Netflix Recommendation System

Shreesh Ravindra Devi
Computer Science and Engineering PES University
Bangalore, India
shreeshrdevi2@gmail.com

Shreesha Seerya Krishna
Computer Science and Engineering PES University
Bangalore, India
skshreesha@gmail.com

Aditya Vijay Kavatage
Computer Science and Engineering PES University
Bangalore, India
write2adityavk@gmail.com

Aryan Kumar
Computer Science and Engineering PES University
Bangalore, India
aryan.kumar.052001@gmail.com

ABSTRACT- With the advent and explosive growth of the Web over the past decade, recommender systems have become at the heart of the business strategies of e-commerce and Internet-based companies such as Google, YouTube, Facebook, Netflix, LinkedIn, Amazon, etc. The purpose of the project is to analyze the various approaches inherited to solve the problem of providing a movie recommendation on Netflix and formulate a model that would provide the best recommendation for the user. Content based filtering and dimensionality reduction help to cluster similar movies to users by generating accurate and highly efficient recommendations. The experimental results show that this new method significantly improves the performance of the recommendation systems.

KEYWORDS-Vectorization(CountVector,TfidVector),Content based Filtering ,IMDB Classification

INTRODUCTION

Nowadays, recommender systems have become one of the most promising techniques for online companies specializing in Internet-related services and products. Google, YouTube, Facebook, Netflix, LinkedIn, and Amazon are typical examples, in which these recommender systems play a vital role into the core of their business model. These systems predict users' preferences based on their behaviors and help improving the satisfaction of users towards the promoted items. Recommender systems are now considered an essential part of many e-business corporations and various domains ranging from movies, books, and news to research articles. These systems use sentiments that are acquired about products and services from communities of users and promote these products for other users with similar interests. For instance, recommender systems can explore the existing connections between users and their friends and automatically recommend new friends for an active user in a social network context

Recommender systems are divided into three categories: content-based filtering, collaborative filtering or hybrid methods. Content-based filtering methods focus on both the profile of the user's preferences and the item description in order to recommend items that are most similar to the items that are highly rated in the past. Collaborative filtering methods take into consideration a variety of criteria such as users' preferences, activities, and behaviors and recommend items based on the similarities to other users. Hybrid methods combine content-based filtering and collaborative filtering and build on their advantages in order to recommend more items that are suitable.

This paper introduces a new content-based filtering recommendation algorithm based on the dimensionality reduction and clustering techniques. The aim is to improve the performance of recommender systems and to overcome their problems of sparsity and cold-start as well as their scalability issues. The experimental results show that our method significantly improves the performance of the recommendation systems.

LITERATURE REVIEW

1) An Approach for Netflix Recommendation System using Singular Value Decomposition

The results show an overall improvement in performance for all algorithms and mathematical improvement models as the data size grow[1]. Moreover, the individual increase in performance of each algorithm implies a great difference in their effectiveness, especially between the matrix factorization algorithms SVD and ALS.SVD produced the best ratings among the algorithms with an RMSE of 0.8190 under the 1M data set and a performance increase of 0.09936% from the 100K to 1M data set. This was expected as the winners of the prize used a modified version of the SVD algorithm in their solution. The RMSE indicate how well these algorithms predict ratings for non-rated items. Based on these results, it is suggested to utilize SVD out of the four algorithms for a recommender system in production. This paper focused on comparing four different collaborative filtering algorithms, in which the aim was to find out which one that produced the best prediction rate. The four algorithms were KNN, SVD, ALS and Slope One. This paper also used two mathematical models, Arithmetic mean and weighted arithmetic mean, in order to determine if the mathematical models could produce better prediction ratings than the four algorithms. Out of the four algorithms the SVD had the best prediction rate whereas ALS the worst. However, ALS had the highest performance improvement when the data increased from100K ratings to 1M ratings. AM had a slightly better prediction rate than SVD, and WAM had the overall worst prediction rate.

2) A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques

This paper proposes a new method for recommender systems that benefits from the potentialities provided by the k-means clustering algorithm and SVD technique. Firstly, the k-means clustering algorithm was adopted to cluster users in the same partition according to their preferences, and then the SVD was used in each cluster not only as a dimensionality reduction technique but also as a powerful mechanism[2], which could efficiently help in finding the most similar users. To evaluate the performance of the proposed method, experimentations were conducted on two real-world datasets for movies recommendation called MovieLens 1M and MovieLens 10M, which contain about 1 million and 10 million ratings made by anonymous users, respectively. In addition, RMSE metric was adopted to evaluate the predictive accuracy of the proposed method in comparison with well-known k-nearest neighbour based recommendation and k-means-based recommendation methods. The experimental results showed that our method improved significantly the performance of the recommendations and remained the lowest values in the RMSE curve in the whole neighbours range.

3)Movie Recommender System Based on Percentage of View

In this paper, an Implicit Opinion Measure (IOM) is proposed to improve the performance of movie recommender systems on implicit feedback. The dataset used is created by Namava, a media service provider[3].
In this paper the authors propose a percentage of view approach to find relevant movies for customers. To predict this metric, collaborative filtering, content-based filtering and a new method which is called residual method was used. To this end, the correlation between like probability and percentage of view of movies is shown and it was illustrated that there is a positive correlation between them. For the evaluation of the recommender system, the average of precision of the top 20 recommendations for all of the users was reported as performance of the recommender system. Prediction results for Content-based Filtering, Collaborative Filtering – Model-based and Collaborative Filtering – Memory-based algorithms are reported.
To improve prediction accuracy, the sign and the value of error was estimated independently at each level. For the comparison, a random recommender system is considered as the baseline and the accuracy of the recommender system will be compared with it.
The recommender system developed using these different prediction methods, works 5 times better than the average random recommendation.

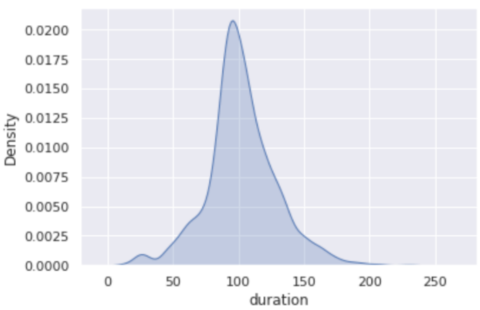
4)Predictive self-learning content recommendation system for multimedia contents

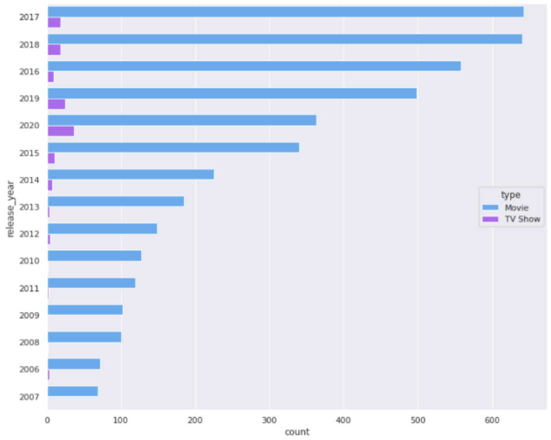
The objective of this paper is to create an algorithm that predicts what users search next by using prior collected information and using machine learning to analyze the user behaviors for the future activity. Recommendation Systems face several challenges such as noisy data or bad data because the users don't want to cooperate. Therefore recommendation systems are general and cannot be changed according to user characteristics after visiting the suggested data. The algorithm predicts what a user searches next by using prior collected information and using machine learning to analyze the user behaviors for the future activity. The Predictive Self-Learnin Recommendation System that uses a Collaborative Filtering Algorithm as well as seven criteria (Popularity, Similarity, Currency, Feedback, Importance, Safety, and Interest) in addition to users' profiles to make predictive recommendations to users. The system is different from traditional recommendation systems because it allows for more diverse suggestions without decreasing the performance of the system in terms of response time and CPU utilization.

DATASET and INITIAL INSIGHTS

For the purpose of providing a recommendation to the user we have taken the Netflix movies and TV-shows dataset. This dataset consisted of 12 columns and 8807 rows. Since we have worked on another aspect or another factor of recommendation of movies considering the IMDb rating and rotten tomatoes we have chosen to merge these two columns from another dataset and joined them over a common attribute or column of Movie Titles.

Initial insights have shown most people prefer to watch shows or movies on Netflix rather than other available platforms and the insights have also shown most movies average around 92 minutes and the United States produces the most number of movies and tv shows. Also the the number of movies and tv shows shown an increase in trend in the recent years.





PROBLEM STATEMENT AND PROPOSED SOLUTION

The aim of this project was to develop a profound recommender model or system for providing recommendations to the user based on his likes of a movie or given a movie provide recommendations for it. Our approach differ from other general trends as we use both content based filtering and tfid vectorization. Also used a soup(bag of words/attributes) considering the CountVetorizer to provide the results .

```
get_recommendations_tf('peaky blinders')
```

```
Out[4]: 7683                    our godfather
        2646                    my stupid boss
        3133                               don
        8293                          the fear
        7140       jonathan strange & mr norrell
        7785                power rangers zeo
        8467                        the prison
        8539                        the tudors
        1510                    the con is on
        8391     the legend of michael mishra
        Name: Title, dtype: object
```

1) Using content based filtering and tfid vectorization

2)Using cosine similarity and CountVector

Our approach also considers the use of cosine similarity which helps overcome the fundamental flaw in the ‘count-the-common-words’ or Euclidian distance approach .mathematically it measures cosine of angle between two vectors projected in multi-dimensional space.

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

```
In [8]: get_recommendations_multiple('peaky blinders', cosine_sim2)
```

```
Out[8]: 4809                 kiss me first
        5032    the frankenstein chronicles
        6922                  happy valley
        3034                    giri / haji
        2184                       get even
        5716                       paranoid
        7333                     london spy
        3789                 killer ratings
        5278                        apaches
        1991                   criminal: uk
        Name: Title, dtype: object
```

2) Using cosine similarity and CountVector

EXPERIMENT RESULTS AND CONCLUSION

Our approach has also considered the given IMDb ratings and worked on generating IMDb ratings for the given movies and have been successful with an accuracy of 0.96 or 96%.

```
In [25]: sum_a = sum_b = 0
         for i in range(100):
             ratings, IMDb = get_similar(i)
             sum_a += max(ratings)
             sum_b += IMDb
         print("Accuracy: ",sum_a/sum_b)

         Accuracy:  0.9612554112554114
```

ACKNOWLEDGEMENT

We would like to express our profound gratitude to Prof Bharathi, for encouraging and providing us with this opportunity to get hands-on experience in the field, and guiding us along the way. We would also like to thank the Computer Science and Engineering department at PES University, for always inspiring us to conduct frequent research and inculcating a problem-solving discipline in us.

REFERENCES

[1] Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub , Yaser Jararweh." An Approach for Netflix Recommendation System using Singular Value Decomposition", 2018.

[2] Ankur A. Ranjan1*, Amod Rai2, Saiful Haque3, Bhanu P. Lohani4 and Pradeep K. Kushwaha "A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques", 2019

[3] Ramin Ebrahim Nakhli, Hadi Moradi, Mohammad Amin Sadeghi "Movie Recommender System Based on Percentage of View", 2019

[4] Ibrahim Hussein Mwinyi, Husnu S. Narman, Kuo-Chi Fang, and Wook-Sung Yoo "Predictive self-learning content recommendation system for multimedia contents",2018

[5] https://www.sciencedirect.com/topics/computer-science/cosine-similarity