Movies TMDb Rate Predictor using Machine Learning

Maher Shahateet
Computer Engineering Department
Princess Sumaya University For Technology
Amman, Jordan
mah20170090@std.psut.edu.jo

Omar Samkari
Computer Engineering Department
Princess Sumaya University For Technology
Amman, Jordan
oma20170259@std.psut.edu.jo

Amjed Almousa
Computer Engineering Department
Princess Sumaya University For Technology
Amman, Jordan
a.almousa@psut.edu.jo

***Abstract*—Movies nowadays have huge effect on people lifestyle, feelings and opinions. It helps them to have a better understanding on their lives and reactions. In this paper, a visualization process of predicting the TMDb rate of movies based on characteristics and metadata of movies, each movie has a specific related data that will help in the process. There are many datasets available online about movies success, rating, and popularity. In this paper, a dataset from Kaggle which has real-gathered data about movies is used for the machine learning regression project. After that, a good quality model is developed to predict a movie's TMDb rating. This model gives high expected results of predictions with good accuracy on different types of regressors.**

Keywords—Machine learning, TMDb, Regression, RMSE, Decision Tree, Random Forest, Correlation

I. Introduction

People love to watch stories and thrilled acts on a daily basis. Since pictures and videos were invented, movies spread quickly worldwide which makes movies one of the important sources of entertainment. With the fast development of technology, it became the most effective element in people's life, and for that reason, movies became so popular online and free most of the time [1]. So, multiple websites were developed to gather information about movies for people like TMDb (The Movie Database), and Rotten Tomatoes etc. So, people can choose "what to watch" from different collection based on what they like . These platforms supports multiple features where people can share their honest reviews about movies. Increasingly speaking, these platforms became very popular and that leads to the availability of huge data and information online about reviews and ratings of movies. In this paper, the process consists of analyzing the huge data and results in predicting the rate of the movie [2].

This dataset originally contained information about 45k movies listed in the Movie Lens Dataset. The dataset was released on July 2017. It contains multiple files including credits, keywords, links, metadata and ratings. These files contain multiple features including cast, budget, original title, production company, cast, production country, spoken language, release date, genres, TMDb vote count and vote average. These files share a common unique id for each movie [3] [4]. The developed model in this project will use these features to predict the TMDb rating of a movie, some of these features will not be included such as, poster and tagline for example, because every movie has a unique tagline so it won't help in the process.

Before the model uses the data set, a process of cleaning data is done because many of the features have null values or 0 values which doesn't make sense. Also, some of these data is written in JSON format, so an extraction process is done also to extract what the model needs exactly, a process of merging files based on the common id for each movie results in one final clean dataset consist of 5k movies with perfect information about them, so the model can fit, train, and test in a good way where the model can have a good accuracy in predicting the TMDb rate

II. Related work

There are a lot of machine learning papers and studies was done before to predict the rate of the movie. Many of researchers on those papers used multiple methods and techniques to achieve best accuracy and they used data about movie features rather than data that is found on social media. Latif et al. (2016) aims to predict the movies popularity using classification algorithms on IMDb dataset using features such as: budget, meta score, number of votes, etc. and achieved an accuracy of 84% using logistic regression classifier [5].

Another journal was done in 2017 to predict the movie success for real world movie dataset [6] , the researchers of this journal used minmax scalar to normalize the numerical attributes, they used fuzzy logic to give more accuracy to the used model in the process of prediction. Although, they concluded that a few predicted cases don’t match with the actual results.

Yoo et al. developed a model to predict the movies revenue from IMDb dataset with 9 numerical features and 5 categorical ones, they used both linear regression and logistic for classification to achieve better results, the used features were not good enough to make strong predictions [7]. However, in their research linear was almost as good as logistic.

A movies recommendation system which is a type of information filtering system was developed by Ibtesam Ahmed in 2020 found in Kaggle website [8], she used the weighted rating feature to have a better performance of the system, she used TMDB 5000 Movie Dataset and created decent recommendation system using demographic, content-based and collaborative filtering.

III. DATASET

The dataset that the model is using was spread to multiple files, after merging them and extracted clean data resulted in one comma separated value (CSV) file that will be used named films.CSV,.

Table 1 Dataset features

Feature	Non-Null/Non-Zero Values	Type	Range
Star	4978	Object	2198 Class
Original title	4978	Object	4892 Class
Budget	4978	Int64	5000 - 380000000
Revenue	4978	Int64	10018 - 2787965087
Original language	4978	Object	61 Class
Popularity	4978	Float64	0.03 - 547.48
Production companies	4978	Object	1277 Class
Production countries	4977	Object	35 Class
Year	4978	Int64	1918 - 2017
Month	4978	Int64	1 -12
Day	4978	Int64	1 - 31
Runtime	4958	Int64	57 - 338
Spoken language	4978	Object	61 Class
Vote count	4978	Int64	1 - 14075
Vote average	4978	Float64	1.7 – 9.1

The data set has 2 more feature which is extracted from other features as follows:

Table 2 extracted features

Feature	Type
profit	Int64
Weighted rating	Floa64

A. *Dataset Filtering*

Originally there was 45 thousand records, after removing the records which contained a noticeable number of null values, remained 5 thousand records [4].

Furthermore, any irrelevant or useless columns such as taglines, poster path, were removed.

Also, one record was removed from the dataset because its production country value was null. Moreover, the runtime had 20 records with zero values which were substituted by the mean of the remaining runtimes, because a zero value for runtime is nonsensical and can produce skewed data.

Lastly, the release date feature was divided into 3 features, and each feature (Year, month, day) was converted into an int64 despite the original feature being a string.

B. *Extracting new features*

- Profit

First of all, the profit was generated from

$$Revenue - Budget \ = profit$$

This was done in order to produce a more accurate precision, by containing the 2 values into one value we have increased the correlation factor, which will be explained why later in the paper.

- Weighted rating

The weighted rating was created by

$$WR = \frac{v}{v+m} \cdot R + \frac{m}{v+m} \cdot C$$

where,

- v is the Vote Count
- m is the minimum votes required to be listed in the dataset
- R is the Vote average
- C is the mean vote average in the dataset
- WR is Weighted rating

Note: m was calculated by finding the Q8 of the Vote counts and secluding all the records which contained Vote Counts less than that.

In order to give higher meaning to the relationship between the vote count and the vote average, compared to each other independently. For example, a movie with a vote average of 10 but only 3 votes do not deserve to be on top of the ratings list.

Conclusively, we used the weighted rating feature as the label for this model.

C. *Feature Description*

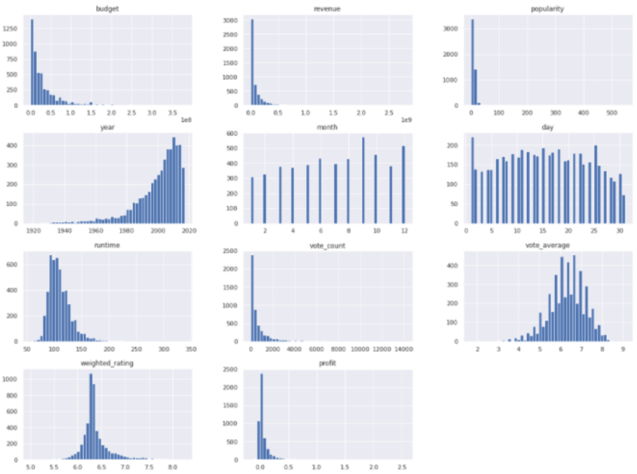


Figure 1 Histogram of Dataset Features

In the first histogram in Figure 1 shows number of movies with their respective budgets, we can deduce that most movies have budgets less than 100 million USD.

From the second histogram find that no movies have a revenue larger than 500 million USD.

The fourth histogram visualize the obvious increase in the number of movies with the increase in the year, and that makes sense because of the development in the film industry and the advance in the technology among the years.

Most movies' runtimes fall in the range of 80-150 minutes as shown by the seventh histogram.

D. *Categorical Features*

1. Original Title

This feature is the title of the movie, it has almost all unique values. In the process of this machine learning, this feature will be dropped because it will be useless.

2. Genres

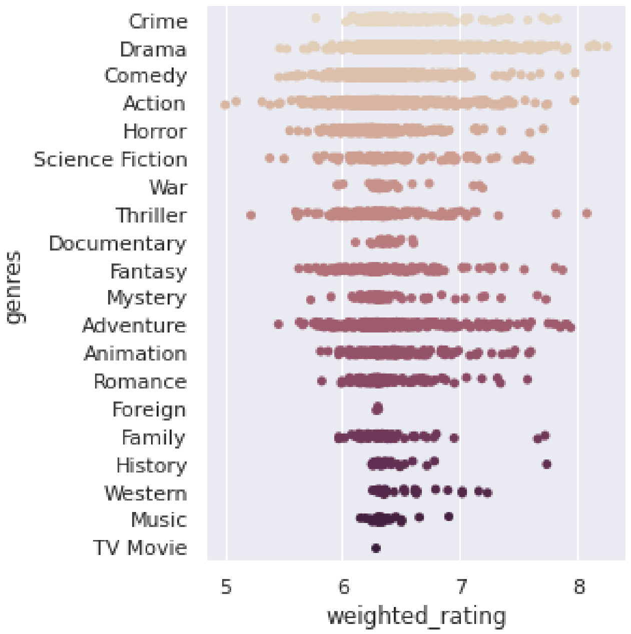


Figure 2 genre's relation with weighted rating

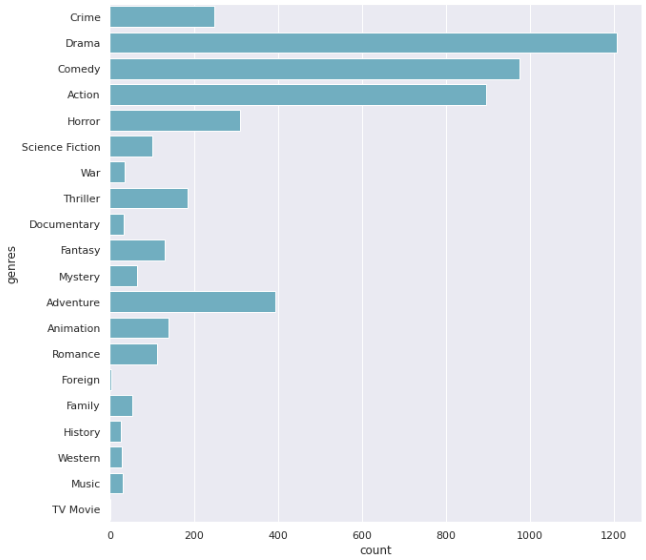


Figure 3: each genre class count

As shown in Figure 3, most of the movies are Drama, Comedy and Action. Also, in Figure 2, Drama has the most weighted rate while Action got the least. Moreover, most of the movies rate relays between 6 and 7.

3. Production Countries

Table 3 movies distribution among countries

Country	Count
US	3256
GB	369
FR	208
CA	175
DE	172
IN	139
Others	<100

High country production is United States as shown in the Table 3, US presents around 65% of the movies in this dataset.

There is a lot of countries which produce less than 100 movies in this dataset categorized under other in the table.

4. Spoken language

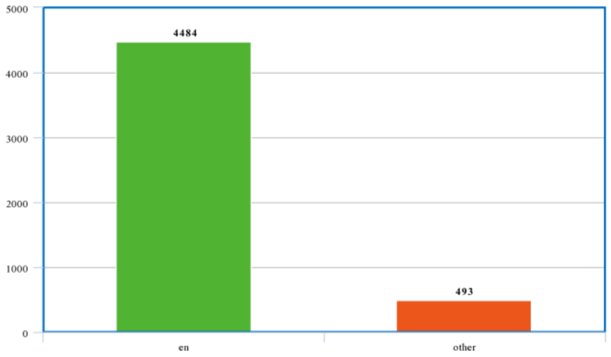
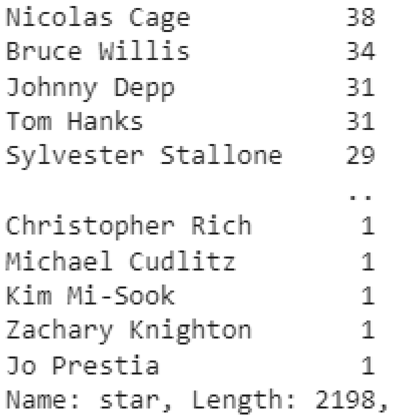


Figure 4 spoken languages in movies

5. Star



```
Nicolas Cage            38
Bruce Willis            34
Johnny Depp             31
Tom Hanks               31
Sylvester Stallone      29
                        ..
Christopher Rich         1
Michael Cudlitz          1
Kim Mi-Sook              1
Zachary Knighton         1
Jo Prestia               1
Name: star, Length: 2198,
```

Figure 5 movie star

In this dataset, there are 2198 different stars on 4977 movies. So, most likely the star feature won't help a lot in the process of prediction because of the abundance of unique values.

As shown in Figure 5, highest actor Nicolas Case with 38 movie records, while there are a lot of actors got only 1 movie.

6. Spoken language

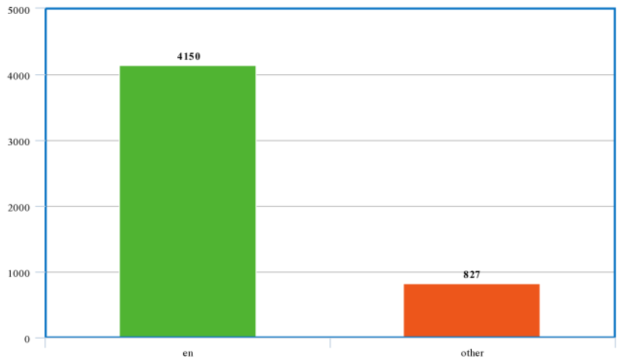


Figure 6 spoken language

The last categorical feature is demonstrated into multiple classes, Figure 6 categorized these classes into 2, 4150 spoken English language and 827 others.

IV. Pearson Correlation

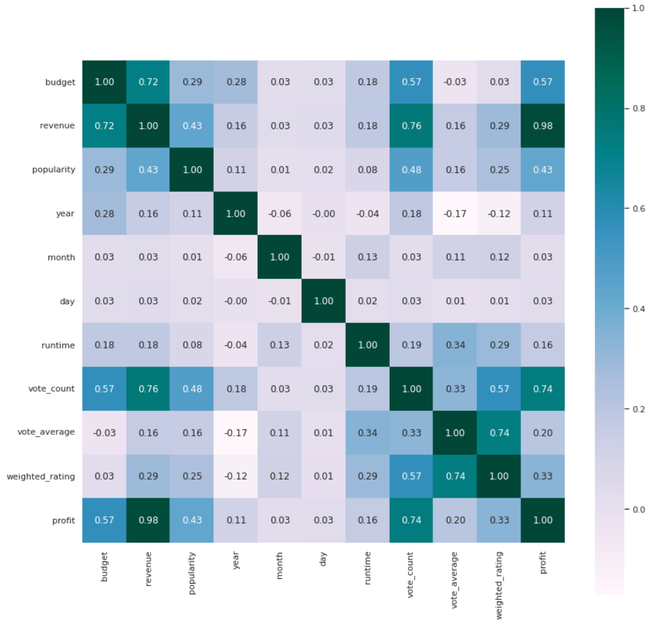
Table 4 features correlation with weighted rating

Feature	Correlation Value
Vote_average	0.738
Vote_count	0.573
profit	0.334
runtime	0.291
revenue	0.288
popularity	0.249
month	0.119
budget	0.027
day	0.009
year	-0.121

From Table 4, with the predicted label, weighted rating, “vote_average” and “vote_count” has the highest correlation, and that makes sense because the label is extracted from both of them.

On the other hand, “profit” feature has the highest correlation among the other features, especially higher than revenue and budget. So, the process of feature engineering (extraction new feature) was very helpful in this model.

The “day” feature has the lowest correlation. So, that leads that whenever day the movie is released, it doesn’t matter much and won’t affect the rating of that movie.



	budget	revenue	popularity	year	month	day	runtime	vote_count	vote_average	weighted_rating	profit
budget	1.00	0.72	0.29	0.28	0.03	0.03	0.18	0.57	-0.03	0.03	0.57
revenue	0.72	1.00	0.43	0.16	0.03	0.03	0.18	0.76	0.16	0.29	0.98
popularity	0.29	0.43	1.00	0.11	0.01	0.02	0.08	0.48	0.16	0.25	0.43
year	0.28	0.16	0.11	1.00	-0.06	-0.00	-0.04	0.18	-0.17	-0.12	0.11
month	0.03	0.03	0.01	-0.06	1.00	-0.01	0.13	0.03	0.11	0.12	0.03
day	0.03	0.03	0.02	-0.00	-0.01	1.00	0.02	0.03	0.01	0.01	0.03
runtime	0.18	0.18	0.08	-0.04	0.13	0.02	1.00	0.19	0.34	0.29	0.16
vote_count	0.57	0.76	0.48	0.18	0.03	0.03	0.19	1.00	0.33	0.57	0.74
vote_average	-0.03	0.16	0.16	-0.17	0.11	0.01	0.34	0.33	1.00	0.74	0.20
weighted_rating	0.03	0.29	0.25	-0.12	0.12	0.01	0.29	0.57	0.74	1.00	0.33
profit	0.57	0.98	0.43	0.11	0.03	0.03	0.16	0.74	0.20	0.33	1.00

Figure 7 correlation matrix between numerical features

Figure 7 visualized 2 noticeable points:

- Vote count correlation with “profit”

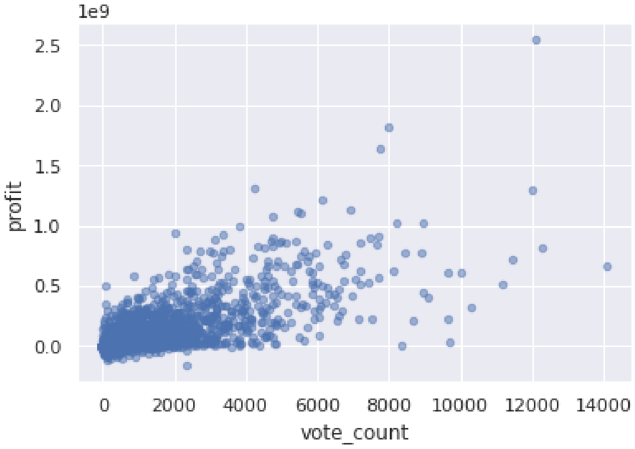


Figure 8 scatter plot between profit and vote count

With more vote counts the profit increase.

- Vote count correlation with “popularity”

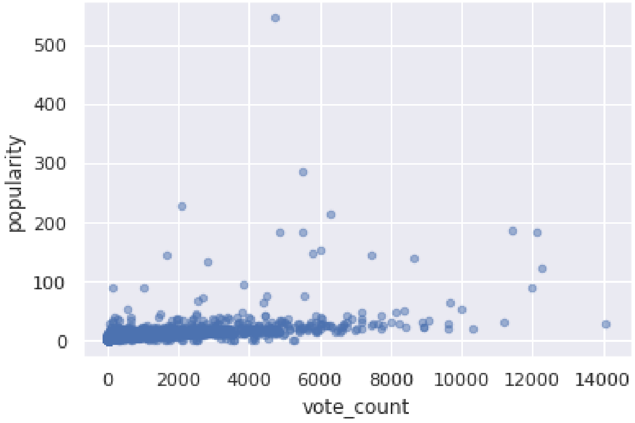


Figure 9 scatter plot between popularity and vote count

Figure 9 demonstrates that there is little to no relationship between the popularity of the movie and the vote count. At 6000 vote counts for example we can find movies with all kinds of popularity. This trend can be explained by the existence of older spectators who watched the movie but didn't rate it on rating web sites.

V. Data Processing tools

A. *Onehot encoder*

The problem with categorical features is that they don't have an inherent order relationship between each other, and machine learning algorithms need to be able to understand and harness this relationship in order to create accurate results [9].

One hot encoding occurs after giving an integer representation to each label, and then the integer variable is removed and is replaced with a new binary variable that is added for each unique integer value.

Let's take genres as an example of categorical features: We give action: 1, drama: 2, and comedy: 3, etc.

Table 5 One Hot encoding example

Action	Drama	comedy
1	0	0
0	1	0
0	0	1

B. *MinMax scaler*

A way to normalize the input. All numerical features will be transformed into the range [0,1] meaning that the minimum and maximum value of a feature/variable is going to be 0 and 1, respectively [10].

$$x_{scaled} = \frac{x - x_{minimum}}{x_{maximum} - x_{minimum}}$$

VI. Measurement tools

A. *R2 Score*

R2 score is the coefficient of determination which is basically a regression score function [11], it is a scikit learn metric that is probably used to measure the accuracy of the predicted value of the machine learning model, highest possible score is 1 and it is the best, but most likely in machine learning cases, 1 is slightly impossible. On the other hand, it can be negative and that's because the model can be bad.

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$$

- SS, is the square summation

B. *RMSE*

Root Mean Square Error is a scikit learn metric that is used to measure the error between the actual value and the predicted one, it can be also helpful to determine how accurate the model is [12].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(x_{actual} - x_{predicted}\right)^2}{N}}$$

VII. Machine learning models

After filtering the data, extracting new features, cleaning it, and then encoding and scaling each feature type, the data was split into two sets: training and test set, where 20% was test and 80% was training. Afterwards, we fit the data for the model.

A. *Linear Regression*

Linear regression creates a target prediction value based on independent features. It is mostly used for finding out the relationship between features and prediction [13].

B. *eXtreme Gradient Boosting regression*

Gradient Boosting regression creates a prediction model in the shape of an ensemble of weaker prediction models, which typically happen to be decision tree, Extreme Gradient Boosting, or XGBoost for short is an efficient open-source implementation of the gradient boosting algorithm [14].

C. *Decision tree regression*

Decision tree builds regression in the form of a tree. It breaks down a dataset into blocks or smaller subsets and creates a decision from the subsets obtained. This results in a tree structure with decision and leaf nodes. The topmost decision node in a tree which corresponds to the best prediction [15].

D. *Random forest regression*

Random forest regression is an ensemble learning method for regression that operates by creating multiple decision trees at training time. The mean or average prediction of the decision trees is returned. Random decision forests correct the overfitting that usually happens in trees [16].

E. *Voting Regression*

A voting regressor is an ensemble meta-estimator that fits several base regressors, each on the whole dataset. Then it averages the individual predictions to form a final prediction [17].

After fitting the data into these 5 models, r2 score was calculated to determine the accuracy of these models, RMSE was calculated also to measure the error between the predicted value and the actual one, both of these measures were calculated for train and test, 3 attempts was done and the average of these 3 is summarized in the below tables.

Table 6 Models score, RMSE on test set

ML Model	R2 Score	RMSE
Linear Regression	0.700	0.053
XGBooster Regression	0.985	0.015
Decision Tree Regression	0.950	0.022
Random Forest Regression	0.990	0.005
Voting Regression	0.960	0.157

Table 7 Models score, RMSE on train set

ML Model	R2 Score	RMSE
Linear Regression	0.740	0.050
XGBooster Regression	0.990	0.011
Decision Tree Regression	0.960	0.020
Random Forest Regression	0.995	0.003
Voting Regression	0.970	0.155

VIII. CONCLUSION

To summarize up, this paper aims to develop a machine learning model using supervised learning methods to predict the rate of the movie. Filtering and cleaning process was done on the dataset, scaling on numerical features and encoding on categorical features was done also. 5 models were used on the dataset in this paper. As shown above, Random Forest Regression was the most accurate one between them followed by eXtreme Gradient Booster Regression. On the other hand, Linear Regression was the worst one with the least accurate results and highest error.

ACKNOWLEDGMENT

The writers would like to thank their university “PSUT” for supporting the publication of this research, which is based on a machine learning class project. Moreover, the writers would like to express their gratitude to their supervisor for his assistance during the process and for allowing them to participate in such a great opportunity.

REFERENCES

[1] https://en.wikipedia.org/wiki/Film , Accessed on December 1st, 2021

[2] https://www.themoviedb.org/ , Accessed on December 1st, 2021

[3] https://grouplens.org/datasets/movielens/ , Accessed on December 1st, 2021

[4] https://www.kaggle.com/rounakbanik/the-movies-dataset, Accessed on December 2nd, 2021

[5] M. H. Latif and H. Afzal, “Prediction of movies popularity using machine learning techniques,” International Journal of Computer Science and Network Security (IJCSNS), vol. 16, no. 8, p. 127, 2016.

[6] S. Pramod, A. Joshi, and A. Mary, “Prediction of movie success for real world movie dataset,” Int. J. of Advance Res., Ideas and Innovations in Technol, vol. 3, no. 3, 2017.

[7] Steven Yoo, Robert Kanter, David Cummings TA, Andrew Maas, “Predicting Movie Revenue from IMDb Data”, pp.1-5, 2011

[8] https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system, Accessed on December 3rd, 2021

[9] Brownlee, Jason. (2017). "Why One-Hot Encode Data in Machine Learning?" https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ , Accessed at Decemeber 20th, 2021.

[10] Brownlee, Jason. (2020). "How to Use StandardScaler and MinMaxScaler Transforms in Python" https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/, Accessed at Decemeber 20th, 2021.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html, Accessed on December 30th, 2021

[12] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html, Accessed on December 30th, 2021

[13] Brownlee, Jason. (2016). "Linear Regression for Machine Learning” https://machinelearningmastery.com/linear-regression-for-machine-learning/, Accessed at Decemeber 22th, 2021.

[14] Brownlee, Jason. (2021). "XGBoost for Regression” https://machinelearningmastery.com/xgboost-for-regression/, Accessed at Decemeber 22th, 2021.

[15] Brownlee, Jason. (2016). " Classification And Regression Trees for Machine Learning” https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/, Accessed at Decemeber 22th, 2021.

[16] Brownlee, Jason. (2020). " How to Develop a Random Forest Ensemble in Python” https://machinelearningmastery.com/random-forest-ensemble-in-python/, Accessed at Decemeber 23th, 2021.

[17] Brownlee, Jason. (2020). " How to Develop Voting Ensembles With Python” https://machinelearningmastery.com/voting-ensembles-with-python/, Accessed at Decemeber 27th, 2021.