International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2063143

Implementation of Movie Recommendation System Using Machine Learning

S. Sridevi[1], Celeste Murnal[2]

[1, 2] Department of Computer Science and Engineering, New Horizon College of Engineering, Bangalore, India

ABSTRACT

As world is evolving, similarly people's desire, trend, interests are also changing. Same way even in the field of movies, people want to watch the movies according to their interest. Many web-based movie service providers have emerged and to increase their business and popularity, they want to keep their subscribers entertained. To improve their business, the service provider should recommend movies which users might like, so that they might watch another movie and be entertained. By doing this there is high possibility that customers will periodically renew the web-based movie service provider application. The objective of this project is to implement the machine learning based movie recommendation system which can recommend the movies to the users based on their interest and ratings. To achieve this, content-based filtering is used to recommend movie based on movie-movie similarity, collaborative based filtering is used to compute features based on user information and movie information. The proposed system uses the new ensemble learning algorithm, XGBoost algorithm to improve the performance. The results show that the proposed system is effective for movie recommendation and the system minimizes the root mean square error (RMSE).

Keywords: Machine Learning, Recommendation System, Content based filtering, Collaborative filtering, RMSE, XGBoost.

I. INTRODUCTION

A recommendation system is a kind of information filtering system which attempts to predict the likes of a user, and provides suggestion based on user preferences [1]. There are lots of variety of applications for recommendation systems. These have become increasingly popular and this is being now used in most online platforms that we use. Often, these systems can collect information about user choices and can use this information to improve their suggestions in the future.

Recommendation systems are generally designed to help users in finding and selecting items which may include books, movies, restaurants that are available in the web or in other digital information sources like Netflix, Hotstar, Amazon etc. Given the details about items and information of the user needs, a movie recommendation system provides suggestions about movies that helps the user better with small set of movie lists that cater to his needs.

Recommendation systems basically use two approaches [1-3]. The first one is based on content based filtering and the second one is based on collaborative filtering. Content based filtering

systems generally analyse the content of the information and finds the similarity among them. In a movie recommendation system based on content based filtering, to recommend a new movie to a particular user, the system will first analyse all the movies watched by that user over a period of time and analyse the content of those movies. It then recommends the movies which are having similar content to the user. On the other hand, the collaborative filtering is based on the ratings given by different users. It works based on the fact that if two users give same rating to some movies, then they are having similar preferences. So it recommends movie to the users based on the rating of the movie by a similar user.

The main purpose of the proposed system is to recommend movies to the users based on their own preferences and also based on the ratings provided by other users after watching a movie. This is achieved by using content-based filtering where list of similar movies is predicted and collaborative filtering is used for generating various features using user ratings and movie ratings. XGBoost algorithm is used to improve the accuracy of the system.

The paper is organized as follows. In section II, we discuss some of the works done by researchers in this area. In section III, we present the design methodology of the proposed system. In section IV, we present the experimental results. Finally, in section V we give the conclusions and the future work.

II. RELATED WORK

Kuzelewska, U. [1] proposed a recommendation system which has two phases offline phase and online phase. Offline phase uses clustering method to group users with similar interests. Representatives are selected for each cluster. In the online phase of the algorithm, to recommend movie to a user his ratings are compared only with the representatives of the clusters that are selected during the offline phase.

Geetha et al. [2] proposed a hybrid approach using collaborative filtering and content based filtering for the movie recommendation system. They used K-means clustering algorithm to improve the accuracy of the system. Pearson Correlation coefficient is used to calculate the similarity between the movies.

De Campos et al. [3] proposed a hybrid recommendation system. Bayesian network was used to find the distribution of probability in the ratings awarded by the users. They have used the static topology information which represent the information about user profile and dynamic topology information which represent the relationship between different movies.

Manoj Kumar et al. [4] presented a recommendation system MOVREC which recommends movies to the users based on collaborative filtering approach that makes use of K-means algorithm. They recommended movies to users based on the information provided by other users. They proposed cumulative weight based approach where weights are assigned to movies based on five parameters namely actor, director, rating, genre and year.

Movie Recommender System proposed by Nupur Kalra et al. [5] used collaborative filtering to provide recommendation which may be helpful to the users to select item of their interest among thousand other items. The intent of their paper is to study the working of collaborative filtering method using film-trust dataset. The results presented a list of recommendation to the users.

Recommendation system with collaborative filtering using big data was a survey done by Sonali et al. [6]. They presented a model that combines recommender system method such as collaborative filtering with

big data technique such as association rule mining. The main focus of their work was to provide a recommendation system that is scalable and robust in nature.

MovieMender[7] is a movie recommendation system for users presented by Rupali Hande et al. They proposed a hybrid system based on content based filtering and collaborative filtering. Similarity between users was measured using Pearson Correlation Coefficient. They didn't give the implementation details of the hybrid algorithm.

III. METHODOLOGIES

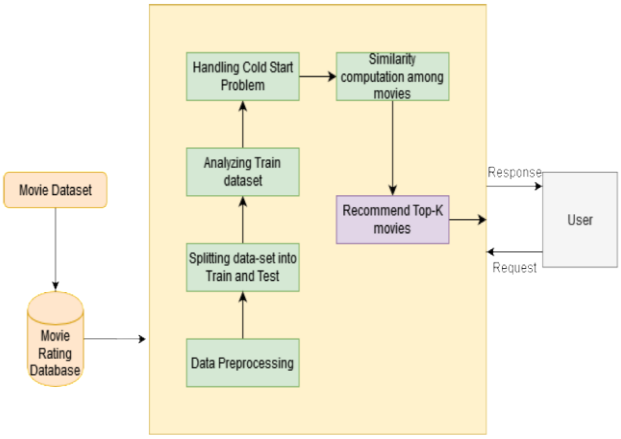

Figure 1. Flow Diagram

Fig. 1 shows the flow-diagram of our proposed movie recommendation method. First we have to get the movie data-set. In our dataset there are 5 files, out of which 4 files contain customer rating information and the fifth file contains movie name and id. Then we performed data preprocessing and cleaning based on the results of analysis. During analysis we came to know that in our dataset there is Cold start problem which is handled accordingly. A list of new features are extracted to check the similarity between the movies based on existing user rating and movie rating. When user watches any new movie, the system will recommend the user movies that are similar to the watched movie. In this section, the various steps involved in the proposed Movie Recommendation system are discussed.

A. Data Gathering

In the proposed work, we used the Netflix Movie Recommendation competition dataset downloaded from Kaggle.com [8]. There are totally five files in the dataset. Out of which 4 files contain CustomerID, Rating, Date, MovieID's of all the movies watched by the customers. The fifth file contains movie ID, name and year of the movies.

The dataset contains MovieID's ranging from 1 to 17770 sequentially, CustomerID's ranging from 1 to 2649429, with gaps. There are 480189 different customers. The dataset has 100480507 movie ratings given by the customers. Ratings are given to movies ranging from 1 to 5. The five files are merged into single csv (comma separated value) file and stored in the database. After getting complete data, we started analyzing the dataset and observed how ratings are distributed.

B. Data Preprocessing

In data preprocessing step NULL or Empty data values in CustomerID, Rating, Date, MovieIDs are identified. In the given dataset there is no empty data values. Duplicate values, if exist are also removed from the dataset. In our dataset there is no duplicate rows.

C. Splitting data into Train and Test sets

To build the machine learning model, first the dataset should be divided into training and test datasets. The machine learning model learns the pattern and predicts the new answers by using the training data set. Then based on the predicted result the efficiency of the algorithm is determined. The test dataset is used to validate how better the algorithm can predict new answers based on its learning done during training phase. The dataset is divided in the ratio of 80:20 for training and testing.

The table 1 explains the overall description of our dataset.

TABLE 1. DESCRIPTION OF DATASET

Total Customers	Total Movies	Date range	Ratings Range
480189	17770	1999-2005	1-5

After splitting dataset into 80% for training and 20% for testing, the description of training dataset and testing dataset are shown in tables 2 & 3.

TABLE 2. DESCRIPTION OF TRAIN DATASET

Total Customers	Total Movies	Total No. of Ratings
405041	17424	80384405

TABLE 3. DESCRIPTION OF TEST DATASET

Total Customers	Total Movies	Total No. of Ratings
349312	17757	20096102

D. Analyzing the training dataset

The distribution of movie ratings in the training dataset is shown in fig. 2. From the figure we can observe that most of the users who watched movies gave the ratings 4, 3 and 5, and few users gave ratings 1 and 2.

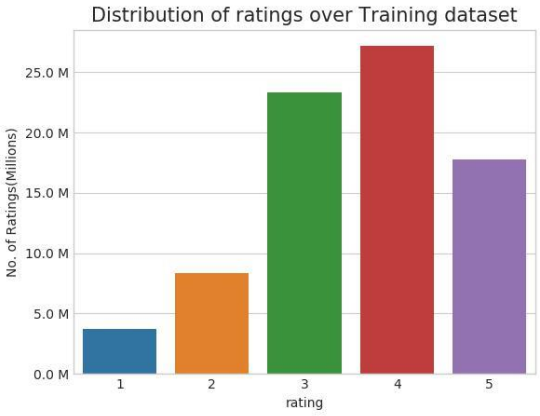

Figure 2. Distribution of rating in Train dataset

We also plotted the Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of the training dataset. From the fig. 3 we can observe that, PDF is the number of rating per user and peak symbol shows that most of the users who watched movies gave only few ratings. In CDF plot it is observed that 90% of the users given only few ratings.

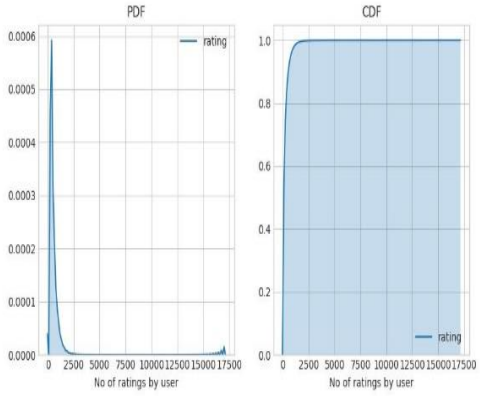

Figure 3. PDF & CDF of rating in Train dataset

E. Cold start problem with Users and Movies

The cold start problem occurs when any new user signs in or any new movie is added into the dataset. Because the new user would not have rated any movie and the new movie would not have been rated by any user. The cold start problem is handled by replacing the rating with 0 for the new user and new movie in the dataset.

F. Finding similar movies

Similarity between two movies is calculated using the Cosine Similarity. Cosine Similarity calculates dot product of one movie vector with another movie vector. Then result indicates how similar these movies are with each other. The definition of similarity between two vectors **A** and **B** is the ratio between their dot product and the product of their magnitudes. Mathematically, it is defined as given by equation (1).

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

The similarity between two movies will be equal to 1 if the two vectors are identical, and it will be 0 if the two are orthogonal. In other words, the similarity is a number bounded between 0 and 1 that tells us how much the two vectors are similar.

We computed the similarity matrix between movie-movie. The similarity matrix looks like fig. 4. All the numbers on the diagonal are 1 because every movie is identical to itself. The matrix is also symmetrical because the similarity between A and B is the same as the similarity between B and A.

$$\begin{pmatrix} 1 & 0.22 & 0.15 & \Lambda & 0.07 \\ 0.22 & 1 & 0.34 & \Lambda & 0.18 \\ 0.15 & 0.34 & 1 & \Lambda & 0.26 \\ M & M & M & O & M \\ 0.07 & 0.18 & 0.26 & \Lambda & 1 \end{pmatrix}$$

Figure 4: Movie-movie similarity matrix

G. Root Mean Square Error Performance Metric

RMSE is a frequently used metric to measure the error rate of the model. It is based on the difference between actual values and predicted values by a model. RMSE is computed using equation (2).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T}(a_i - p_i)^2}{T}} \quad (2)$$

In equation (2), T is the total ratings, p_i is the predicted value and a_i is the actual value.

H. Generating new features using ratings

From the training dataset, user rating information and movie rating information are used to generate 13 new features, so that with these new features XGBoost algorithm can predict ratings for users in the test data.

The 13 new features are listed below.

- GAvg: Global average rating of all the ratings
- Similar users rating of the movie: sur1, sur2, sur3, sur4, sur5 (top 5 similar users who rated that movie)
- Similar movies rated by the user: smr1, smr2, smr3, smr4, smr5 (top 5 similar movies rated by a user)
- UAvg: Average rating given by a user
- MAvg: Average rating given to a movie

I. Collaborative filtering using XGBoost algorithm

Content-based system is only capable of suggesting movies which are close to a certain movie. It is not capable of capturing tastes and providing recommendations. Collaborative filtering system recommends items based on similar tastes between the users. Here by using cosine similarity, the system will compute most similar users for the movie. For example, if user1 has rated movie1 and user2 has rated to movie1 and movie2, then system will recommend user1 with movie2. It is like recommending movie to users based on similar users who watched the movie.

J. XGBoost

XGBoost (Extreme Gradient Boosting) is well known to provide better solutions than other machine learning algorithms [9]. XGBoost is a kind of boosting algorithms and uses the gradient boosting framework at its core.

Boosting technique is based on the principle of ensemble. It combines several machine learning techniques into one predictive model. The algorithm is generally used to improve the accuracy. The XGBoost algorithm now finds its application in many machine learning projects due to the following advantages. It generally performs much faster than other machine learning algorithms like Linear Regression, KNN algorithm. It can be executed in multi-core computers thereby it can get the benefits of parallelism.

IV. RESULTS AND DISCUSSIONS

To evaluate the performance of the proposed system, we conducted experiments on movie datasets with Python3 Jupyter Notebook. The proposed system is running on Intel Core i7-5600U@ 2.60GHz CPU with 16.00 GB RAM and 64-bit Windows 10 Operating System.

The proposed system will recommend Top-10 similar movies for a user based on the movie he/she watched. Using XGBoost algorithm with the 13 features the RMSE (Root Mean Square Error) is also minimized.

For example, if a user gives input as movie id 1 that represents the movie 'Dinosaur Planet', the proposed recommender system gives top 10 similar movies list as shown in fig. 5. From the figure, we can observe the similarity in the movies based on the preference of the users.

movie_id	year_of_release	title
694	2000.0	When Dinosaurs Roamed America
5302	2003.0	Chased by Dinosaurs: Three Walking with Dinosa...
1084	2001.0	Walking with Prehistoric Beasts
13586	2001.0	Allosaurus: A Walking with Dinosaurs Special
1173	1999.0	Walking with Dinosaurs
4181	2003.0	Walking with Cavemen
8800	2003.0	Prehistoric America: A Journey Through the Ice...
10656	2003.0	Before We Ruled the Earth: Mastering the Beasts
15648	2002.0	National Geographic: Dinosaur Hunters: Secrets...
10257	2002.0	Prehistoric Planet: The Complete Dino Dynasty_

Figure 5. Output of Recommendation system

Table 4 given below shows the output from the proposed system for some other movies according to the IDs.

TABLE 4. SAMPLE OUTPUTS

Movie ID	Movie Title	Ratings from Customer	Total Similar Movies
5	The Rise and Fall of ECW	842	17323
777	Sherlock Holmes and th e Spider Woman	477	17294
2020	The Winds of War	1093	17320

```
Done. Time taken : 0:00:04.934970

Done

Evaluating the model with TRAIN data...
Evaluating Test data

TEST DATA
------------------------------
RMSE :  1.076373581778953

<IPython.core.display.Javascript object>
```

Figure 6. Error value obtained from the system

From the result of XGBoost algorithm as shown in fig. 6, we can observe that it has taken 4.9 seconds to train the model and RMSE on test data is 1.076.

V. CONCLUSION

We have implemented a recommendation system based on content-based filtering and collaborative filtering. Cold start problem in the dataset is addressed by adding 0 ratings. The proposed system used 13 features consisting of user information, movie information and predicted top-10 movies that are similar to user interests using content based and collaborative based filtering. The performance of the

system is improved by applying XGBoost algorithm. We obtained the RMSE value as 1.076.

Our future work will be implementing the recommendation system using deep learning algorithm and analyze the improvement in the accuracy of the system.

VI. REFERENCES

[1]. Kuzelewska, U., 2014. "Clustering algorithms in hybrid recommender system on movielens data" Studies in logic, grammar and rhetoric, vol. 37, no. 1, pp.125-139.

[2]. Geetha, G., M. Safa, C. Fancy, and D. Saranya. "A hybrid approach using collaborative filtering and content based filtering for recommender system" Journal of Physics: Conference Series, vol. 1000, no. 1, p. 012101, 2018.

[3]. De Campos, L.M., Fernández-Luna, J.M., Huete, J.F. and Rueda-Morales, M.A., 2010. "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks" International journal of approximate reasoning, vol. 51, no. 7, pp.785-799.

[4]. Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta; "A Movie Recommender System: MOVREC" International Journal of Computer Applications, vol. 124, no. 3, pp. 7-11, 2015.

[5]. Nupur Kalra, Deepak Yadav, Gourav Bathla; "Movie Recommender System using Collaborative Filtering" International Journal on Future Revolution in Computer Science & Communication Engineering, vol. 4, no.12, pp. 88-92, 2018.

[6]. Sonali R. Gandhi, Jaydeep Gheewala; "A survey on recommendation system with collaborative filtering using big data" in IEEE International Conference on Innovative Mechanisms for Industry Applications, pp. 457-460, 2017.

[7]. Rupali Hande, Ajinkya Gutti, Kevin Shah, Jeet Gandhi, Vrushal Kamtikar; "MOVIEMENDER A Movie Recommender System" International Journal of Engineering Sciences & Research Technology, pp. 469-473, 2016.

[8]. https://www.kaggle.com/netflix-inc/netflix-prize-data/data

[9]. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system" Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, pp. 785-794. 2016.

Cite this article as :

S. Sridevi, Celeste Murnal, "Implementation of Movie Recommendation System Using Machine Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6, Issue 3, pp.587-593, May-June-2020.
Available at
doi : https://doi.org/10.32628/CSEIT2063143
Journal URL : http://ijsrcseit.com/CSEIT2063143