COMP4211 Project – TMDB Box Office Prediction

Shayan Imran

20316792

2019

Contents

[Introduction 2](#_Toc8309395)

[Dataset and Preprocessing 3](#_Toc8309396)

[Dropped Columns 3](#_Toc8309397)

[One Hot Encoding 4](#_Toc8309398)

[original\_language 4](#_Toc8309399)

[genres 5](#_Toc8309400)

[production\_countries 5](#_Toc8309401)

[production\_companies 5](#_Toc8309402)

[Keywords 6](#_Toc8309403)

[Binary Transformation 6](#_Toc8309404)

[belongs\_to\_collection 6](#_Toc8309405)

[homepage 6](#_Toc8309406)

[Release Date Transformation 6](#_Toc8309407)

[Machine Learning Task 7](#_Toc8309408)

[Machine Learning Methods 8](#_Toc8309409)

[LASSO 8](#_Toc8309410)

[Results 8](#_Toc8309411)

[Ridge Regression 8](#_Toc8309412)

[Results 8](#_Toc8309413)

[Decision Tree Regression 8](#_Toc8309414)

[Results 8](#_Toc8309415)

[Random Forest Regression 8](#_Toc8309416)

[Results 8](#_Toc8309417)

[XGBoost Regression 8](#_Toc8309418)

[Results 8](#_Toc8309419)

[MLPRegressor 9](#_Toc8309420)

[Results 9](#_Toc8309421)

[Discussion 9](#_Toc8309422)

[Computing Environment 9](#_Toc8309423)

# Introduction

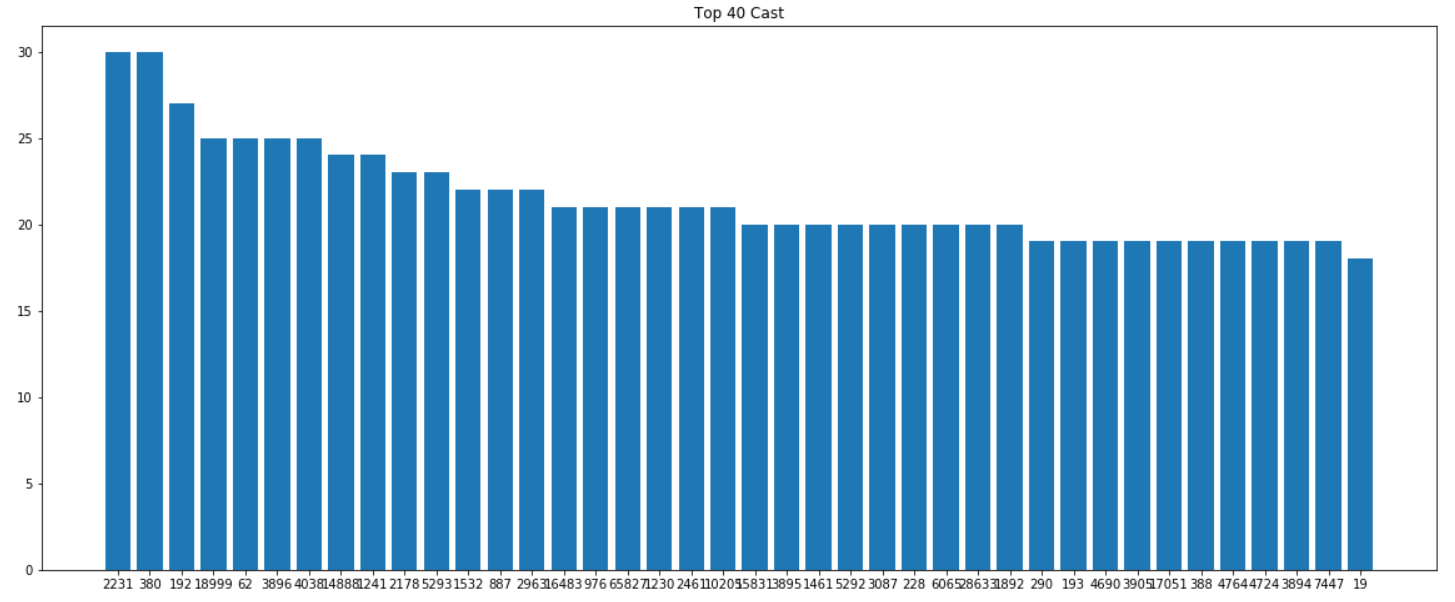
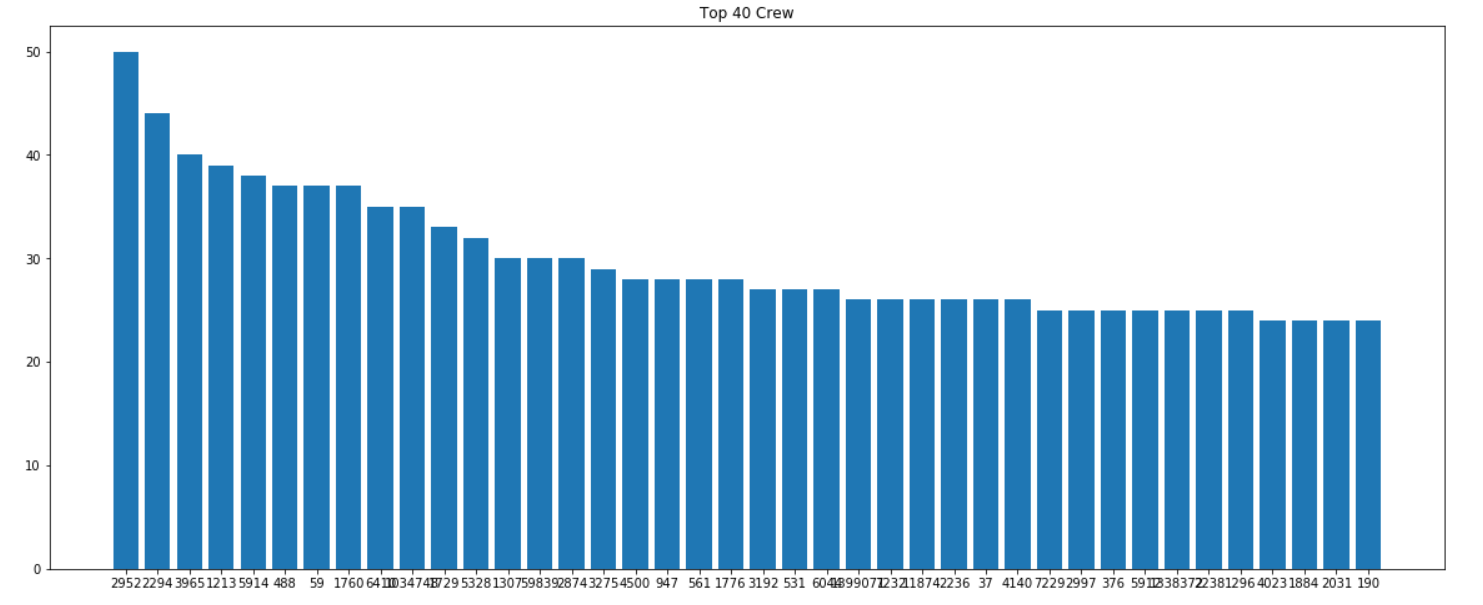
This project presents experimentation using multiple types of machine learning techniques on the [TMDB Box Office Prediction Kaggle Competition](https://www.kaggle.com/c/tmdb-box-office-prediction/overview) (<https://www.kaggle.com/c/tmdb-box-office-prediction/overview>). The aim of the competition is to predict the revenue for movies given metadata for over 7000 past movies.

# Dataset and Preprocessing

The dataset provided in this competition consists of 22 columns of metadata for 7398 movies from the TMDB movie database. Only 3000 of the movies in this list contain a 23rd column containing the revenue earned by the movie, predicting which is the target of this competition. These 3000 rows are therefore used to train the machine learning models that are used.

## Dropped Columns

The following columns were dropped as they are believed to be irrelevant and do not contribute much to the machine learning task:

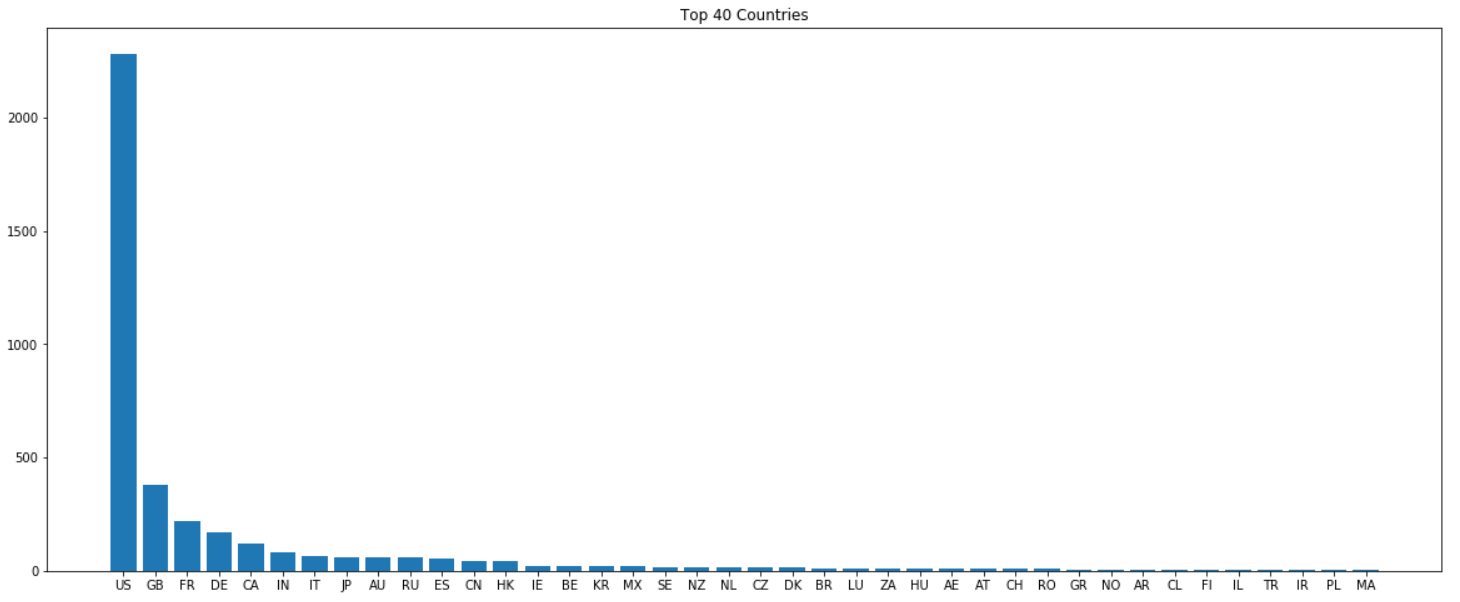
* id
* imdb\_id  
  The id of the movie on the Internet Movie Database
* original\_title  
  The original title of the movie in its original language
* overview  
  A few sentences introducing the movie
* poster\_path  
  A web location for the poster in the TMDB website
* status  
  Whether the movie has been released or not
* spoken\_languages  
  The languages spoken in the movie eg.  
  [{'iso\_639\_1': 'ar', 'name': 'العربية'}, {'iso\_639\_1': 'en', 'name': 'English'}, …]
* tagline
* title  
  The title of the movie in English
* Cast  
  A list of the cast in the movie eg. [{'cast\_id': 4, 'character': 'Lou', 'credit\_id': '52fe4ee7c3a36847f82afae7', 'gender': 2, 'id': 52997, 'name': 'Rob Corddry', 'order': 0, 'profile\_path': '/k2zJL0V1nEZuFT08xUdOd3ucfXz.jpg'}, …]. The data contains 38,760 unique cast names, but each of these have very low occurrences, with the top 40 cast only occurring between 15 and 30 times in the entire data. Due to this sparsity, this feature is dropped.  
  
* Crew  
  A list of the crew in the movie eg. [{'credit\_id': '59ac067c92514107af02c8c8', 'department': 'Directing', 'gender': 0, 'id': 1449071, 'job': 'First Assistant Director', 'name': 'Kelly Cantley', 'profile\_path': None}, ...]. The data contains 38,897 unique crew members, but the occurrences of the crew members is low, with the top 40 crew occurring between 20 and 50 times in the data.  
  

## One Hot Encoding

original\_language  
This is the original language of the movie, the original data consists of around 40 different languages, but the occurrence of these languages is quite skewed; 85% of the movies were English movies, while around 25 of the other languages occurred less than 10 times in the data. In order to get good data representation, while minimizing the number of columns required for one hot encoding, only languages that occurred in at least 1% of the data were chosen. Altogether, this represented 94.1% of the entire dataset. The languages chosen are English, Spanish, French, Hindi, Japanese and Russian.

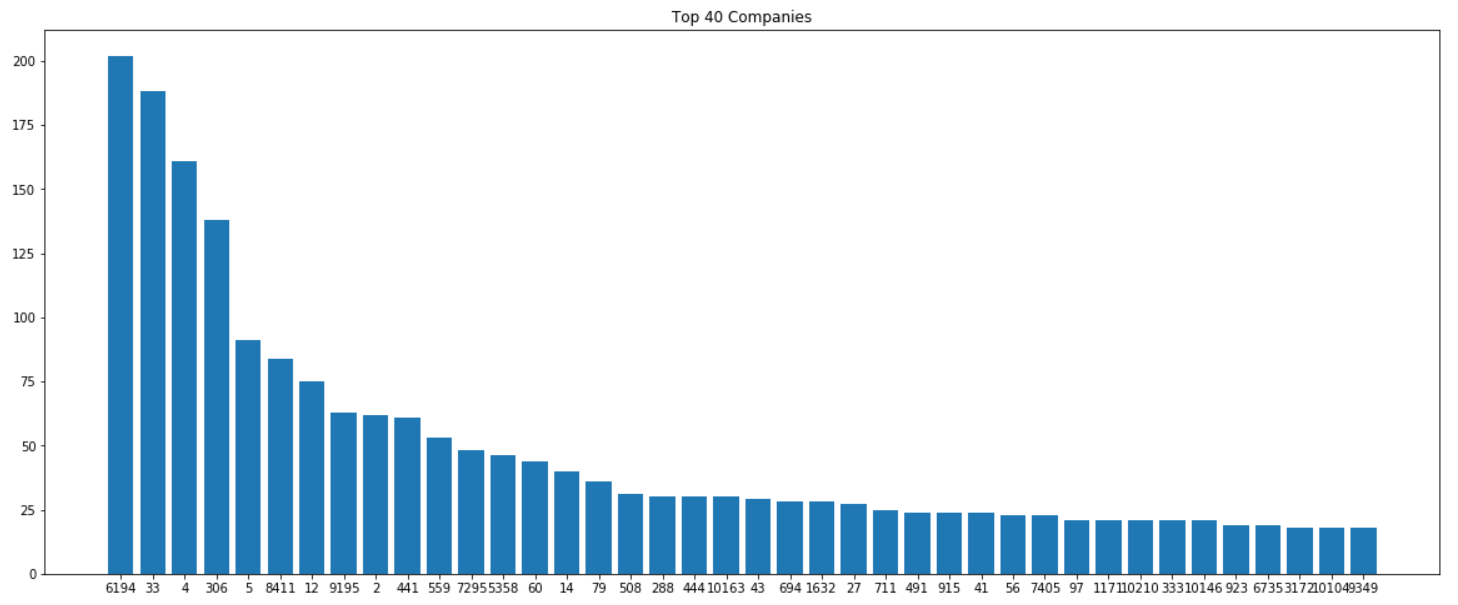
genres  
A list of genres this movie belongs to eg. [{'id': 28, 'name': 'Action'}, {'id': 35, 'name': 'Comedy'}, …]. There are a total of 20 unique genres, so all of these are one hot encoded.

production\_countries  
A list of the countries where the movie was produced eg. [{'iso\_3166\_1': 'US', 'name': 'United States of America'}, {'iso\_3166\_1': 'CA', 'name': 'Canada'}, …]. There are 74 unique countries, but the distribution of these countries is quite skewed as shown by the following table:

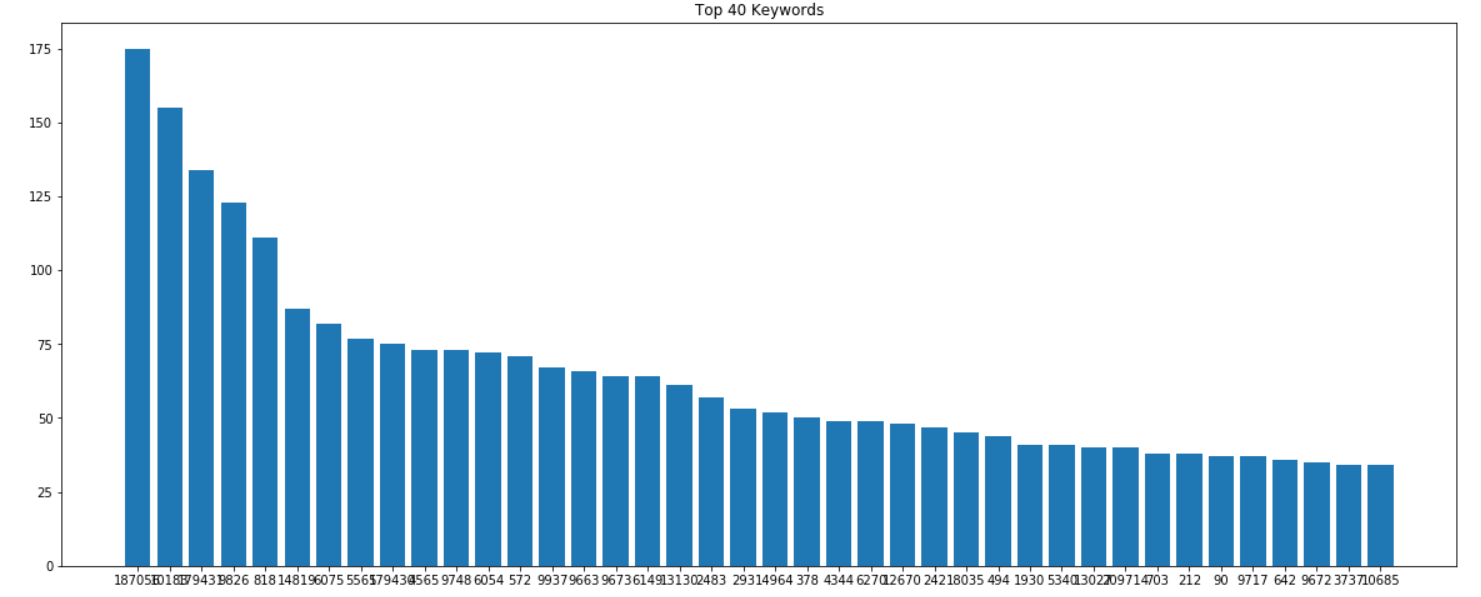


54 out of the 74 top occurring countries are chosen and one hot encoded.

production\_companies  
A list of the companies involved in the production of the movie eg. [{'name': 'Paramount Pictures', 'id': 4}, {'name': 'United Artists', 'id': 60}, …]. There are 3712 unique companies, out of which the top occurring 1034 are selected for one hot encoding and only the id is kept. This is because the occurrences of companies rapidly drop off, as show below, so keeping values that occur very few times in the entire dataset may result in overfitting.



Keywords  
Keywords for the movie eg. [{'id': 4379, 'name': 'time travel'}, {'id': 9663, 'name': 'sequel'}, …]. There are 7400 unique keywords in the dataset out of which only half are used, and only the id of each keyword is kept. The distribution of Keyword occurrences follows a similar trend to the previous two features, as shown below.



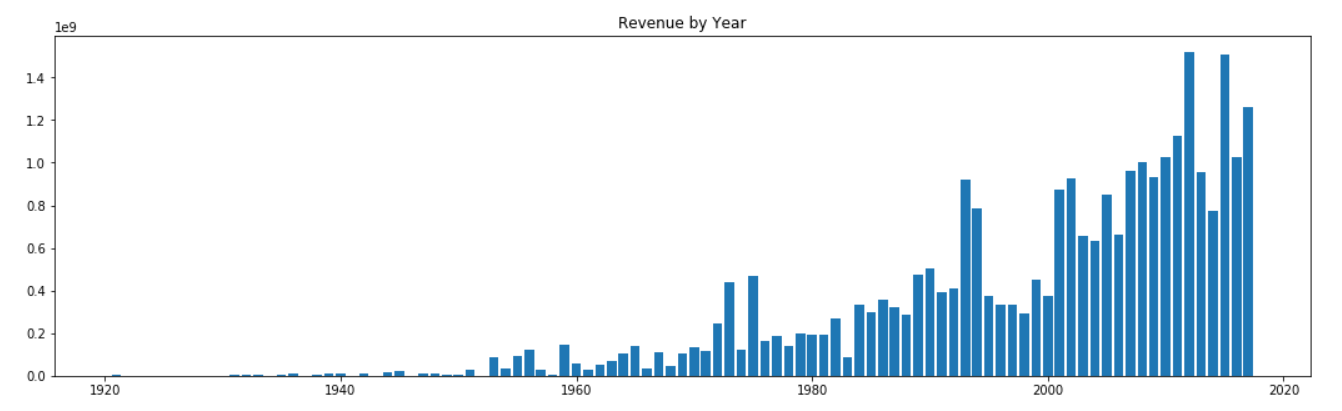
## Binary Transformation

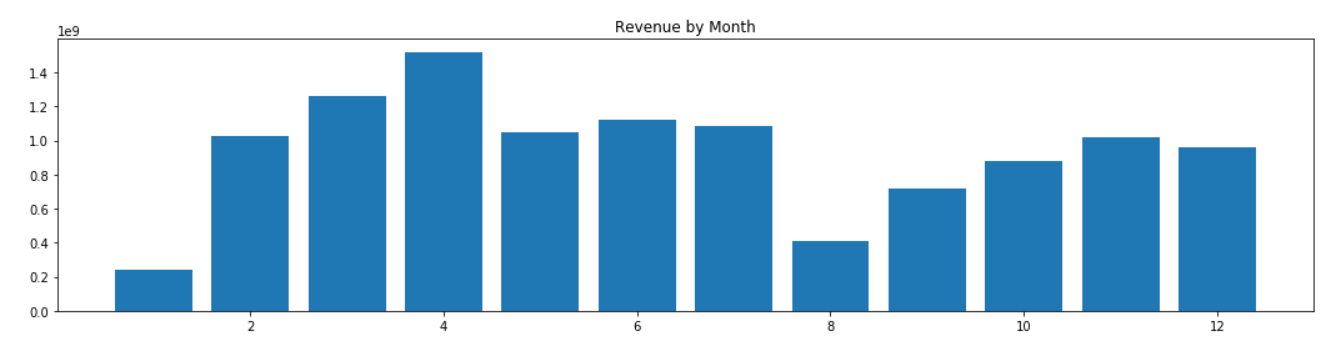
belongs\_to\_collection  
The name of the series this movie belongs to, if it belongs to one eg. a Transformers movie belongs to the Transformers movie collection. Since this column is sparse and the collection names are mostly unique, the column was transformed to a binary column with a value of 1 if the movie belongs to a collection, otherwise 0.

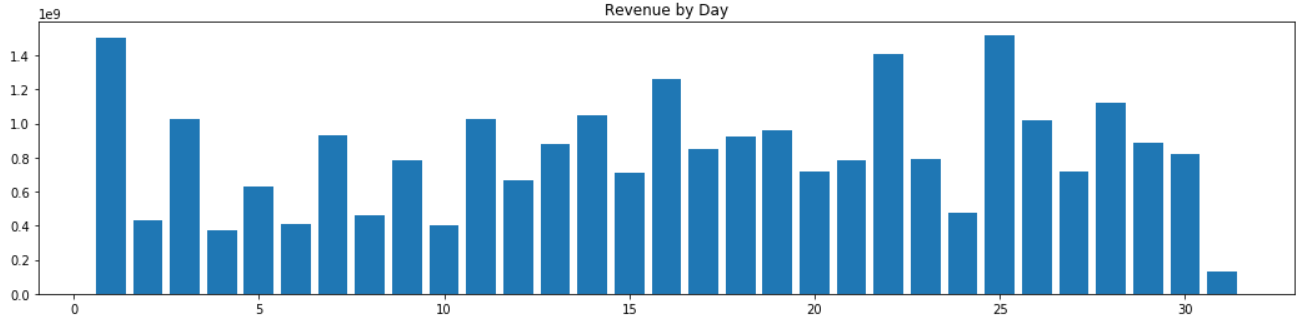
homepage  
A link to the website of the movie and it too is very sparse, so the values are transformed to 1 if there is a homepage, otherwise 0.

## Release Date Transformation

The release date is given as a string of the form MM/DD/YY







* budget
* popularity  
  A real number which seems to be higher the more popular the movie is, but it is not explained how this is obtained
* runtime  
  The runtime of the movie in minutes

In addition to this, the training data consists of an extra column for the output of the task:

* revenue

# Machine Learning Task

# Machine Learning Methods

## LASSO

### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 2.52961 |  | 2.522926 |

## Ridge Regression

### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 2.726012 |  | 2.154292 |

## Decision Tree Regression

### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 2.553579 | 2.198857 | 2.190488 |

## Random Forest Regression

### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 2.150786 | 2.074492 | 2.096104 |

## XGBoost Regression

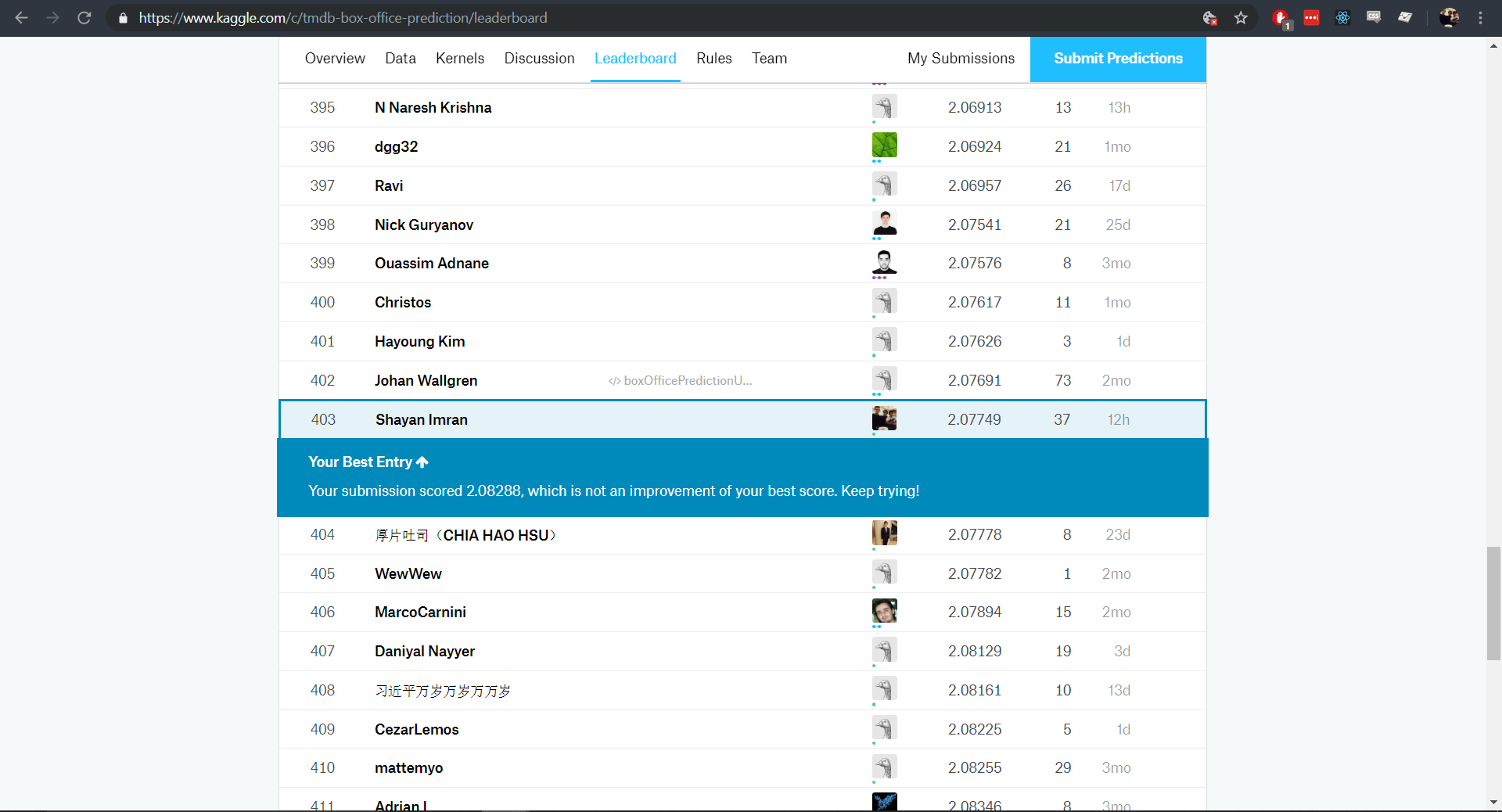
### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 2.122015 | 2.023341 | 2.03813 |

## MLPRegressor

### Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | Default | Bayes Search CV | Randomized Search CV |
| Root Mean Squared Log Error | 3.176259 |  | 2.252303 |



403/1007

# Discussion

Cast that occurs together

Lowering number of features used

Day of week

# Computing Environment

OS: Windows 10 Home 64 Bit  
OS Version: 17134  
CPU: Intel Core i7-4720HQ  
GPU: Nvidia GeForce GTX 960M  
RAM: 16 GB  
Programming Language: Python 3.7.1  
Programming Environment: JupyterLab 0.35.3