Q1

MSE: $6.16*10^{15}$

correlation: 0.503

The lower the MSE or the higher the correlation, the better the performance

Because it is a regression task, I use MSE and Correlation to evaluate the performace.

First do feature enginnering for some features. I plot graphs to show the relation between each column and the revenue.

1. Runtime: Category the runtime, do one hot encoding with these categories.
2. Genres: Find the top K most common genres.
3. Keywords: For each film, I only consider the first 5 keywords, extract all film's keywords and do one hot encoding for the most common 5 ones.
4. Release date: By plotting the graph, I find the higher the year, the higher the revenue. And the higher revenue is mainly appeared in the middle and the end of the year. Hence add two new columns, one for month, one for year.
5. HomePage: New feature to show whether a film has the homepage.
6. Language: Do one hot encoding for most common languages.
7. Actor, Director, Production Companies: I use the same logic to preprocess the three features. For each film, I only consider the first M actors/directors/productions, then find the most common actors/directors/production companies and do one hot encoding.

Then I fit three models on the train dataset, include RandomForestRegressor, Linear Regression and XGBoost, then predict the validation dataset, usually the RandomForestRegressor has the best performance.

After done feature enginnering and apply these models, the correlation is only around 0.25, and the MSE around $7.3 * 10^{15}$

Improvements

1. Using K actors/directors/productions with highest mean revenue instead of K most common actors/directors/production companies, but with at least M films they made. This can improve the performance a litte, raise around 0.05 for correlation
2. Apply log on budget. After applying log on the budget, there is the linear relation between the budget with the revenue, and the performance is improved a lot by applying log on the budget. (correlation is between 0.39 – 0.45 now)
3. Model Selection. Originally I use linear Regression and xgboost, but I found sometimes the predicted revenues are negative, then I try to use RandomForestRegreesion, and the predicted values are valid and has lower MSE
4. Adjust hyperparameters I mentioned above, like K, M, N. I turn up the K (how many actors/directors/productions) and the correlation is gradually increased from 0.43 to 0.48. The reason I add more features is if the number of features is not so many, then there will be many zeros (column's value) when feeding the data into model.

Problems:

1. Json format: Use list(eval(x)) to transform some columns from string format to json arrays.

Q2

Accuracy: 0.73
Average_precision: 0.679
Average_recall: 0.663
The higher the accuracy/average_precision/average_recall, the better the model

Originally I use the same features I use in Q1, and the model I select is RandomForestClassifier. The accuracy on the validation dataset is 0.69, and I check the predicted ratings, nearly all of them are 3

Improvements:

1. Recatogories the 'runtime' column. By observing the column 'runtime' and the column 'rating', I find the rating usually is 2 when the runtime is smaller than 90mins, hence I recatogories the 'runtime' column, then the accuracy is increased from 0.69 to 0.71
2. Try different models. Instead of RandomForestClassifier, I try to use Xgb Classifier and Logistic Regression, the performance of this two models are same (0.72), both of them are higher than RandomForest Classifier
3. Remove features. I use budget as the feature in Q1, but the rating can usually be 2 if the budget is high, also the keywords has little influence for rating, and finally I keep runtime, companies, genres, actors and directors.
4. Adjust hyperparameters. After remove some features, I try to adjust the number of actors/directors/companies, and the accuracy is increased from 0.72 to 0.73.