Python task – Danesh kar

TMDB Movie Dataset

MohammadAmin Omidzadehnik

Contents

1 Introduction

Success is one of the most concerns amongst any business in the world. Movie has become an industry these days. It is better to forecast its benefit and sale before starting the project. In addition, if you are able to define some features to increase chance to be successful.

In this project, we are going to utilize movie dataset for further research about movie, budget, cast and crew, genre, etc. The main purpose of the project is What we can say about the success of a movie before it is released. Is there any formula?! The aim of this project is to gain some insight about the data for further modeling it for investigating the profitability of the movies.

2 Data loading and gain info

Since Python has been used in the project, so before loading data, we need to import essential modules. There are so many modules in Python, for this project, we are going to import the most famous libraries. For example, Pandas, Numpy, Sklearn, json, time, matplotlib, category_encoder, xgboost, scipy, seaborn, and warning. Although most of modules are built-in functions, you need to install some packages before importing them.

First of all, movie and credit datasets are called. It seems they have connection between each other. We predict "id" in movie dataset and "movie id" in credit dataset are relational. For this theory, we check equality of these features and flag any extra record (out of this theory, if exist). Fortunately, these features are completely matched. Now, we can merge both datasets based on these variables. For further analysis, we add two features in new dataset. Benefit is our goal in the project. So, we add this feature based on a simple equation:

$$Benefit = Revenue - Budget$$

Next, we add another feature related to "benefit" called "benefit_status." If a movie has positive benefit, "benefit_status" will have "Profitable" values. Otherwise, it will get "Non-profitable." Secondly, we check dimension of the dataset. Final dataset contains 4803 rows and 25 columns.

3 Data cleaning and preprocessing

Data cleaning and preprocessing is the most important part of any project related to data science. Because this section is like a raw material to create peace of art.

At first, we check missing values. According to the output we can say most of features are valid without any missing values.

Column	Non-Null Count	Dtype
movie_id	4803 non-null	int64
title_movies	4803 non-null	object

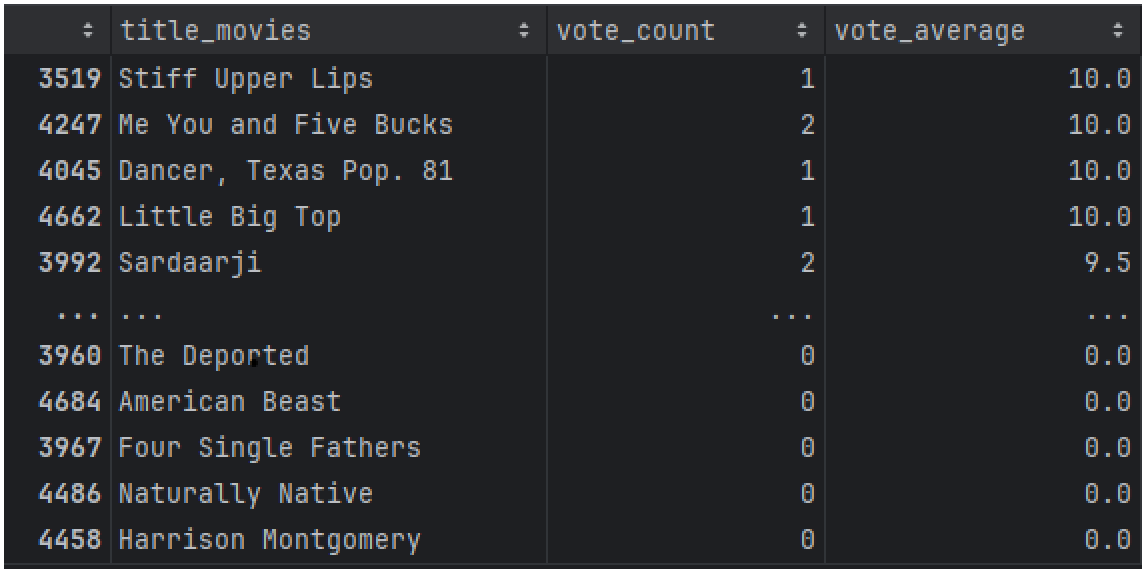
```
cast                 4803 non-null   object
crew                 4803 non-null   object
budget               4803 non-null   int64
genres               4803 non-null   object
homepage             1712 non-null   object
keywords             4803 non-null   object
original_language    4803 non-null   object
original_title       4803 non-null   object
overview             4800 non-null   object
popularity           4803 non-null   float64
production_companies  4803 non-null   object
production_countries  4803 non-null   object
release_date         4802 non-null   object
revenue              4803 non-null   int64
runtime              4801 non-null   float64
spoken_languages     4803 non-null   object
status               4803 non-null   object
tagline              3959 non-null   object
title_credits        4803 non-null   object
vote_average         4803 non-null   float64
vote_count           4803 non-null   int64
benefit              4803 non-null   int64
benefit_status       4803 non-null   object
```

According to this table, homepage, overview, release_date, runtime, and tagline have missing values. Since, we are just going to utilize runtime variable, so its missing values are replaced by the average method.

Now, we check duplicated value. For this matter we change type of “movie_id” to str. Then, check duplicated value on them. As a result, there is no duplicated value in dataset.

4 Exploratory Data Analysis (EDA)

Firstly, I am going to analyze dataset descriptively. In this part, we can have tables, graphs, etc. The more insight you have, the better information you gain. Before visualizing features, we check status of the highest movie based on votes.



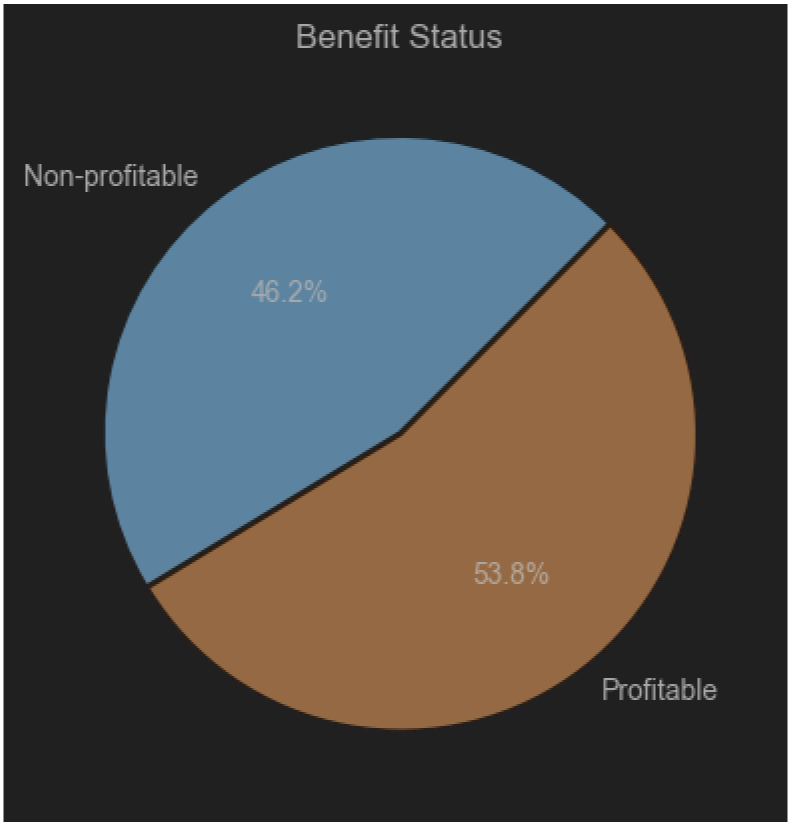
	title_movies	vote_count	vote_average
3519	Stiff Upper Lips	1	10.0
4247	Me You and Five Bucks	2	10.0
4045	Dancer, Texas Pop. 81	1	10.0
4662	Little Big Top	1	10.0
3992	Sardaarji	2	9.5
...	...	...	...
3960	The Deported	0	0.0
4684	American Beast	0	0.0
3967	Four Single Fathers	0	0.0
4486	Naturally Native	0	0.0
4458	Harrison Montgomery	0	0.0

It seems odd. Because a movie with a single vote and 10 rate is the best movie. It is not accurate. For solving the problem, I am going to define another feature called “score” with the following formula[1] with WR (Weighted rating).

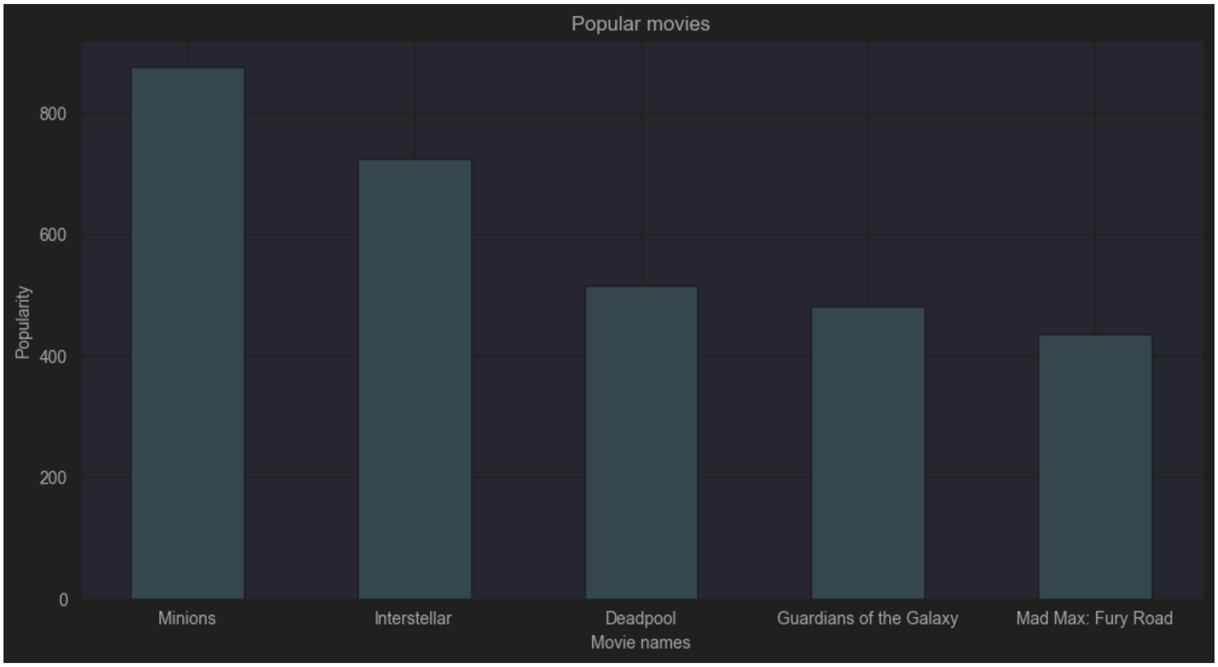
$$\text{Weighted Rating (WR)} = \left(\frac{v}{v+m} \cdot R\right) + \left(\frac{m}{v+m} \cdot C\right)$$

Now, we can draw some graphs based on last version of dataset.

[1] It is calculated based on IMDB website calculation.



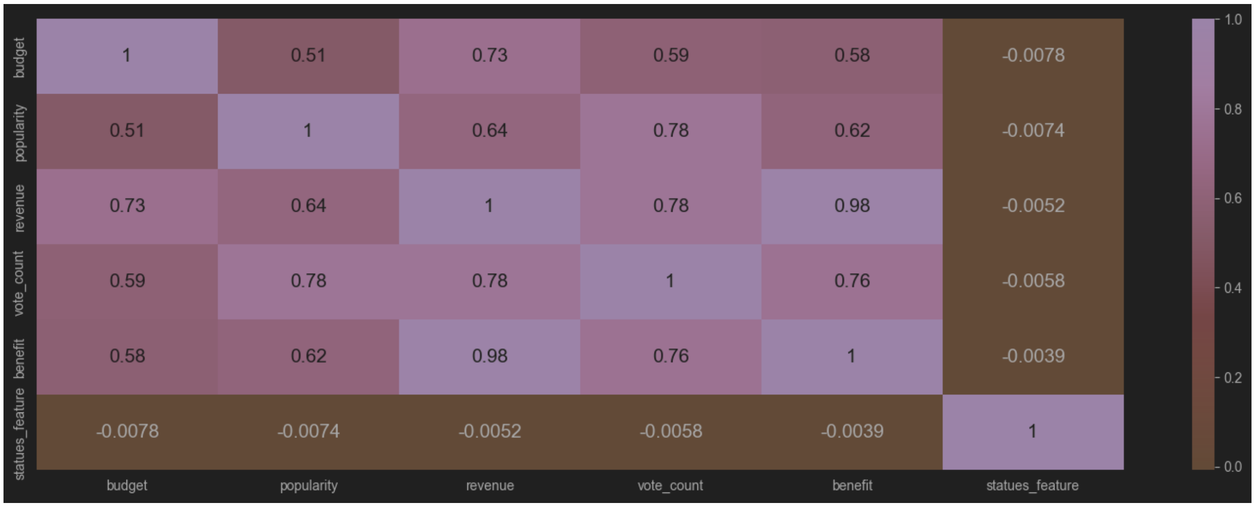
In the pie chart, we can observe that the majority of movies was successful and have positive profit. On the other hand, almost 46 percent of movie in the dataset are non-profitable. It is a significant amount for unsuccessful movie.



How about popularity? Beside of score of a movie, its famousness is another feature. It is very important for success of a movie. Due to bar chart, Minions is the most popular movie among the dataset. In addition, Interstellar is the second famous movie with almost 700 popularities. Deadpool, Guardian of the galaxy, and Mad max: Fury Road are in third to fifth place which are close to each other.

5 Data Modeling (Using different ML models)

Before using ML method, we checked correlation and add another feature called "status_feature" which is related to status variable. So, we label all values of status for new variable. Now, correlation:

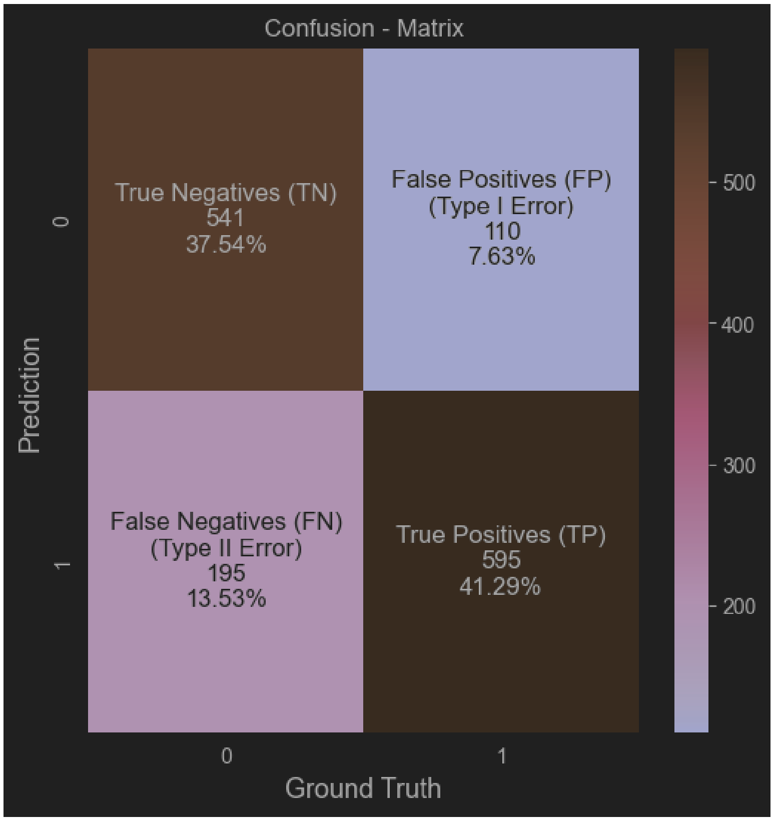


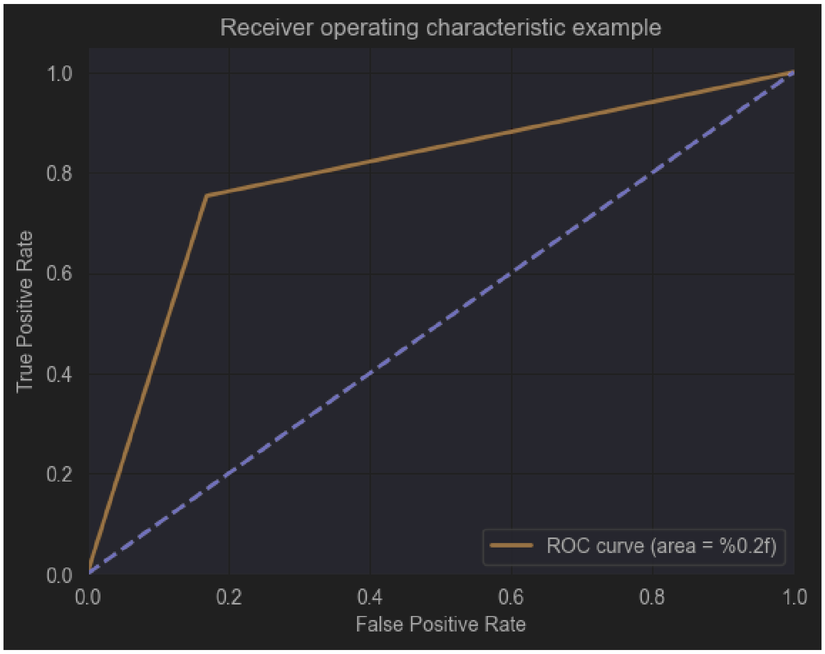
It is very clear that all features have positive correlation, but feature status. However, feature status has few negative relations with other variables that we can ignore it. The most correlation belongs to benefit and revenue. It is expectable, because benefit is calculated from revenue and budget. Then, we have popularity and vote count, it means 78 percent differential of popularity related to vote count and vise versa. Vote count has the exact same correlation with revenue. It shows 76 percent of revenue of a movie is connected to its vote count. The other interesting point is all of correlations are more than 50 percent (positive, except status feature). It can make a point that these features are very correlated. In this situation, popularity and budget have less values (%51) which is high in fact.

Now, we are ready for machine learning. In this project, we are looking for successful and unsuccessful movie. For this matter, we add features (benefit and benefit_status). We assume "benefit_status" as response and some other features as feature. Popularity, runtime, vote_average, vote_count, score, and status_feature are independent variables. Since these feature have different measurement, we standardize features before running classification. In classification, we have variety of method to classify target such KNN, Logistic Regression, Random forest, decision tree, etc. For comparing these methods, we are going a function named use_model. The function split dataset to train and test parts with 30 percents of test size. Then it fits a model based on train data and predict response with model and test features. Next, we have classification report including precision, recall, f1-score, support, and accuracy. After that, we have ROC AUC score and confusion matrix. Finally, we can observe ROC curve. After analyzing many methods with use_model function, Logistic Regression is chosen. It includes the highest value in each part of the report.

```
The Classification report :
              precision    recall  f1-score   support


           0       0.74      0.83      0.78       651
           1       0.84      0.75      0.80       790


    accuracy                           0.79      1441
   macro avg       0.79      0.79      0.79      1441
weighted avg       0.79      0.79      0.79      1441


roc_auc_score :  0.7920968714149604
```

According to the output, generally this model has almost 80 percent roc_auc_score. In addition, 84 percents of profitable movies are predicted correctly. Also, this method recognizes unsuccessful movie in benefit with 74 percent precision.



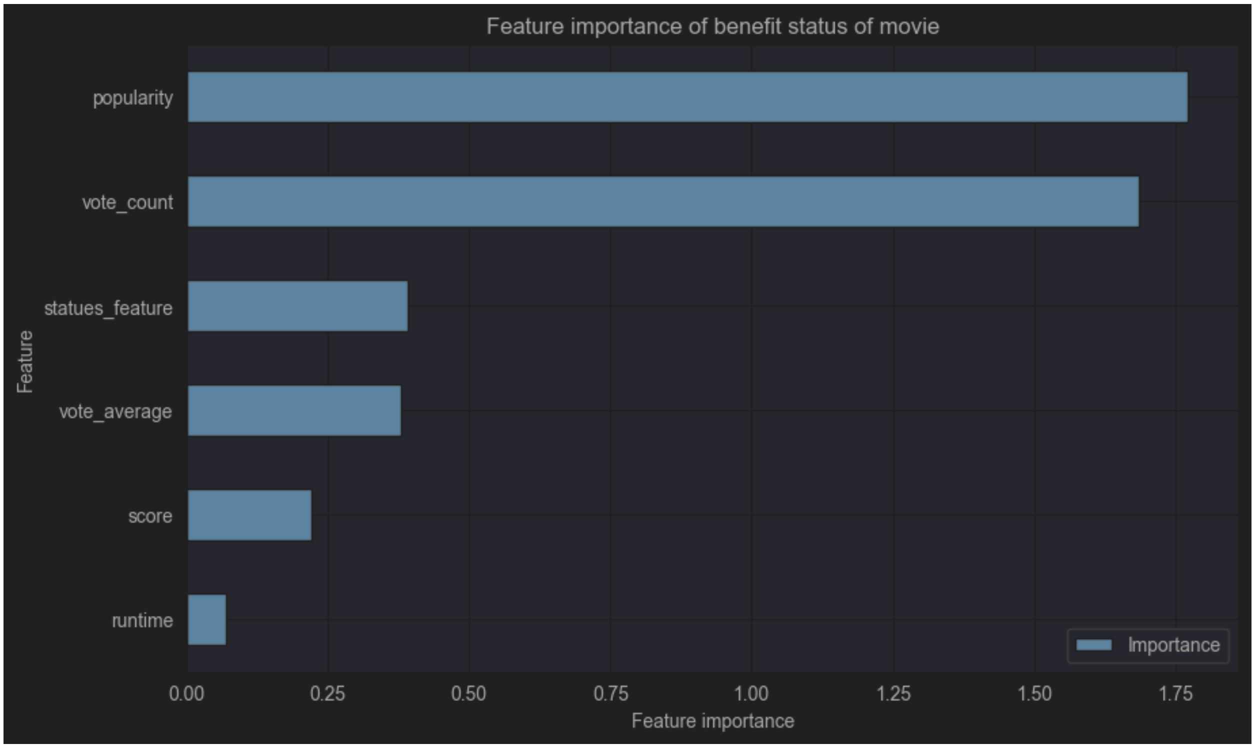


In confusion matrix, we can see less than 8 percent of classification is related to Type I error which means we predict the movie will be unsuccessful, but it will be successful in fact. The most important error is Type II. we have less than 14 percents in this error. We predict the movie will be profitable. Unfortunately, in fact it will not be. Also, ROC curve confirms this statement.

As we choose the model, we need to select its parameters in detail such as solver, penalty, max iteration and so on.

```
{'solver': 'liblinear',
 'penalty': 'l2',
 'max_iter': 200,
 'class_weight': None,
 'C': 0.1}
```

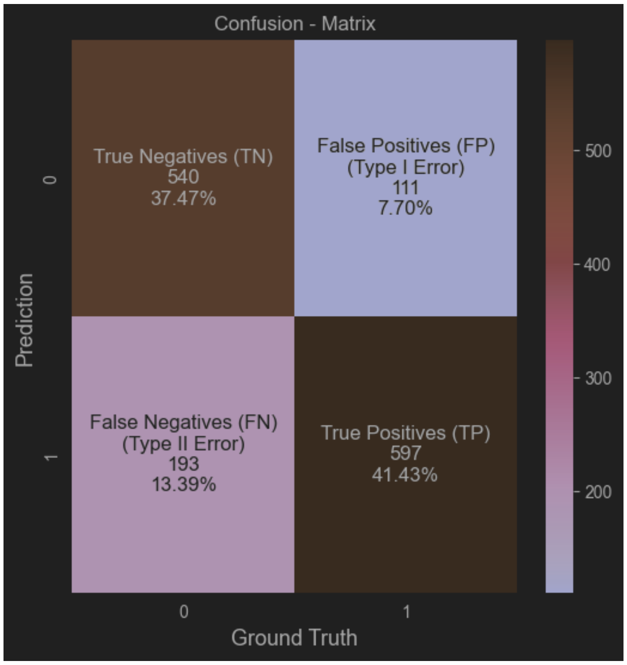
After choosing the best parameters, we are going to know which feature are more important than others. So, we run feature importance function.

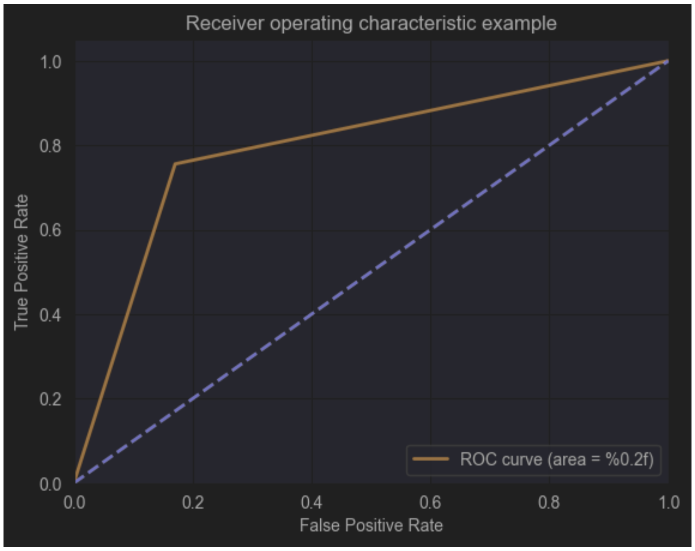


It is very clear that popularity is the most effective feature among all features. Also, vote_count is very close to it. status feature and vote average are almost similar in next steps in order. According to this graph, we can ignore score and runtime because of small importance. In final part, we run the model again with 4 features.

```
The Classification report :

              precision    recall  f1-score   support
```

```
           0       0.74      0.83      0.78       651
           1       0.84      0.76      0.80       790


    accuracy                           0.79      1441
   macro avg       0.79      0.79      0.79      1441
weighted avg       0.80      0.79      0.79      1441


roc_auc_score :  0.7925946450446246
```





So, the previous and current model are very similar. Hence, we can use the new model with four features.

6 Conclusion

All in all, in this project, we prepared dataset to run the classification model. Meanwhile, we have some issues with missing values, we add some features which are very useful for us. Then, we analyzed dataset descriptively. In final section, we run classification method to separate response to profitable and non-profitable parts. Logistic regression with lib linear solver, L2 penalty, 200 maximum iteration and 0.1 C is the most suitable method for this project.