Survey on Recommendation System for Movies using Content Based Filtering and Collaborative Filtering

Madhurima Chakraborty 19MCA0087

Pragya Yadav 19MCA0125

GeetaMunjal 19MCA0147

PROJECT GUIDE: Prof. Senthil Kumar N

School of Information Technology & Engineering (SITE)
Vellore Institute of Technology (VIT)
Vellore, Tamil Nadu, India

Abstract—**A recommendation engine is a tool that predicts suggestions of a list of items, to the users, based on the items that they may like or prefer. The items may be movies, books, songs etc. As the online market is expanding enormously; hundreds of movies are being made available every year and there are huge number of viewers for the same. Recommendation tools benefits both organizations and end users. This paper is a survey on content based recommendation system and collaborative filtering recommendation system, which tries to provide the best suggestions of movies a user would like. The algorithms taken in account for the same are 'content-based filtering' and 'item-to-item-collaborative filtering'. The content based recommender engine would make predictions based on user's preference for various genres of movies, whereas, collaborative filtering would recommend movies on basis of the preferences of other similar users.**

Keywords—Recommendation System, Content Based Filtering, Content Based Filtering, Count Vectorizer, Cosine Similarity

I. INTRODUCTION

In this age of Internet, where there is a booming expansion of data, which is available online, not all the data is useful to the user. A recommendation engine is a tool that is capable of predicting suggestions of a list of items like books, movies, songs etc.

A very crucial reason that we require a recommendation system is that, in this online environment, there is way too much choices and possibilities available to the user to choose from. This makes it difficult for us to choose the best suited option that would cater to our needs or interests. Recommendation engines help not only the customers but organizations as well. Organizations collect a large amount of real time data, and hence wish to provide customized services to the users. There can be recommender systems for various fields. In online video streaming sites these days, there are thousands of movies available and users find it hectic to select the movie that they might like the most. This is where a movie recommendation tool comes to use. It presents a list of movie suggestions to the users based on their preferences and like trends in the past. The movies are filtered according to the user's interest. Organizations like Amazon, Netflix, Youtube, Spotify, etc. use the recommendation system to provide their best services.

Recommendation engines adopt two ways; either collaborative filtering or content based filtering. In content based filtering, also termed as cognitive filtering, the recommendations are made on the basis of analysis of the content of an item; in this case, it is a movie. Then, predictions are made on the user's preferences, which match the content of the item in the list.

II. LITERATURE SURVEY

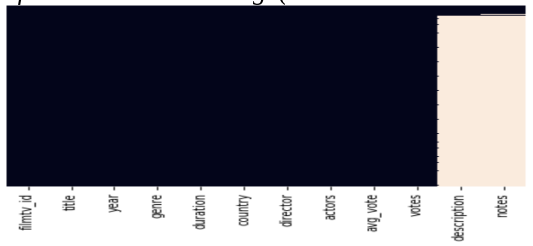
In this paper[1] the author has been recommending movies according to their genres. The used methodology is content-based filtering by genre correlation. The movie lens is the data set used for this system. It uses the R tool for analysis of data. It forms a matrix of genre and rating based on past data of a user and performs dot products of the matrix. In the resultant value of two matrices, Euclidean distance is calculated. Results having the minimum value are suggested to the user. It suggested movies on the user's past data patterns.

In paper[2] focuses on video recommendation system. It automatically evaluates the content of video and abstract some features uses a method Applied Media Theory. It procedures the grouping of traditional content-based method that explicit content features concomitant to it, so as to improve the accuracy. While several experiments they selected few movies of same genre and applied classification algo and decision table gives the best result. In second experiment they use cosine similarity on level video and suggested the full movie of that video. It concluded that low level stylistic provide better suggestion comparing to high level semantic features.
In paper [3] they recommended the scientific papers.it suggested the paper based on user rating or likes.LSA algorithm used to suggest the no of component. Rocchio Algorithm is used to classify the relevant and non-relevant document. As a result, it successfully recommends the relevancy and non-relevancy of the scientific paper on the basis of votes.
In paper [4] author shares about the importance of recommendation system and perform a comparative study on it. They combine two methods and perform in it. They use content with collaborative filtering and collaborative with demographics and so on. It is concluded that combination of two filter is more effective and provide appropriate results the best one among these filter is Collaborative and content filtering.
In this paper [5] it based on neuro-fuzzy approach. It decides that recommend or not to user. It simulates and learn the possibility of users vast on their pervious actives. The two approaches are applied in this paper which is neuro fuzzy and deep learning. In conclusion the neuro fuzzy is not as good as DNN but the hybrid methodology is best for this system.
In this paper [6] focuses on marketing and sales using the improvised recommendation system. It performs to gives the high customer satisfaction. The two methodology is used in this system is collaborative and content to get the latest trends in the market. It achieves the privacy, robustness, trust, speed, accuracy and relevancy in the system. So combining these two methodology they achieve the increase in sales and it can able to handle the vast database using python.
In this paper [7] the authors have offered recommendation system based on Doc2Vec model. This model converts the textual document in the form of numbers. The model analyses the document by it's semantics and converts into a vector. The Vector is then used in collaborative filtering. Their approach have provided them with better results and accuracy.
In paper [8] , a user based recommendation system is built. To build an intellectual system, various techniques like Pearson's correlation, matrix factorization etc. have been used, so that the UCF algorithm can be upgraded. For comparing the accuracies, techniques like SVD, cosine similarity has been used. From their experiments, they have concluded that, UCF based recommender system performs better than a normal recommender system.
Since most of the recommendation systems nowadays provide correct suggestions. This paper [9] aims on increasing the accuracy of the prediction. They have used collaborative filtering with the k nearest neighbor algorithm along with k clique. They have compared the personalized information of all users with each other. Their predictions have proven to be more accurate than the other system's prrediction.
In paper [10] a novel approach of collaborative filtering using neural networks is proposed, where they use several sklearn tools to build their recommendation system. They aim to discover the optimum criterion for perfect predictions with their recommendation system.

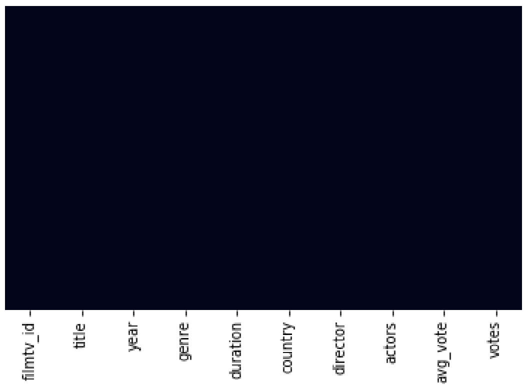
III. Datasets

The two recommendation system approaches are implemented on two different datasets taken from **Kaggle** website.
For content based filtering we've used **'filmtv_movies-ENG.csv'**, which has several features like movie-index, name, genre, description, notes, director, title, year, duration, country, actors, etc. to describe the movie. On analyzing the dataset, the features with null or missing values are dropped in order to attain higher accuracy for recommendations. Finally for this recommendation system approach, the selected features are genre, country, director and actor. Data pertaining to the selected features are hence extracted for further proceeding.
For collaborative filtering approach we've used dataset: **'Movielens 20M Dataset.csv'**. It contains the movie rating for about 2000 users on various movies. Out of those, the irrelevant movies are either dropped, or ratings are replaced by the mean ratings for better accuracy. It is used to provide predictions based on the ratings provided in ratings.csv file.

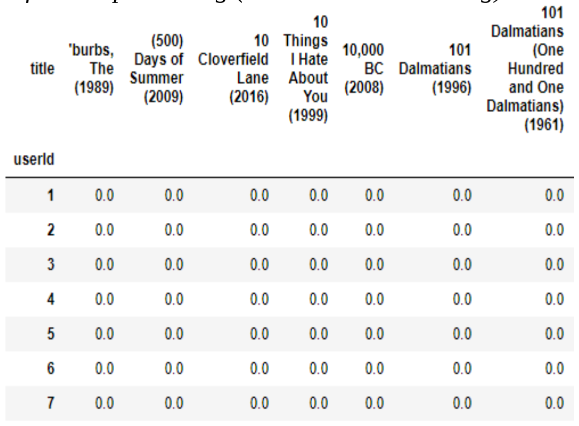
Before Dataset Cleaning (Content Based Filtering)



After Dataset Cleaning (Content Based Filtering)

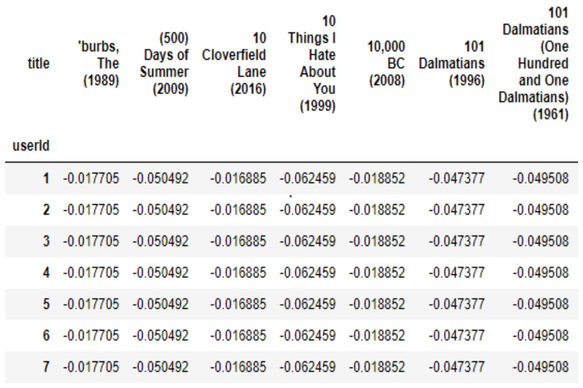


Before Preprocessing (Collaborative Filtering)



title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)
userId							
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

After Preprocessing (Collaborative-Filtering)



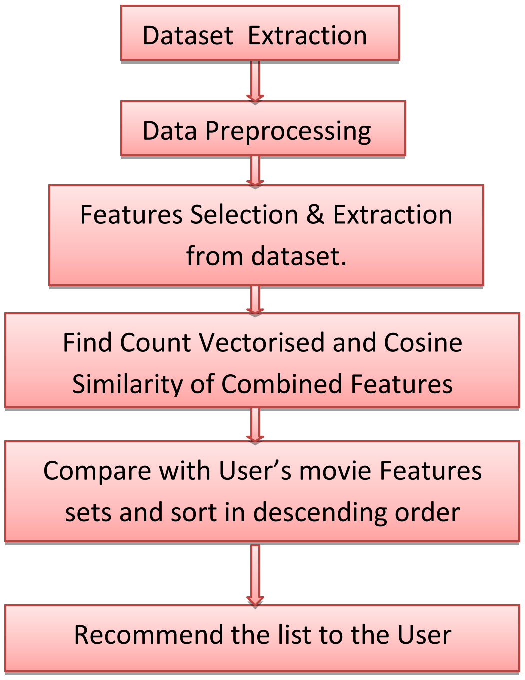
title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)
userId							
1	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
2	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
3	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
4	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
5	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
6	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508
7	-0.017705	-0.050492	-0.016885	-0.062459	-0.018852	-0.047377	-0.049508

IV. CONTENT-BASED FILTERING

Using this movie recommender model, we aim to suggest the movies best suited to the user's interest and preferences. It is also termed as cognitive filtering. In this approach the characteristics of the movie is matched with user's preference of films.

Architecture

Following is the architecture that we have followed.


Dataset Extraction
Data Preprocessing
Features Selection & Extraction from dataset.
Find Count Vectorised and Cosine Similarity of Combined Features
Compare with User's movie Features sets and sort in descending order
Recommend the list to the User

Data Extraction

We proceed by including the dataset and preprocessing it.

Data Preprocessing

For cleaning the data, we first visualize the features with null values. The irrelevant features from the dataset are discarded for better and accurate results.

Feature Selection & Extraction from dataset

We analyze the dataset free from null values and select the relevant features for our recommender system. It is best to use the required features to build our model, as unnecessary features might lead to complexities and deviations from the actual predictions.

Find Count Vectorised and Cosine Similarity of Combined Features.

Then the dataset with the required features is taken up to implement the approach. Further, we "count-vectorize" the data using the **CountVectorizer** class of **scikitlearn** module of Python language. Using that class we get the count of each word present in the document. It is a technique of numerically representing the dataset containing textual data. Further, on the count-vectorised data cosine similarity is calculated to find the similarity of each film with other films.

Compare with user's movie features

The results thus obtained are then compared with the movie feature set of user's taste. The obtained set of results is then sorted in descending order providing the most suitable movies at the top of the set.

Recommend the list to user

The list of movies is hence recommended to the user. The user now has movies similar to his taste assorted from the vast collection of movies.

i. Count Vectorization

Count Vectorization is a metric that refers to counting the number of times a word has occurred in the text. This is one of the elementary ways of representing textual data in the form of numbers. In this method we define vectors whose dimensions are equal to the words in our document. If the text in our document matches the vocabulary, then we assign a count one to the dimension of that vocabulary. It provides the accurate count of the words in our document. In this system we have used CountVectorizer class of scikitlearn module present in Python language

ii. Cosine Similarity

It is a measure that is used to find the likeness among two components regardless of their sizes. In this method the angle amid two vectors, which have been plotted in a multi-dimensional-space is measured. The similarity between two vectors is assessed in this method. Hence we can conclude that the similarity rate would be higher if the vectors have small cosine angle. So, for movies which will have cosine similarity closer to 1 would be more similar and appear higher on the list of recommended movies.

Mathematically, cosine similarity can be expressed as

$$similarity(A,B) = \frac{\sum_{i=1}^{n} Ai * Bi}{\sqrt{\sum_{i=1}^{n} Ai^2} * \sqrt{\sum_{i=1}^{n} Bi^2}}$$

While implementing in Python, for the same, we use cosine_similarity method of scikit_learn module.

V. COLLABORATIVE FILTERING

Collaborative Filtering is also termed as **social filtering**. In this approach a user is suggested recommendations based on similar preferences by other users. In this methodology, a larger set of data is consulted and a smaller subset similar to user's interest is taken up for generating ranked results.

i. Item-to-Item based Collaborative Filtering

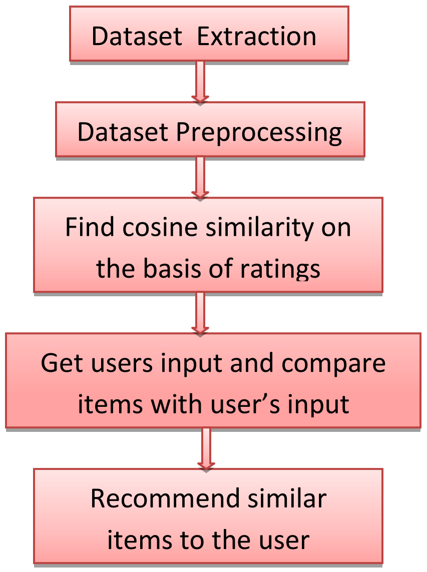
In this recommendation system we have used item-to-item collaborative. It is based on likeness between items that has already been given ratings by other users.

	M1	M2	M3	M4
U1	5	1	4	4
U2	2	4	1	1
U3	4	1	?	?

Here, we would recommend movie M3 to user U3 as Ratings provided for M1 is more similar to M3 than M4. Hence, according to the similarity of items, M3 would be suggested.

Architecture

The architecture adopted for the above approach is


Dataset Extraction
Dataset Preprocessing
Find cosine similarity on the basis of ratings
Get users input and compare items with user's input
Recommend similar items to the user

Dataset Extraction

In the above approach, we have two datasets, one for movies and one for ratings. The movies file consists of movie names along with their genres. Ratings file consists of the ratings of those movies. We merge the files to get a singular files consisting of movie names and their ratings only.

Dataset Preprocessing

We preprocess the merged dataset. The movies rated by less than 10 people are dropped, as they will not contribute to the predictions. Movies which have been rated by more than 2 users are then taken up and the null values are replaced with zero. We further preprocess the data by normalizing our dataset. Since, movies not rated by user do not mean that they will not be rated more than 0 by other users. We use the **mean formula** to **standardize** the dataset. We use the long tail plot to visualize the movies with highest ratings. Lastly, retain the relevant attributes and features, so that the approach can work well.

Find cosine similarity based on ratings

Since the approach used here is item to item collaborative filtering, we find similarity index of the movies based on ratings. This we get a matrix of values of likeness of each movie with other.

Get user input and compare items with that film

After that we compare the user's input with the similarity indices. The items are then arranged in Descending order.

Recommend similar items to user

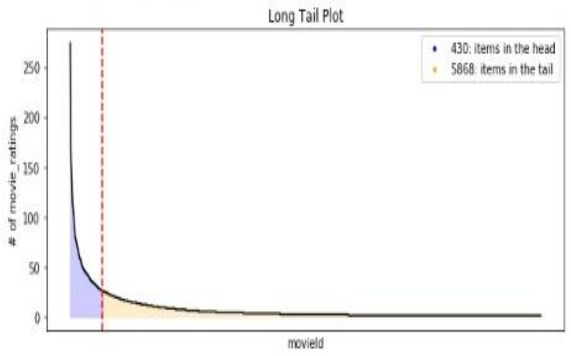
Movies with closest similarity are suggested.

VI. EVALUATION

The item-to-item approach of collaborative filtering is being evaluated to check the accuracy of the recommendation system. For evaluation purposed we have taken up the **RMSE and MSE** methods.
We have visualized the popular movies on the basis of film ratings using the **long tail graph**.
Further in our recommendation system, we use python's surprise package and the **SVD** algorithm, to perform valuation.

i. Long Tail Plot

This plot helps in visualizing the highest rated movies in our dataset. The items in the longer tail are the items which are less popular while items at the smaller part of the tail are the more popular items. The graph below describes the plotting for our dataset. We have **430 movies popular** according to ratings, which are in the blue section. While the movies less competitive are in the yellow section.



ii. Surprise

This package is present in scikit of Python language. It will help us in the analysis of our recommendation system.

iii. Singular Value Decomposition (SVD)

This is a technique used to shrink the dataset, that is used in machine learning for reducing features. In context of CF , a matrix of dimensions user x items is used. Matrix is reduces in accordance with the following relation:

$$A = USV^T$$

Uand V are singular orthogonal matrices representing user and features and items and features respectively.
The rating is calculated as follows:

$$Min(x, y, bi, bu) \sum_{(u,i)\in K} (r - y.x^T - \mu - bi - bu)^2 + \lambda(||x||^2 + ||y||^2 + bi^2 + bu^2)$$

bi and bu are biases for user and items.

iv. RMSE

Root mean squared error or RMSE is a standard approach to evaluate our model. It's resemblance to the Euclidean formula of distance proves that this formula is a kind of measure of distance between our actual and predicted data. It defines the deviation of our predictions of the model from the original data. Mathematically it can be expressed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(predicted - observed)^2}{n}}$$

For our DataSet, the RMSE and MSE scores are as follows:

RMSE	0.407421597085228
MSE	0.1659923577714779

An RMSE between 0.2 to 0.5 is considered good result. We have set a constraint that **rating>=4** would be taken up for evaluation, which is giving us an RMSE ≅ 0.4

VII. CONCLUSION

Recommendation systems ease up the task of choosing what's likeable from a plethora of options. There are various approaches to a recommendation system, out of which we have surveyed two here. From the experimental setups , we understand that in a CBF recommender system, the individuals previous choices are taken in account. It is totally relying upon the features of the movie. Our recommendation system have successfully predicted similar movies based on user's choice of movie. On the other hand, CF is based completely on peer rating for the same movie. This method needs a vast data on ratings for a particular movie by different users. Our recommendation system based on SVD have proven to deliver accurate predictions.

VIII. REFERENCES

1. "Reddy, Srs & Nalluri, Sravani & Kunisetti, Subramanyam & Ashok, S & Bachu, Venkatesh. (2018). Content-Based Movie Recommendation System Using Genre Correlation".
2. "Yashar Deldjoo ,Mehdi Elahi ,Paolo Cremonesi ,Franca Garzotto ,Pietro Piazzolla ,Massimo Quadrana. Content-based Video Recommendation System based on Stylistic Visual Features(2019)"
3. "Titipat Achakulvisut, Daniel E. Acuna, Tulakan Ruangrong, Konrad Kording. Science Concierge: A Fast Content-Based Recommendation System for Scientific Publications(2016)"
4. "Nymphia Pereira, SatishKumar Varma. Survey on Content Based Recommendation System(2019)"

5. “Tomasz Rutkowski, Jakub Romanowski, Piotr Woldan, Paweł Staszewski,Radosław Nielek, Leszek Rutkowsk. A Content-Based Recommendation System Using Neuro-Fuzzy Approach(2016)”

6. “SHRUTI SUNIL KUMBHAR, OLIVIA PARASAR SHARMAROY, OLIVIA PARASAR SHARMAROY, SHRUTI DILIP SOHANI. Improvised Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining(2016)”

7. “Gaojun Liu , Xingyu Wu. Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation, 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)”

8. “Chen, Vito & Tang, Tiffany. (2019). Incorporating Singular Value Decomposition in User-based Collaborative Filtering Technique for a Movie Recommendation System: A Comparative Study. PRAI '19: Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence. 12-15. 10.1145/3357777.3357782.”

9. “Vilakone, Phonexay, Khamphaphone Xinchang, and Doo-Soon Park. Personalized Movie Recommendation System Combining Data Mining with the k-Clique Method. Journal of Information Processing Systems 15, no. 5 (2019).”

10. “Lin, Chu-Hsing & Chi, Hsuan. (2020). A Novel Movie Recommendation System Based on Collaborative Filtering and Neural Networks. 10.1007/978-3-030-15032-7_75.”