|  |  |
| --- | --- |
| Faculty of Computer & Information Sciences  Ain Shams University  Subject: Machine Learning & Pattern Recognition  Year: (3rd year) undergraduate (CS)  Academic year: 2nd term 2021-2022 |  |

**Milestone (1)-Movie Revenue Prediction**

**Regression & Preprocessing**

Data Collection:

We started by collecting missing directors names from the Internet using **get\_director()** function in which we used googlesearch library to retrieve directors names. Then, the returned names were put inside a new data frame using **fill\_new\_director()** function. It succeeded in retrieving 75% of missing names.

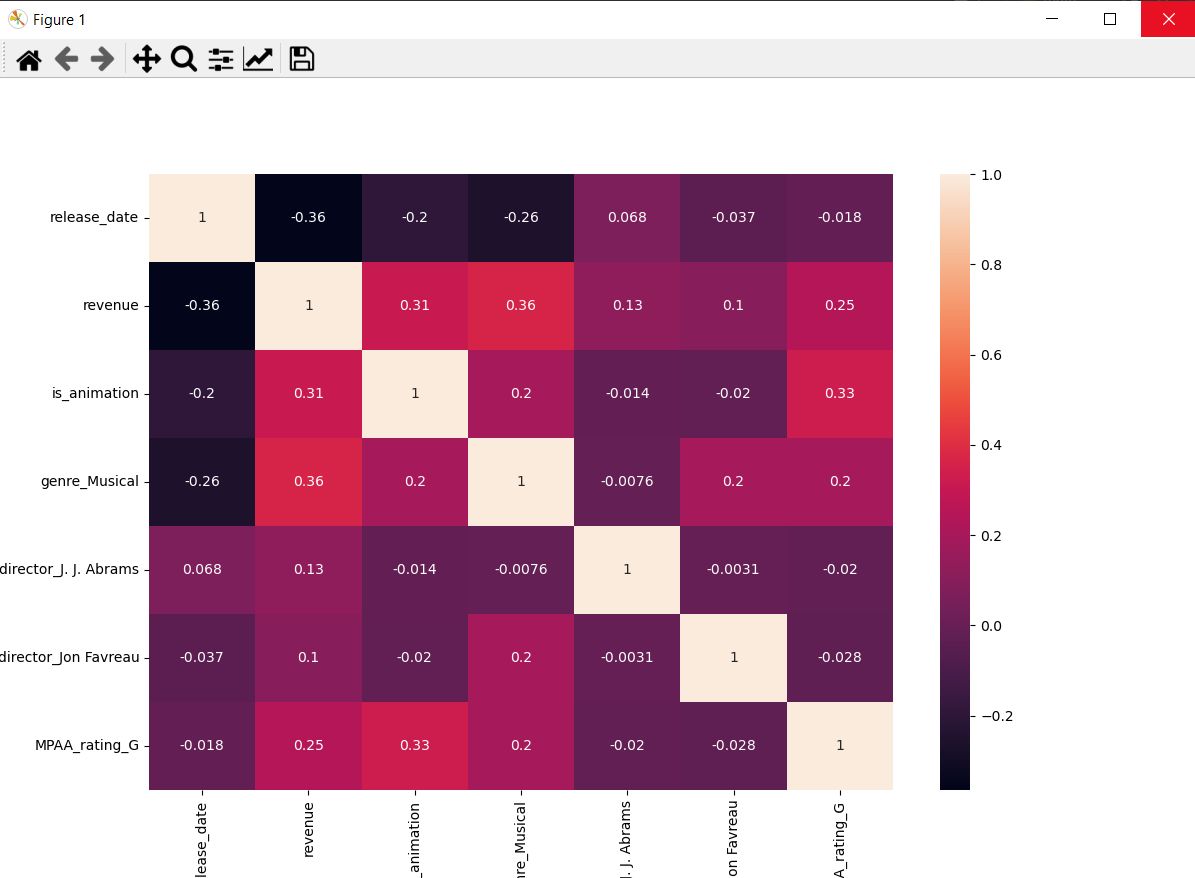
Merging Tables:

At first we had 3 different datasets (csv files); ”movie-director.csv”, “movies-revenue.csv” and “movie-voice-actors.csv” which we had to merge in a single dataset. We started by merging the new movie director data frame and “movies-revenue.csv” (using inner merging on “movie\_title” column). In order to avoid the presence of many null values, we decided to extract a new feature “is\_animation” from the third dataset “movie-voice-actors.csv” because the only movies that contain voice actors are the animated ones. So, if “is\_animation” is 1 it means the movie is animated and 0 otherwise. After doing the whole merging process, the final dataset info became as follows:

# Column Non-Null Count Dtype  
--- ------ -------------- -----  
0 movie\_title 463 non-null object  
1 release\_date 463 non-null object  
2 genre 449 non-null object  
3 MPAA\_rating 416 non-null object  
4 revenue 463 non-null object  
5 director 337 non-null object  
6 is\_animation 463 non-null object

Preprocessing techniques:

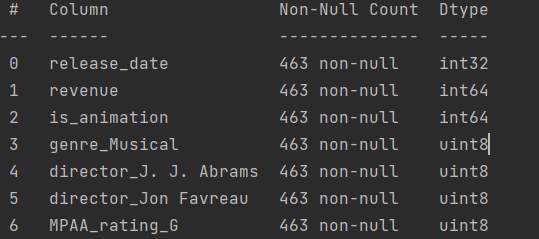
* Feature Selection: we get the correlation between X features and the single Y feature we want to predict(revenue). After plotting the correlation graph, we conclude that all the features like “genre”, ”MPAA\_rating”, ”director”, etc. have weak correlation as all their absolute values are < 0.5. But in order to reduce the number of X features, we pick the features with correlation > 0.1 using **correlation()** function. Correlation function takes the whole data frame and column of Y feature “revenue” as parameters then returns all X features with correlation > 0.1 in “top\_feature” data frame. The following figure shows the heatmap showing the correlation:



1 Correlation heatmap

* Feature Encoding: One-hot encoding is applied on the following X features; “genre”, “MPAA\_rating” & “director” using **one\_hot\_encoder()** function. We have used it due to its accuracy and in order to reduce the number of null values among the dataset features as different columns for all genres are created (same with MPAA\_ratings and directors names). The value is 1 if the movie belongs to this genre and 0 otherwise. In this way, we ensure the absence of null values.
* 4 lambda expressions: they were used for different reasons; remove ‘$’ and ‘,’symbols from revenue data, numerize revenue data and modify the release date format.
* Replacing null values: the remaining null values where replaced by 0 including (the 25% of directors without names).

After applying all preprocessing techniques, the resulted dataset info as follows:



Splitting Dataset:

**Train\_test\_split(**X, Y, test\_size=0.2,shuffle=True, random\_state=28**)** function: used to split dataset into training set and test split with being shuffled (80% Training and 20% Testing) and setting random\_state to 28 to lead to the best result.

Regression techniques and their results:

* Multiple Linear Regression: is the first applied regression model using **multi\_reg()** function which takes x\_train, x\_test, y\_train& y\_test datasets after being split to fit them in linear model and apply prediction on x\_test to return the mean square error and accuracy of this model.

Results:

-Mean Square Error: 1.6378414312388248e+16

-Accuracy: 18.68%

-Training Time of the model: 0.003989696502685547

* Polynomial Regression: the second regression model using **poly\_reg()** function which takes degree, x\_train, x\_test, y\_train& y\_test datasets after being split as parameters to transform the existing features to higher degree features then fit them to linear model and apply prediction on x\_test to return the mean square error and accuracy of this model.

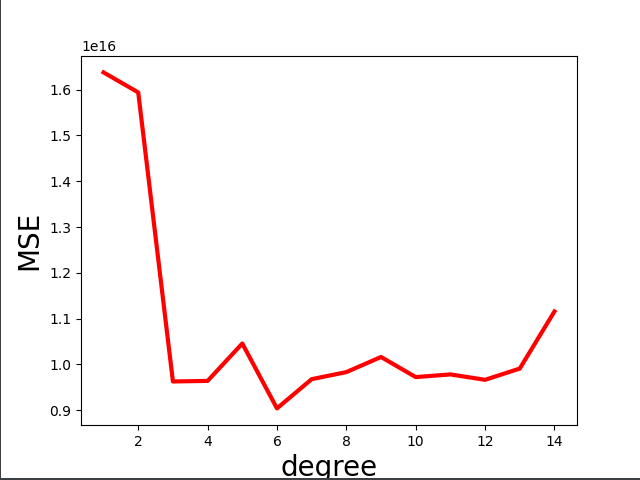
Results:

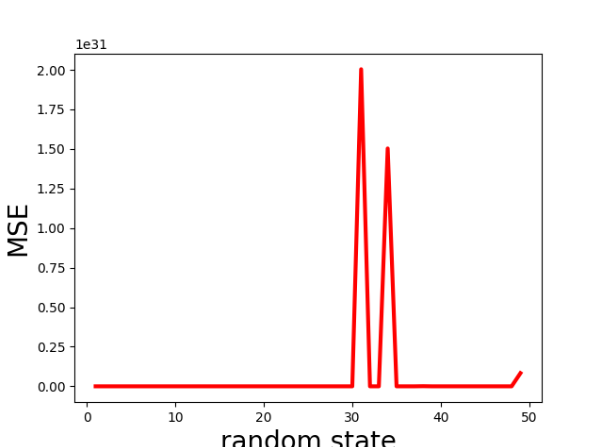
-Mean Square Error: 9662219641496994.0

-Accuracy: 52.03%

-Training Time of the model: 0.20043492317199707 ms

🡪We achieved these results after several tests and from the following figures, we concluded that the best degree for polynomial regression is 6th degree with random state 28.





Conclusion:

Polynomial Regression fits the dataset better than Multiple Linear Regression. This statement is proved by comparing both accuracies and mean square errors. The Multiple Linear Regression has bigger error (MSE=1.6378414312388248e+16) with less accuracy (18.68%). On the other hand, the Polynomial Regression has less error (MSE=9662219641496994.0) and bigger accuracy (52.03%). After applying several tests, we concluded that the best degree for the polynomial regression is the 6th degree with random state 28 as it led to the least mean square error we achieved.