare to the traditional method i.e. cosine similarity. In this TF-IDF vectorizer is used, which is more efficient as compare to previous one.

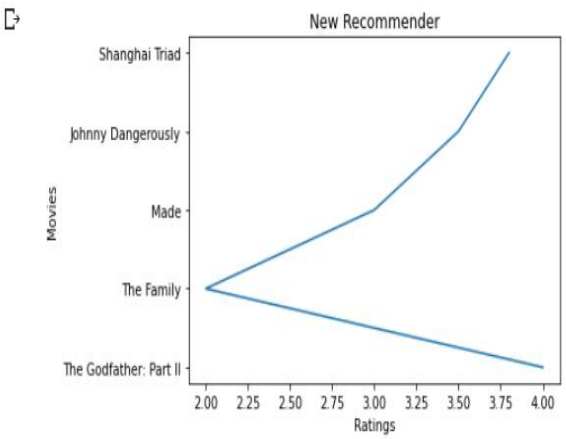

Figure 14. Recommendation graph using TF-IDF vectorizer

5. Conclusion

In this paper, we used a movie recommendation system based on machine learning algorithms. Consequently, users receive better suggestions as a result of collaborative filtering, which is based on their prior experiences and activities. To suggest movies to the user, we used the SVD algorithm in Collaborative Filtering. The fundamental problem with collaborative filtering is that if a new user has no previous experience, the recommender cannot provide relevant recommendations. It is also possible that collaborative filtering will fail to produce meaningful suggestions if the data becomes too large. By comparing the attributes of the specified item with those of other items, content-based filtering makes suggestions. A TF-IDF vectorizer and Cosine Similarity were employed for Content-based filtering. Due to its ability to count every word in movie genres, actors, and directors, TF-IDF vectorizer provides better results than cosine similarity.

References

[1] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems HandbookRecommender Systems Handbook, Springer, 2011, pp. 1-35J.

[2] "playboy Lead Rise of Recommendation Engines - TIME". TIME.com. 27 May 2010. Retrieved 1 June 2015.

[3] R. J. Mooney & L. Roy (1999).Content-based book recom mendation using learning for text categorization.In Workshop Recom.Sys.Algo.and Evaluation.

[4] Hosein Jafarkarimi; A.T.H. Sim and R. Saadatdoost A Naïve Recommendation Model for Large Databases, International Journal of Information and Education Technology, June 2012

[5] Prem Melville and Vikas Sindhwani, Recommender Syste ms, Encyclopedia of Machine Learning, 2010

[6] M. M. Reddy, R. S. Kanmani and B. Surendiran, "Analysis of Movie Recommendation Systems;with and without considering the low rated movies,"24-25 Feb,*2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Vellore, India, 2020, pp. 1-4, doi:10.1109/ic-ETITE47903.2020.453.

[7] Nagamanjula. R.; Pethalakshmi, A.; "A Novel Scheme for Movie Recommendation System using User Similarity and Opinion Mining", International Journal of Innovative Technology and Exploring Engineering, vol: 8, 2019, pp: 316-322

[8] Shaik, I.; Nittela, S.S.; Hiwarkar, T.; Nalla, S.; "K-means Clustering Algorithm Based on E-Commerce Big Data", International Journal of Innovative Technology and Exploring Engineering, vol: 8, 2019,pp: 1910-1914

[9] Pavithra, M.; Sowmiya, S.; Tamilmalar, A.; Raguvaran,S.; "Searching an Optimal Algorithm for Movie Recommendation System", International Research Journal of Engineering and Technology, vol: 6, 2019,pp:216-221

[10] Wu, C.-S. M., Garg, D., &Bhandary, U. (2018). Movie Recommendation System Using Collaborative Filtering, 23-25 Nov, 2018, IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018. doi:10.1109/icsess.2018.8663822