Predicting Movie Revenues

DS 5110 Course Project

Yiyang Jiang, Jinxuan Zhang, Yu-Hsuan Lin
Group 6
Northeastern University

I. SUMMARY

In today's fast-paced social life, more and more people will go to the cinema to watch movies to relax themselves. According to statistics, thousands of movies are released every year around the world, and the average production cycle of each movie takes more than two years. Therefore, whether each film can be profitable after its release is the most concerning issue of every film producer.

This project could be divided into two problems:

Problem 1. We will predict the effect of different factors on the commercial value of a movie. For example, the genre, the votes, the duration, the release date…etc.

Problem 2. Famous actors in a film production will have high appearance fees but will attract the audience to watch and thus increase the movie gross. However, not the more famous actors hired, the higher revenue will be. In our second problem, we will examine the relationship between the number of famous actors and the movie revenues, while predicting how many famous actors should be hired to maximize movie gross.

II. METHODS

2.1 DATA PREPROCESSING

We first converted the formatting of the **credits.csv** file which includes the information about the cast name and movie id, because although it is a **csv** file, the actual storage format of the data inside it is **json**, we first used a loop to iterate through each group of json data and put it into a list, then created a new dataframe and imported the array in the list into the dataframe, so that we completed the conversion from json, format to dataframe format.

Then we processed **movies_metadata.csv** file. We first kept only these column names: **budget**, **genres**, **title**, **revenue**, **movie_id** in our dataframe. For the **genres** column, we first stored all the unique genre names into an array and created new columns of these unique genres in our dataframe. Then we use a loop to iterate through each movie to convert the genre labels from variable names to numeric form, with a "1" indicating that the genre label is present and a "0" indicating that it is not.

Finally, we cleaned the **budget** and **revenue** columns by first converting the values of these two columns from string to integer and then removing all rows with 0 or null values.

2.2 PROBLEM 1

In this section, we present and compare the models that will be used in solving the first problem.

1. Linear Regression

Our first model is Linear Regression, traditionally and commonly used in various fields for prediction. Specifically, it attempts to model the relationship between multiple explanatory variables and a response variable by fitting a linear equation. The population regression line for n explanatory variables $x_1, x_2, \cdots, x_n$ Is defined below. This line then describes how the mean response changes with the explanatory variables.

$$\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

We will utilize the implementation of LinearRegression in the scikit-learn library. Since the model is relatively simple and easy-to-interpret, it could serve as a baseline model that provides preliminary insights and a reasonable benchmark to improve upon.

2. Random Forest

A Random Forest estimator fits a number of decision trees on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. Denoting $S := (x_1, y_1), \cdots, (x_n, y_n)$ as our training set, F as the set of features, and B as the number of trees in forest, the pseudocode for a random forest algorithm is illustrated in Algorithm 1.

The advantage of Random Forest is that, based on the bagging algorithm and the use of ensemble learning technique, it mitigates the problem of overfitting in decision trees and significantly reduces the variance, leading to an improvement in accuracy. In this project, we will utilize the

implementation of RandomForestRegressor in the scikit-learn library. Hyperparameter tuning will be subsequently performed using GridSearchCV and RandomizedSearchCV.

Algorithm 1 Random Forest

function RANDOMFOREST ($S, \mathcal{F}$)
  $H \leftarrow \emptyset$
  for i $\in 1, \ldots, \mathcal{B}$ **do**
    $S^{(i)} \leftarrow$ a bootstrap sample from S
    $h_i \leftarrow$ RANDOMIZEDTREELEARN ($S^{(i)}, \mathcal{F}$)
    $H \leftarrow H \cup \{h_i\}$
  end for
  return H
end function
function RANDOMIZEDTREELEARN ($S, \mathcal{F}$)
  At each node:
    $f \leftarrow$ small subset of $\mathcal{F}$
    Split on best feature in f
  return learned tree
end function

2.3 PROBLEM 2

In Problem 2, we first visualize the relationship between the number of times all actors act in a movie and the number of actors. We then determine which actors are famous based on their influence. Then we tried single linear regression, but due to the large data difference, the performance of single linear regression was not good, so we chose polynomial regression.

Polynomial regression is a form of regression analysis in which the relationship between number of famous actors and profit is modeled as a polynomial of degree n with respect to number of famous actors. Polynomial regression fits the nonlinear relationship between number of famous actors and the corresponding conditional mean of profit. Polynomial regression more accurately describes the trend of the relationship between number of famous actors and profit.

In the process of polynomial regression, we first sort the data according to the number of famous actors, then use PolynomialFeatures() and poly_reg.fit_transform() to process the number of famous actors, and finally use linear_model.LinearRegression for linear regression.

III. RESULTS

3.1 PROBLEM 1

Table 1: *Model Performance Summary*

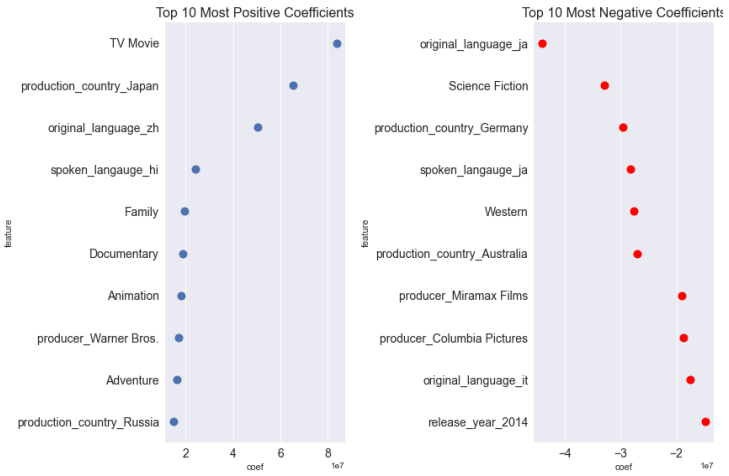
Model	MSE	RMSE	MAE	R^2
LINEAR REGRESSION	6.088180e+15	7.802679e+07	4.601996e+07	0.682504
RF BASELINE	4.903117e+15	7.002226e+07	3.763279e+07	0.744304
RF WITH NORMALIZATION	4.877961e+15	6.984240e+07	3.754272e+07	0.745616
RF WITH GRIDSEARCHCV	4.369570e+15	6.610273e+07	3.688613e+07	0.772129
RF WITH RANDOMIZED SEARCHCV	4.389555e+15	6.625371e+07	3.677771e+07	0.771086

In the previous section, the modeling toolbox utilized in the development of predictive models was presented. In this section, we present the final results and performance of the models fitted.

As a first attempt, we fit the Linear Regression model on all 83 features (after preprocessing and one hot encoding). As shown in the first row of Table 1, this gives poor results, with an RMSE and R2 of 7.8e+07 and 0.68, respectively. This illustrates some of the main disadvantages of Linear Regression, especially how it fails to capture more complex, possibly non-linear relationships. Additionally, the presence of multicollinearity between features also creates problems.

In addition to the model's prediction power, we would also like to know what features contribute the most in predicting movie revenues. Figure 1 shows the top 10 most positive and most negative coefficients based on the fitted model. It seems that, on average, revenue is higher by approximately 80 million for movies of the genre TV Movie than for other genres, holding all other variables constant. Other factors that have a positive effect on movie revenue include whether the movie is produced in Japan and whether the original language of the movie is Chinese or Hindi. Common genres such as Family, Documentary, and Animation also have a somewhat positive effect on increasing movie revenue.

Figure 1: *Most Positive and Negative Coefficients in the Linear Regression Model*



Interestingly, while the production country of Japan has the second most positive coefficient, the Japanese language has the most negative coefficient based on the model. A possible explanation might be that people enjoy translated Japanese movies.

Random Forest, on the other hand, is naturally capable of handling a high dimensional set of variables. The results of fitting a Random Forest model on all features are shown in the second row of Table 1. It could be observed that, without any hyperparameter tuning, the performance of Random Forest already surpasses that of the Linear Regression model. However, the advantages of Random Forest come at a cost of longer computational time and lower interpretability. We then fit the model again on normalized data using MinMaxScaler, which did not yield any improvements as shown in the third row of Table 1.

Subsequently, the Random Forest model was calibrated by tuning the following hyperparameters: n_estimators, max_depth, min_samples_split, and max_features. We now examine the performance of the model again, using the four best parameters determined by GridSearchCV and RandomizedSearchCV, as shown in the fourth and fifth row of Table 1, respectively. While there is no significant difference between the results of Grid Search and Randomized Search CV, all four metrics of performance exceed that of the previous model using the default values of hyperparameters by quite a large margin. This illustrates the critical importance of hyperparameter tuning in further improving model performance.

Figure 2 shows the most important features based on the Random Forest model. The impurity-based feature importance is computed by measuring how effective the feature is at reducing uncertainty when creating decision trees. As we can see, **vote counts** appear to be most important in explaining movie revenue, followed by **budget**, **popularity**, **runtime**, and **vote average**. The results are consistent with the conclusions from the correlation analysis between revenue and features, as shown in Figure 3. The budget and vote count / average are indeed critical features that are highly correlated with movie revenue and are very important in contributing to the prediction. Additional plots are included in Appendix 2-4.

Figure 2: *Top 10 Most Important Features based on Random Forest*

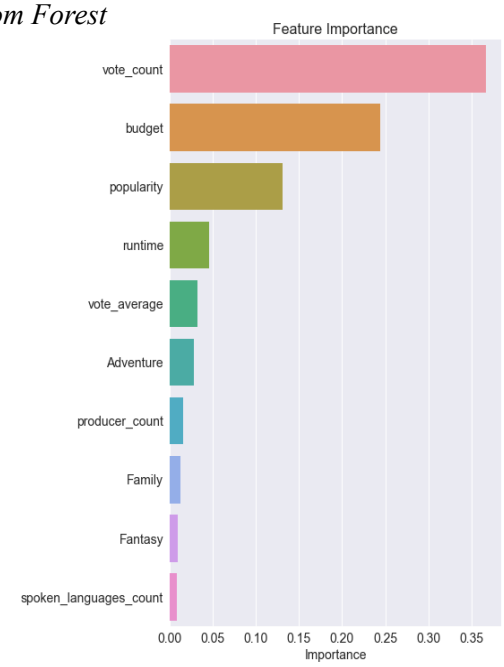
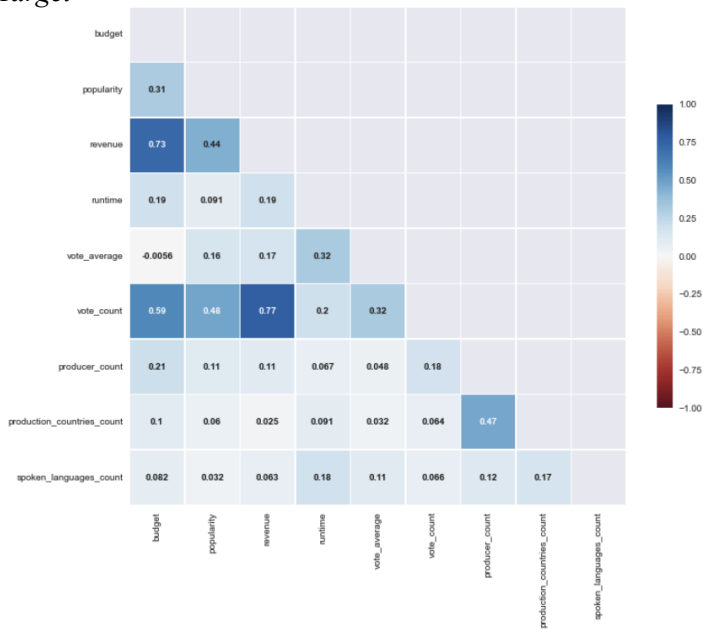


Figure 3: *Correlation Heatmap Between Features and Target*

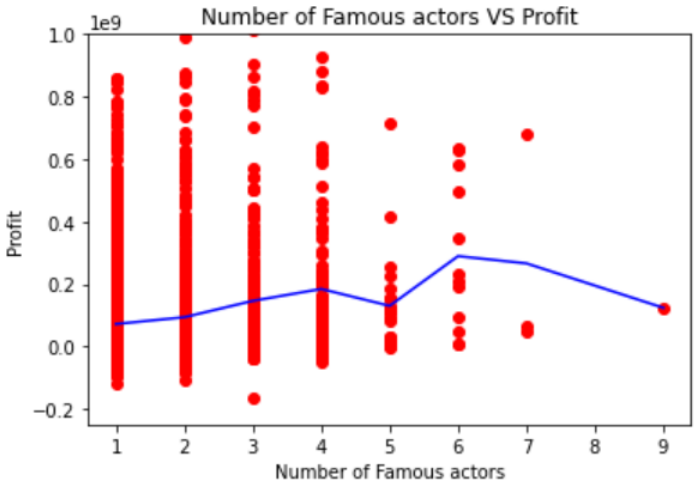


In summary, Random Forest outperforms Linear Regression, with a higher R^2 and lower MSE, RMSE, MAE. The best performance is achieved with Random Forest after data is normalized and hyperparameters are tuned. Both

GridSearchCV and RandomizedSearchCV produced similar results.

3.2 PROBLEM 2

After data denoising and min-max normalization, and Polynomial regression with some deviations. However, from the images after data analysis, we can see that from one famous actor to four famous actors, the movie profits are steadily rising. Films with five famous actors were bottlenecked, but films with six famous actors were the most profitable.



IV. DISCUSSION

The main purpose of our project is to use the information in the existing movie database to mine the relationship between movie gross and other variables. We believe that the most judging indicator of a film's success in the film industry is its commercial value, so our project predicts what factors will affect the commercial value of a movie in the past and maximizes it.

In Problem 1, comparisons between the proposed models were performed, with an emphasis on the differences between the traditionally adopted Linear Regression model and the relatively advanced Random Forest model. The poor performance of Linear Regression highlights its disadvantage when dealing with a complex, high dimensional dataset with the presence of multicollinearity and nonlinearity. We conclude that the best performance in predicting movie revenues is achieved with Random Forest after normalization and hyperparameter tuning, and the most important factors in affecting revenue are vote counts (and average), budget, popularity, and runtime.

For the second question, we can conclude that when each movie invites 6 famous actors who have participated in movies more than 30 times in the past, the profit of this movie will theoretically reach the maximum.

I think the producer of the film may benefit from referring to our findings because they can spend their budget on variables that have a greater impact on profits in future film production. There is also a benefit for the investors of the films, who can refer to our findings and invest in films that match our findings and are still being made, because these films with great potential may bring more profit for the investors.

V. STATEMENT OF CONTRIBUTIONS

Yiyang Jiang: Present the definition of the second problem and the definition of the famous actor, clean the raw data: convert the genre labels from variable names to numeric form, cleaned the budget and revenue columns by first converting the values of these two columns from string to integer and then removing all rows with 0 or null values. Merge all columns into a dataframe that can be finally analyzed.

Yu-Hsuan Lin: Data cleaning and processing for problem 1; EDA and visualizations; correlation analysis of features; model fitting for problem 1; feature importance analysis

Jinxuan Zhang: Read cast data from json and pair them with movies, data visualization, analyze the relationship between the number of famous actors and movie profits, Polynomial regression for number of famous actors and profit in a movie.

VI. REFERENCES

https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

Pseudocode for Random Forest algorithm:
https://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf

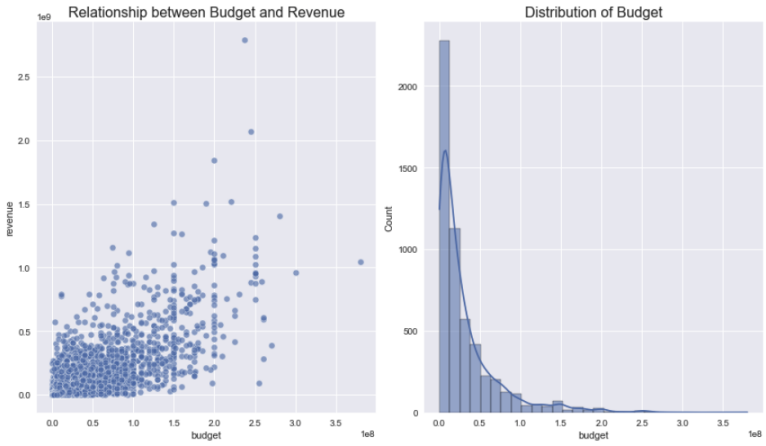
VII. APPENDIX

Appendix 1: Polynomial regression for number of famous actors and profit in a movie

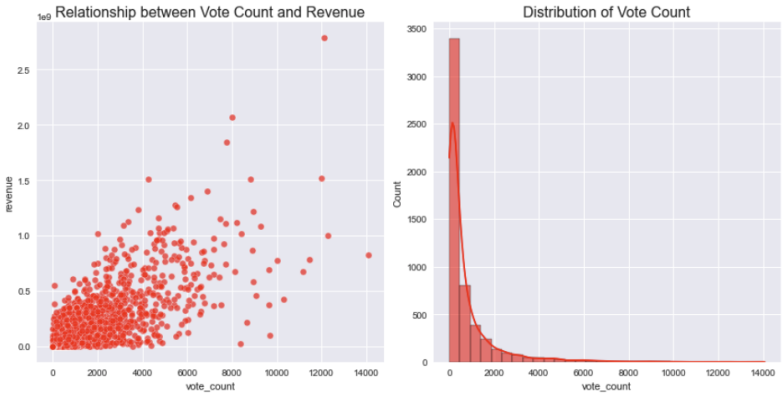
```
X = list(df_movie_revenue_budget_cast['num_famous_actors'])
y = list(df_movie_revenue_budget_cast['profit'])
d = {y[i]:X[i] for i in range(len(X))}
sorted_d = sorted(d.items(), key = lambda x: x[1])
X = []
y = []
for p in sorted_d:
    X.append(p[1])
    y.append(p[0])
X = np.array(X).reshape(-1,1)
```

```
#fitting the polynomial regression model to the dataset
poly_reg=PolynomialFeatures(degree=10)
X_poly=poly_reg.fit_transform(X)
poly_reg.fit(X_poly,y)
lin_reg2=linear_model.LinearRegression()
lin_reg2.fit(X_poly,y)
```

Appendix 2: Relationship between Budget and Revenue



Appendix 3: Relationship between Vote Count and Revenue



Appendix 4: Relationship between Vote Average and Revenue