Table of Contents

Mini Project 1. TMDB Box Office Prediction 1

Introduction and Objective3

Model Selection ……………………………………………………………………………………………………………………………3

Preprocessing the data……………………………………………………………………………………………………………….…3

Important Observations7

Predictive Models………………………………………………………………………………………………………………………….8

Linear Regression ………………………………………………………………………………………………………………….8

SVR ……………………………………………………………………………………………………………………………………….8

Random Forest Regressor ……………………………………………………………………………………………………9

Decision Tree Regressor ……………………………………………………………………………………………………….9

KNN Regressor ……………………………………………………………………………………………………………………10

Standardization …………………………………………………………………………………………………………………..11

Saving to csv ………………………………………………………………………………………………………………………11

Conclusion……………………………………………………………………………………………………………………………………11

References…………………………………………………………………………………………………………………………………..11

**MINI PROJECT 1. TMDB Box Office Prediction**

* 1. **Introduction and Objective:**

The Movie Database (**TMDb**) is a community-built movie and TV database. Data points provided include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries. The objective is to use machine learning algorithms to predict the overall worldwide box office revenue of a movie. This project is part of a Kaggle competition, the main goal is focused on answering the question, what will be the box office revenue of a movie.

* 1. **Model Selection:**

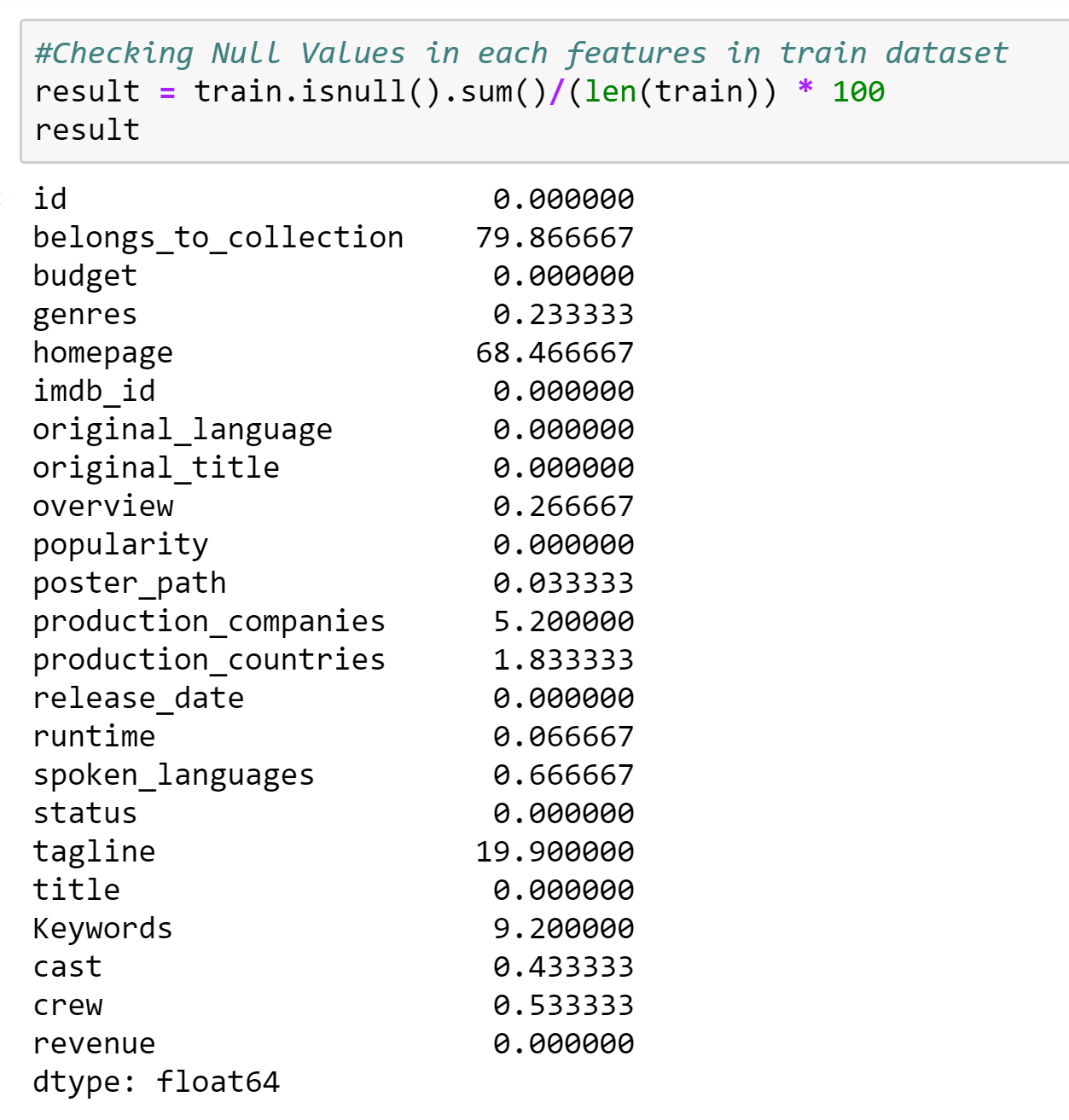
For the purpose of this project we have selected the following models to predict the best scoring model. Each of these models are discussed more in length, in the later parts of this report.

1. Linear Regression
2. Random Forest Regressor
3. SVR
4. Decision Tree Regressor
5. KNN Regressor
   1. **Preprocessing the data:**

Pre-processing the data is one of the most critical and the first steps before we can proceed creating our model. Cleaning up the data for missing values and finding out the features which will improve our score is important. Pre-processed data is used to build out models We typically utilized two separate files viz., train.csv and test.csv available on the Kaggle to train and use it on test and build our models respectively.

In order to pre-process our data, the following steps were employed:

1. From the train.csv file, we first identified that there were NULL values in some features of the data. We used the following code to identify the NULL values in train.csv



1. The genre feature was pre-processed by using only the genre value like drama, action, comedy etc and removing all the other values.

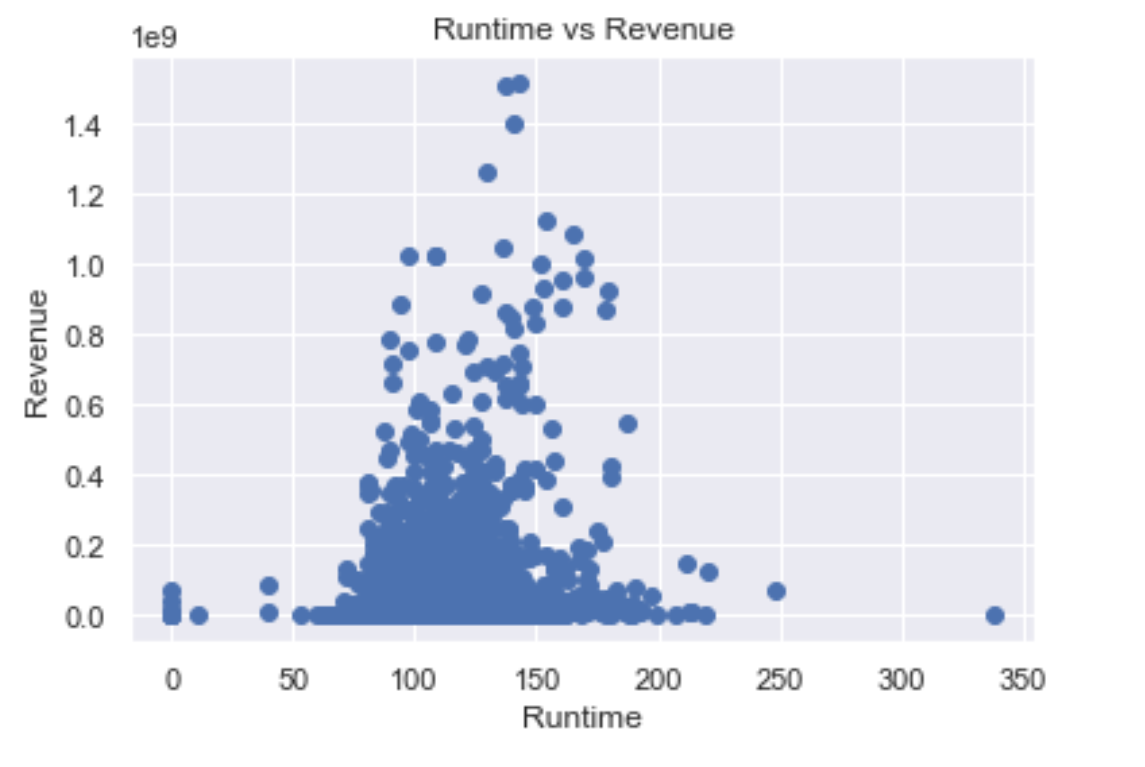


1. We tried world cloud to find out the values in genre feature . The result of the genre fetaure after pre-processig it :

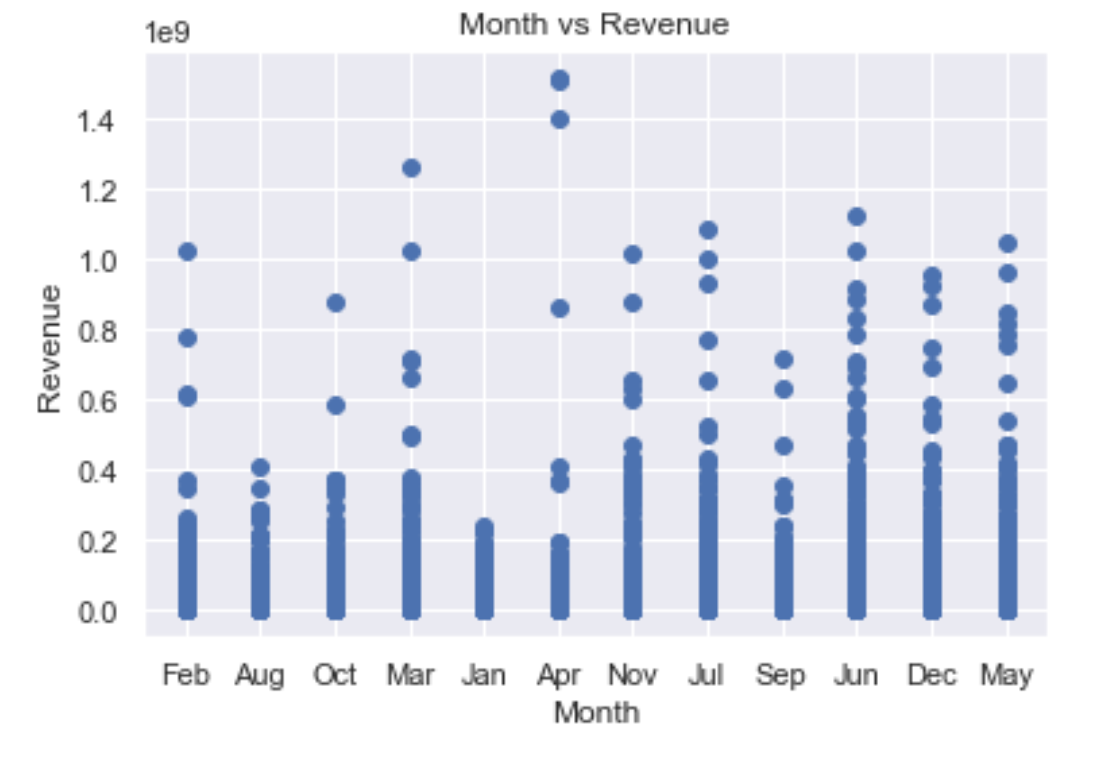


1. We tried to visualize how the runtime and month has an impact on the revenue.

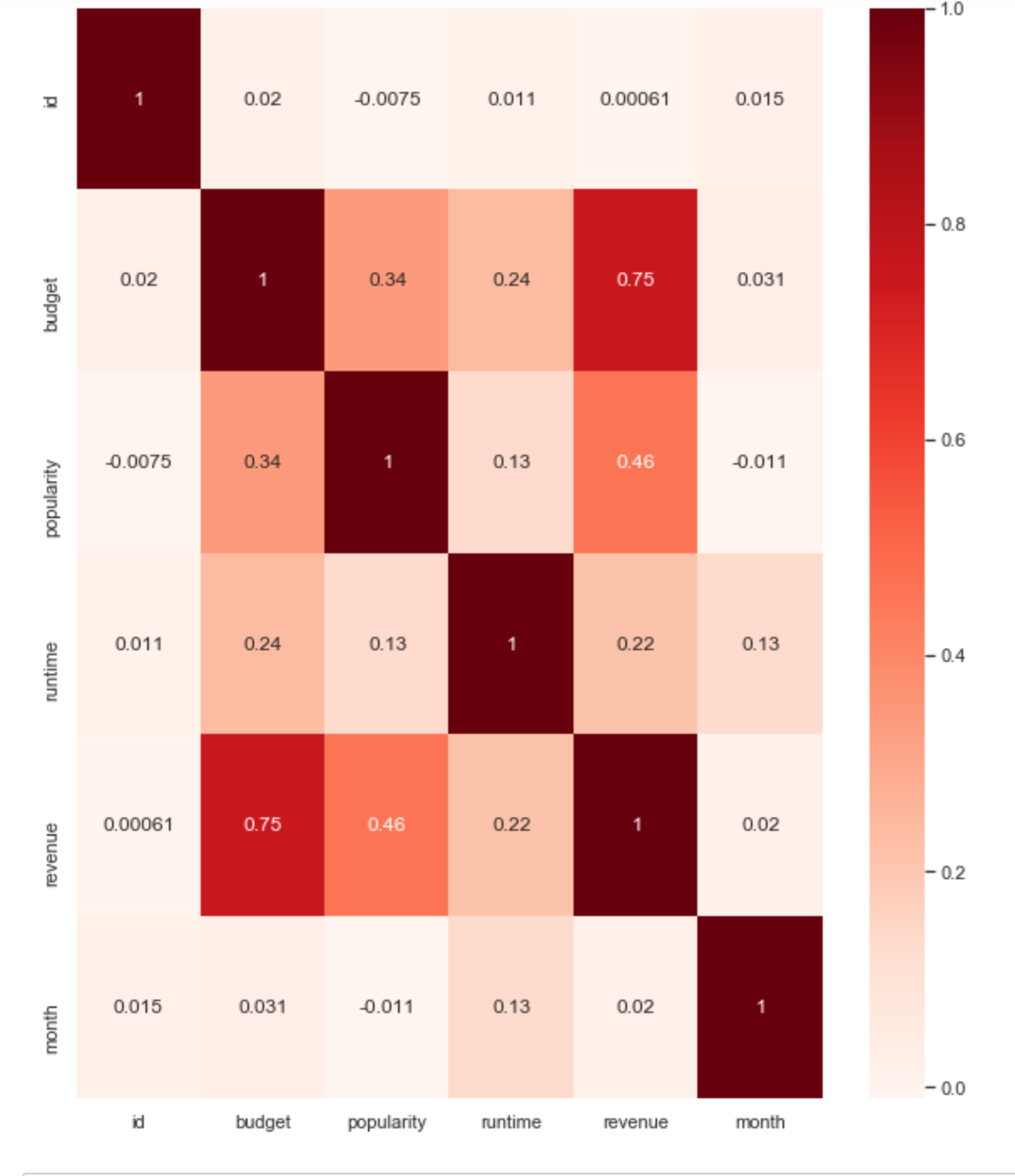
Our visualization result: scatterplot of runtime vs revenue



Scatterplot of month vs revenue

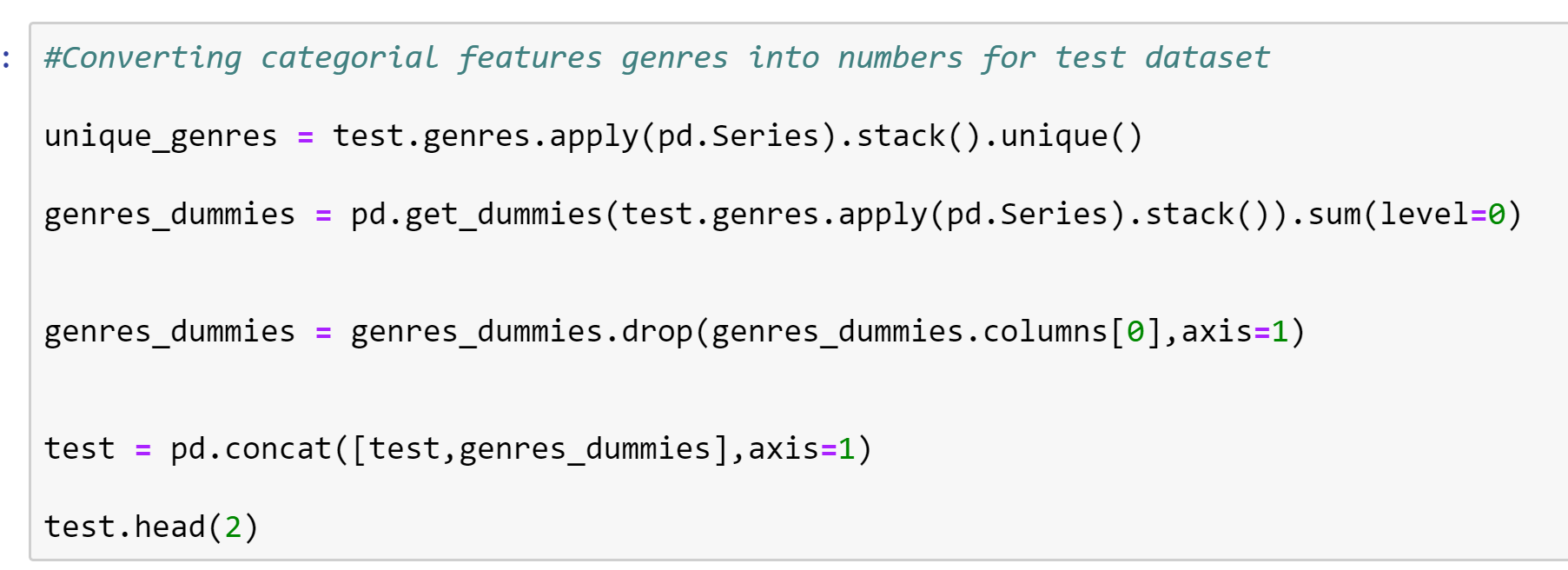


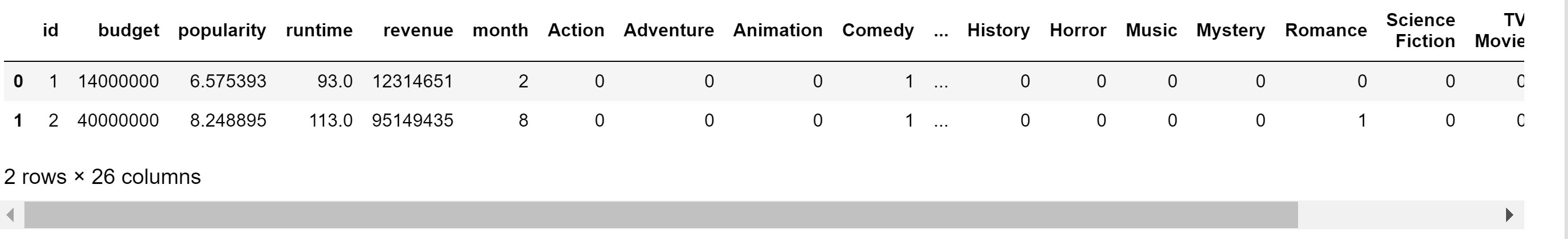
1. Using heat map, budget and popularity were correlated with revenue when compared with other features



1. Extracted only the genre value in that feature and dropped all the values.

1. Created dummies for the genre feature, and it will take a value of 1 if that movie belongs to that genre or else its zero.





1. Filled in the missing values in the feature runtime with the mean of its value.



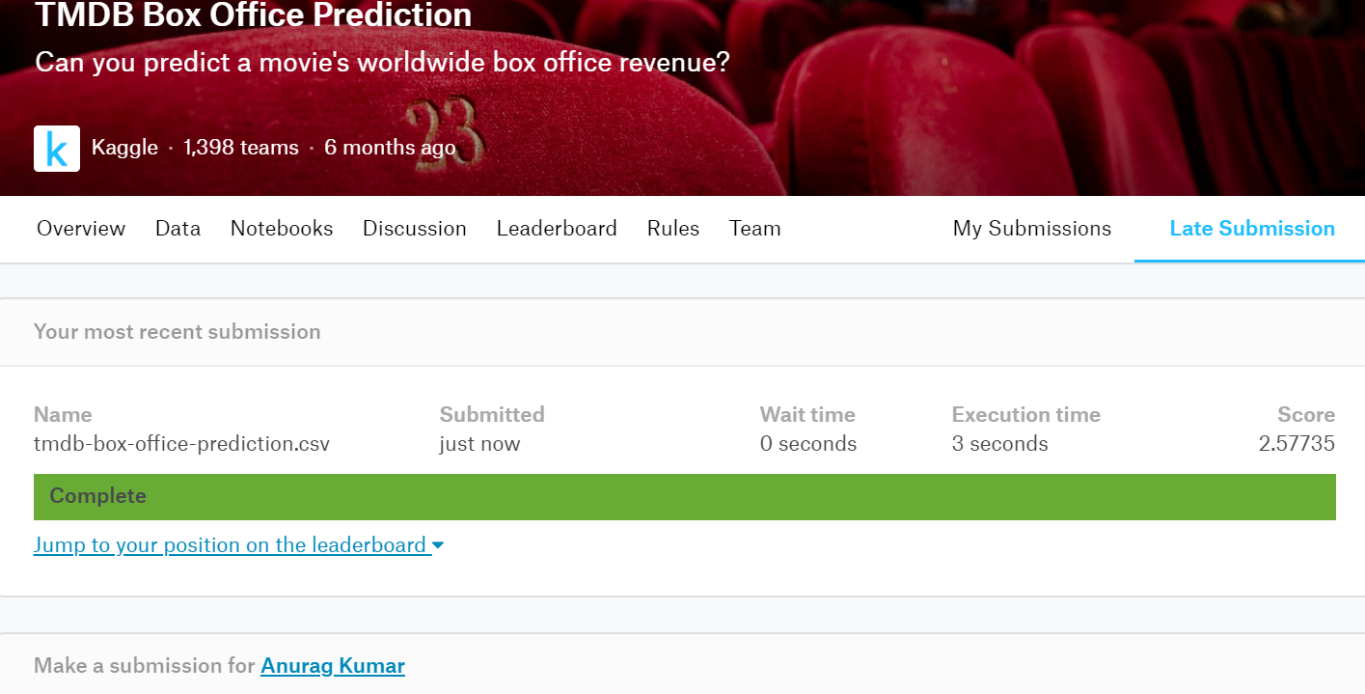
1. The final dataset after pre-processing looks like



**Important Observations:**

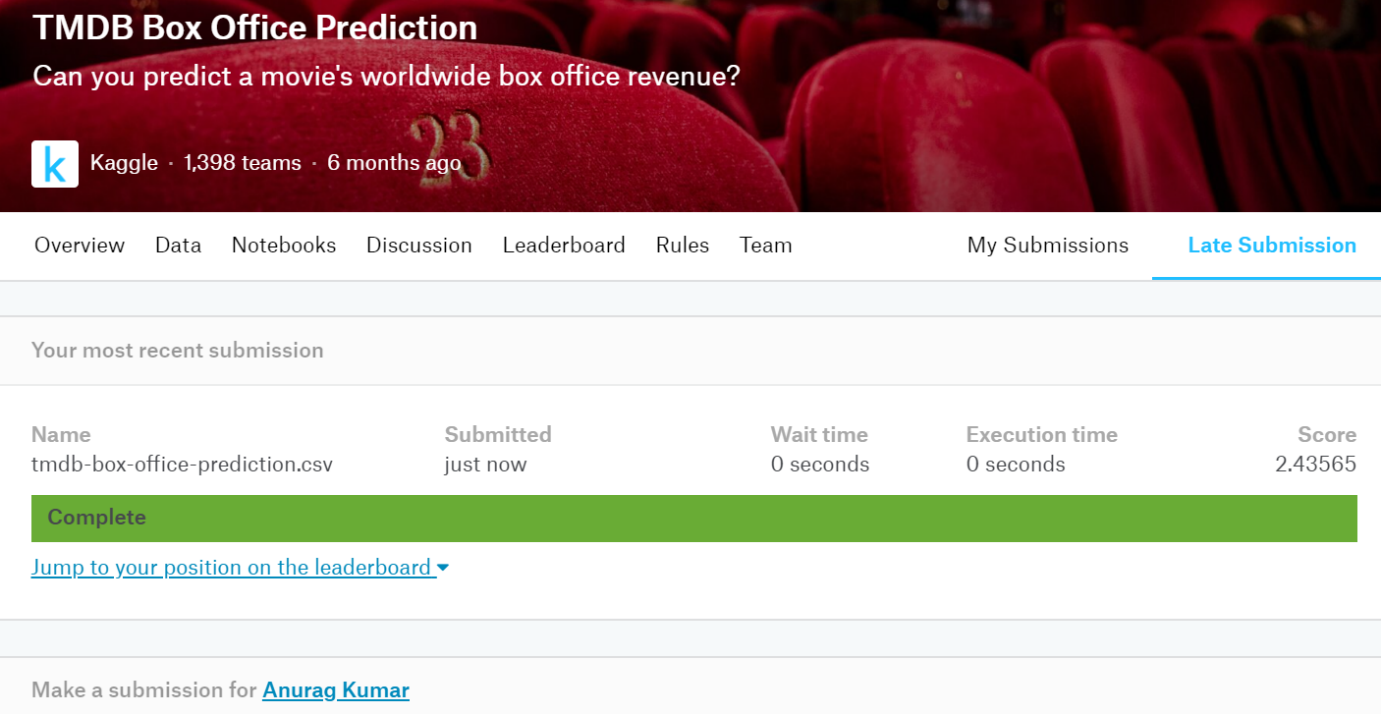
1. Only 2 features belongs\_to\_collection,homepage having 50% of their values are null.
2. Features with titled production\_companies,tagline,keywords having null values percentage as 5.2%,19.9% and 9.2% respectively.
3. The bar plot of unique gender values vs frequency shows genre Drama with maximum count
4. The scatter plot of revenue vs budget shows high correlation between these features.
5. From the correlation heatmap the feature budget is highly correlated(0.75) with revenue followed by feature popularity(0.46).
   1. **Predictive Models:**
      1. **Linear Regression:**

Linear regression is a linear approach to modelling the relationship between a target response (or dependent variable) and one feature (or independent variable). We predict the dependent variable (Revenue) using one or more independent features. We found it to be suitable and used the same. The Kaggle score was 2.57735 for linear regression.



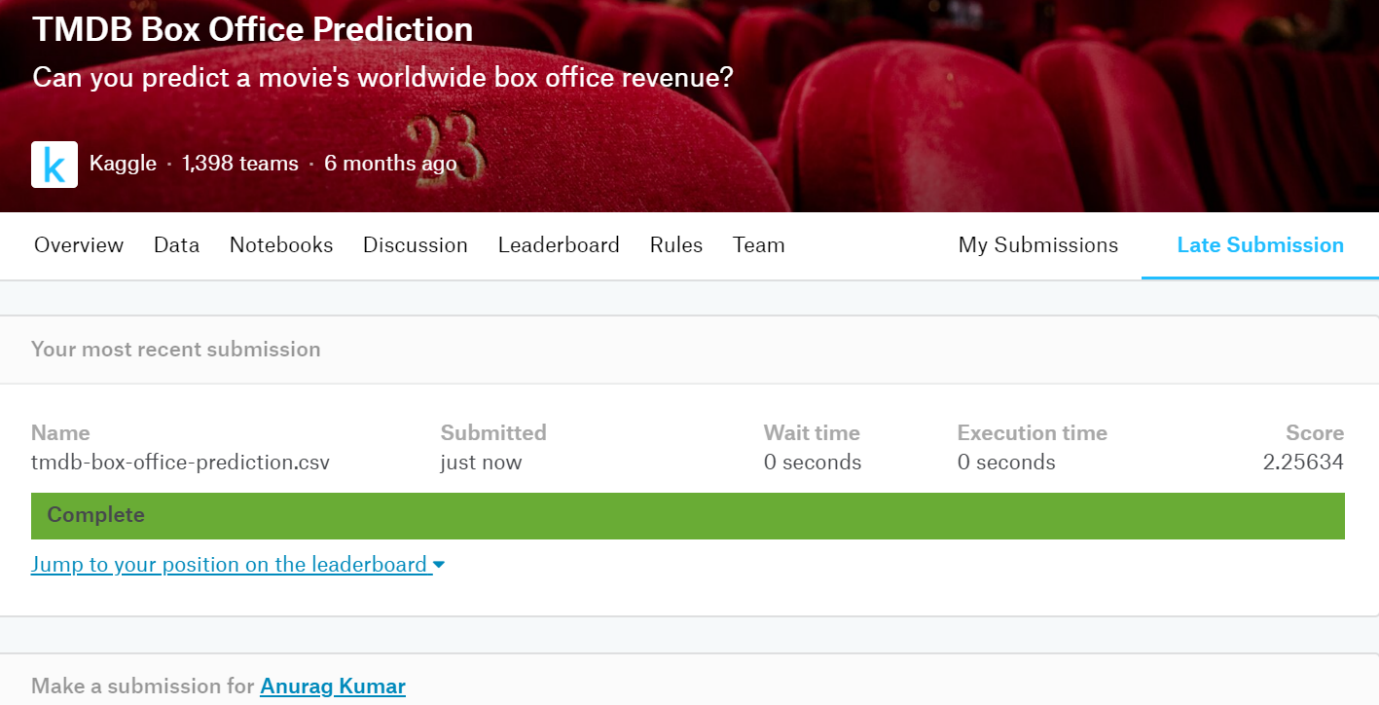
* + 1. **Support Vector Regressor:**

Support Vector Machines (SVMs) are a popular machine learning method for classification, regression, and other learning tasks. SVM regression is considered a nonparametric technique because it relies on kernel functions. In simple regression we try to minimize the error rate. While in SVR we try to fit the error within a certain threshold. The Kaggle score was 2.43565 for SVR.



* + 1. **Random Forest Regressor:**

An ensemble of randomized decision trees is known as a random forest. This algorithm on a random basis selects random samples and get the prediction based on the samples. We give the no of estimators we need to predict the model. The Kaggle score was 2.25634 for Random forest.



**1.4.4 Decision Tree Regressor:**

Decision trees are predictive models that use a set of binary rules to calculate a target value. Decision trees regression normally use [mean squared error (MSE)](https://en.wikipedia.org/wiki/Mean_squared_error) to decide to split a node in two or more sub-nodes. The Kaggle score was 2.91063 for Decision Tree Regressor.

A screenshot of a cell phone

Description automatically generated

**1.4.5 KNN Regressor**

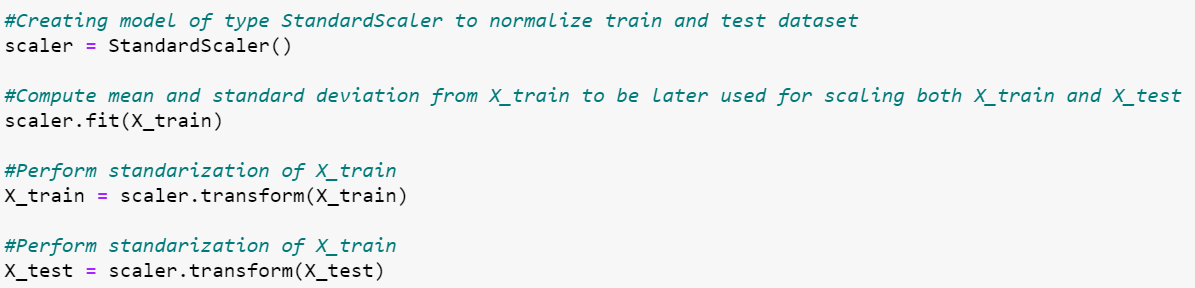
K-Nearest Neighbors algorithm (KNN) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). In KNNregression, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors. The Kaggle score was 2.91182 for KNN Regressor.

A screenshot of a cell phone

Description automatically generated

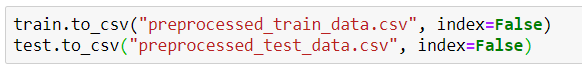
**Standardization:**

1. We standardized the entire data using StandardScaler() after preprocessing the data.



**Saving to csv:**

1. The preprocessed train and test files have been saved to “preprocessed\_train.csv” and “preprocessed\_test.csv” respectively.



**1.6 Conclusion:**

Random Forest produces the highest score on Kaggle. Furthermore, it prevents overfitting the data by using only a limited set of samples for each estimator. All other regression models like Linear Regression,DecisionTreeRegressor,KNNRegressor,Support Vector Regressor also performed well on this regression task. Random Forest Regressor being an ensemble learning algorithm that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model, hence this model having highest score among all the other machine learning models.

**1.7 References:**

* <https://www.themoviedb.org/>
* <https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4>
* <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa>
* <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
* <https://gdcoder.com/decision-tree-regressor-explained-in-depth/>
* <https://www3.nd.edu/~rwilliam/stats2/l25.pdf>
* <https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f>
* <https://scikit-learn.org/>
* <https://matplotlib.org/>
* All the materials provided in this course.