# Machine Learning for the Analysis and Prediction of Film Popularity

**Ziang Liao**
1412182
COMP20008
zialiao@student.unimelb.edu.au

**Hao Tian**
1342982
COMP20008
httia1@student.unimelb.edu.au

**Liu baojun**
1280146
COMP20008
Baojun@student.unimelb.edu.au

**Shengning Ding**
1342782
COMP20008
shending@student.unimelb.edu.au

## Executive Summary

**This study aims to predict the popularity of Netflix movies and TV shows by utilizing relevant information such as description, genres, production country and credit list. Firstly, we have done some pre-analysis on the data by using the histogram to compare the distribution of production countries and actors and get the top 20th by average TMDB popularity and IMDb score. Word cloud is applied to show the frequency of keywords in the description. Secondly, we processed the missing data based on various indicator characteristics and missing data volume. After exploratory analysis, it was confirmed that the core of the study is to predict its score on IMDb. To this end, our group used Scaling, TF-IDF, PCA analysis and One-hot encoding to pre-process the indicators, preparing for the application of machine learning models. During the analysis phase, linear regression, decision tree models, and K-nearest neighbor models are used for prediction. Additionally, we explore the possibility of building a supervised model based on the cluster number of a K-means clustering model on description instead of using principal components directly to let our clients interpret the model better. Finally, a decision tree regressor model with the best performance and a decision tree classification model with the highest accuracy was selected as two optimal models. This study provides a reliable and detailed methodological framework for predicting the popularity of Netflix movies and TV shows.**

## *1. Introduction*

### 1.0 Data Description

The datasets we analyze are two separate CSV files. The main dataset (Titles.csv) contains information about Netflix movies and TV shows, while another file (Credits.csv) contains information about the 'actors' and 'directors' for the titles in the main dataset. The datasets contain approximately 5850 different movies from 1945 to 2022 and focus on 8 quantitative and 11 qualitative variables. These are split up into the id, title, type, description, release_year, age_certification, runtime, genres, production_countries, seasons, _id, imdb_score, _votes, tmdb_popularity, tmdb_score, person_id, name, character and role. Data cleaning and preprocessing helped remove missing instances. The data and the analysis below allows for exploring and learning more about the underlying trends in the movie industry.

### 1.1 Research Question:

**How do relevant information such as description, genres, production country and credit list impact the popularity of Netflix movies and TV shows.**

With the rise of digital media platforms, there is an increase in film or TV show platforms which share data resources. They also play an important role in customers' decision on whether or not they are going to watch a movie and TV show. This question helps movie producers and financiers to make a decision on funding a movie or TV show project. Furthermore, it can also assist casting directors when selecting actors for the various roles in the film. Understanding this will help to identify a successful movie project and might even help to predict which movies and TV shows may have a higher popularity.

## *2. Data Exploration and Analysis*

Firstly, we analyze the distribution of movies and TV shows by types, release year and genres separately and get the following diagrams. The first graph(Figure 1), shows that around two thirds of the data are movies, while a third of them are television programs.

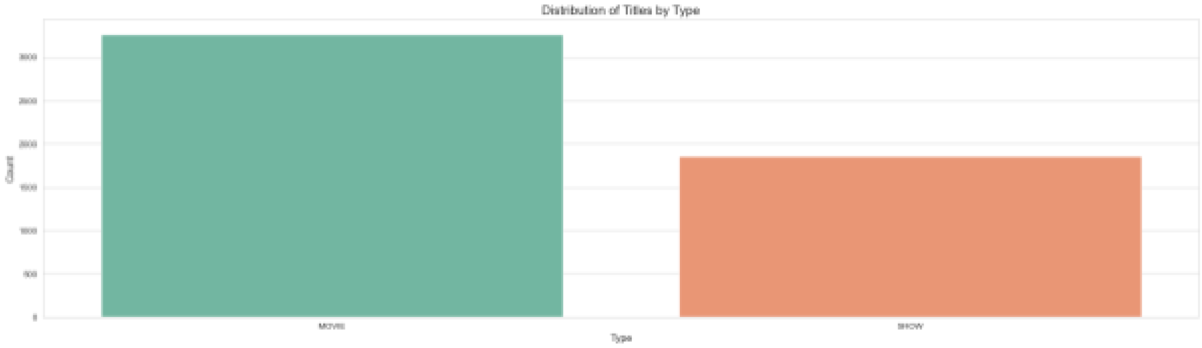

**Figure 1:** Plot the distribution of types (Movie vs. Show).

Next we find out that most movies and TV shows were released after the 20th century, a majority of which was released around 2020(Figure 2).

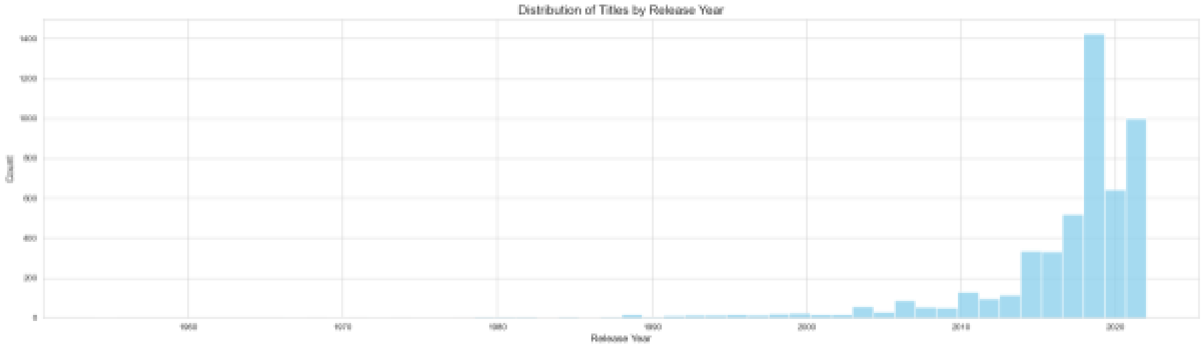

**Figure 2:** Plot the distribution of genres by release year.

Then the bar chart(Figure 3) shows the distribution of movies and TV shows with different genres. It also suggests that a large proportion of  movies and TV shows belong to “drama” and “comedy” genres.

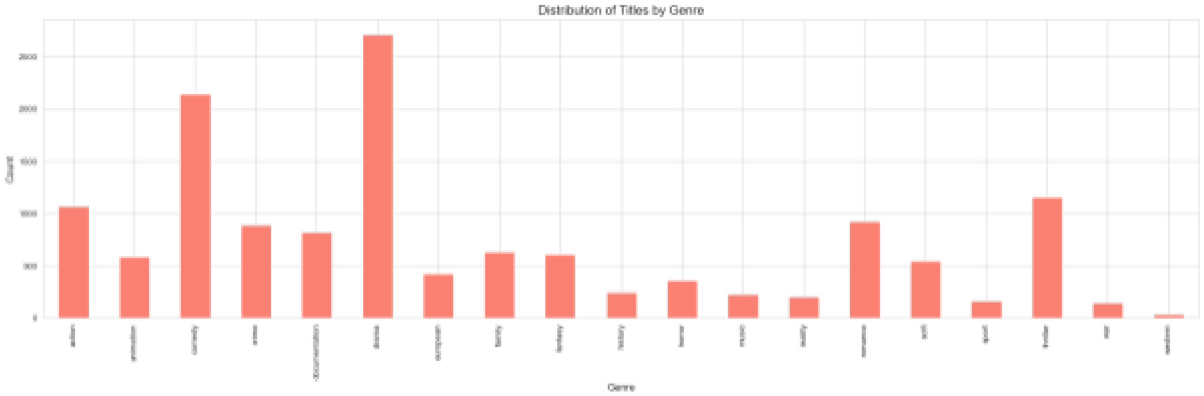

**Figure 3:** Plot the distribution of titles by genre.

Then, we figure out the top 20 production countries by the number of movies and TV shows(Figure 4).

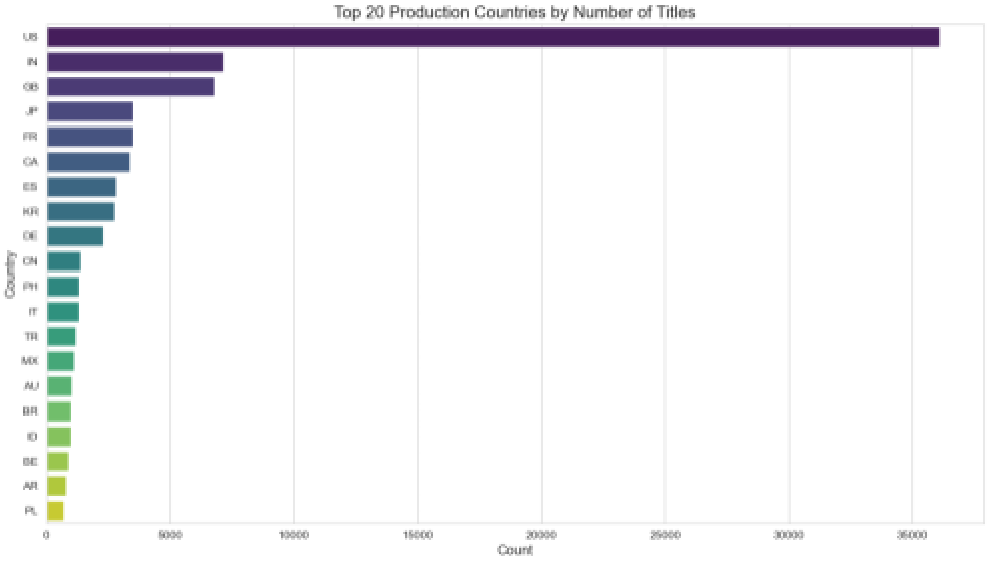

**Figure 4:** The visualization of the distribution of titles by production country.

We also plot the top 20 countries by the Average TMDB popularity and  IMDb score of the movies and TV shows they produced(Figure 5).

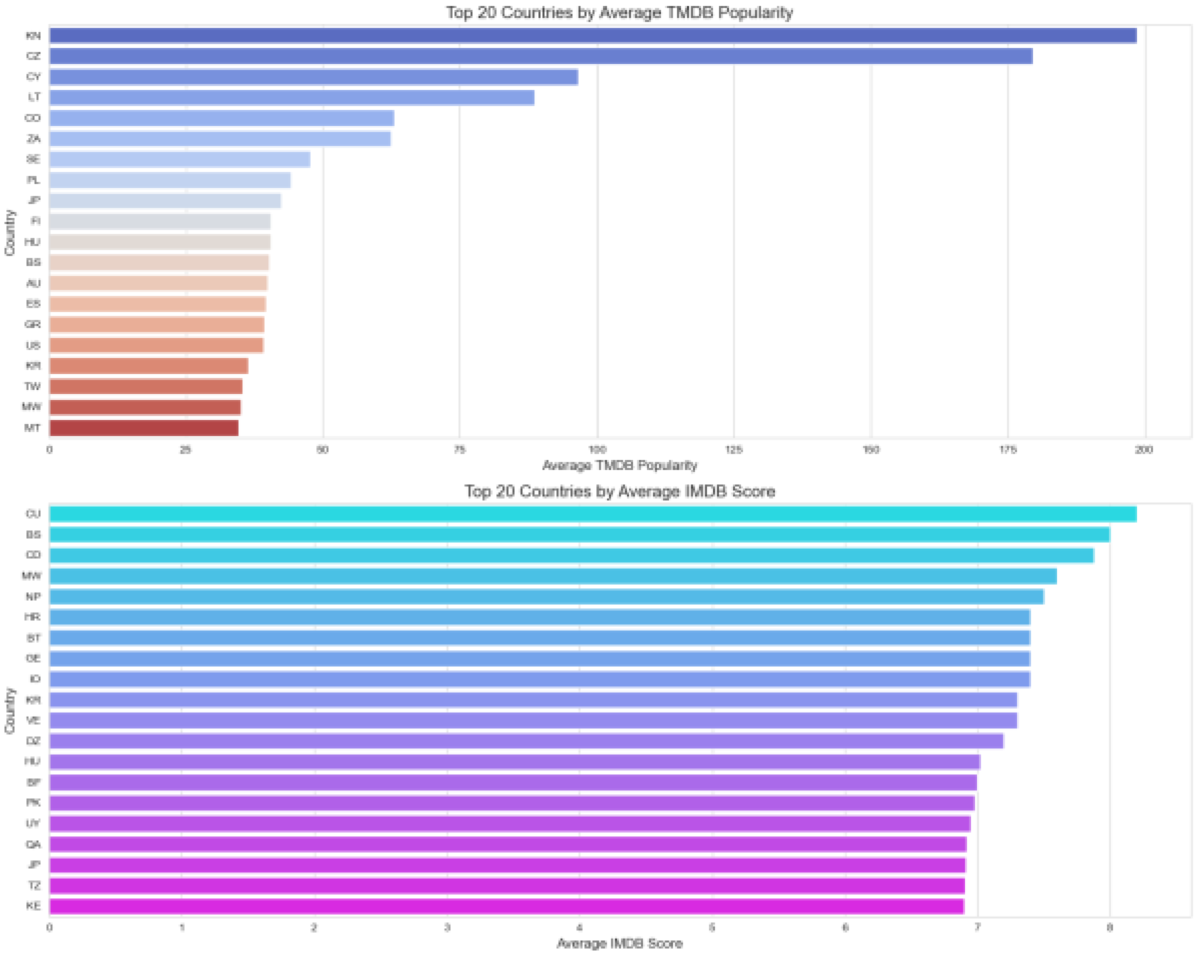

**Figure 5:** The visualization of the top 20 countries with the highest TMDB popularity and  IMDb score

After we inner join the titles.csv and credits.csv on movie id  we can use the same method to rank the actors' popularity, then we get the following diagram(Figure 6).

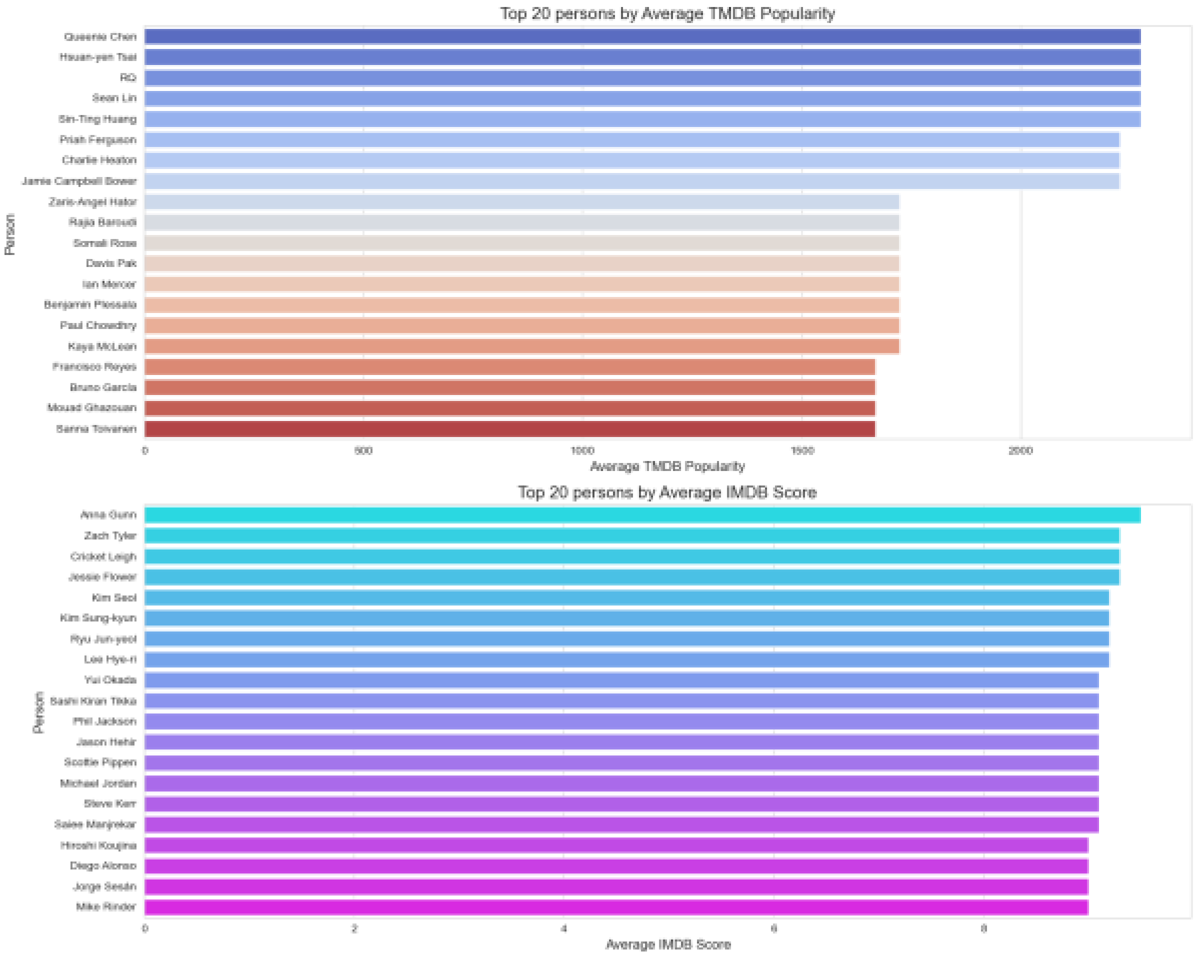

**Figure 6:** The visualization of the top 20 actor/actress with the highest TMDB popularity and IMDb score

Next is a textual analysis of the description of movies and TV shows, as it is necessary to see what kind of description would correlate with which rating.

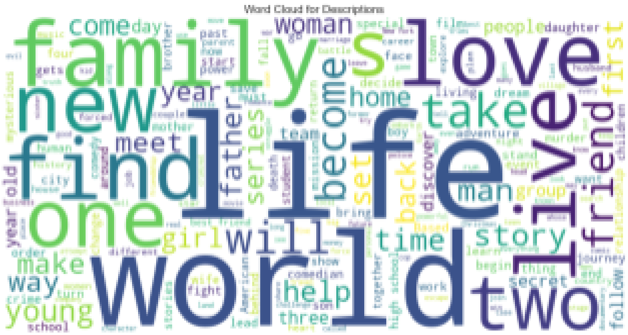

**Figure 7:**  Word cloud for description

The above word cloud(Figure 7) shows that the most common words are ”life”, ”world”, ”family”, ”love”, etc.

In conclusion, these charts provide some context on how different features affect the TMDB popularity as well as the  IMDb scores of the movies and TV shows.As all the above features may affect our predictions, we have included all these factors in our model for better performance.

Popularity is an important metric on TMDB. It is impacted by many attributes like number of votes, views, “Favorite” and the score of the movie of the day which means a newly released movie is more likely to get a high TMDB score. IMDb score is a 1 to 10 voting system in which individual votes are then aggregated and summarized as a single  rating. We decided to choose IMDb score as the target variable which represents the popularity of a movie or a TV show to examine.

## *3 Methodology(Data Preprocessing)*

### 3.1 Data Imputation (Handle the missing data)

Firstly, we analyze the missing values of the data in the title.csv file, and the results are as follows(Table 1):

**Table 1:** The missing values of the title.csv file

| Column name | The number of missing data |
|---|---|
| id | 0 |
| title | 1 |
| type | 0 |
| description | 18 |
| release_year | 0 |
| age_certification | 2619 |
| runtime | 0 |
| genres | 0 |
| production_countries | 0 |
| seasons | 3744 |
| _id | 403 |
| imdb_score | 482 |
| _votes | 498 |
| tmdb_popularity | 91 |
| tmdb_score | 311 |

From Table 1, (1) there are fewer missing values in the "title" and "description", so we directly removed these missing values. (2) We consider that some movies or TV shows have season numbers, while some movies or TV shows do not have season numbers. For seasons, we filled it with 0 to indicate that it is a movie and not a TV show. (3) Column "age_certification" refers to the age rating or rating set by film censorship agencies for movies. The data column "age_certification"  has a huge number of missing values, so we populated it with a special value "Unknown" as a "placeholder" to fill in the missing values to ensure we have enough data to train our model. (4). For rating and voting columns (e.g., imdb_score, _votes, tmdb_popularity, tmdb_score), these missing data might be caused by those Movies or TV shows, due to the exclusive copyright, only released on one of the  or TMDB platforms. If we choose some values to fill in these columns, the result will lead to the skewing of practical variance, thus causing bias. For a trade-off, we choose to remove these missing value instances to ensure the model's accuracy.

### 3.2 Text Processing & Dimensionality Reduction

For the "description" column, the majority of machine learning models cannot process text-type data. The description of a movie or TV show has a significant impact on the popularity of TMDB. Therefore, it is necessary to vectorize the "description" and transform it into a digital vector for subsequent machine-learning methods. Therefore, we use the TF-IDF method to vectorize the description. After the description is vectorized by the TF-IDF method, the matrix obtained would be a sparse matrix with high dimension. Therefore, we used the PCA method to reduce the dimensionality of the data and obtained 3 principal component features (pc_1 to pc_3) from it.

### 3.3  Encoding

For the columns with categorical value, we employ encoding techniques. Due to the fact that both genres and production countries in movies or TV shows are a list with multiple values, we applied one-hot encoding to genres to produce a vector of length 19. In this vector, the position corresponding to a specific genre is set to 1, and all other positions are set to 0. The same one-hot encoding method is used for the production countries. For type, age_certification and person_id, we converted these columns to categorical codes.

### 3.4 Data Scaling

After completing the above processing, all indicators have been transformed into numerical data. Owing to the huge difference in the number range between every attribute, we choose to use the StandardScaler method of Sklearn to standardize the data range. This can consolidate these indicators to the same scale and distribution to avoid some imbalanced range indicators having a significant impact on the machine learning models.

## *4. Result*

Our goal is to make predictions on the tmdb_popularity field, which is a continuous variable. Therefore, we initially regard this prediction as a regression task. We first tried a linear regression model. Considering the fact that one-hot encoded categorical variables are not very suitable for a linear regression model and some of the features may be correlated with each other(e.g., imdb_score, _votes, tmdb_score).  We used it as a baseline model for our analysis. We then explored the possible use of two other regression models: KNN regressor and decision tree regressor. We use RMSE (Root Mean Square Error) as our evaluation metric. To see whether the classification method was also suitable for this prediction task or not, we also explored the possibility of building a classification model based on the cluster number of a K-means clustering model on description instead of using principal components directly to help the clients better understand the model.

### 4.1 Linear Regression

For linear regression,we just fit the model as there aren't hyperparameters to tune for a basic linear regression. We got a root mean squared error(RMSE) of 0.7 and run-time is approximately 0.2 seconds.

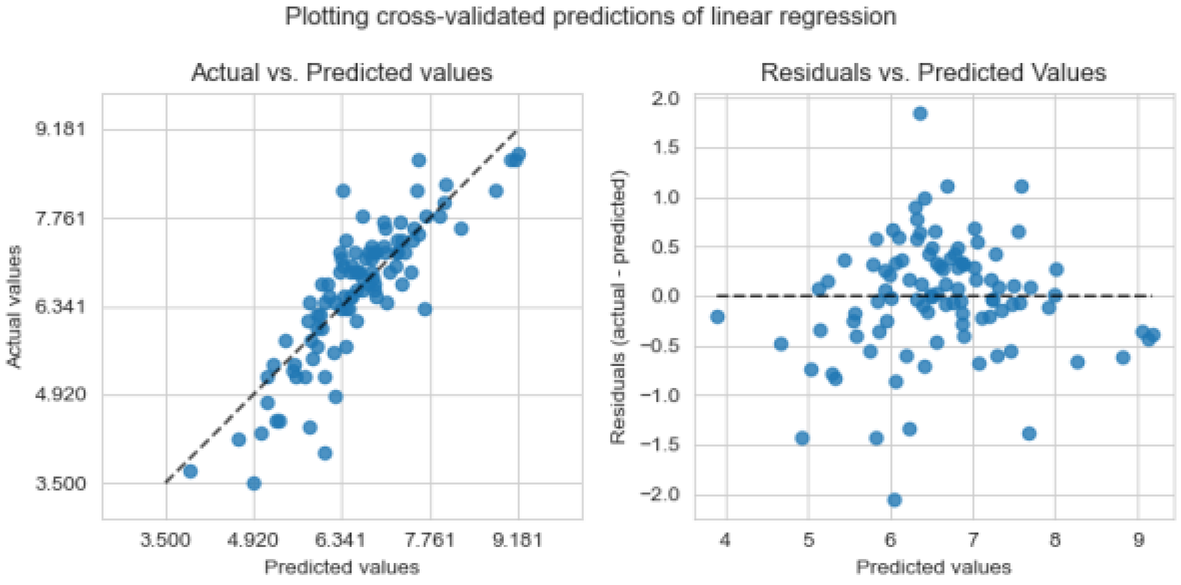

**Figure 8:** Visualization of the prediction error of a linear regression

The left part of the plot(Figure 8) above compares the actual value and our predictions, the right one shows us how the error fluctuates. From the graph, we can see that the predictions are sometimes correct, but often there are residuals of maximum 2 which is quite huge considering the score is a 1 to 10 voting system. As mentioned previously, this model is selected as our baseline model.

### 4.2 KNearestNeighborsRegressor

Here is a brief introduction of the mechanism of KNN regressor. Similar to its classification counterpart,the first step in using KNN regression is to choose the number of neighbors (K) to consider when making predictions. K represents the number of data points closest to a new data point that will influence the prediction.The choice of K is a hyperparameter, and it can impact the model's performance. A small K value may lead to a noisy prediction, while a large K value may lead to overly smoothed predictions. The next step is calculating Distance. For a given data point that you want to make a prediction for, KNN calculates the distance between that data point and all other data points in the training dataset. The most common distance metrics used are Euclidean distance and Manhattan distance, but other distance metrics can be used as well. Once distances are calculated, the algorithm selects the K data points (neighbors) with the smallest distances to the new data point. These neighbors are the ones that will contribute to the prediction. For regression, the predicted value for the new data point is typically calculated as the average (mean) of the target values of the K nearest neighbors. In some variations of KNN regression, you can assign weights to neighbors based on their distance, giving more influence to closer neighbors. Finally, the calculated average (or weighted average) of the target values of the K nearest neighbors is the prediction for the new data point.

We use GridSearch to try different combinations of values and search for the best parameters through cross-validation. The best parameters to use are {'metric': 'manhattan', 'n_neighbors': 7, 'weights': 'distance'}. The RMSE of this model is roughly 0.9, which is close to that of the baseline model. However, this model runs for roughly 10 minutes, which is much worse than the baseline model.

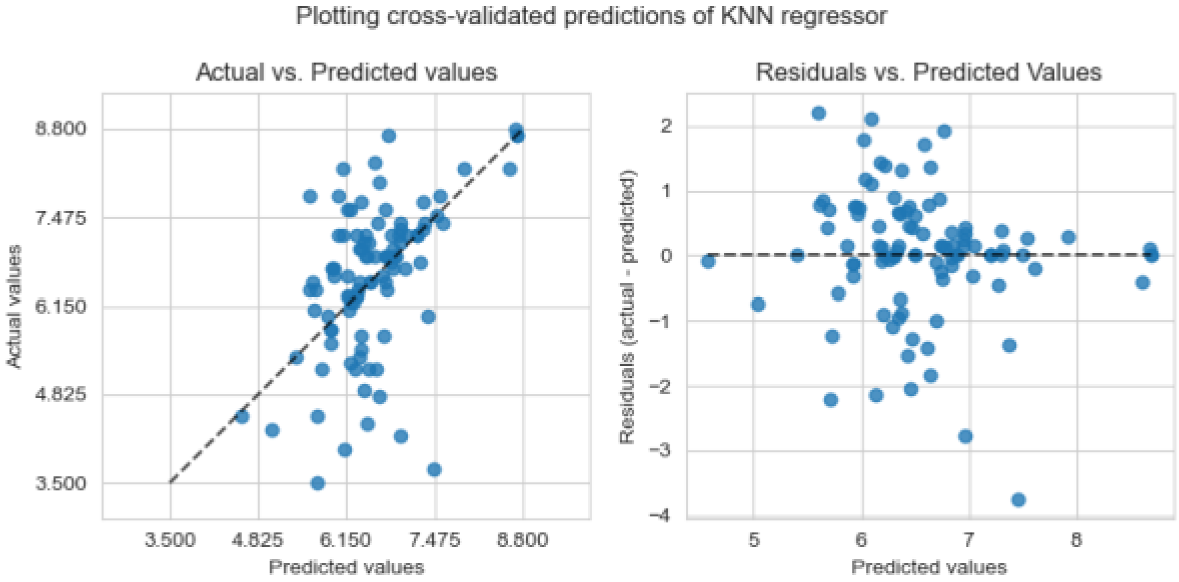

**Figure 9:** Visualization of the prediction error of KNN regression

We can observe that the prediction errors plot(Figure 9) is similar to that of the baseline model. We conclude that KNN regressor is not a good model to use in this case

### 4.3 Decision Tree Regressor

Similar to decision tree classification, a decision tree regressor selects a feature from the dataset to split the data into two subsets. It chooses the feature and split point that minimizes the mean squared error (MSE) or another suitable criterion. The dataset is recursively divided into subsets, and the process continues until a stopping criterion is met, such as a maximum depth of the tree or a minimum number of samples in a leaf node. At each leaf node of the decision tree, the algorithm assigns a predicted value. In the case of a Decision Tree Regressor, this value is typically the mean (average) of the target values in that leaf node.

In this model, we used 5 cross-validation folds to tune the hyperparameters of the decision tree, after which the GridsearchCV gives the best parameters of {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}. Finally,we got a model with a RMSE of 0.08 roughly, it is much smaller than that of  the simple linear regression which means its prediction is much more accurate.The prediction errors for decision tree regressor are shown below(Figure 9).

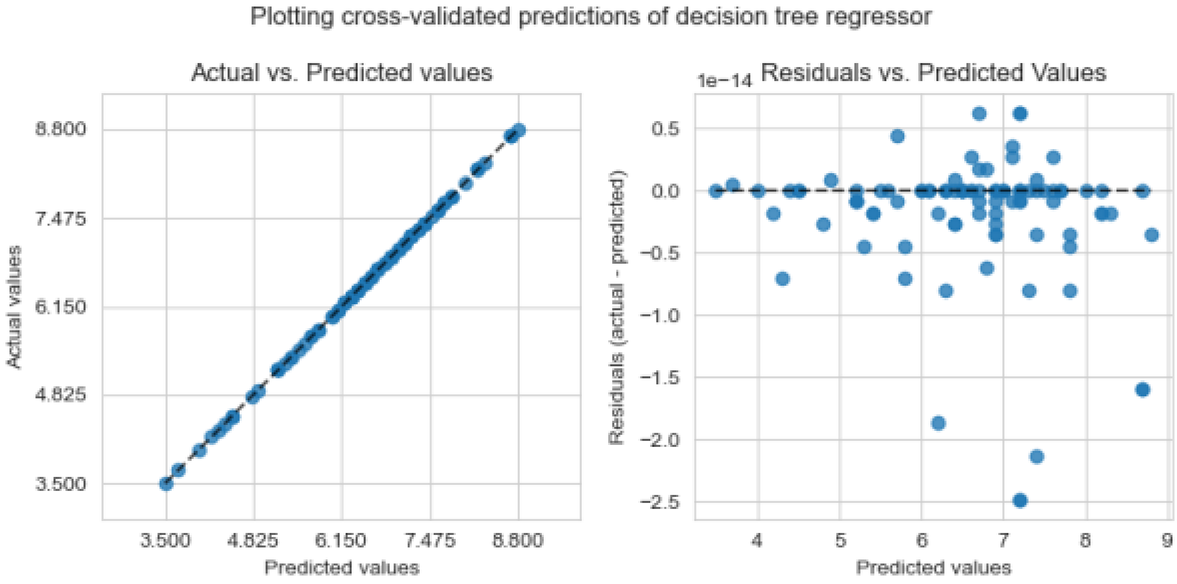

**Figure 10:** Visualization of the prediction error of the decision tree regressor

According to Figure 10, the actual value is very close to our results. And residuals are all less than 0.00025 which is marvelous, therefore this model is the one we are most likely to provide to our customers.

## 4.4 K-means Clustering

From PCA(Principal Component Analysis)we reduce the dimensionality of the Bag-of-Words(BoW)to 3 principal components. Since it may be non-intuitive to interpret the principal components, We decided to cluster the description and build a supervised model based on the cluster number instead of principal components. The Elbow method is used to plot the within-cluster sum of squares (WCSS) against the number of clusters(Figure 11).

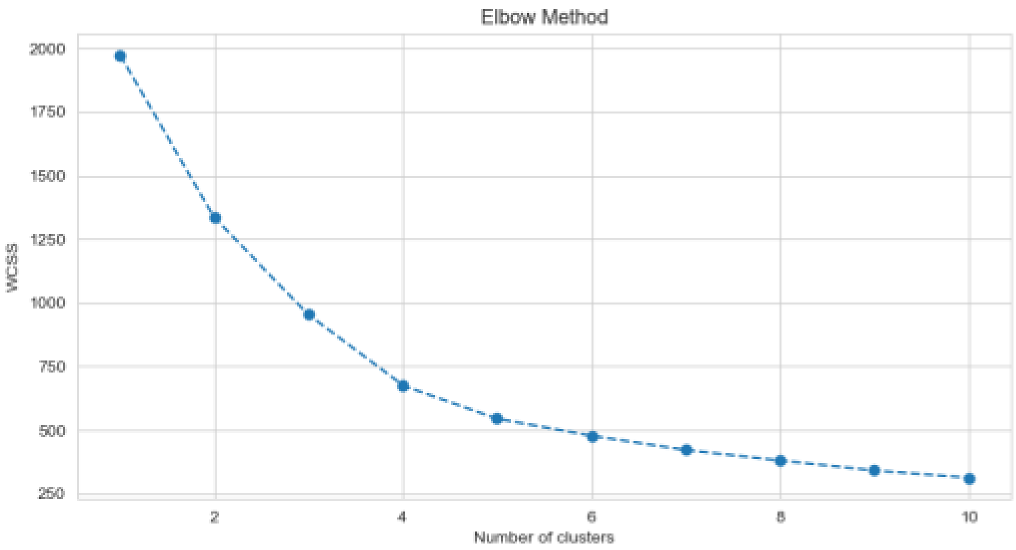

**Figure 11:** Elbow method

We can read that 4 is the point where the reduction in WCSS significantly slows down(Elbow point). Hence we select 4 as our k value and use it to split description into 4 clusters.

### 4.5 Classification model:Decision tree and KNN

As we considered that the regression model needs to use continuous values,we have tried to use classification models as a backup. Firstly, We use equal width binning to discretize the IMDb score into three bins which are low, medium and high respectively and map it to three labels 0, 1, and 2. The original regression task is transformed into a classification task, and we use KNN and Decision Tree algorithms for classification. Like the previous regression task, we use GridSearch to try different combinations of values and search for the best parameters through cross-validation. The specifics of the algorithms will not be repeated. The best parameters we got for the two classification model are {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2} and {'metric': 'manhattan', 'n_neighbors': 5, 'weights': 'distance'}. The accuracies we got are 0.997 and 0.6 respectively. The runtime of the decision tree model is 225 seconds whilst the runtime of the KNN model is 517 seconds. We visualize the predicted results by using the confusion matrix as below(Figure 12).

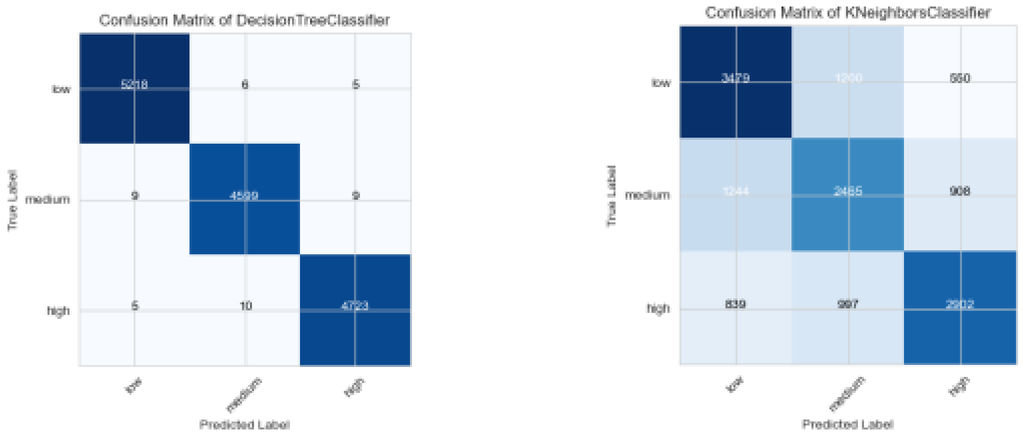

**Figure 12:** Confusion matrices for two classification models

## 5. *Findings and Interpretation*

To sum up, we can build a list of findings from what we have collected up there:

- Around half of the movies and TV shows are classified as drama and comedy. Action and thriller are also two genres which are popular to movie producers.

- A growing number of movies and TV shows were produced after the twentieth century, especially around 2020.

- In terms of movie and television production, the United States leads in quantity, while countries like Cuba, the Bahamas, and the Democratic Republic of the Congo are known for their high-quality productions, as indicated by IMDb scores.

- Anna Gunn, Zach Tyler, Cricket Leigh, and Jessie Flower have consistently garnered high IMDb scores for the movies and TV shows in which they starred in. Their involvement can serve as a reliable indicator of a production's potential popularity among audiences. We've integrated the credit list as a feature in our machine learning models, allowing casting directors to receive predicted scores for various combinations of cast members. This predictive capability will aid them in making informed decisions when selecting actors for productions.

- Words like ”life”, ”world”, ”family” and “love” appear frequently in description. Hence they get penalized in the TFIDF matrix. The principal components we got after PCA analysis help us mitigate the curse of dimensionality of the models. These models will be valuable in assisting screenwriters in crafting more compelling descriptions and providing producers with insights to make better choices when selecting film projects.

## *6. Limitations and improvement opportunities:*

### 6.1 Limitations

1. The processing time of the decision tree models and KNN models are relatively too long because they employ m-fold cross-validation.

2. The BoW is built before the test-train data split which might cause feature leakage.

3. We didn’t apply feature selection in our model which might increase the model complexity and computational costs.

4. We didn’t evaluate the generalization of our model. It is very likely that our models are overfitting due to the extensive features.

### 6.2 Improvement opportunities

1. Construct the BoW representation solely using the training set and avoid using the test set.

2. After splitting the training dataset and the test dataset, a BoW representation is created for each of them. This ensures that the model does not inadvertently learn information from the test data during training.

3. Apply filtering methods like ranking features based on pearson correlation for regression model and mutual information for classification models.

4. Diagnosis Overfitting by plotting a learning curve.

## *7. Conclusion:*

We have made many key takeaways relating to the analysis of how relevant information such as description, genres, production country and credit list impact the popularity of Netflix movies and TV shows. We understand the distinct changes in production countries, genres, actors in credit list and word in description impact the potential success and outreach of movies. These conclusions are essential for our understanding of how the movie rating in digital platforms operates and changes over time. There are still some unsolved questions about how attractive movie or TV show's titles lead to higher IMDb scores, and so on. Overall, the analysis and the machine learning models contribute to movie producers and casting directors to make better decisions based on historical data on Netflix, thus creating high-achieving popularity movies and TV shows.