Prediction of Movie Gross using Regression Model from Movie Industry Datasets

Project Report for DATA 1030, Fall 2020 at Brown University

Supervised by **Prof. Andras Zsom**

Yurui Zhang

https://github.com/yuruizhang9734/Brown_DATA1030_MovieProject_Fall2021.git

1 Introduction

a. Motivation

Last year, due to the pandemic caused by covid-19 around the world, there was a huge impact on the traditional movie industries. On the other hand, it was a great chance for the development for streaming media like Netflix and Hulu. The Los Angeles box office, a key movie market and local economic backbone, was projected to fall by 20% in April 2020 compared to its 2019 figures.[1] In order to protect the traditional movie industries from unexpected impacts, a model which can predict the movie gross based on the other features is essential. The model can help the movie companies to predict how much a movie can make and when is the best time to release the movie based on the different unique characteristics of the movie itself. Even more, this model can help the new movie companies to have a clue about how to make popular movies in both reputation and business.

b. Simple cleaning process and feature description

The dataset used for this project came from Kaggle datasets called Movie Industry. There are a total of 7668 data points and 15 features. [2] The target variable is the gross variable for a movie which is in $ US dollars. Since the target variable is continuous, this problem is a regression problem. However, there are 189 missing values in gross which is our target variable. After dropping the missing value in target value, a simple data cleaning process needed to extract data from some text features.

Features Names	Explanation	Data Type
Name_Length	The length of the name for the movie	Continuous Measures the length of movie names.
Rating	Rating of the movie (R, PG, etc.), after replacing “Not Rated” and “Unrated” with NaN	Categorical 10 levels
Genre	Main genre of the movie.	Categorical 18 levels
Released_year	Year of release	Categorical 40 levels
Released_month	Month of release	Categorical 12 levels
Released_weekday	Weekday of release	Categorical 7 levels
Score	IMDb user rating between 0 to 10.	Continuous Measures the rating of movie
Votes	Number of user votes.	Continuous Unit: times
Director	Name of the director.	Categorical 11 levels
Writer	Name of the writer.	Categorical 11 levels
Star	Name of the star.	Categorical 11 levels
Country	Country of origin	Categorical 11 levels
Budget	The budget of a movie.	Continuous Unit: US dollars
Company	The production company.	Categorical 11 levels
Runtime	Duration of the movie.	Continuous Unit: minitues

Figure 1. This table explains the features after simple cleaning and their types and units

After cleaning the data, there are 15 features left and a target variable.

c. Previous projects and results

Several authors have applied several cluster analysis over the movies datasets. Their purposes were to classify the movies into clusters in order to see the nature of movie clusters in the industry. [3] Most of them used K-nearest- neighbors algorithms to perform the analysis. There is also another project which performs a regression onto the Budget variable. [4] The purpose of that study is trying to predict the planned cost (budget) based on the characteristics of the production. After choosing from the linear regression, Deep Learning (keras) and Gradient

Boosting Regression, the author got an overall 56% score in predicting the budget using the linear regression.

2 Exploratory Data Analysis

a. Categorical Variable vs Target Variable

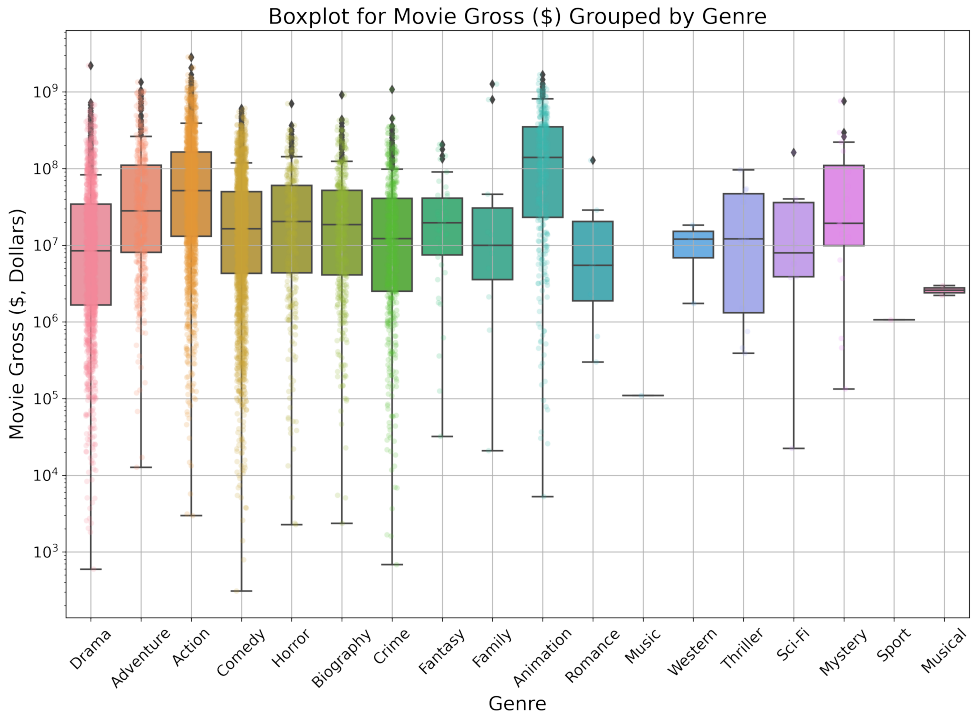

Figure 2. This figure displays the box plot of movie gross in dollars which is our target variable, grouped by genre. The y label is scaled by log due to the huge difference in values of gross. From the plot, it is obvious that the movie with genre animation has the highest median and Q3 gross. Given the difference between the genres, this feature has the potential to play an important role in the machine learning model.

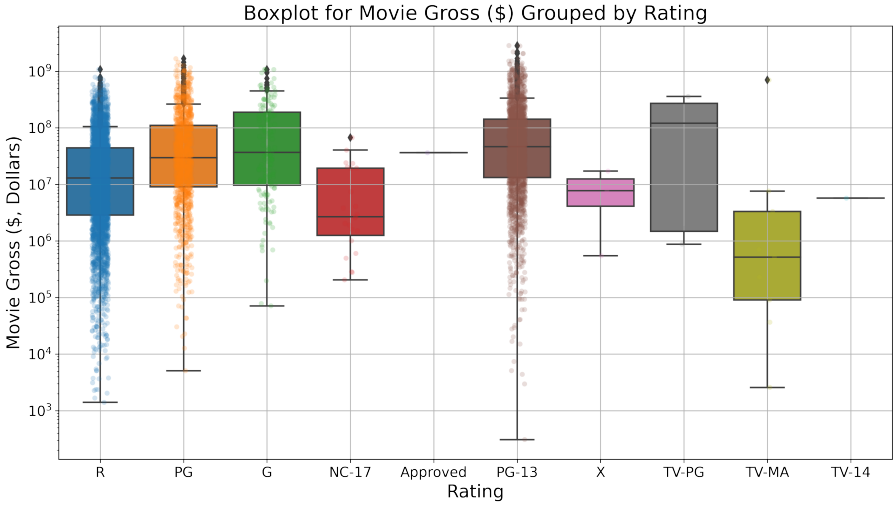

Figure 3. This figure displays the box plot of movie gross in dollars which is our target variable, grouped by rating. The y label is also scaled by log due to the huge difference in values of gross. From the plot, it is obvious that the TV-PG rating has the highest median movie gross and G has the highest Q3. The reason why the Approved and TV-14 rating are two very thin bars because they have too few data points. The different ratings have much differences in move gross. This variable has potential to play important role in the regression model.

b. Continuous Variable vs Target Variable

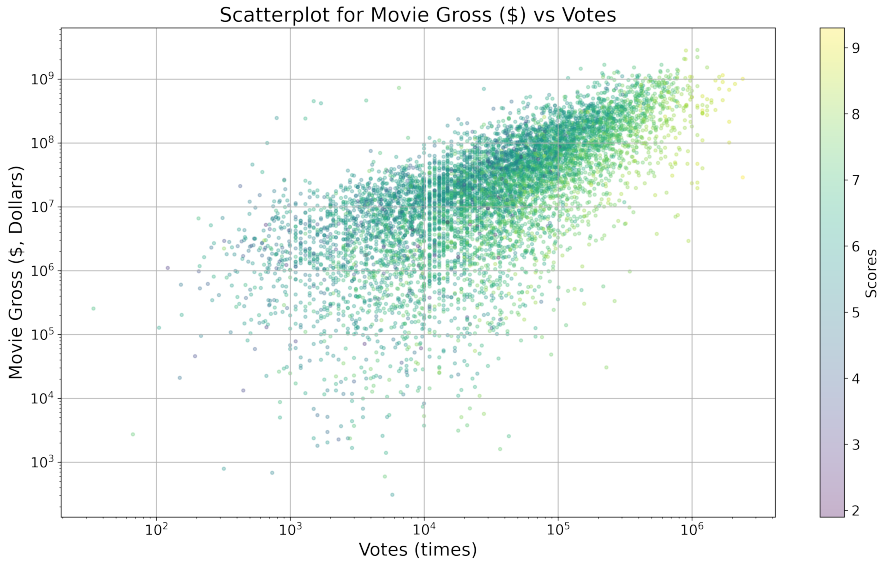

Figure 4. This figure displays the scatter plot of movie gross ($) as y and Votes (vote times on IMDB) as x axis, and add in the scores as color bar. Both the y label and x label are scaled by log. From the plot, it seems that there is a linear relationship between number of votes and movie gross. Also, the higher score movies seems have a more steep linear relationship between Votes and Gross. Based on the plot, these features have the potential to play an important role in the machine learning model

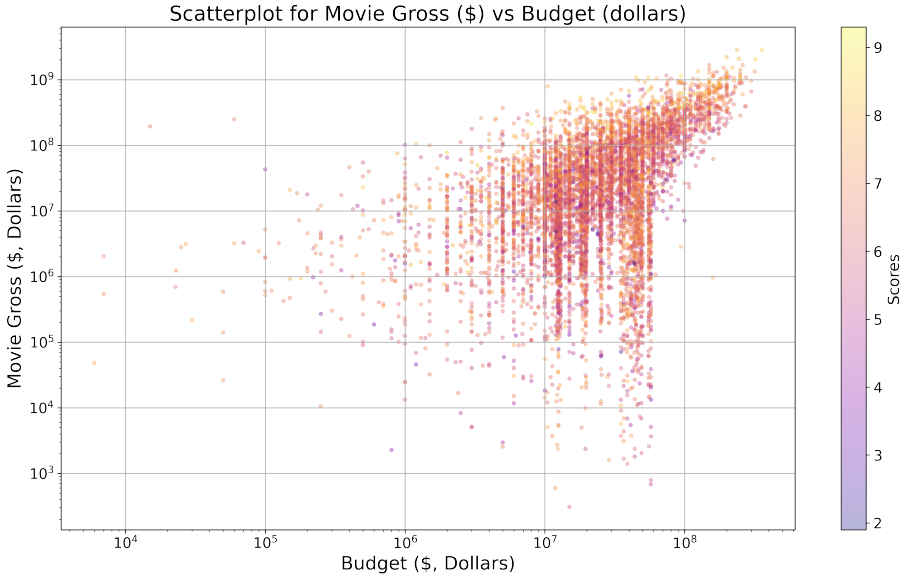

Figure 5. This figure displays the scatter plot of movie gross in dollars as y and budget in dollars as x axis, Both the y label and x label are scaled by log. From the plot, it seems that there is a linear relationship between budget and movie gross. Lower score movies seems have a more steep linear relationship between Budget and Gross since the light color points are floating above the dark color points. Based on the plot, this feature has the potential to play an important role in the machine learning model.

c. Time Variable vs Target Variable

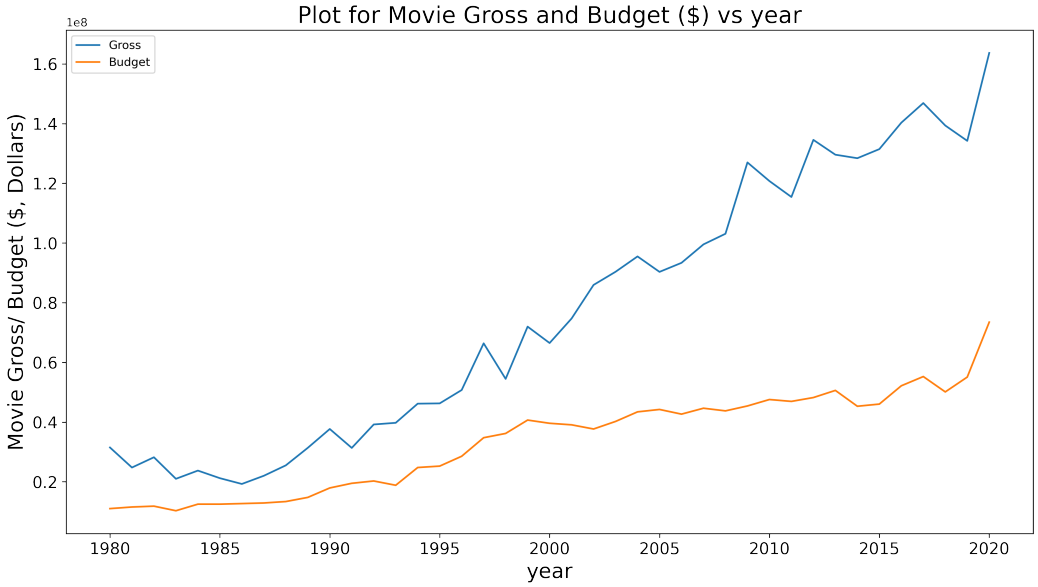

Figure 6. This figure displays the line plot of the grow of movie gross and budgets in dollars each year from 1980 to 2020. From the plot, there is an obvious linear relationship between movie gross each year and year. And there is a linear relationship between budgets and movie gross. As time passed, the profit of Movies have generated more, this might be due to the more popular demand and the effect of social media on propaganda. Based on the plot, these features have the potential to play an important role in the machine learning model.

3 Methods

a. Data Splitting and Preprocessing

First, the datasets are not time series models since each data point is a different movie that measures gross after box office debut at a fixed time, the movie will not change over time. Second, the datasets are not group structure because each data point is a unique and separate independent movie. In order to split the datasets into train datasets, validation datasets and test datasets, we first split 20% of the observations to testing and then use the 5 fold cross-validation onto the rest 80% observations. During the 5 fold cross-validation process, first using the processing pipeline to fit transform onto the four folds used for training sets, then apply

transform onto the one fold validation set and the test set. The 5 fold cross-validation can better use the datasets and gain several score while using different validation sets.

While preprocessing the datasets, one hot encoding technique is applied on features: rating, genre , director , writer , star , country , company , released_year , released_month , released_weekday, because all of them are categorical variables. Then standard scaler technique is applied on votes, budget, name_length because they are continuous variable with tail distribution. Last, min_max scaler technique is applied on variable score , runtime because both are continuous variable and score is between 0 to 10, and runtime is between 0 to 400, both with no tail distribution. There is also SimpleImputer to take care of missing value in categorical values and there is only one missing value in continuous value, so I dropped the row. As a result, after preprocessing, there are 4785 rows in training, 1197 in validation and 1496 in test set and 154 features.

b. Model Selection

After applying the splitting and preprocessing pipeline, I have trained and compared four totally different machine learning models with one baseline model and compared across them: Ridge regression, Random Forest regression, K-nearest neighbors regression and XGBoost regression. All of the five machine learning models have been hyper-parameter tuned by GridSearchCV over a parameter grid for the best parameter combination. Since it is a regression model and I want to know how close my model fit, I used RMSE to evaluate. Also, I used ten different random states to do 10 different 5-folds CV to lower uncertainties due to splitting. Here are the models and parameter grid I tried.

Models	Parameters	Parameter Meaning	Best Parameters
Ridge	alpha: np.logspace(-10, 3, 51); max_iter: 100000000	alpha:Regularization strength;	alpha=15.1356
Random Forest	max_depth: [1, 10, 20, 40, 60, 100, 140, 150]; max_features: [0.2,0.4,0.5,0.6,0.8,1.0]	max_depth:max depth of the tree max_features: fraction of features in each split	max_depth=100 max_features=0.4
KNN Regressor	p: [1, 10,20,60,100,140]; n_neighbors: [1, 3, 10, 20, 50, 100]	p: Power parameter for the Minkowski metric n_neighbors: number of neighbors	p= n_neighbors=
XGBoost	learning_rate: [0.03]; max_depth: [1, 2, 6, 10]; subsample: [0.66]	Learning rate shrinks the contribution of each tree Max depth of each estimator	max_depth=6
BaseLine	no parameter	-	-

Figure 7. This tables contains all of the models and the parameters I tried for each different model

After picking the best parameters for each random state, I stored the all RMSE scores and models. Here is the plot of the mean of RMSE values for the best of each model across the ten random states. The Baseline model here I used is taking the mean of y of training data and used as our prediction for y test.

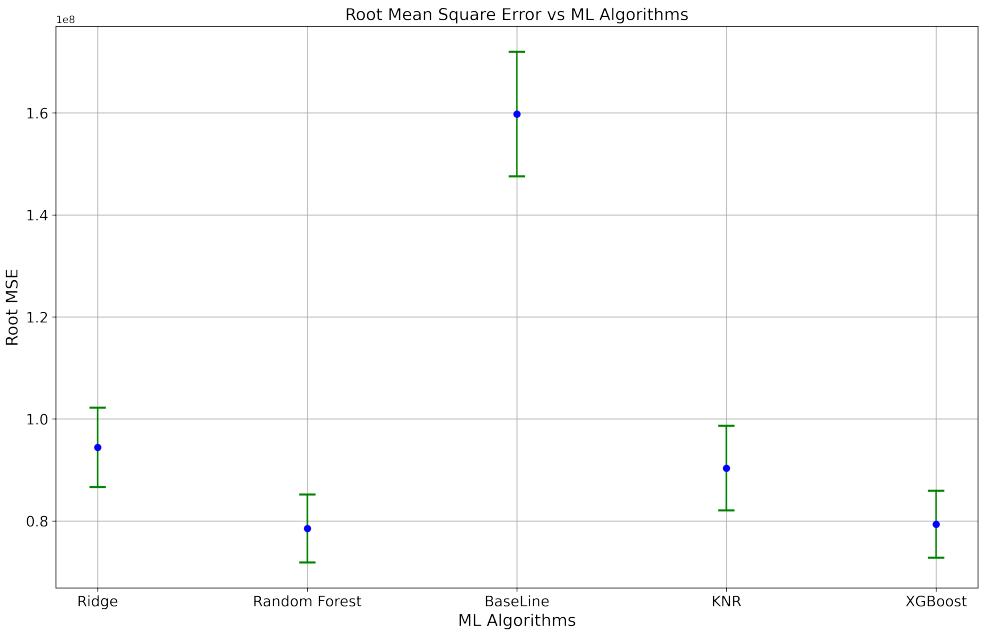

Figure 8. Mean of RMSE scores for the best model over ten random states with Baseline

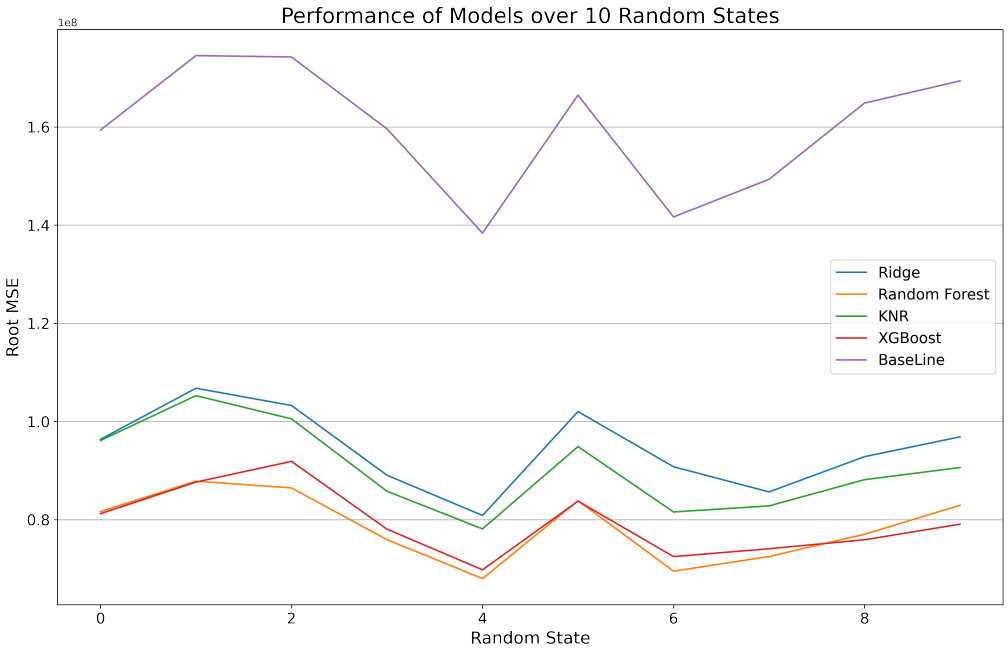

Figure 9. RMSE scores for the best model over ten random states to show sensitivities and uncertainties in the model to splitting

From the two plots above, we can see that the Random Forest model achieved the minimum Root MSE which means it has the highest accuracy. Also, for different random states, the best model is different, but mostly RF. After looking at the best parameters over the 10 random states, we chose max_depth as 100 and max_features as 0.4.

c. Final Model Formulation

Now the best model and matching parameters are ready, so we retrained random forest models with 50 new splits over 50 different random states. For each split, 80% of the datasets were assigned to the training datasets and 20% were assigned to test datasets. For every split, we recored the baseline score and model score.

4 Results

a. Evaluation of Models

Among these 50 random states, the average RMSE for the baseline models is 164717974.42342967 and the standard deviation of the RMSE for the baseline models is 11670558.738955721. And for the trained random forest models have the average RMSE as 80823011.15508816 and the standard deviation of the RMSE for the random forest models is 7668682.711652616. The trained models has RMSE that is 7.2 standard deviations below baseline. Similarly, the baseline model's RMSE was 10.9 standard deviations below the average of the trained models.

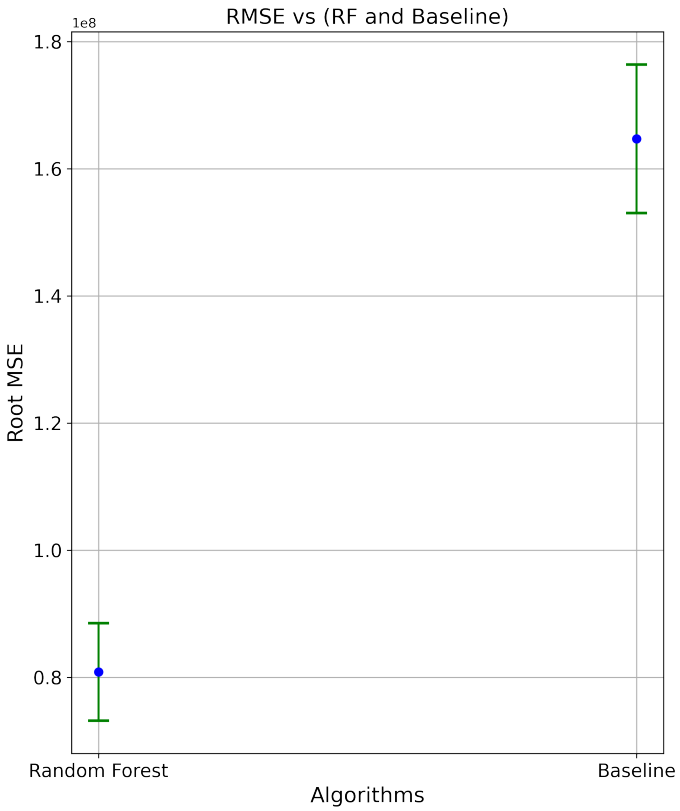

Figure 10. Mean of RMSE scores for the Random Forest vs Baseline

b. Interpretation of Findings

The Global Feature importance for the random forest model was calculated using both the random forest feature importance and the permutation importance. Here are the results plots.

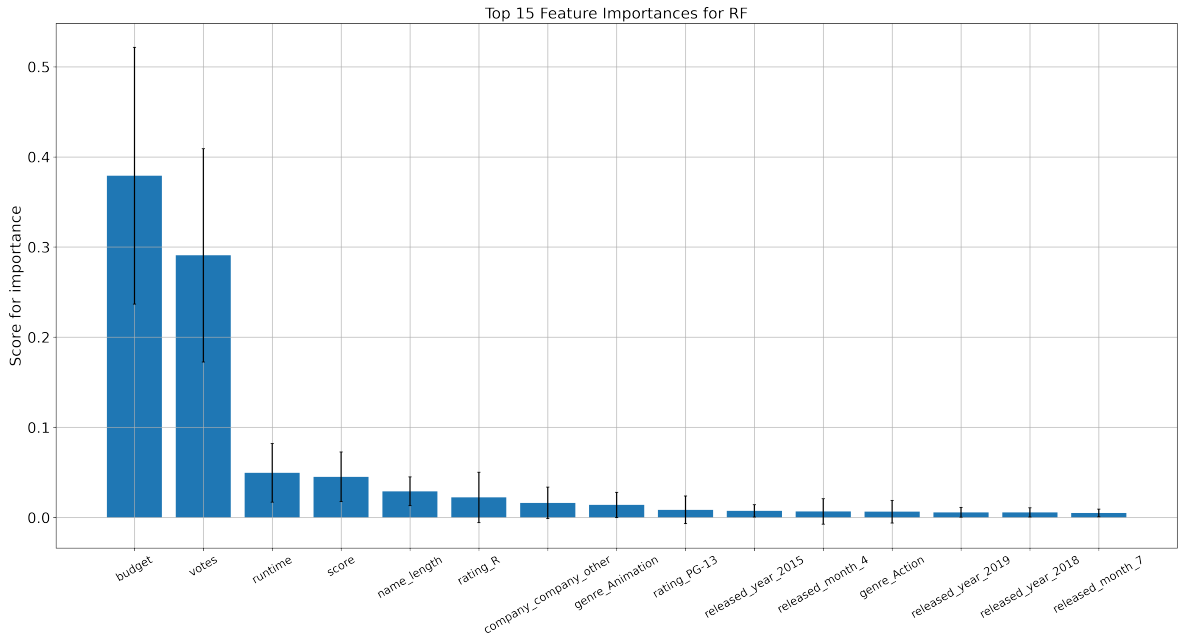

Figure 11. Most important 15 features from the Random Forest Model

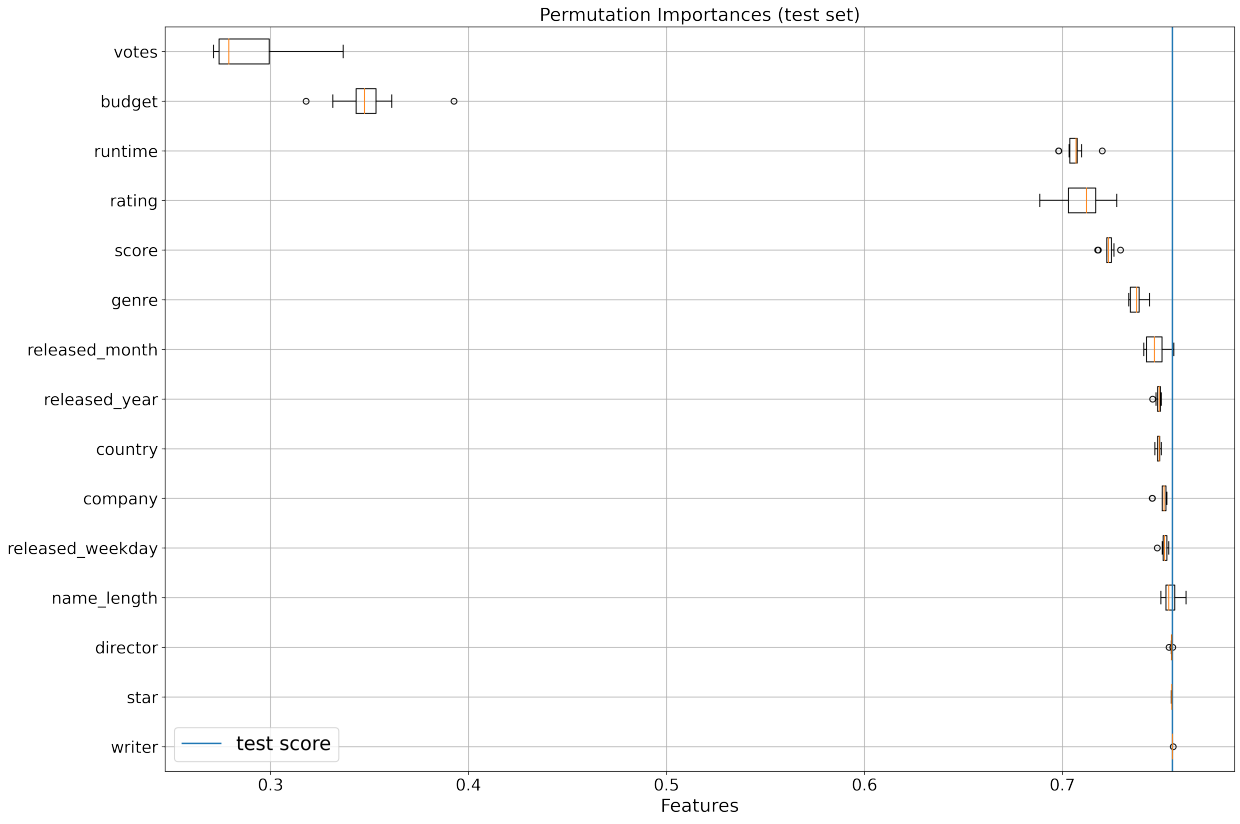

Figure 12. Permutation Importances for features from the Random Forest Model

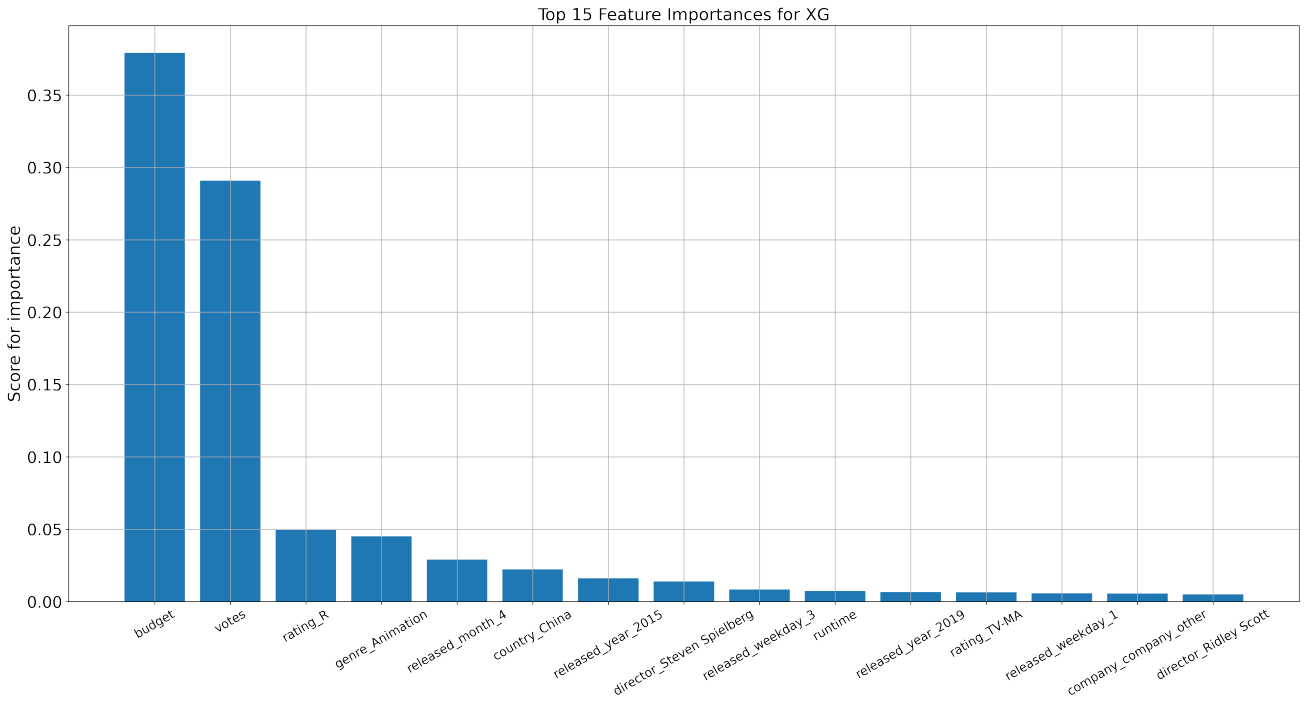

Figure 13. Feature Importances for features from the XGBoost Model

From the above plots, the two most important features from all three global feature importance are budget and votes, but the votes are more important than budgets as mentioned in permutation importance. The reason is random forest feature importance is impurity-based feature importance which can inflate the importance of numerical features such as budget. Also, by comparing the results from feature importance of RF and XGboost, rating_R is more important in the XGBoost model than that in Random forest model. Also, director, writer and star are the least important from permutation importance.

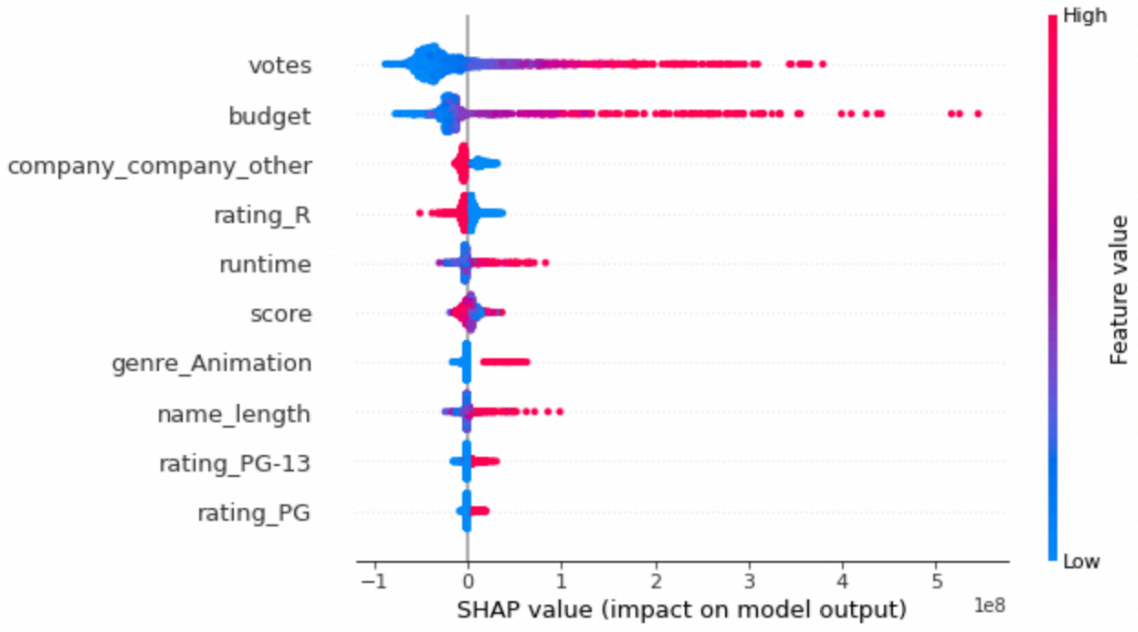

Figure 14. SHAP value for top 10 features from the Random Forest Model

Now, using the SHAP value to take a look at the local importance.

Figure 15. SHAP value for local importance for the 100th row

Here are two force plot visualizing shapley values for the features. For the 100th data, feature values votes, runtime increase the prediction and feature values budget, company_company_other decreases the prediction. Among all, the biggest effect is budget and it has decreased the predicted movie gross.

5 Outlook

The first point that can be improved about the model is that we have not employed the feature engineering part which can group or modify the features. For the next time, I will group the movie score and votes because they shows a correlation in EDA. The second part is that during the first process of cleaning data, there are many different levels in star, writer and director. What we did was just picked the 10 most common levels in these three features and combine the rest levels into one called other. We can improve the data by choosing another better way to clean it. For additional datasets, we can collect more features that can analyze the content of the move more quantitive such as the sum of actor's pay, the genre of the most popular movie in that year.

6 Reference

[1] En.wikipedia.org. 2021. *Impact of the COVID-19 pandemic on cinema - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/

[2] Grijalva, D. G. (n.d.). Movie Industry Four decades of movies. Retrieved from https://www.kaggle.com/danielgrijalvas/movies.

[3] MOZAHEM, N. (2021, September 28). Movie Industry Cluster Analysis. Retrieved from https://www.kaggle.com/najibmozahem/movie-industry-cluster-analysis

[4] FERREIRA, A. (n.d.). Retrieved from https://www.kaggle.com/alanhenriqueferreira/budget-prediction/notebook#5—Training

GitHub Repository: https://github.com/yuruizhang9734/Brown_DATA1030_MovieProject_Fall2021.git