COMP9321

Assignment3 report

z5141401 Xu GAO

Part 1

- Performance and evaluation

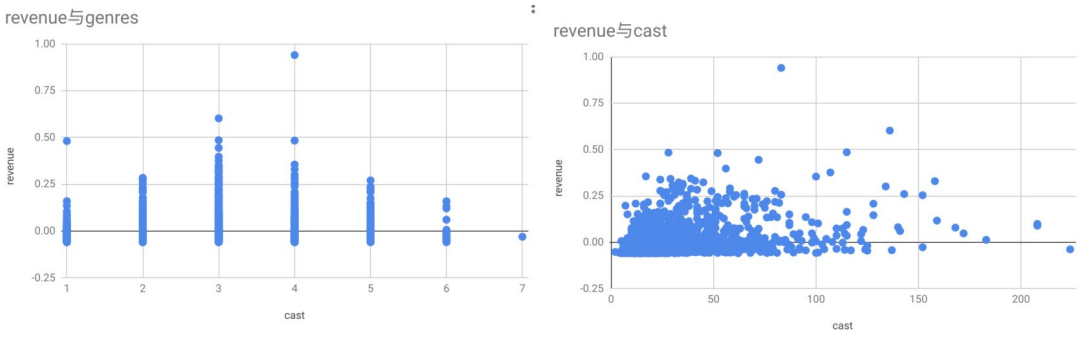
To start with, I got a brief overview of the whole data frame, start to consider how to do preprocessing and which row can be implemented for this part.

Firstly, I tried to count numbers of certain types, like the number of genres, the number of cast and crew, or even the number of male/female casts in a movie, and try to find if there is any correlation between this number and the revenue.

After a simple plotting, I found that the relationship between the number of genres and revenue is like a normal distribution. Hence we can use the number of genres as one of the features. However, for the number of casts, the dots on the graph is too distributed, and we can hardly find a linear separate line to fix this relationship, so I did not use the number of casts as one of the features.

Column	Value
Budget	convert to budget_year ratio
Genres	convert to one-hot encoding
Homepage	convert to boolean (has a homepage or not)
Tagline	convert to boolean but not useful (abort)
original_language	not relevant (abort)
release_date	convert to year
release_date	convert to weeks
release_date	convert to day of weeks (one-hot encoding)
Crew	convert to crew number

The final performance of the model is around 0.4 Pearson correlation coefficient.



- Problems you have faced in predicting.

For me, the CSV file I printed always got an index column starts with 0, which is not the same format as the assignment specification. The solution is quite simple, the default value for pandas function pd.to_csv() has a default value to set index column. So set the index parameter to false could solve this issue.

P2

- Performance and evaluation

I tried another way to format the dataset as features. I separated the release_date to *release_year*, *release_month*, *release_weeks* and *release_dayofweeks*. And try to find the relationship between the revenue and these numbers, by plotting graphs, I found that *release_year*, *release_dayofweeks*, *release_weeks* has a close relationship to the revenue.

Furthermore, though the *budget* has a close relationship with the revenue. However, it is not a stable feature for the dataset. For some old movies, the budget was limited by the economic condition which means it is unfair to measure it with the recent movie's budget. Since it is still valuable for those decades, I used a budget_year_ratio as a new feature to replace the budget feature. The value is budget / release_year, and it has a dramatic effort for the model.
The final performance of the model is around `['z5141401', 0.66, 0.55, 0.71].`

- Problems you have faced in predicting.

The first problem I faced is the model selection, in order to compare models, I wrote a python script to calculate the scores of models and pick the highest result.

```
the classifier is : svm
the score is : 0.6925
the classifier is : decision_tree
the score is : 0.555
the classifier is : naive_gaussian
the score is : 0.7
the classifier is : K_neighbor
the score is : 0.675
the classifier is : bagging_knn
the score is : 0.7
the classifier is : bagging_tree
the score is : 0.6875
the classifier is : random_forest
the score is : 0.6475
the classifier is : adaboost
the score is : 0.345
the classifier is : gradient_boost
the score is : 0.7075
```

Another question I faced is, when I try to calculate the precision score, it returns this error: "ValueError: pos_label=1 is invalid. Set it to a label in y_true.". Since the tutor told us to calculate unweighted average mean, we need to set "average='macro'" to fix this issue.