A report of analytical research on IMDB dataset using scikit learn library

Pankaj Mehar

CONTENTS

LIST OF FIGURES

A report of analytical research on IMDB dataset using scikit learn library

I. Introduction

Given dataset is the amalgamation of two datasets 1.IMDB, 2.The Numbers. Mr Chuan Sun scraped this websites with motive of finding important inferences. IMDB, also known as Internet Movie Database, is an online database of information related to world films, television programs, home videos and video games, and Internet streams, including cast, production crew, personnel and fictional character biographies, plot summaries, trivia, and fan reviews and ratings. The Numbers website hosted the similar information but with business perspective.

Motive of any analytical study is to answer the question of curiosity. End business of such study could be revealing truth or finding way to make money. In this study we would like to explore Movie industry: " How to make the film successful as a business project, that is, to get the most out of the invested capital?"

Let's assume the Chinese guy, who is ultra rich and would like to diversify his money in different industry. He thinks Hollywood could be best bet. The best way is to do crunch data and get the answers for safe investment.

In this study I used Anaconda IDE, scikit learn library and python as language. After loading data and importing all required, first thing in any data science process is to find data type of features. The data type is as follows.

The dataset consist of 5043 rows and 28 columns/features

II. Problem Definition

Our main target for any analytics is, how to earn money or how to increase profit. In this case the perspective of investor is as follows, one who bet his million dollar. The strategy for this perspective involves finding answers for below questions.

- **Who gave more hits?**
- **What if I have an algorithm, who can predict blockbuster movie?**
- **Is IMDB score reflects good movie?**
- **How to Predict blockbuster movie based on Director, Actor, Genre, Budget?**
- **What make movie profitable?**
- **How are these features related to each other?**
- **Who is the big shot in industry?**

III. Proposed Solutions

In our data analysis the best features can be define as, who gave maximum information regarding revenue. Exploration of such features could be the best strategy.

IV. Research Methodology

In the data science project, before doing any machine learning, we need to do preprocessing, exploratory data analysis, feature engineering and feature creation.

A. Preprocessing

In preprocessing I check the data type, shape of data frame and null values.

B. Exploratory Data Analysis

In EDA as per our goal I tried to find correlation of features in our dataset. The strategy I used is 'Pearson correlation'. Then I tried to find how many movies made 500 million and more. Same for most earning actors, most earning means more successful. I tried to figure out which Genre gave most successful movies. Social media is always best marketing tools, movie makers have nowadays. Common peoples opinion can be judge by using Facebook, I tried to find relationship using scatter plot.

C. Feature Engineering

Supervised machine learning is process of predicting and generalization on unknown labels or target. I thought directly using 'IMDB score' or 'gross' as target variable is not a wise way. Even though this is crude algorithm but I choose to generate new target variable and using encoding, so that our needed classification could be better.

1) Machine Learning: First I have generate profit column, by subtracting budget from gross. Then encode the positive value as '2' negative as '1' and zero as '0'. Such encoding makes classification easy.

I tried different algorithms and found out the respective generalization capability. The algorithms is as follows.

- **KNN algorithm**
- **LogisticRegression**
- **DecisionTreeClassifier**
- **RandomForestClassifier**

V. Analysis and Interpretation

The Pearson correlation shows relationship between features. Its range is in between (-1 to 1) highly related score toward 1, while -1 is no correlation at all. I have chosen relevant features for machine learning, which shows strong correlation. Some features are totally irrelevant in our dataset as per correlation matrix.

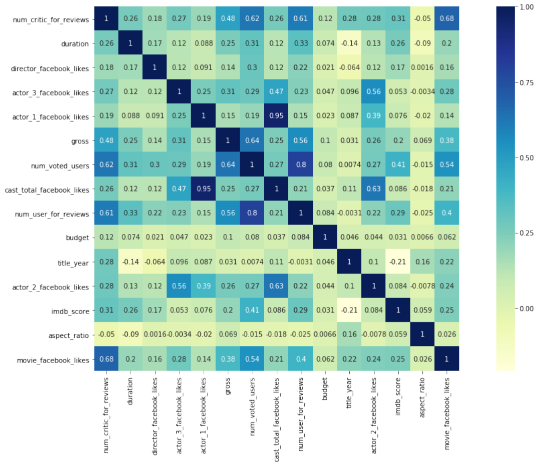


Fig. 1. Correlation matrix

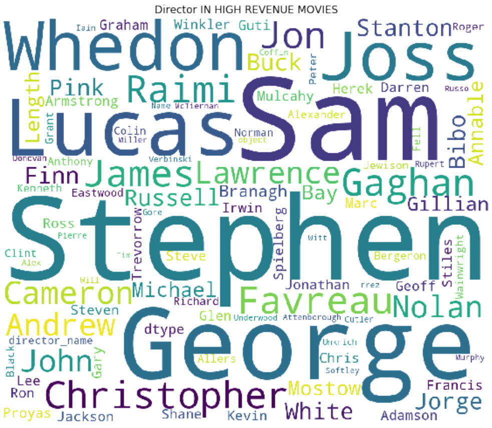


Fig. 2. Highest earning directors word cloud

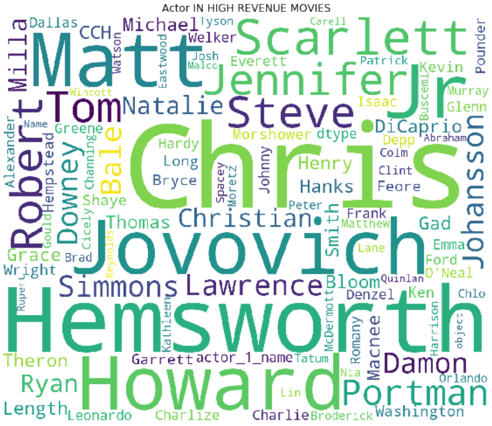


Fig. 3. Highest earning actors word cloud

While doing EDA we figure out which director is earning most. The word cloud describe the earning as per size of font in cloud.

Similarly I tried to find which actor have highest revenue. Wordcloud can reveal the richness as per size of cloud.

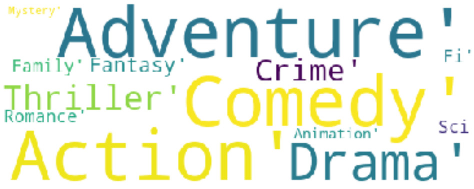


Fig. 4. Genre word cloud

Genre is a category of movie. After doing analysis I figure out that people most pay for Action, Adventure, Comedy, Drama, Thriller and Crime. this category is safe bet for our investor.

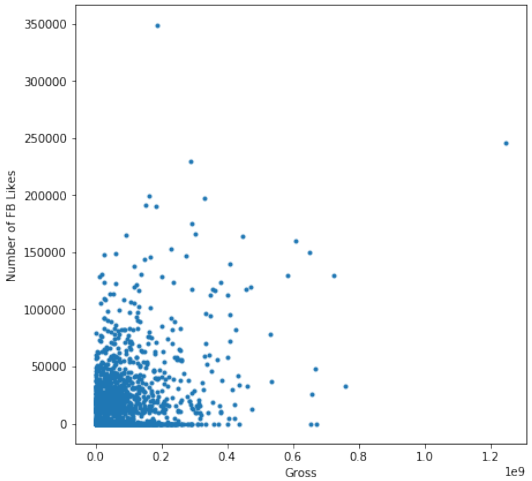
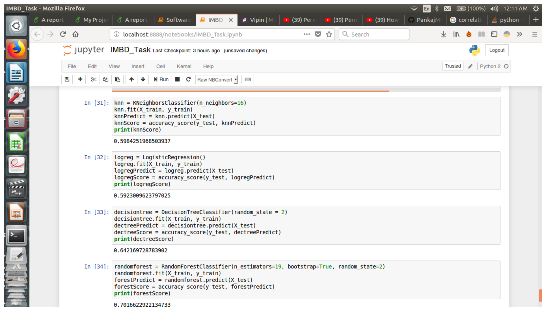


Fig. 5. Scatter plot between gross revenue and facebook likes

Social media is great platform for getting genuine feedback, if there is no bots how-erring over it, still facebook could be used to validate the hypothesis, based on other features. I found the strong relation between facebook likes and gross earning by movie.


IMBD_Task - Mozilla Firefox
localhost:8888/notebooks/IMBD_Task.ipynb
jupyter
IMBD_Task Last Checkpoint: 3 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 2
Run
Raw NBConvert

```
In [31]: knn = KNeighborsClassifier(n_neighbors=16)
         knn.fit(X_train, y_train)
         knnPredict = knn.predict(X_test)
         knnScore = accuracy_score(y_test, knnPredict)
         print(knnScore)

         0.5984251968503937

In [32]: logreg = LogisticRegression()
         logreg.fit(X_train, y_train)
         logregPredict = logreg.predict(X_test)
         logregScore = accuracy_score(y_test, logregPredict)
         print(logregScore)

         0.5923009623797025

In [33]: decisiontree = DecisionTreeClassifier(random_state = 2)
         decisiontree.fit(X_train, y_train)
         dectreePredict = decisiontree.predict(X_test)
         dectreeScore = accuracy_score(y_test, dectreePredict)
         print(dectreeScore)

         0.642169728783902

In [34]: randomforest = RandomForestClassifier(n_estimators=19, bootstrap=True, random_state=2)
         randomforest.fit(X_train, y_train)
         forestPredict = randomforest.predict(X_test)
         forestScore = accuracy_score(y_test, forestPredict)
         print(forestScore)

         0.7016622922134733
```

Fig. 6. Machine learning prediction accuracy

While doing machine learning I tried different algorithm. Highest accuracy i got for Random Forest, which is 70 percent, least I got for KNN algorithm which is 60 percent.

VI. Conclusions and Further Work

Conclusion : As code showing accuracy percentage, which is not up to mark, I haven't tried any hyper parameter tunning nor Gridsearch CV. This is just crude algorithm, my motive was to project proper approach towards problem. By doing feature engineering, accuracy can be brought to the mark.

Further Work : We could use IMDB score for recommending similar score in different category by using cosine similarity metrics.