Master's Programme in Data Science

Movie Recommendation

Andrew Zaki, Borna, Sanish

April 17, 2024

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

Contents

1. Introduction

Movies have become a part of our daily lives, including various aspects such as movie nights at home, thrilling experiences at cinemas, and streaming during commutes. Choosing the best suitable movie is overwhelming in the vast sea of movies available on numerous platforms. In this context, our project emerges as a solution to this challenge by introducing a website equipped with a recommendation system. This system is designed to suggest five movies that closely align with the user's input movie, thereby simplifying the process of movie selection.

Our project is designed to process 2 data-sets: Netflix and IMDB data-set, which include 7815 and 1000 rows of unique movies. Afterward, we performed EDA techniques on the data-sets separately to understand the data features. Moreover, we combined these data into one data-set, which enables us to create a clustering model that will be then used to recommend the best five movies.

Deploying the machine learning model is crucial for the users to have full access to our clustering model. We designed a user-friendly website that allows the user to input one movie and return the closest five movies.

2. Data Sets

2.1 Netflix Data Set

Netflix popular movies dataset contains several columns that provide detailed information about movies available on Netflix. The title column contains the name of the movie. The stars column lists the actors and actresses who starred in the movie. The description column provides a brief description or synopsis of the movie's plot. The duration column indicates the length of the movie in minutes. The rating column shows the movie's rating out of 10. The votes column shows the number of people who have rated the movie. The year column indicates the year the movie was released. The genre column lists the categories or genres that the movie falls under, such as action, comedy, drama, etc. Finally, the certificate column indicates the movie's age rating or certification [1].

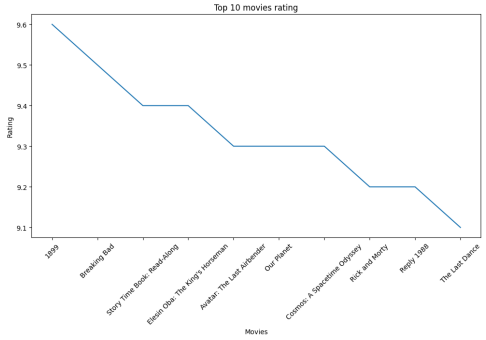
2.2 IMDB Data Set

The IMDB Dataset is a comprehensive collection of data that includes a variety of information such as the link of the poster that IMDB uses, the name of the movie or TV show, the year at which that movie or TV show was released, and the certificate earned by that movie or TV show. It also includes the total runtime of the movie or TV show, its genre, its rating on the IMDB site, and a mini story or summary. Furthermore, it provides the score earned by the movie or TV show, the name of the Director, and names of the Stars. Lastly, it includes the total number of votes that a movie or TV show has received and the money earned by that movie or TV show. This dataset provides a rich source of information for anyone interested in analyzing trends and patterns in popular movies and television shows [2].

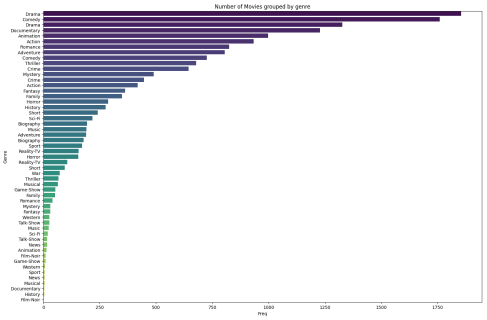
3. Netflix EDA and preprocessing

3.1 EDA

In order to be able to ensure the accuracy of any of our analysis, we started by removing duplicate movies from the Netflix data set. Having unique entries is important since duplicate rows can skew the results and result in incorrect conclusions. Moreover, we tried to create some visualizations to understand the data we have.



(a) Top 10 movies rating



(b) Number of Movies grouped by genre

Figure 3.1: Visualizations for Netflix Data

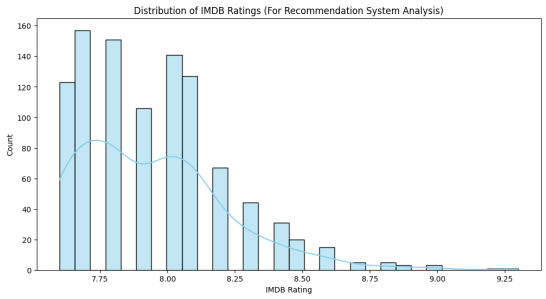
3.2 preprocessing

In the preprocessing of the Netflix data set, several transformations were applied to make the data more suitable for analysis. Firstly, the year column was extracted from the title and relabeled as Released_Year. This was done by extracting the first occurrence of a year in the title using regular expressions and converting it to a float data type. For example, (2018 -) would be 2018.0. Next, the duration column, which originally contained values like 30 min, was converted to a new column duration_in_min. This was achieved by extracting the numeric part of the duration and converting it to a float. For instance, '30 mins' would be 30.0. Finally, the votes column was transformed into a numeric value. The original column contained vote counts as strings with commas, such as '177,031'. These were converted to float values by removing the commas. These preprocessing steps have effectively transformed the Netflix data set into a more analyzable format. The first approach to impute the missing data in the Netflix data set is to look for the same movies in the IMDB data set. After This preprocessing stage of the Netflix data set, the certificate column had the most missing values at 40.44%, followed by duration_in_min at 18.44%, rating and votes each at 13.18%, Released_Year at 7.50%, and 'genre' at 0.81%. The conclusion is that there are few common movies in both data sets, and handling missing data will require a different approach due to the large amount of missing data. However, the genre column has negligible missing data which can be removed without causing severe problems.

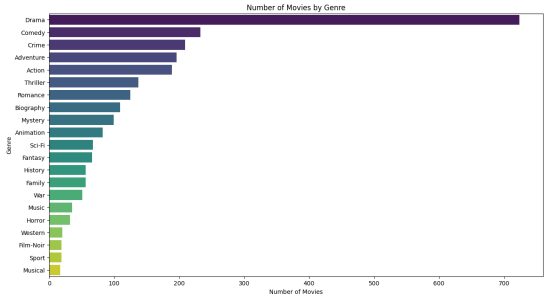
4. IMDB EDA and preprocessing

4.1 EDA

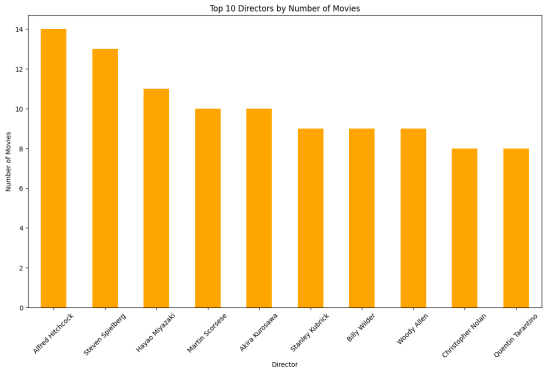
We removed any duplicate movies before exploring the data and then created some visualizations



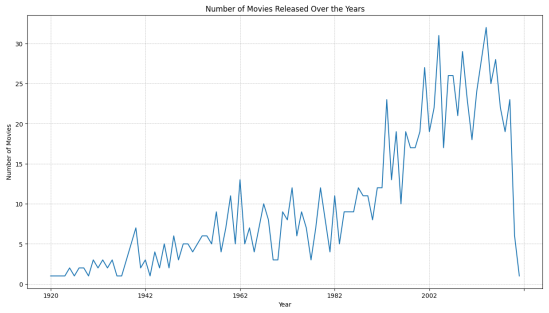
(a) Distribution of IMDB Ratings



(b) Number of Movies by Genre



(c) Top 10 Directors by Number of Movies



(d) Number of Movies Released Over the Years

Figure 4.1: Visualizations for IMDB Data

4.2 preprocessing

In the preprocessing of the IMDB dataset, missing data was identified in the Meta_score and Certificate columns. The Meta_score column had 15.7% missing values, which were filled with the median value of the entire column. The Certificate column had 10.1% missing values, which were replaced with a placeholder value Not Specified. After these preprocessing steps, there were no remaining missing values in the dataset. Additionally, the Runtime column was transformed from a string format like xxx min to a float data type representing the runtime in minutes. This transformation makes the Runtime column more suitable for numerical analysis and machine learning algorithms.

5. Machine Learning Model

In the data preparation stage for machine learning, Netflix and IMDB data sets were combined to form a single dataset. This was achieved by standardizing the column names across both datasets. Furthermore, the genre column was converted to lowercase in both datasets to ensure consistency. Subsequently, the datasets were concatenated, resulting in a combined dataset that has movies from both Netflix and IMDB.

The first step in the machine learning process was feature extraction, where each movie was represented as a binary vector indicating the presence or absence of each genre. This transformation allowed for the calculation of distances between movies based on their genres. Following this, the K-means clustering algorithm was used to group movies into clusters. To determine the optimal number of clusters, the elbow method was used. This involved iterating over a range of cluster numbers and calculating the within-cluster sum of squares (WCSS). A plot of WCSS against the number of clusters revealed an elbow point at around 5 or 6 clusters, indicating that adding more clusters beyond this point would result in poorer clustering results. Based on this analysis, K-means clustering was performed with 5 clusters. The resulting clusters were then added to the combined data set. This process effectively grouped movies based on their genre profiles, providing a basis for further analysis and prediction. We then used t-SNE, a dimensionality reduction technique, to visualize the clusters.

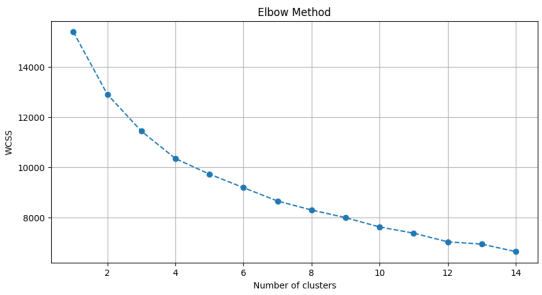


Figure 5.1: Elbow Method

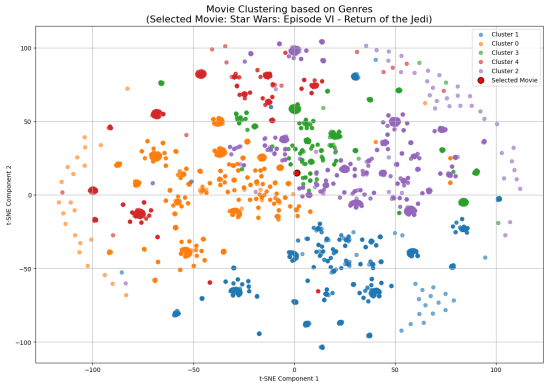


Figure 5.2: Clustering based on genres

Once the clusters are formed, we can calculate the distance between the given movie and all other movies in its cluster. Initially, we used Euclidean distance, which is a common method for calculating distances in high-dimensional spaces. However, considering the high-dimensional nature of our data, we found that cosine similarity might be a more appropriate measure. Cosine similarity measures the cosine of the angle between two vectors and is often used in high-dimensional spaces. By calculating the cosine similarity between the given movie and all other movies in its cluster, we can identify the top five movies that are most similar to the given movie.

6. Conclusion

In conclusion, our project has successfully developed a robust movie recommendation system that simplifies the process of movie selection. By combining two data sets from Netflix and IMDB, we have created a single data set of unique movies. Subsequently, this data set underwent thorough preprocessing and cleaning to be used in the application of a machine learning model to generate recommendations.

Looking ahead, there is significant potential for further development and enhancement of our movie recommendation system. While our current model primarily utilizes the genre feature to group similar movies together, future iterations of the system could utilize additional features.

6. References

[1] Narayan63. Netflix popular movies dataset, 2022.

[2] H. Shankhdhar. Imdb dataset of top 1000 movies and tv shows, 2020.