Part 1: Regression

Performance on validation set:

MSE is about 6.2 x 10^{15}.
Correlation coefficiency is about 0.49.

How to evaluate the performance of the model?

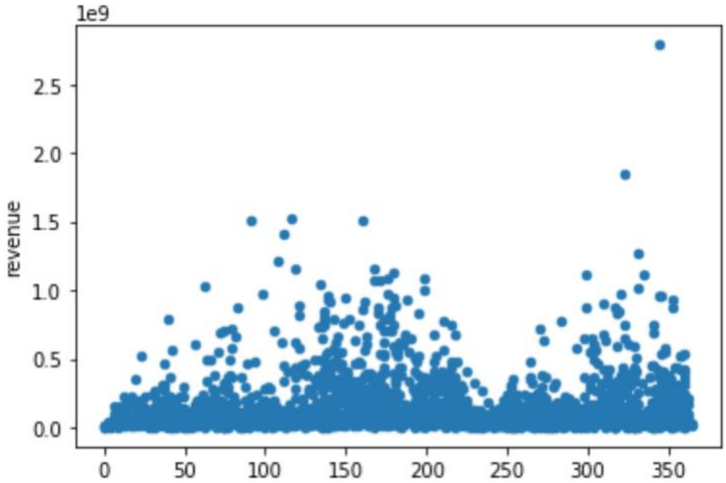
MSE value and Pearson correlation coefficient.
Also use cross-validation to see the performance when using different combinations of trainning sets and test sets to get a more general understanding of how the model is performed

How to select the best model?

For model selection, I use cross validaiton to test on different regression models from a set of different linear regression models to DecisionTree regression, RandomTree regression and Kneighbor regression. By using the neg_mean_squared_error as the scoring parameter, I chose RandomTree regressor as the best model because it has the highest socre.

The problems in predicting:

1. We have different attributes in the original dataset. Some of them are not singleton value but data in json formats. Since we cannot pass json formatted data into any model, we need to first process those json strings into a string of one attribute value in each json element. I chose the attribute 'name' as the selected one in column cast, crew, genres, keywords, production_companies and 'iso_639_1' in spoken_languages.
2. For the column release_date, the format is not a valid format to pass to a regression model. Instead of get the year, I tried to get the day of the year. Because the revenue could be different in different period of the year.



4. After I well cleaned the data, how to generate a feature vector is another problem. I used countVectorizer to vectorize those data from the first problem and set a 'max_feature' to control the number of features in each feature vector. At the end concat them together with the original features and this become the data that can pass to a model.
5. Then I tuned the hyperparameters by using random grid search. The parameters I chose are n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, bootstrap and warm_start.

Part 2: Classification

Performance on validation set:

Precision socre: 0.69
Recall score: 0.62
Accuracy score: 0.73

How to evaluate the performance of the model?

Using precision score, recall score and accuracy score to evaluate the model and print the confusion matrix to see the preformance.

```
[[ 42  81]
 [ 26 251]]
```

From the confusion matrix, the correctly classified instances are 293. So the accuracy is 293 over 400 which is around 0.73.
When we look at the precision score, the true positive instances for class 1 are 42 and the false positive number is 26. For class 2, the true positive instances are 251 and the false positive number is 81. After calculation we can get the precision score is 0.62 for the minority class and 0.75 for the majority class. This means the ability of getting the fraction of the retrived documents that are relevant is not bad. But when we look at the recall score, for class 1 false negative is 81 so the recall of class 1 is about 0.34, which is really low and that means the ability of getting the fraction of relevant instances that are correctly classified is pretty weak when this class is the minority class in the real data set.

The problems in predicting?

1. The data processing problem has beed dealed with in part 1. But the data model that pass into the classification model is not exacly the same with the one in part 1.
2. For model selection, I used the same methods in the regression part which is using cross-validation to check the accuracy and select the best model with highest accuracy. Then I chose RandomForest classifier as the model.
3. Overfitting problem. After using this model to predict the validaiton set, the accuracy is about 73%, but the accuracy of training set is about 99%, which means this model overfitted too much information. The method I used is the combination of random grid search and grid search to tune the parameters and find the model that can reduce the overfitting problem. In the end, this model can still get 73% accuracy on validation set but this time, the accuracy on training set is about 83% which clearly reduced the overfitting problem a lot.

Further improvements:
1. Since I used random forest in both problems, it takes some time to train. In further improvement, I want to find a way to reduce the training time of random forest and improve the efficiency.
2. Find a way to improve the recall score is also a further improvement of my model in the future.