|  |  |
| --- | --- |
|  | Movies Revenue Prediction |
|  |  |
|  | CS\_13  Milestone 1  4/21/22 |

**Preprocessing:**

* **Filling missing directors** using TMDb API by matching the movie name and its release date with the search results to ensure getting the corresponding director, included in “director\_filling.py”.
* **Determining director popularity** using TMDb API and instead of using normal encoding techniques we replaced each director with his popularity, included in “director\_pop.py”.
* **Determining voice actors popularity** using TMDb API and instead of using normal encoding techniques we replaced each voice actor with his popularity, then calculated the average popularity of all the voices actors for each movie, then added a new feature to represent the voice actors popularity for each movie. Included in “voiceActors\_pop.py” and “movie\_score.py”.
* **Format release date** to the universal format included in ”format\_date.py” then calculated the date into one number by the following formula:

Included in “calc\_date.py”.

* **Changing revenue type** from currency to decimal value by using regular expressions, pandas and decimal libraries included in “revenueTransform.py” file.
* **Encoding the MPAA\_rating and genre** columns using dummies. Included in “rating\_genre\_Encoding.py”.
* **Feature Engineering**: Adding two new features 'CharactersCount' which represents number of voice actors in the movie and 'IsAnimation' which represents the type of the movie. Included in “feature\_Engineering.py”.
* **Joining** ‘movie-revenue.csv’ table and ‘movie-director.csv’ table by using pandas merge method. Included in “join.py”.
* **Scaling features**: by using the preprocessing file included in lab2
* **Filling missing data:** by getting the mean of the columns that have NaN such as directors and Actors then we fill the NaN values with the mean of each column using fillna() function in python. Included in “fill\_empty\_cells.py”.

**Regression Techniques and Results:**

1. **Linear Regression polynomial**: Implemented in “basicModel.py” using ‘linear\_model.LinearRegression()’ to apply the technique on the dataset ‘movie-revenue.csv’ to predict the revenue using the provided features after preprocessing.

***Training MSE****: 9922079237943064.0.*

***Testing MSE****: 2.8434148573054884e+16*

***Random state****: 20.*

***Degree****: 3*

***Time****: 0.01795220375061035 seconds*

1. **Regularization Models**: Implemented in ’regularization\_model.py’
   1. **Lasso Regression** using ‘linear\_model.Lasso()’.

***Testing MSE****: 2.3744553873741133e+17*

***Training MSE****:* *2120176549424832.2*

***Degree****: 3*

***Random state****: 10*

***Time****: 0.012951374053955078 seconds*

* 1. **Ridge Regression** using ‘linear\_model.Ridge()’.

***Testing MSE****: 1.90529529477161e+19*

***Training MSE****: 3829522426942718.0*

***Degree****: 3*

***Random state****: 10*

***Time****: 0.015958070755004883 seconds*

* 1. **Bayesian ridge** Regression using ‘linear\_model.BayesianRidge()’.

***Testing MSE****: 2.3744553873741133e+17*

***Training MSE****: 5.162516674040349e+17*

***Degree****: 3*

***Random state****: 10*

***Time****: 0.01647210121154785 seconds*

**Feature selection:**

* **Dropping title movie column**: it has no effect on the revenue.
* **Dropping director name column:** replaced by their popularity score.
* **Getting the top features:** using the absolute correlation more than ‘0.2’ and plotting the heat map.

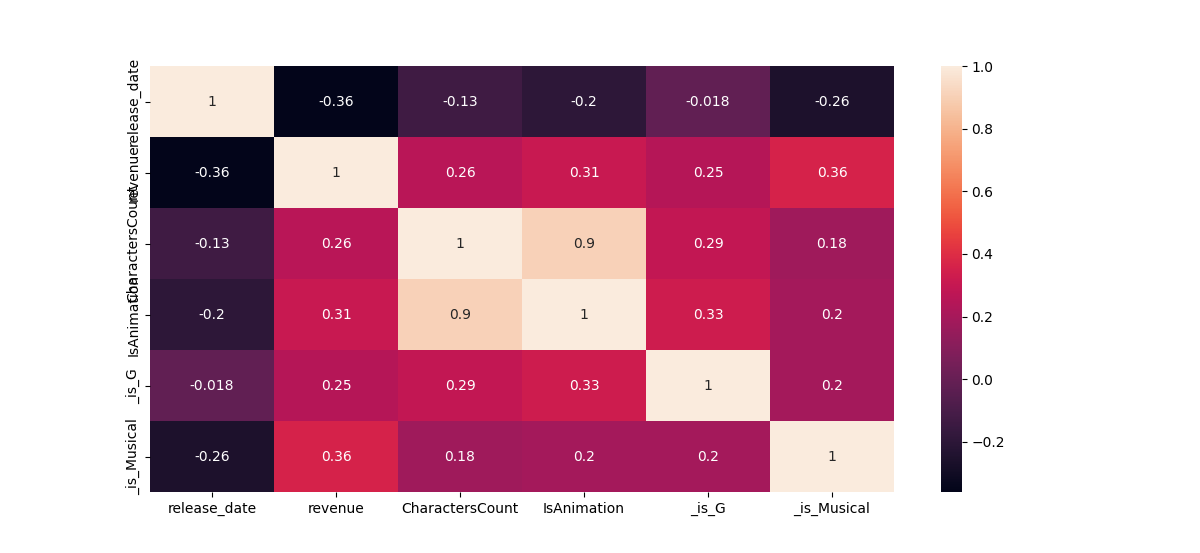
**Size of the training and testing sets:**

After several experiments we concluded that the best splitting is:

* testing set: 0.2 -> 20%
* training set: 0.8 -> 80%

**Screenshots:**

* Correlation heat map:



**Conclusion:**

In conclusion we can say that to be able to predict movie revenue more accurately you would need all the features and datasets you can get.

Working on this project was a worthy experience trying to figure out how to get the most accurate model with so little data about a movie.

We did our best trying to figure out how to process the initial datasets and join them together into a meaningful dataset that can develop a dependable model to predict what movie revenue would be.

Our trials started with figuring out what preprocessing techniques are the best to use on every categorical data provided, from the vast variety of encoding techniques we started to assess what is best for every column and try it to see if it works and what would produce the least mean square error possible.

It didn't end there; we then proceeded to choose which regression model should be applied to the outcome of the refined dataset.

Top features were release date, MPPA rating G, and Animation.

Release date was the least surprising outcome as it was easily anticipated because of the currency inflation throughout the years, the continuous development in the movie industry and the undeniable growing number of cinemas built every year.

MPPA rating G can be explained by simply breaking down what it means; G stands for General audience which means anyone from all ages can enter with no limitations whatsoever.

Animation was a bit surprising for me to jump over other features like action, thriller, drama and even comedy but when numbers don't lie and it can be explained by the revolutionary studios built to make these movies by big names like Steve Jobs. Also it targets young audience that can be very pressing.