Movie Popularity and Quality Prediction Based on Multi-Modal Learning

Bu Jiaxin
NUS Business School
National University of Singapore
A0262748U

Jiang Jinjing
NUS Business School
National University of Singapore
A0231986U

Li He
NUS Business School
National University of Singapore
A0262704J

Pragati Sangal
NUS Business School
National University of Singapore
A0262745Y

Wang Yihe
NUS Business School
National University of Singapore
A0262735B

Abstract

In this project, we use Machine Learning, Deep Learning and NLP models to understand and predict the popularity and quality of a given movie/series. The final selected model is the XGBoost model with inputs of both basic features and poster features. And the results show that posters, awards, and director are the most important features that reflect both the quality and the popularity of the movie. In the end, we deploy the prediction model as a real-time API service and source code can be found at https://github.com/azuretime/Netflix

-Prediction.

1 Problem Definition

The rise of streaming services like Netflix has transformed the way people consume movies and TV shows, offering a vast library of content that can be watched on-demand. As of 2022, Netflix has over 230 million subscribers in more than 190 countries, making it one of the largest streaming services in the world. Netflix annual revenue for 2022 was $31.616B, a 6.46% increase from 2021. Netflix annual revenue for 2021 was $29.698B, a 18.81% increase from 2020.

The potential revenue growth of the streaming industry is significant, as more and more people around the world continue to shift away from traditional cable and satellite TV and towards on-demand streaming services.According to a report by Grand View Research, the global streaming market was valued at USD 89.03 billion in 2022 and is expected to grow at a compound annual growth rate (CAGR) of 21.5% from 2023 to 2030. This growth is driven by factors such as the increasing popularity of on-demand content, the proliferation of mobile devices, and the availability of high-speed internet.

However, competition among streaming platforms is also intense, with companies like Disney, CNN, Apple, Amazon, and Warner Bros all vying for a slice of the market. To maintain subscriber loyalty and generate a high return on investment, it's crucial to identify projects that have both high-quality and latent popularity. This is where machine learning can play a crucial role in predicting the potential value of candidate projects.

This project aims to leverage the power of natural language processing and computer vision techniques to predict the quality and popularity of films based on

IMDb movie data. To achieve this, we will conduct feature engineering on both the movie summary text and poster images, using methods such as natural language processing and CNN image processing. Next, we will train and evaluate several machine learning models, including classification models such as Decision Tree, Random Forest, and XGBoost, as well as a neural network, which can effectively capture non-linear and complex patterns in the data. By combining these models with the extracted features, we aim to predict the scores and votes of

films accurately. The insights obtained from this project can have practical applications in Netflix business, such as informing decisions around movie production, purchase, and marketing.

2 EDA & Data Pre-processing

The dataset used in this project is taken from Kaggle. It contains 15,480 records and 29 features including the target variables "IMDb Score" and "IMDb Votes". Each observation in the dataset represents a movie/series record with other basic information of the content such as genre, language, release date, country, director, actor and so on. The following [Table 1] shows information of variables used in this project.

Data variables	
Title	**IMDb Score**
Genre	Rotten Tomatoes Score
Tags	Metacritic Score
Languages	Awards Received
Series or Movie	Awards Nominated For
Hidden Gem Score	Boxoffice
Country Availability	Release Date
Runtime	Netflix Release Date
Director	Production House
Writer	Netflix Link
Actors	IMDb Link
View Rating	Summary
Poster	**IMDb Votes**
TMDb Trailer	Image
	Trailer Site

Table 1: Data Variables

As the objective of this project is to predict the content popularity and quality based on given IMDb votes and IMDb scores respectively, we classify this problem as a multi-class classification problem having two target variables with three classes 'Low', 'Medium' and 'High' by converting continuous target labels into categorical labels. Before we do so, based on our initial exploration, we observe that the "IMDb Score" feature is normally distributed whereas "IMDb Votes" follows a power law distribution [Figure 1]. By further discovery, we will illustrate how we predict the popularity and quality for a given content.

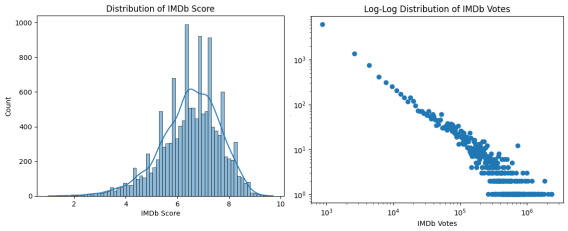

Figure 1: Distribution of Target Variables

2.1 Exploratory Data Analysis (EDA)

In our project report, we initially conducted preliminary data cleaning tasks, including renaming feature names and eliminating any missing values from the features. The specifics of these tasks will be outlined in the Data pre-processing section. Subsequently, our exploratory data analysis focused on examining the independent features with regards to their ability to predict the popularity and quality.

- Movies Vs Series: The most popular content type on Netflix is movies. According to popularity and quality of the content, over 70% of the audience favors movies over series [Figure 2]

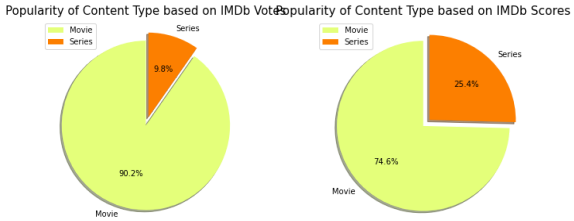

Figure 2: Distribution of Content Type on Netflix

- Genre: Among all genres, "Drama" is the most favored one by the audience. In terms of content quality, It's followed by the "Documentary" genre [Figure 3]. However, in terms of content quality, the audience favors "Drama" followed by "Action" [Figure 4].

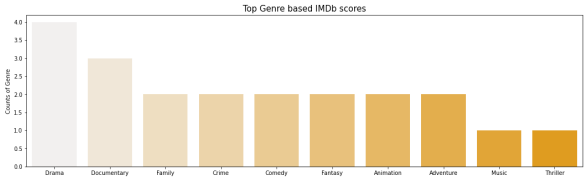

Figure 3: Top Genres based on IMDb scores

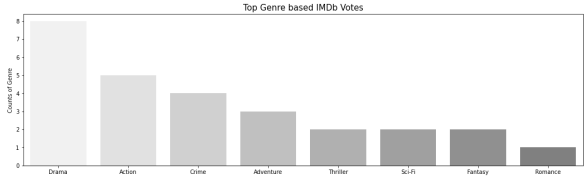

Figure 4: Top Genres based on IMDb votes

- Content Released over Years: The highest number of content (2,500+) were released in 2015 and 2020. Due to the covid, releases of content were significantly dropped to less than 500 in 2021 and also we don't have complete data of 2021. [Figure 5].

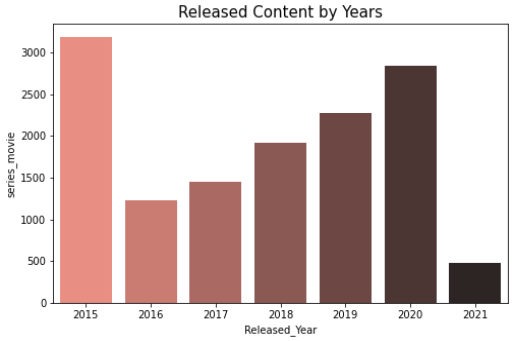

Figure 5: Distribution of content released over years

- Content Released over Months: Netflix tends to release the most content in April (with over 3,000 releases), followed by December, among all the months. Moreover, there has been a consistent increase in the production of content by Netflix from June through December, over the years [Figure 6]

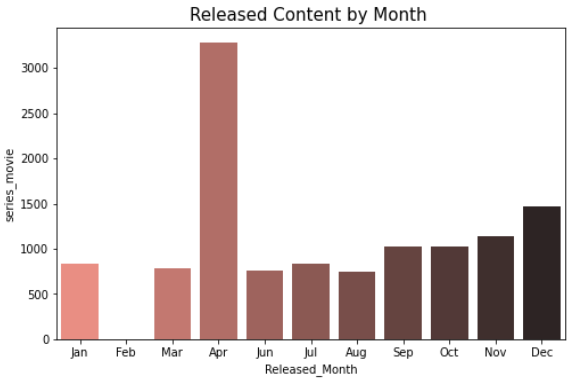

Figure 6: Distribution of content released over month

- Title: When analyzing the titles of shows on Netflix, it was found that the most frequently used words are Love, Life, Girl, and Christmas. This supports the idea that December is the most popular release month among all months because it is associated with Christmas, which is a time of love and warmth [Figure 7].

Figure 7: Word Cloud on Title

- Language: "English" language content is the most popular one among all languages followed by "Spanish" language [Figure 8]

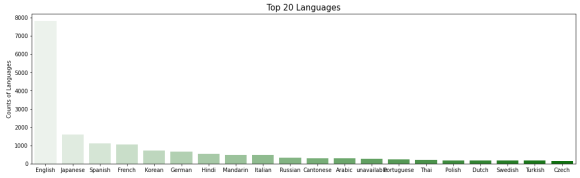

Figure 8: Top 20 Content based Languages

- Country: The "United Kingdom" takes the lead in producing the largest number of content titles on Netflix, with over 5,000 productions, followed by the "Czech Republic". However, when considering both popularity and quality ratings, countries like Romania, Turkey, Czech Republic, Poland, and Hungary is among the top 10 rated countries [Figure 9]

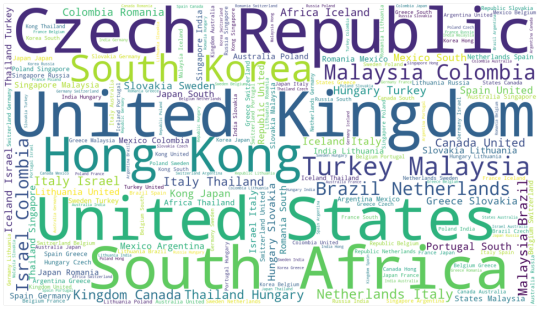

Figure 9: Word Cloud on content produced Countries

- Rating: Rating with the largest number of Movies content is "R" with 2,000 content titles and the rating with the largest number of Series content is "TV-MA" with 500++ content titles. In conclusion, most of the audience is of mature age [Figure 10].

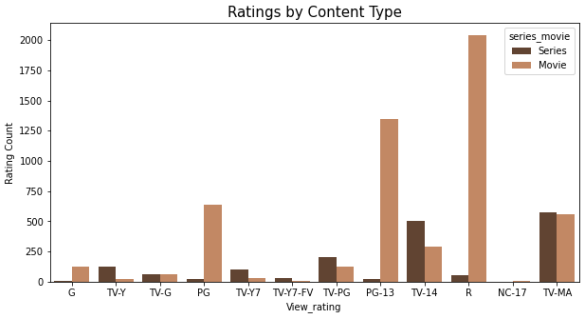

Figure 10: View Ratings by Content type

2.2 Data Pre-processing

In the given dataset, as we addressed the issue of numerous null values in the features, our initial approach involved dropping rows based on the 'IMDb_votes', 'IMDb_score', 'Summary', and 'Genre' features since they contained null values and might be important features in predicting the content popularity. Further, we filled the null values for other categorical variables with the label "unknown" and for numerical variables with 0. Additionally, we converted the column format from string to integer and proceeded to perform feature engineering.

3 Feature Engineering

There are 29 features in the dataset as mentioned above. Few features such as "Runtime", "country", "Hidden Gem Score","Rotten Tomatoes Score","Metacritic Score" are dropped as they do not help to classify the target variables and few variables (Realeased_Month, Realeased_Year) are being added. The features with their type are shown in the below attached table [Table 2].

Numerical	Categorical
Awards_received	series_movie
Awards_nominated_for	Director
Released_Month	Actors
Released_Year	View_rating
	Genre

Table 2: Feature Engineering Features

As mentioned above, we converted both target variables into multi-class labels using “Quantile-based binning” technique. All three generated classes ‘Low’, ‘Medium’ and ‘High’ are balanced in both target variables.

Subsequently, we encode the categorical variables into numerical variables. That is, we apply Label Encoder on features like series_movie, Director, Actors and One-Hot Encoder on ‘Genre’ feature to transform them into numerical variables. Furthermore, in order to facilitate the analysis and comparison between different models, we apply the scaled dataset in both linear and non-linear models.

We also added features from “Summary” and “Image” into our dataset, the details are mentioned below.

3.1 Text Features

In the original dataset, there’s a “Summary” feature that provides the summary of the movie. To get the text features from the summary, we first cleaned and tokenized the summary text, then fine-tuned a Bert model using the downstream IMDb score and votes prediction labels, and used the hidden state of the CLS token as the sentence embedding. The sentence embedding has 768 dimensions. We generated different sets of embeddings for different prediction tasks.

We also tried to generate text features using TF-IDF, but that resulted in a very large feature with 344295 dimensions, so we will use Bert embeddings for later model building.

3.2 Image Features

The ‘Image’ column in our dataset provides URL links referring to movie posters. Similar to text, we also utilized a pre-trained model, Inception V3, as an image feature extractor. The Inception V3 model was released in 2015 by Google [1], with a total of 42 layers. There are a couple of reasons why we chose Inception V3.

(1) One of the advantages of using Inception V3 is its ability to handle input images of various sizes and aspect ratios. In our case, movie posters are typically rectangular in shape. If we were to crop or pad the images to make them square, this could introduce irrelevant and misleading information, while resampling could distort the images to a large extent, both of which could negatively impact the accuracy of model inference. However, Inception V3 is designed to accept non-square images, allowing us to set the input size to a rectangular shape, with a height-to-width ratio of 1.5. This minor modification preserves the original image structure, helping to prevent potential negative effects of image distortion.

(2) Secondly, a common challenge in computer vision models is overfitting, which usually occurs when deep layers of convolutions are stacked together. However, Inception models address this issue by using multiple filters of different sizes on the same level, creating parallel layers that make the model wider rather than deeper. This structure helps to reduce the risk of overfitting, making Inception V3 a reliable choice.

(3) Thirdly, in comparison to VGGNet and ResNet, Inception Networks have proved to be more computationally efficient, both in terms of the number of parameters generated by the network and the economic cost incurred (memory and other resources). This is an ideal feature given the limited computing resources we have.

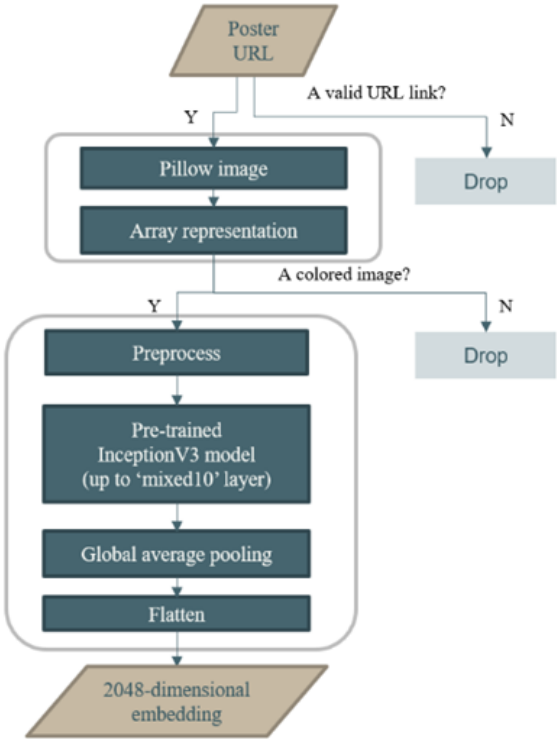

Figure 11 The flow chart of image processing

To prepare our dataset for input into the Inception V3 model, we took several preprocessing steps [Figure 11]. First, we removed any movie records that had invalid poster links, as these URLs returned a '404 Not Found' error and were unusable. Additionally, we removed grayscale images, since Inception models require images to have three channels representing the RGB color space. Colored images can be represented as a rank-3 array (i.e. height, width, and color), whereas grayscale images are single-channeled and only convey information about the intensity of light in each pixel. Moreover, we rescaled the pixel values of each image to lie between -1 and 1, and zero-centered each color channel with respect to the ImageNet dataset. These preprocessing steps helped to ensure that our model was trained on full-color, normalized images.

For our model implementation, we utilized the pre-trained Inception V3 model provided by Keras applications [2]. We set the input size to a fixed dimension of 240×160×3, and excluded the final fully-connected layer at the top of the original network. Instead, we selected the 'mixed10' layer as the final layer, as it captures high-level image patterns. The weights learned from the ImageNet dataset were directly applied to generate embeddings.

To reduce the output dimensions, we added a global average pooling layer to summarize the information from each feature map, and then flattened the results to generate 2048-dimensional vectors.

In subsequent steps, we combined the image embeddings with structured data and text embeddings to train our movie classification models. By using this approach, we were able to leverage the power of transfer learning and optimize the use of our available data to achieve better performance in our classification tasks.

4 Model Building and Evaluation - Popularity Prediction

Since the goal is to predict the popularity of the movie, which is reflected in the IMDb votes, we first convert the continuous variable "IMDb votes" into a categorical variable, so that we have a multi-class classification problem. We categorize the IMDb votes into three categories: Low, Medium and High, and the cutting thresholds are the 33% and 67% percentile. Additionally, to avoid overfitting, we further dropped a few features such as IMDb score, Netflix Released_Year, Netflix_Month features for the prediction of IMDb votes.

We then build a multi-modal model [Figure 12] where input features consist of three parts: structured features about the movie itself, like Director, Writer, View Rating and Genre; text features extracted from the movie summary, and the image features extracted from the movie poster.

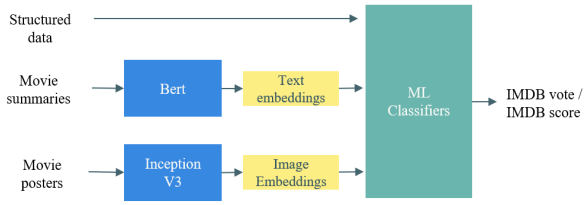

Figure 12 Multimodal learning structure

We then standardized all the features and performed a train-test split with a test size of 20%.

First, we used image features and structured features (2085 features in total) and built the model. Further, we also added features extracted from text data and it turns out that the model performance with or without text features are very close to each other. Therefore, we only use image+structured features in the later model training to avoid overfitting. The following [Table 3] shows the model performance (Test

Accuracy) for both types of dataset used for model building and evaluation.

Model	**Image + Structured**	**Image + Summary + Structured**
BernoulliNB	0.470	0.393
KNeighborsClassifier	0.386	0.364
LogisticRegression	0.572	0.575
MLPClassifier	0.557	0.564
DecisionTreeClassifier	0.578	0.586
RandomForestClassifier	0.603	0.620
AdaBoostClassifier	0.668	0.689
XGBClassifier	0.690	0.709
Neural Network	0.606	0.619

Table 3: Model Performance for Votes Prediction

It is observed that XGBoost performs the best with accuracy of 69%, so we further fine-tuned the XGBoost model and reached a test accuracy of 70.49%. The following graph [Figure 13] shows the confusion matrix on test data. We see that most wrongly predicted labels are only one class away from the true label, for example High class is predicted as medium class, very few predictions are drastically different from the true class (the case where high is predicted as low), so the prediction result is quite reliable.

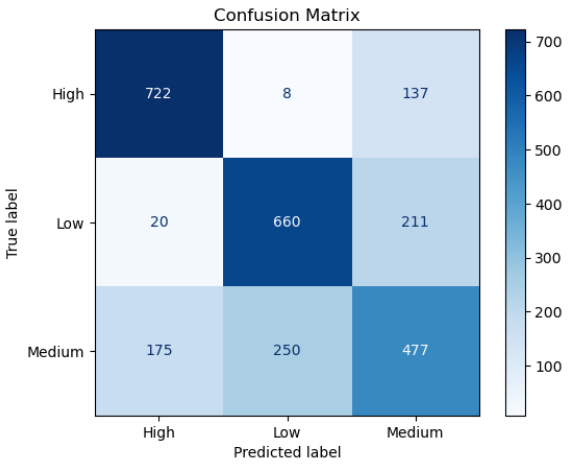

Figure 13: XGBoost Model Confusion Matrix

As the number of features we have is huge, we further check which features are contributing most to the model. We use the built-in variable importance of the XGBoost model, and the following graph [Figure 14] shows the relative feature importance.

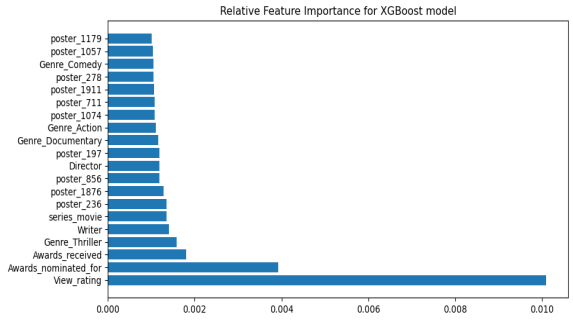

Figure 14: XGBoost Model Important Features

We see that View Rating, Awards nominated and received, Genre, Writer, Director, Series or Movies and some image features from posters are the main features contributing to the model.mod

We then build a new machine learning models, where we only include the above mentioned important features as input features and as a result, we see that the result for boosting based models (AdaBoost and XGBoost) are the same however, the other models specially non tree based models performance get better as shown in the following [Table 4].

Model	**Test Accuracy (Image + Structured)**
BernoulliNB	0.623
KNeighborsClassifier	0.563
LogisticRegression	0.657
MLPClassifier	0.645
DecisionTreeClassifier	0.605
RandomForestClassifier	0.692
AdaBoostClassifier	0.692
XGBClassifier	0.705

Table 4: Final Model Performance for Votes Prediction

5 Model Building and Evaluation - Quality Prediction

For the quality prediction task, we use IMDb score as the target variable. Similar to popularity prediction, we convert the IMDb score into three classes: High, Medium and Low based on 33% and 67% percentile.

We built another multi-modal model that combines all the structured, text and image features, and trained the standardized data on the following models.

Model	**Image + Structured**	**Image + Summary + Structured**
BernoulliNB	0.423	0.376
KNeighborsClassifier	0.376	0.371
LogisticRegression	0.497	0.497
MLPClassifier	0.474	0.484
DecisionTreeClassifier	0.461	0.461
RandomForestClassifier	0.501	0.471
AdaBoostClassifier	0.552	0.528
XGBClassifier	0.566	0.549
Neural Network	0.484	0.492

Table 5: Model Performance for Score Prediction

Again XGBoost turns out to be the best performing model with image + structured data features. The following graph [Figure 15] shows the confusion matrix of the XGBoost score prediction model. The high class and low class have a precision and recall around 60%, but for medium class the precision and recall are around 45%, so the model tends to misclassify the medium class. Generally, the wrongly predicted labels are only one class away from the true label.

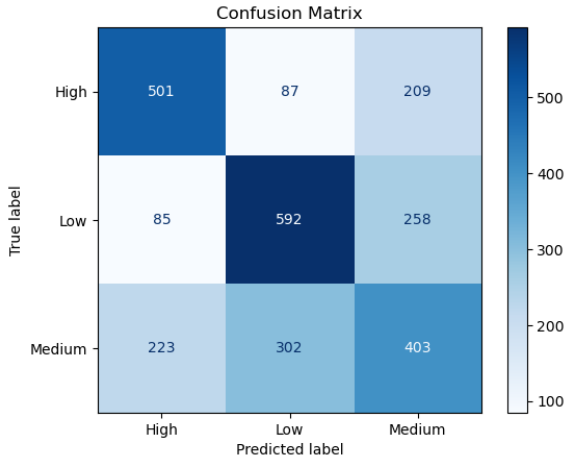

Figure 15: XGBoost Model Confusion Matrix

We then check the feature importance to see which features contribute most to the movie quality. From the following plot [Figure 16], we see that Director is the most important feature in predicting quality, followed by Awards received, Genre, Series or Movie and some image features from posters.

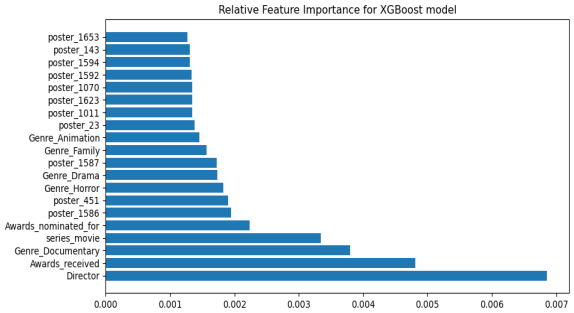

Figure 16: XGBoost Model Important Features

We then build a new model that only takes the above mentioned features as input and trained on the following models. As a result, all models' accuracy improved, especially for non-tree-based models. We then fine-tuned the best performing XGBoost model, and achieved a test accuracy of 58.53%.

Model	**Test Accuracy**
BernoulliNB	0.539
KNeighborsClassifier	0.474
LogisticRegression	0.565
MLPClassifier	0.534
DecisionTreeClassifier	0.473

RandomForestClassifier	0.571
AdaBoostClassifier	0.578
XGBClassifier	0.585

Table 6: Final Model Performance for Score Prediction

6 End-to-end ML Pipeline

We built a Flask web app that enables real-time prediction of movie popularity and quality. It takes in movie information that analysts want to query about, including movie genre, poster, and director, etc. Upon receiving user requests, our website will automatically process the inputs and generate popularity and quality predictions based on the learned classification models [Figure 17]. Accordingly, Netflix could refer to the provided results to support their procurement (for buying distribution rights) decisions.

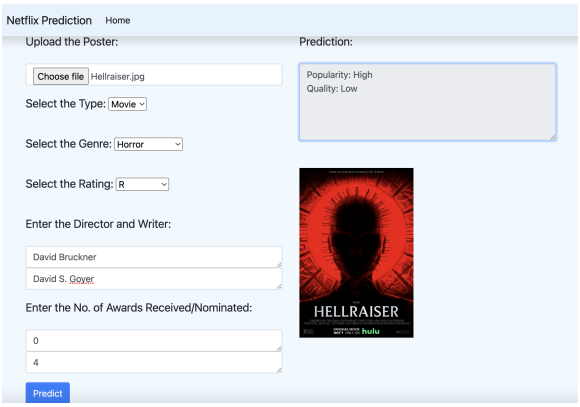

Figure 17 Low Quality and High Popularity Predicted (*https://github.com/azuretime/Netflix-Prediction*)

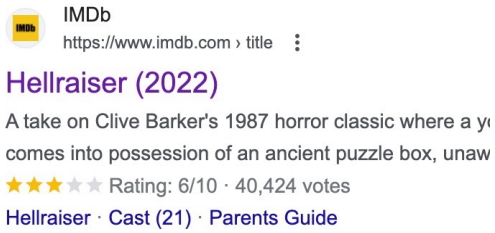

Figure 18 Actual Quality and Popularity

As the prediction results for the movie *Hellraiser* matches the actuality score and votes ranges as seen in Figure 18 and Table 7. We successfully predicted the quality and popularity of the movie.

Cutoff Points	Low	Medium	High
Score	< 6.1	6.1 - 7.1	> 7.1
Votes	< 743	743 - 9323	> 9323

Table 7 Quality and Popularity Cutoff Points

7 Conclusion and Business Improvement

In this project, we are trying to use Machine Learning, NLP, and Deep Learning models to understand and predict the movie score and votes in IMDB. The conclusions that we gain through this project are shown below:

(1) In the votes prediction part, View Rating, Awards nominated and received, Genre, Writer, Director, Series or Movies and image features from posters are the most important features.
(2) In the score prediction part, Director is the most important feature, followed by Awards received, Genre, and image features from posters.
(3) In the trial of classification with text features, the results show that summary may have little impact on viewers' rating and tendency to view the movie.
(4) For both prediction tasks, among all of classification models and Neural Network, XGBoost can achieve highest accuracy. Our final prediction model is the XGBoost model with inputs of both basic features and image features.

Based on the data from trial performance, Netflix can pay more emphasis on certain characteristics of the projects, such as Genre, Writer, Director, Poster features, etc. With our algorithm predicting whether it will gain high ratings and popularity among viewers, it can guide the selection of future projects for the platform.

1) If the projects belong to other production companies, Netflix can purchase the distribution rights as soon as possible to save costs.
2) If the projects belong to Netflix Original, Netflix can then increase the budget on marketing and advertising.

This will help to increase the visibility and exposure of the movies on the platform, potentially leading to higher ratings and greater popularity.

For creating its own movies, Netflix can:

1) cooperate more with famous directors and writers to create compelling and well-written storylines. This will help to ensure that the movies are of high quality and engage viewers.
2) explore more on poster features and design the poster in a way that can arouse the audience's interest most. This can include utilizing image features from posters that were found to be important predictors in the project, as well as other design elements that are known to be effective in catching the viewer's attention.

Overall, these strategies should be centered around identifying the important features for predicting high ratings and popularity, as well as investing in high-quality production, marketing, and advertising to increase visibility and engagement among viewers.

Reference

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[2] Team, K. (n.d.). *Keras Documentation: Inceptionv3*. Keras. Retrieved April 23, 2023, from https://keras.io/api/applications/inceptionv3/

[3] rz0718. (n.d.). *RZ0718/spam_detection*. Spam Detection. Retrieved April 23, 2023, from https://github.com/rz0718/spam_detection