Predicting Movie Revenue by Implementing Linear Regression
Machine Learning Project Report

Group Members:
Juan Farell Haryanto - 2301855072
Justin Mudita Kristian - 2301860854
William Sunjaya - 2301851925

1. Introduction

Watching a movie with the family is a very common recreational activity for people who live in most cities. The development of the film industry has prospered throughout the centuries, and some disruptions have caused the change of the business field of the movie industry. In 2018 alone, the box office and home video had a revenue of approximately $136 billion. The high number does have a boost on a country's economic revenue. As an innovation to further develop the movie industry itself, we will conduct research to predict the revenue of a movie by implementing linear regression.

Linear regression is a machine learning algorithm that is based on supervised learning. This algorithm performs a regression task whose models are based on independent variables. The independent variable is x, while the dependent variable is y. By using the algorithm, we are able to predict the revenue of movies that are yet to come.

2. Methodology

2.1 Data Profiling

In the first stage, data profiling is firstly implemented by gathering dataset information. The dataset is gathered from kaggle, which provides the required dataset for our project. Furthermore, we are going to check for any not a number or NaN value. The result of checking the NaN value is in percentages. After we retrieve the NaN percentage, we are going to retrieve the data description, which is using Pandas. .describe() is a function that is used for showing the data description such as count of total data amount, mean, std (standard deviation measures the spread of the data about the mean value), min (the minimum value contained in each column), 25% (the first quartile value of count from each column), 50% (the second quartile value of count from each column), 75% (the third quartile value of count from each column), and max (the maximum value contained in each column). Lastly, we are going to retrieve the data correlation matrix, which can discover relationships between columns. The data parts which are interrelated are popularity, revenue and vote_count by having 0.78 correlation value, and budget and revenue by having 0.73 correlation value. The whole correlation value between columns can be checked by Pandas .corr().

2.2 Data Cleaning

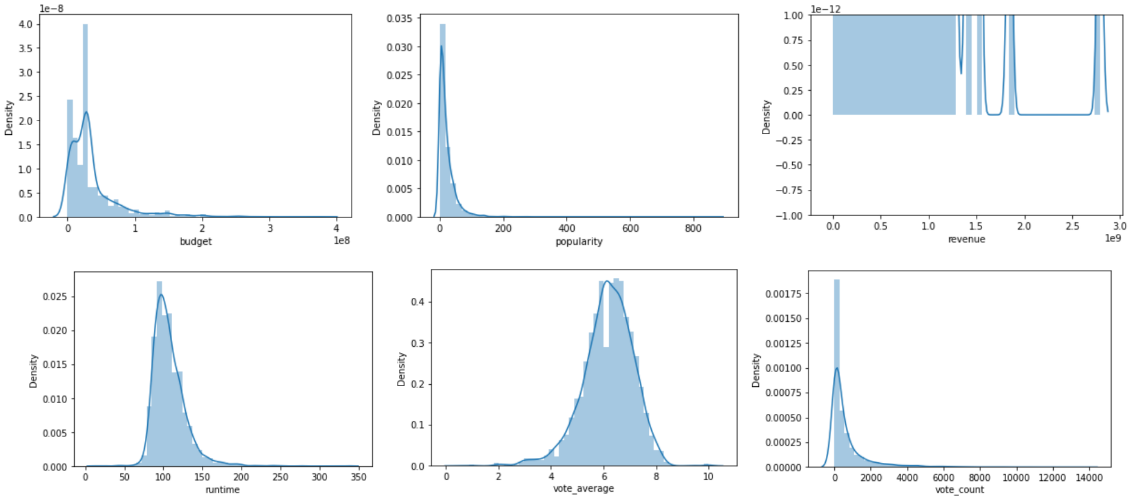

Figure 1.1 Data Distribution

In the data cleaning section, the process is firstly conducted by removing the unrelated columns. Which means that we will get rid of irrelevant attributes or columns that have no relation in predicting the data. The irrelevant columns that are identified are: *'genres', 'homepage', 'id', 'keywords', 'original_language', 'original_title', 'overview', 'production_companies', 'production_countries', 'release_date', 'spoken_languages', 'status', 'tagline', and 'title'.* The next step is by checking the NaN or 0 values. Afterwards, the data which has 0 and NaN as its value will be changed into the mean of the data in each column. Then finally, we retrieve the data distribution information and probability density function (PDF). Regarding the data distribution and PDF, the histogram is plotted by utilizing Seaborn .distplot(), and matplotlib.pyplot .show() to generate the histogram, which can be seen above.

2.3 Feature Engineering

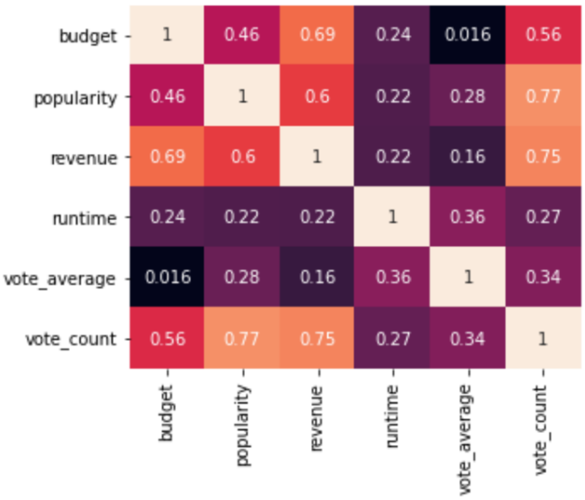

Figure 1.2 Dataset Correlation Matrix

Pandas .corr() is used to retrieve the correlation matrix which contains correlation values between each column. In this case, revenue can be considered as the Y since it is dependent with budget, popularity, vote count and has the suitable correlation values, which are 0.69, 0.6, and 0.75 consecutively (above or equal with 0.6). It also means that budget, popularity, vote_count will be the most suitable predictor for revenue since it has equal or higher correlation values than 0.6, which are 0.69, 0.6, and 0.75 consecutively. Meanwhile, runtime and vote_average are not included as the predictor of revenue since they have correlation values below 0.3 and close to 0.

2.4 Training

In the training section, the dataset into train data and test data by utilizing sklearn.model_selection train_test_split library. In the following step, data fitting is performed by fitting the data into a linear model by implementing .fit(). A fitted linear regression model can be utilized to distinguish the connection between a predictor variable x_j, which are the budget, popularity, vote count and the response variable y, which is the revenue when the wide range of various predictor variables in the model are fixed. Furthermore, the revenue in the test data is predicted based on the data that has been trained, by utilizing .predict() function.

3. Result and Discussion

Data	R^2	MAE	MSE
Training	0.6585	53449454.55158375	7675263070504818
Testing	0.706	56202611.16459572	8702114743637363

Table 1.1 Training and Testing Result

To evaluate the learning model, R^2, Mean Absolute Error (MAE), and Mean Squared Error (MSE) are utilized which can be obtained by using sklearn.metrics r2_score, mean_absolute_error, mean_squared_error. R^2 shows the level of the variance in the dependent variable that is explained by the independent variable. R^2 commonly measures the relation strength between your model and the reliant variable on 0 – 1 scale. From the test result above, the R^2 score obtained is 0.706. Mean Absolute Error (MAE) measures the absolute average distance between the predicted and the real data. From the test result above, the MAE score obtained is 56202611.16459572. Mean Squared Error (MSE) is an estimator that measures the average of error squares such as the average squared difference between the estimated values and true value. From the test result above, the MSE score obtained is 8702114743637363.

Regarding underfitting and overfitting, underfitting means that the model or the algorithm does not fit the data well enough. It usually happens when the data is not enough to build an accurate model. Meanwhile, overfitting occurs when a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. Hence to check that whether my model is underfitting, overfitting, or not, the evaluation result with the dataset train result can be compared. The model is not underfit since the predictors which correlate with revenue are budget, popularity, vote count, which has correlation value is above 0.6. On the other hand, the result of the dataset train R^2 value is 0.6585, and does not exceed the dataset test R^2 value 0.706. Therefore, this means that this model is not overfitted.

4. Conclusion

To sum up, movie revenue can be predicted by implementing linear regression. Our methodology is divided into four sections which are data profiling, data cleaning, feature engineering, and training. For the result, this model shows strong connection between reliant variables and the model by having 70.6% as R^2 score. Other than that, the Mean Absolute Error score obtained is 56202611.16459572 according to the test result above. Furthermore, the Mean Squared Error score obtained from the test result above is 8702114743637363. In addition, this model is not underfitted and overfitted since the predictors are suitable and train R^2 value does not exceed the dataset test R^2. As an additional, this method has the potential to be useful in many more aspects in the future.

5. References

[1] Source Code:
https://colab.research.google.com/drive/1DXNivbr1_rKmJr31J-c0gZqZwlgIHW3S?usp=sharing
[2] Dataset: https://www.kaggle.com/tmdb/tmdb-movie-metadata