OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH

DUBLIN

TECHNOLOGICAL
UNIVERSITY DUBLIN

TMDB Box Office Prediction

Harsh Vardhan Rai
D20123653
TU059

Machine Learning SPEC9270: 2020-21
Full Time
02nd May 2021

Introduction

The film industry boasts one of the most successful business models, with demands meeting results more often than not. Box offices host audiences in large numbers, and when the objectives of the movie align with the target audience, the income goes through the roof. For instance, a successful annual business at the box office was recorded in 2018, with sales marginally crossing a cool $41.7 billion. The numbers speak for themselves, and indicate the potential for growth in the film industry, and thereby, the revenue collected by box offices. The research aims at cornering the very reason/reasons that induce the prosperity of the box offices, by analysing metadata from over 7000 films as an attempt to not only understand the major contributing factors to a successful business, but also to predict the worldwide box office revenue. Spearheaded by several influencing parameters such as data points of cast, crew, plot keywords, budgets, posters, release dates and several more, the project will collect said data and administer it to predict the revenue based outcomes.

Herein, in this project a comparative analysis was performed on **Linear Regression, Random Forest**, **XG Boost** and **LG Boost** models to find the best suitable model for revenue prediction. The exploratory data analysis included data cleaning, label encoding, and data pre-processing that is introduced in the exploratory data analysis section further, the four regressor models are discussed. The evaluation metrics used to show the superiority of our models is talked about in the evaluation and the result section . Finally, the last section draws conclusion to this research and describes future scope.

Literature Review

Movie revenue prediction is said to be one of the most important problems faced by the film industry that administers decision making by the directors and producers . Work has been conducted on identifying patterns and relationships between factors defining movies and the revenue it could generate. Models like linear regression, polynomial regression and support vector regression (SVR) were applied on the entire movie data to predict the movie revenue. In addition to this, clusters were generated on features like genre by using techniques like Expectation Maximization and K-means [2]. Evaluation metrics like **R square** and **RMSE** were used as performance indicators. In yet another work, a neural network approach was used to predict the probability of movies(between 2010 to 2015) to fall under classes like “Profit” and “loss” [3] .

Research conducted in [1] concluded that algorithms like multiple linear regression and support vector machines are the most commonly used prediction models. Besides, mean percentage error, root-mean-square error are commonly used evaluation indicators.

Data Description

TMDB Box Office Prediction - Can you predict a movie’s worldwide box office revenue ?

Link: https://www.kaggle.com/c/tmdb-box-office-prediction

The main objective of this research is predicting the revenue of box office as denoted by the target variable revenue. The dataset is taken from “TMDB Box Office Prediction” sourced from Kaggle. The dataset is broken into two files train and test. In this dataset around 7398 movies along with their metadata is described. Movies are labelled with an ID . Metadata includes features like cast, crew, keywords, poster path, budget and a few more. The prediction is done on the test file for around 4398 movies. In addition to this, supplementary features are added to both test and train which is aligned to this competition aiming to make our models more accurate. It is very important to analyse and understand the data , explore data, describe data and verify data quality. The shape of the dataset includes around 3000 instances with over 26 features. But few features like Genre, Production companies, Production countries and many more were in JSON format containing metadata within itself. It became important to extract the desired information. This will be discussed in the later sections.

The dataset is a mixture of both categorical and discrete values (as seen in Figure1) which makes it important to understand significance of these features.

Type of features	Features
Categorical	Belongs_to_collection, genres, homepage, original_language, original_title, overview, status, tagline, title, keywords, cast, crew
Numerical	Budget, popularity, popularity2, revenue, rating, totalVotes

Figure 1: Feature Description

Exploratory Data Analysis

Exploratory Data Analysis is an approach of examining and summarising datasets to generate insights about their characteristics with the help of statistical graphs and other data visualisation methods. The primary steps of this process is to gain intuitions about the data, conduct sanity checks, figure out missing values, identify outlier and then summarizing the data.

Data Pre-processing

The dataset consists of two main tables, train and test. For convenience and better exploration additional features data has been added to them . The primary goal of data preparation is to organise and clean the data for further modelling. In this project a lot of quality checks were performed:

- A lot of entries in the dataset were incorrect and would lead to decrease in quality information but discussions in the competition forum provided information about rectifying these **erroneous data**. The entries were straightaway taken or both the datasets without any modifications.
- The **format** of the release date column was unclear for the requirements. It became necessary to segregate them into release_year, release_day and release_month for both the datasets.

- A lot of machine learning algorithms perform poorly with categorical data this is where **Feature Encoding** comes into picture. In the dataset features were encoded with 0/1 (**One Hot Encoding**).

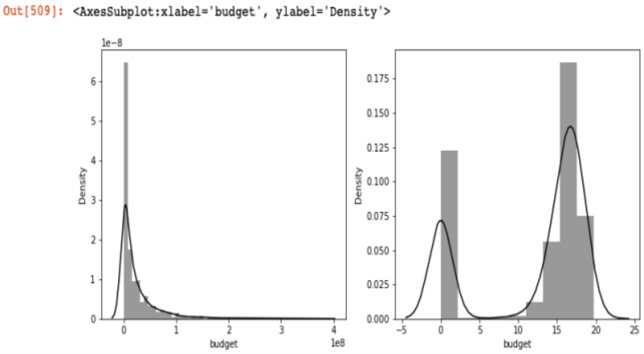
- As discussed previously a lot of features were in **JSON format** with metadata within them. This data cannot be crawled through a basic data frame, so these type of entries were e converted initially to dictionary dtype and later the data which was required for data modelling was extracted into individual columns.

- A lot of dataset may contain **missing values** for various reasons such as undefined values, data collection errors etc. A machine learning model can be drastically impacted by these missing values. In this project missing values were imputed by either mean or mode based on their variable type and the intervals in which they were missing.

- Corelation can help predict one feature from another. Relationships were analysed between the features and the target variable. Feature engineering is performed which resulted in adding **additional calculated features** to increase accuracy.

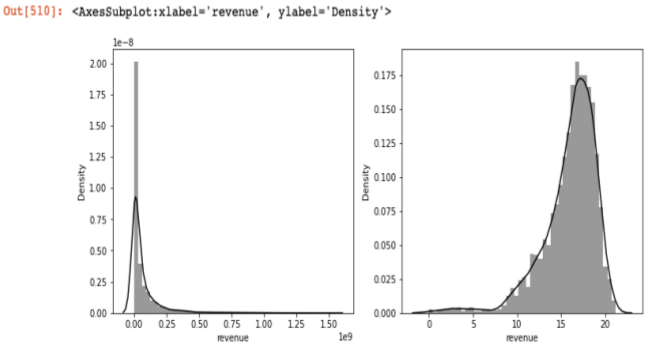
- The next quality check performed was to identify **duplicates**. However no redundant values were found.

- The budget and revenue feature were highly skewed (as seen in Figure 2 and 3) respectively which could be representative of the entire dataset. .To normalise these log functions are used. But, basic log function can result in values like infinity and Nan . Hence, the approach was to use **log1p** .



```
Out[509]: <AxesSubplot:xlabel='budget', ylabel='Density'>
```

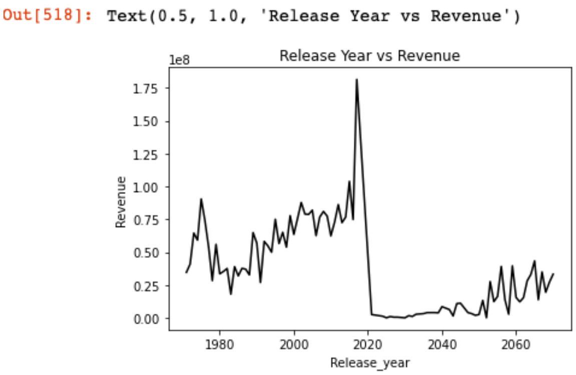
Figure 2: Skewness of budget and revenue



```
Out[510]: <AxesSubplot:xlabel='revenue', ylabel='Density'>
```

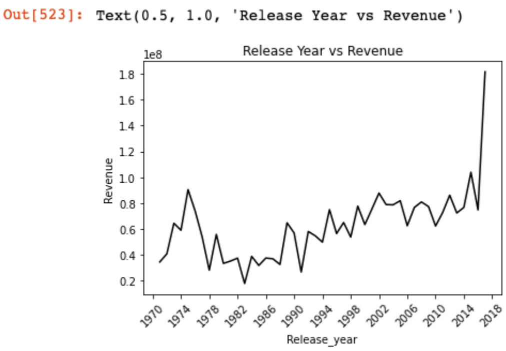
Figure 3: Budget and Revenue Normalization

- **Data visualization** provides underline insights which can otherwise be sometimes missed. For example in this dataset a lot of **entries were made for the year till 2070** (figure 4). But, predicting a revenue for such movies is quite impossible. It raises a question of how one can find a movie's revenue that has not yet been released. Therefore such records were removed (figure 5).



```
Out[518]: Text(0.5, 1.0, 'Release Year vs Revenue')
```

Figure 4: Release year vs revenue till ***2060***



```
Out[523]: Text(0.5, 1.0, 'Release Year vs Revenue')
```

Figure 5: Corrected data for release year vs revenue

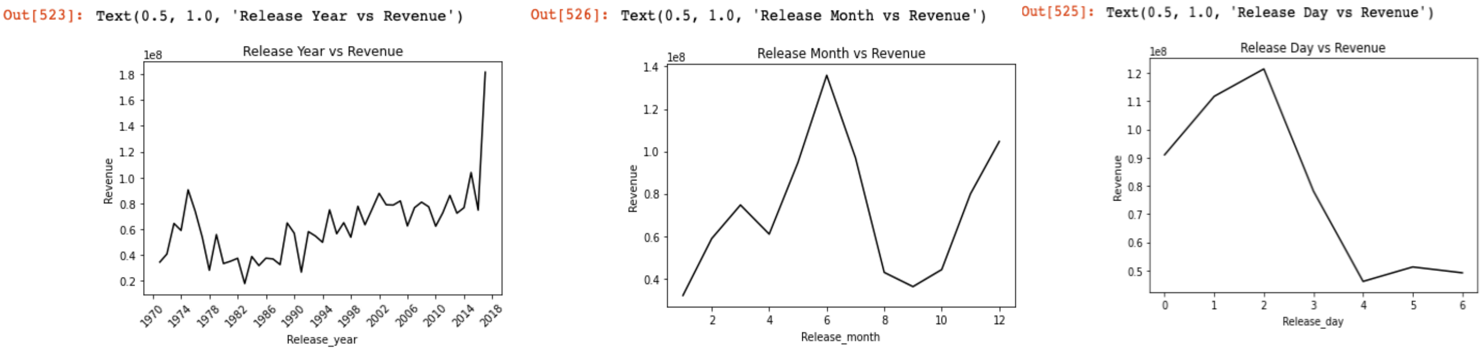
- With the help of **descriptive statistics** it was observed that revenue had a minimum value of 1 which is unusual. Since this is our target variable it is incorrect to modify this feature by imputing it for any other value as it may affect our models. Around a **minimum 25% instances of budget** were identified as 0. This seems irrelevant, the data however was handled in a similar manner as those of incorrect entries described in the first point.

- Finally, unwanted columns were dropped.

Data Insights

One of the most important strategy to turn data into actionable insights is to use clear visualisations to convey the message. Now that the data is cleaned and normalised it becomes easier to understand how the target variable is affected by the features. Some of the important insights are as follows :

- **Release year VS Revenue** : The plot shows a decreasing trend ,from the year 1970 to 1982 where it dips to the minimum nevertheless the revenue has been increasing since and has peaked in 2018. The probable reason in this increase would be improved infrastructure, advance advertising techniques, foreign sales and many more. Similar plots are generated for release _month and release_day against revenue (Figure 6).



```
Out[523]: Text(0.5, 1.0, 'Release Year vs Revenue')
```

```
Out[526]: Text(0.5, 1.0, 'Release Month vs Revenue')
```

```
Out[525]: Text(0.5, 1.0, 'Release Day vs Revenue')
```

Figure 6: Visualisation of release year, release month and release day vs revenue

- Another important analysis was to understand the number of movies released vs the day of the week. It can be seen that majority of the movies were released on Friday. This could be one of the tactics to improve the revenue over the weekends (Figure 7).

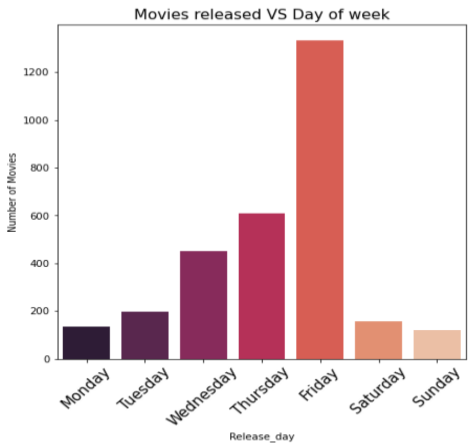


Figure 7: Plot of count of movies released for each day

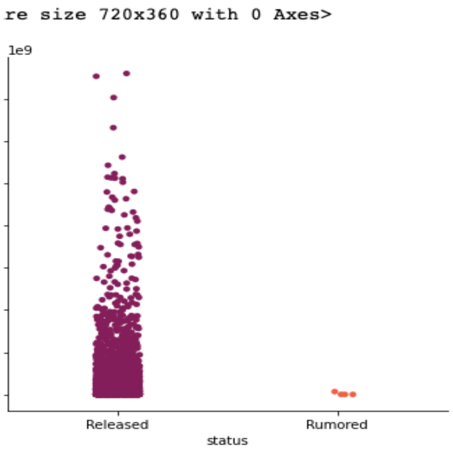
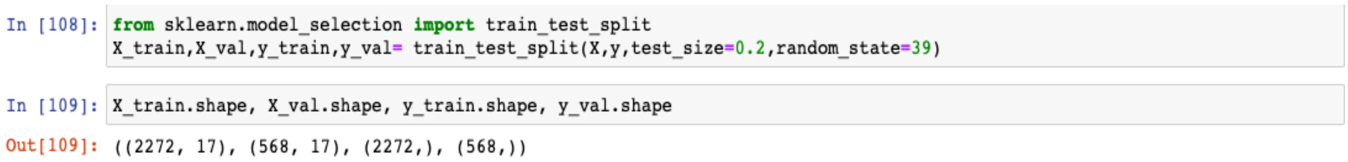


Figure 8: Distribution of movies for movie status

- 99.8% of the movies are released while 0.2% have the status of rumoured. This could lead to feature being imbalanced with 99.8% of data under a variable (Figure 8).

Data Modelling Implementation and Evaluation – Machine Learning Algorithms

The dataset for initial local evaluation has been divided into: **(80:20 split)**
X_train, X_val
Y_train, y_val



```
In [108]: from sklearn.model_selection import train_test_split
          X_train,X_val,y_train,y_val= train_test_split(X,y,test_size=0.2,random_state=39)

In [109]: X_train.shape, X_val.shape, y_train.shape, y_val.shape
Out[109]: ((2272, 17), (568, 17), (2272,), (568,))
```

This was done so that we can evaluate our models locally. Furthermore, this split was used for every model training to get our evaluation metrics "**RMSE**" values.

Multi-variate Linear Regression

Linear Regression is the most basic and widely used technique when we are predicting a target variable with reference to dependent variable. Before proceeding for the technique there are a few assumptions that we should be aware of ; the variables should follow a normal distribution here should be a little/no threshold for multicollinearity, there should be no autocorrelation, the data points are homoscedastic (the variables have same finite variance) and we are exploring a linear relationship

I choose this algorithm as this is the basic when we are trying to predict a continuous variable in this case it's "Revenue". The features that we are passing to train this model are:

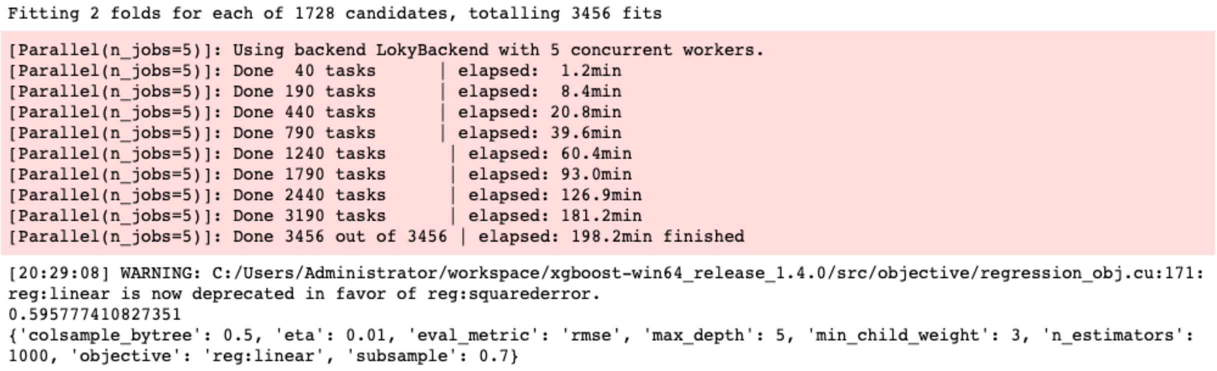
```
Index(['id', 'budget', 'homepage', 'popularity', 'runtime', 'rating',
        'totalVotes', 'release_year', 'release_month', 'release_day',
        'numberofgenres', 'lang_english', 'bud_runtime', 'bud_year',
        'bud_popularity', 'runtime_year', 'popularity_year'],
       dtype='object')
```

Random Forest Regressor

It's a supervised learning algorithm which uses ensemble learning for our regression task. It uses a bagging technique (aggregating the outputs in end reducing the variance). It's a meta-estimator as it combines the results of different predictions. This involves developing a collection of decision trees by focusing on the records and features of the training data, and classifying records based on the maximum votes for each class. This method minimises overfitting in decision trees and helps improve the accuracy. However, the process is quite slow, as it combines many trees. I choose this as it can solve both type of problems of classification and regression as it provides a decent estimation at both fronts. The only drawback I found was we have a very little control over the model, there's nothing specific you can play around except from the parameter and seed.

eXtreme Gradient Boost (XGBoost)

XGBoost is a supervised learning algorithm and works on the concept of decision trees. It is an **ensemble approach** that utilizes a gradient boosting framework. The advantages of this approach are high scalability and accurate implementation with good computing power for boosted trees. **Grid Search cross validation** for this algorithm derived the best parameters that would perform better. XGBoost showed good results with the **least RMSE value.** Furthermore, in order to optimise the model we have performed hyper-parameter tuning with the use of Grid Search Cross Validation. This was done after the data was split into training and test sets. Using this method, different hyper-parameters were tested on the training data and after the optimum parameters were selected, which resulted in improved performance.



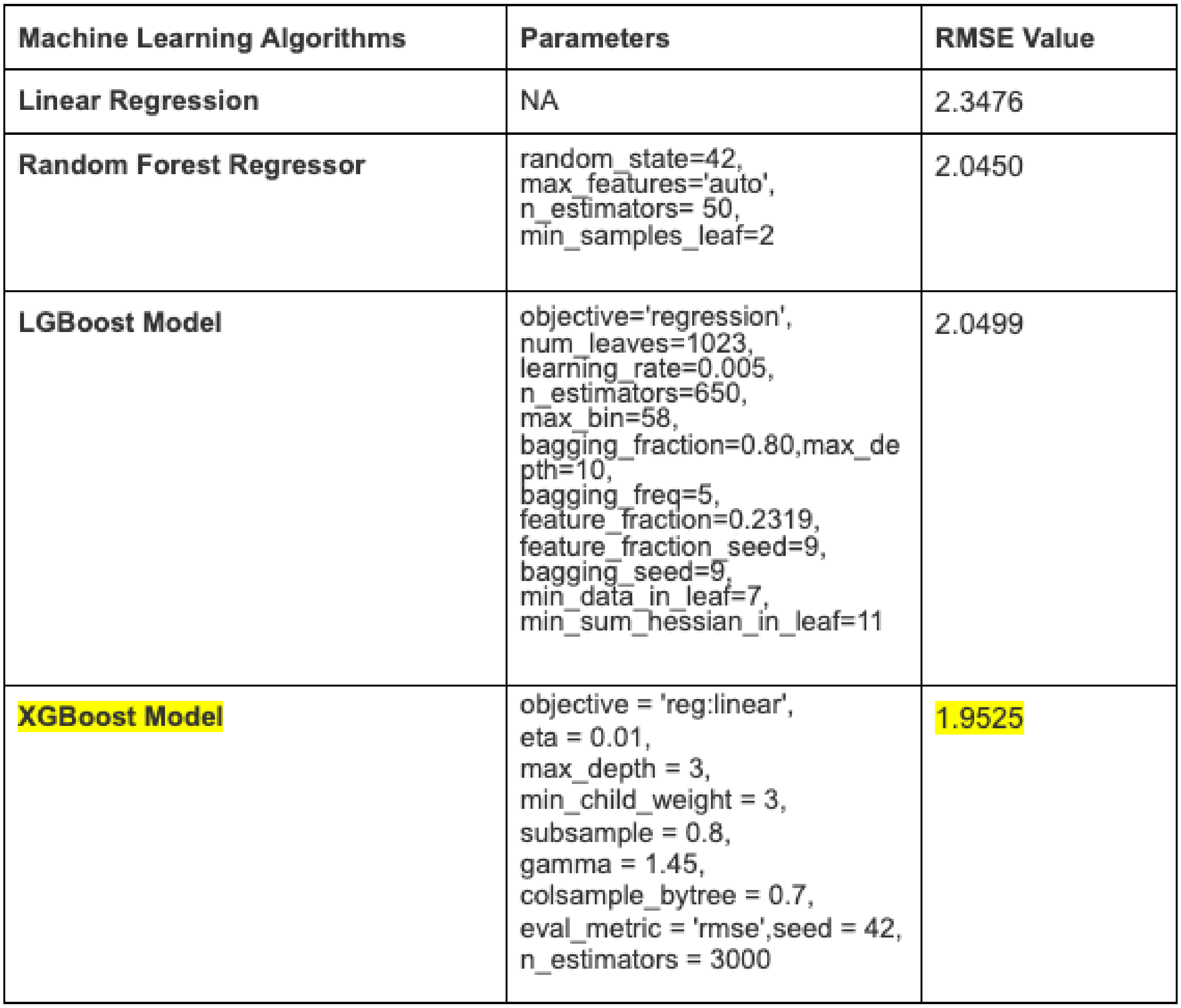
```
Fitting 2 folds for each of 1728 candidates, totalling 3456 fits

[Parallel(n_jobs=5)]: Using backend LokyBackend with 5 concurrent workers.
[Parallel(n_jobs=5)]: Done  40 tasks      | elapsed:  1.2min
[Parallel(n_jobs=5)]: Done 190 tasks      | elapsed:  8.4min
[Parallel(n_jobs=5)]: Done 440 tasks      | elapsed: 20.8min
[Parallel(n_jobs=5)]: Done 790 tasks      | elapsed: 39.6min
[Parallel(n_jobs=5)]: Done 1240 tasks      | elapsed: 60.4min
[Parallel(n_jobs=5)]: Done 1790 tasks      | elapsed: 93.0min
[Parallel(n_jobs=5)]: Done 2440 tasks      | elapsed: 126.9min
[Parallel(n_jobs=5)]: Done 3190 tasks      | elapsed: 181.2min
[Parallel(n_jobs=5)]: Done 3456 out of 3456 | elapsed: 198.2min finished

[20:29:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/objective/regression_obj.cu:171:
reg:linear is now deprecated in favor of reg:squarederror.
0.595777410827351
{'colsample_bytree': 0.5, 'eta': 0.01, 'eval_metric': 'rmse', 'max_depth': 5, 'min_child_weight': 3, 'n_estimators':
1000, 'objective': 'reg:linear', 'subsample': 0.7}
```

The best thing with XGB is that we don’t have to normalise the features as you need to do with other models. However, we need to keep a check on the noisy data as they may overfit.

Light Gradient Boost (LGBoost)

LGB is a histogram based algorithm that generates highly complex trees and is scalable when the number of records is high. For optimising the parameters after the data was split into training and test sets, we can implement hyper-parameter tuning using Grid Search Cross Validation (as in the case with XGB). The advantages of using LGBM is they provide faster speed with low memory usage with compatibility with large datasets.

Important parameters: max_depth(depth of tree), min_child_samples(minimum data to group in a leaf), num_leaves(leaf nodes to use)



Machine Learning Algorithms	Parameters	RMSE Value
Linear Regression	NA	2.3476

Random Forest Regressor	random_state=42,
	max_features='auto',
	n_estimators= 50,
	min_samples_leaf=2
		2.0450

LGBoost Model	objective='regression',
	num_leaves=1023,
	learning_rate=0.005,
	n_estimators=650,
	max_bin=58,
	bagging_fraction=0.80,max_depth=10,
	bagging_freq=5,
	feature_fraction=0.2319,
	feature_fraction_seed=9,
	bagging_seed=9,
	min_data_in_leaf=7,
	min_sum_hessian_in_leaf=11
		2.0499

XGBoost Model	objective = 'reg:linear',
	eta = 0.01,
	max_depth = 3,
	min_child_weight = 3,
	subsample = 0.8,
	gamma = 1.45,
	colsample_bytree = 0.7,
	eval_metric = 'rmse',seed = 42,
	n_estimators = 3000
		1.9525

Conclusion and Discussion

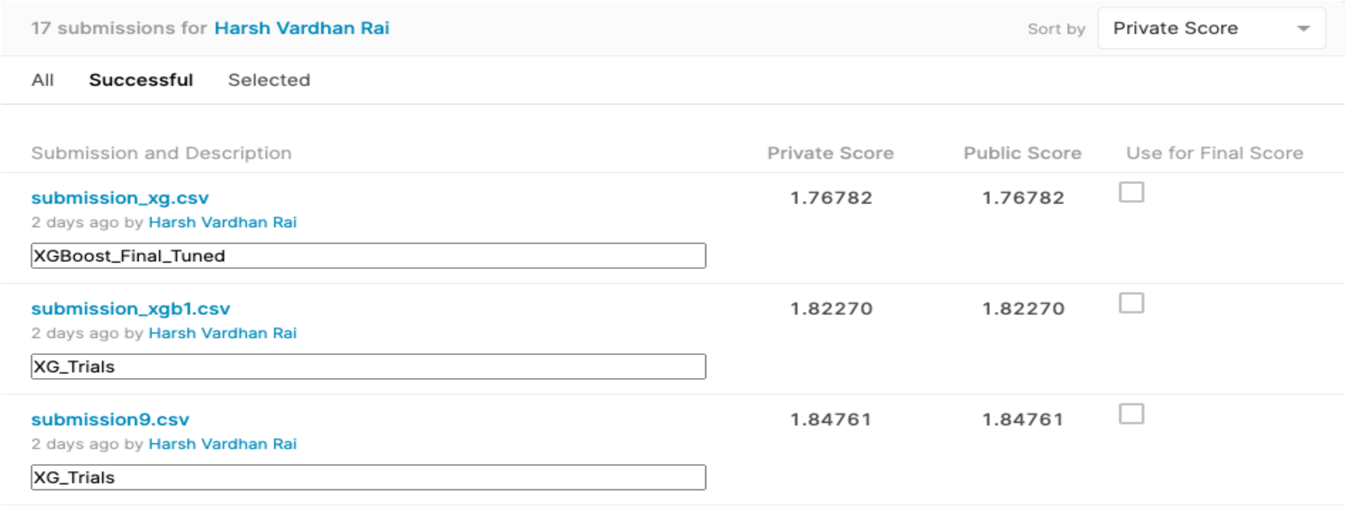
The main aim of this research was to answer the question about movie prediction. Initially, the research identified this problem as that of regression with multiple features to predict the target value called revenue. The most popular algorithms identified for movie revenue predictions were Linear Regression, SVR, NN, Random Forest and XGBoost. In addition, **Grid search Cross Validation** technique helped identify the critical parameters used for **Hyperparameter Tuning** to improve accuracy. The final stage was to identify the evaluation metric for evaluating the prediction algorithm. Consequently, the **RMSE metric** was used for this problem. In conclusion, **XGBoost** performed better when compared to other algorithms(**1.76 RMSE**) with hyper parameter tuning.

However, as part of future work, we could consider user opinions and conduct NLP and sentiment analysis of the feature. This could predict the revenue that a movie would make on the day of the release.

In conclusion, this work has presented some important aspects in the domain movie revenue prediction.

Kaggle Ranking

Kaggle Standing: As the competition was closed there was only evaluation which was possible. There were **1615** competitors and my rank manually found was **286** which calculates it to be in **Top 20%.**



17 submissions for Harsh Vardhan Rai

Sort by Private Score

Submission and Description	Private Score	Public Score	Use for Final Score
submission_xg.csv 2 days ago by Harsh Vardhan Rai XGBoost_Final_Tuned	1.76782	1.76782	□
submission_xgb1.csv 2 days ago by Harsh Vardhan Rai XG_Trials	1.82270	1.82270	□
submission9.csv 2 days ago by Harsh Vardhan Rai XG_Trials	1.84761	1.84761	□

References

[1] I. Said Ahmad, A. Bakar, M. R. Yaakub, and S. Hassan Muhammad, “A Survey on Machine Learning Techniques in Movie Revenue Prediction,” *SN Computer Science*, vol. 1, Jul. 2020, doi: 10.1007/s42979-020-00249-1.

[2] P. Walanaraya, W. Puengpipattrakul and D. Sutivong, "Movie Revenue Prediction Using Regression and Clustering," *2018 2nd International Conference on Engineering Innovation (ICEI)*, 2018, pp. 63-68, doi: 10.1109/ICEI18.2018.8448610.

[3] T. Rhee and F. Zulkernine, *Predicting Movie Box Office Profitability: A Neural Network Approach*. 2016, p. 670.