CineMatch: A Content Based Recommendation System For Movies and T.V Shows

Avantika Prativadhi

April 10th, 2023

Abstract

Recommendation systems are very useful in the world today with every platform using them to enhance user experience. The aim of this research paper is to develop a content-based recommendation system for movies and TV shows called CineMatch. The system utilizes features such as actors, directors, plot, number of seasons, age certification, production country, and genre, along with the popularity and score data from TMDB and IMDb to accurately recommend a movie to a user and also predict the rating of a movie or a TV show. The motivation behind this project is to improve user experience and increase engagement on entertainment platforms. The paper discusses the importance of recommendation systems and the hope to learn more about applying various analysis methods and algorithms on multiple datasets. The project involves using different datasets from Netflix, Hulu, and Amazon Prime, which were cleaned and combined into one major data frame. Statistical modeling and machine learning techniques such as feature engineering, similarity measures, clustering, and matrix factorization were used for the analysis. The software used for this project is Python and Jupyter Notebook. Overall, this project provides a learning experience and an opportunity to gain insight into recommendation systems while applying various techniques and algorithms.

1 Introduction

Recommendation systems have become very evident in our daily lives. From online shopping to social media and job searching to entertainment platforms, these systems have become an integral part of our online experience. In particular, streaming services such as Netflix, Hulu, and Amazon Prime heavily rely on recommendation systems to personalize content for their users and improve overall user experience.

Despite their prevalence, recommendation systems are still a complex and challenging field of research. One of the main goals of these systems is to accurately predict a user's preference for a given item based on their past interactions and behaviors. To achieve this, recommendation systems typically use content-based filtering, collaborative filtering, and/or hybrid filtering techniques. Collaborative filtering (CF) approach is where, "a model learns from a user's past behavior as well as similar decisions made by other users, and predict items (or ratings for items) that users may be interested in."[1]. On the other hand, Content based filtering," takes some information which can descript the characteristic of item as its training data input." [2]. The hybrid approach is a combination of the two methods and is most commonly used. Although the systems we have in place today are very accurate to a certain extent, they can still more research and additional improvements to make user experience even more better.

Out of the three major ways of filtering, this paper proposes a content-based recommendation system for movies and TV shows called CineMatch. Rather than using personal and individual user data, this system utilizes a wide range of features of the movie/show itself such as actors, directors, plot, number of seasons, age certification, production country, and genre, along with popularity and score data from TMDB and IMDb.The research question is whether this approach, combined with appropriate data processing and analysis techniques, can not only recommend a perfect show/movie, but also accurately predict a user's rating of a movie or TV show.

The motivation for the research stems from my personal experiences with recommendation systems. While these systems have greatly improved the ability to discover new content, they are not always perfect. We have all encountered situations where the recommendations we receive are far away from what we are interested in, leaving us to waste a lot of time starting a show/movie that we definitely are not captivated by. Additionally,I'm also curious regarding the work put into establish something like this and would like to get a better understanding on the inner workings of recommendation systems and explore ways to improve their accuracy.

With this research, I hope to contribute to the broader field of recommendation systems and machine learning by exploring the effectiveness of different techniques in a real-world setting. The rest of this paper is organized as follows. In section 2, the data description and resources will be provided. Part 3 contains the analysis and the methods along with all the insight obtained through the models. Finally, the conclusion will go over the potential future works.

2 Data Description

The datasets used in this project were obtained from Kaggle and were scraped from a streaming guide called "JustWatch". The collaborator for this project is Victor Soeiro. The project includes three datasets: Netflix, Hulu, and Amazon Prime. The Hulu dataset contains two CSV files, one for credits and one for titles. The titles data frame contains information about the movie like the title, description, release year, runtime, genre, etc. The credits file contains information about the name of the actor/director and their role in the show/movie. Both these CSV files were combined into one with a new column called "service" added, which specifies which of the three platforms (Prime, Hulu, Netflix). The type of variables in the Hulu, Prime, and Netflix datasets can be seen in figure 1. These three datasets were combined into one major data frame called "df", which serves as the metadata for this project.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 157366 entries, 0 to 1216
Data columns (total 20 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   id                    157366 non-null  object
 1   title                 157366 non-null  object
 2   type                  157366 non-null  object
 3   description           157270 non-null  object
 4   release_year          157366 non-null  int64
 5   age_certification     81836 non-null   object
 6   runtime               157366 non-null  int64
 7   genres                157366 non-null  object
 8   production_countries  157366 non-null  object
 9   seasons               18012 non-null   float64
 10  imdb_id               150945 non-null  object
 11  imdb_score            150126 non-null  float64
 12  imdb_votes            150098 non-null  float64
 13  tmdb_popularity       157351 non-null  float64
 14  tmdb_score            146844 non-null  float64
 15  service               157366 non-null  object
 16  person_id             157366 non-null  int64
 17  name                  157366 non-null  object
 18  character             138802 non-null  object
 19  role                  157366 non-null  object
dtypes: float64(5), int64(3), object(12)
memory usage: 25.2+ MB
```

Figure 1: The combined dataset and the attributes.

Before starting the process of creating the model, the data was cleaned and explored to understand the initial shape of the data and to see if any patterns could be discovered as this would help in the variable selection process. To treat the missing values, they were either completely removed depending on their importance in the research or were replaced using a process call mean imputation where the null values were replaced by the mean of the column. The figure below shows the visualizations obtained through the data exploration process.

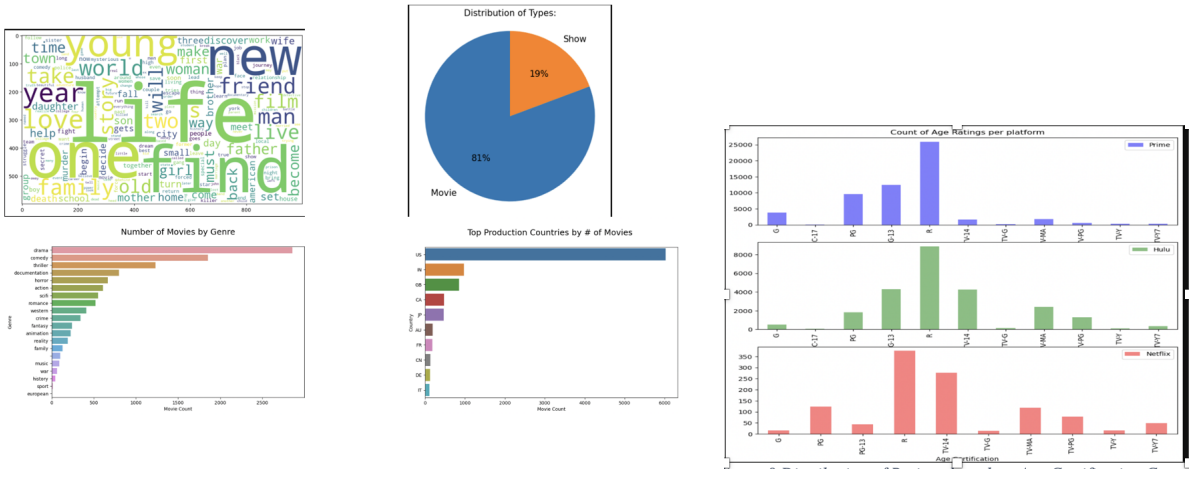

Figure 2: Visualizations of the movies/tv shows data.

3 Analysis

The coding for this project was entirely done on Jupyter Notebook using the Python kernel.Various python packages like pandas, numpy, matplotlib, seaborn, and sklearn were used to clean, visualize, analyze, and model the data. The main aspect of the project was cosine similarity as it was utilized as a similarity metric to find similar movies based on various features such as title, type, genre, description, actor, director, IMDB score, TMDB popularity, and TMDB score. In the International Journal of Engineering and Advances Technology, the authors of the article state," Cosine similarity among two objects measures the angle of cosine between the two objects. It compares two documents on a normalized scale. It can be done by finding the dot product between the two identities." [3]

To vectorize the data, the Scikit-learn package's CountVectorizer() function was used to convert the textual data into a matrix format. Then, cosine similarity was computed using the cosine similarity formula to produce a similarity matrix. To evaluate the performance of the model, the dataset was split into training and testing data with a 80:20 ratio where 20 percent of the data was used for testing and 80 percent for training. Precision, recall, and F1 score were calculated to evaluate the performance of the model. The results showed that the model achieved a precision of 0.8933, recall of 0.8963, and F1 score of 0.8867.

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

Figure 3: Formula for Cosine Similarity

Subsequently, cluster analysis was carried out using PCA and k-means clustering. PCA was utilized to transform the data into a lower-dimensional space while retaining as much variance as possible. K-means clustering was then applied to cluster the transformed data into 900 groups. The silhouette score, a measure of how similar an object is to its own cluster compared to other clusters, was 0.7923.

Lastly, based on the clusters generated, a function was developed to recommend a movie. The function accepts a movie as input and recommends a similar movie from the same cluster. Figure 4 gives a look at what the output is.

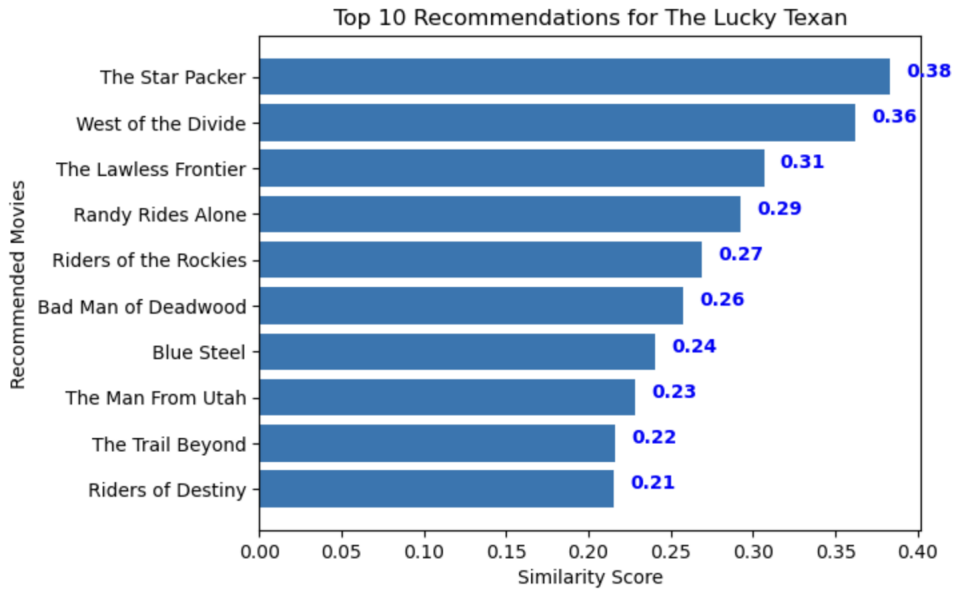

Figure 4: The recommendation system presents a list of 10 similar movies along with their similarity score

4 Conclusion

In conclusion, this study used cosine similarity and cluster analysis to generate movie recommendations based on various features. The precision, recall, and F1 score metrics showed that the model performed well in finding similar movies. However, it is important to note that the recommendations are limited to the movies in the given dataset.

Future work for this study involves connecting the system to an open source database to expand the recommendations to a larger collection of movies. Additionally, I plan to add a prediction element to the system to generate recommendations based on the user's viewing history.

Despite the limitations, this study provides a strong foundation for the development of a robust movie recommendation system. With further improvements and expansion, this system could provide valuable recommendations to a broader audience of movie enthusiasts.

References

[1] Wang, D., Liang, Y., Xu, D., Feng, X., andamp; Guan, R. (2018, May 17). A content-based recommender system for Computer Science Publications. Science Digest. Retrieved April 10, 2023, from `https://www.sciencedirect.com/science/article/pii/S0950705118302107`

[2] H. -W. Chen, Y. -L. Wu, M. -K. Hor and C. -Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017, pp. 504-509, doi: 10.1109/ICMLC.2017.8108968.

[3] Singh, R. H., Maurya, S., Tripathi, T., Narula, T., and Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. International Journal of Engineering and Advanced Technology, 9(5), 556-559

[4] Choi, S.-M., Han, Y.-S.: A content recommendation system based on category correlations. In: The Fifth International Multi-Conference on Computing in the Global Information Technology, pp. 1257–1260 (2010)

[5] D. Das, H. T. Chidananda and L. Sahoo, "Personalized Movie Recommendation System Using Twitter Data" in Progress in Computing Analytics and Networking, Singapore:Springer, vol. 710, 2018.

[6] o N. Mishra, S. Chaturvedi, V. Mishra, R. Srivastava and P. Bargah, "Solving Sparsity Problem in Rating-Based Movie Recommendation System" in Computational Intelligence in Data Mining, Singapore:Springer, vol. 556, 2017.