Movie Prior Release Box Office Prediction A Machine Learning Based Approach

V. Gangadhara Reddy[1], K. Radheer Reddy[2], A. Krishnamoorthy[3*], R. Kannadasan[4], P. Boominathan[5]

[1]UG Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

[2]UG Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

[3]Assistant Professor (Senior), School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.*

[4]Assistant Professor (Senior), School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

[5]Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

[3]krishnamoorthy.arasu@vit.ac.in*

Abstract: The "Movie Box Office Prediction" includes different factors that influence the movie revenue at the Box Office. Some of the factors include Budget, Genres, Spoken languages, Cast, Crew. In this paper, various plots are made in order to understand and observe the relations between the variables and the amount of effect of factors on the Revenue. Linear Regression, Random Forest and XGBoost are the models used for Training and Testing the Data.

Keywords: Linear Regression, Random Forest, XGBoost.

1. INTRODUCTION

The Global box office revenue had hit a record of $42.5 billion in 2019. According to this, Movie Industries has become one of the major industries in the world which includes a lot of Budget, Crew, and Cast. So the prediction of movie revenue makes a great deal. Predicted revenues can be used to decide on planning the production and the distribution stages. For an instance, the projected revenue will be helpful to decide the remuneration of actors, members, and also the costs to sell the copyrights. Movie success or failure can depend mainly on the Stars that are acting in the film, release day, Budget, and also the Genre of the film. For an instance, the Adventure Genre films usually get more Gross revenue compared to other Genre. These all the factors have an impact on the Movie's revenue. So it is not easy for humans to predict the Collections. Since this era became a Computer and Data Science era it is easier for the machines to perform mathematical computations in predicting the Movie's revenue based on the historical Databases.

2. LITERATURE SURVEY

Matt Vitelli [1] has designed a model for the movie revenue prediction by observing the relationships by the combination between the features using graphs such as actor-actor relationship graphs, actor-movie relationship graphs, and movie-movie relationship graphs and concluded that by combining these features and observing the relationships using the actor-actor relationship graphs and actor movie relationship graphs, they were able to design more accurate model than using the features in the Dataset alone.

Prediction of movie box office success based on the Wikipedia Activity Big Data[3] used a simple model based only on few variables, but the efficiency could be enhanced using statistical computations and also more on the content related parameters[4][5], for example, the controversial measure in the article. There can be a diverse number of changes that can be made to the defined model in this paper.

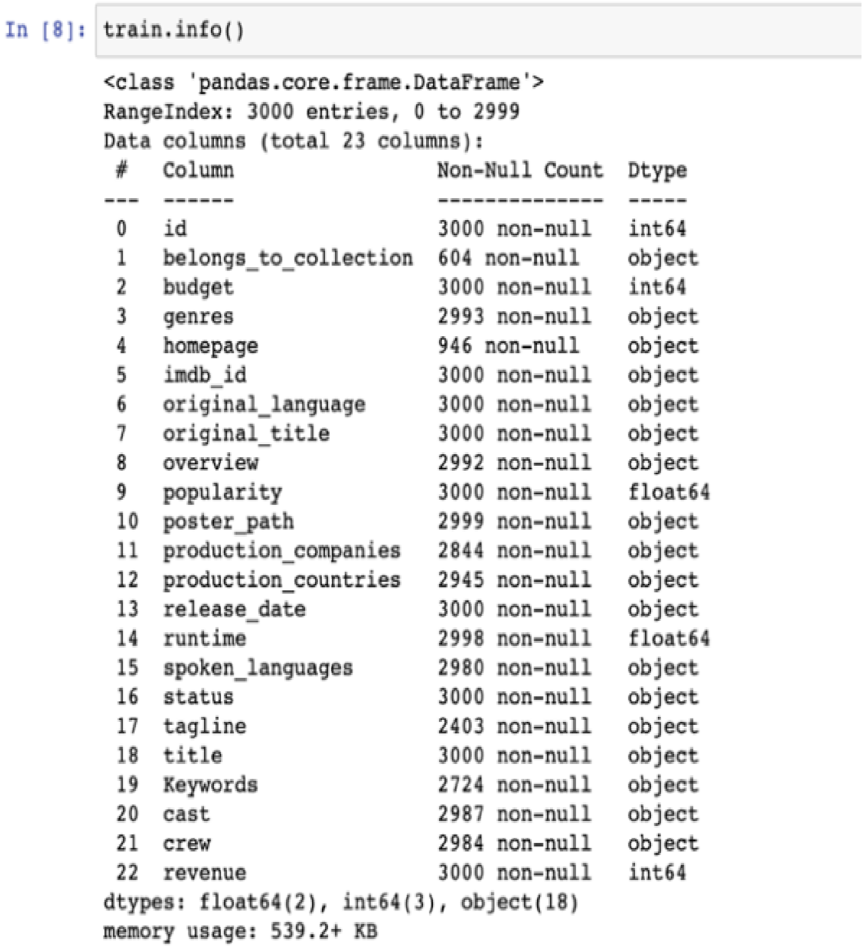
The model designed in this paper does not consider the movie genre and the actor's popularity [6][7][8] involved in the movies which might lead to accurate predictions if these are considered in the paper [9][10]. One biggest change that can be made to this model, is that increasing the Data in the Dataset to obtain more accurate prediction [11][12].

3. DESCRIPTION OF DATA

Training dataset has 3000 Movie records and 23 variables (including revenue), Testing dataset has 4398 Movie records with 22 variables.

4. EXPLORATORY DATA ANALYSIS

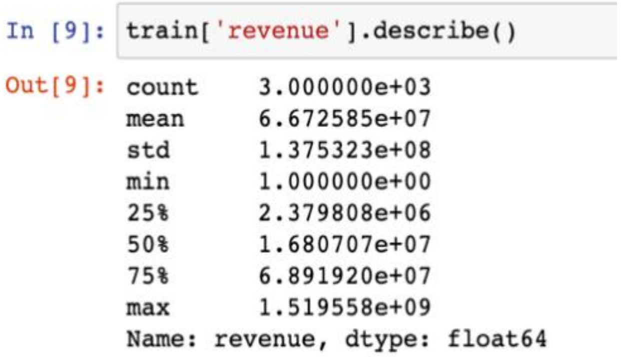
Exploratory data Analysis is a process of analysing the Datasets and understanding the relations between the features in Dataset using visualizations and also remove the unnecessary features that are not influential on the dependant feature.



```
In [8]: train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     3000 non-null   int64
 1   belongs_to_collection  604 non-null    object
 2   budget                 3000 non-null   int64
 3   genres                 2993 non-null   object
 4   homepage               946 non-null    object
 5   imdb_id                3000 non-null   object
 6   original_language      3000 non-null   object
 7   original_title         3000 non-null   object
 8   overview               2992 non-null   object
 9   popularity             3000 non-null   float64
 10  poster_path            2999 non-null   object
 11  production_companies   2844 non-null   object
 12  production_countries   2945 non-null   object
 13  release_date           3000 non-null   object
 14  runtime                2998 non-null   float64
 15  spoken_languages       2980 non-null   object
 16  status                 3000 non-null   object
 17  tagline                2403 non-null   object
 18  title                  3000 non-null   object
 19  Keywords               2724 non-null   object
 20  cast                   2987 non-null   object
 21  crew                   2984 non-null   object
 22  revenue                3000 non-null   int64
dtypes: float64(2), int64(3), object(18)
memory usage: 539.2+ KB
```

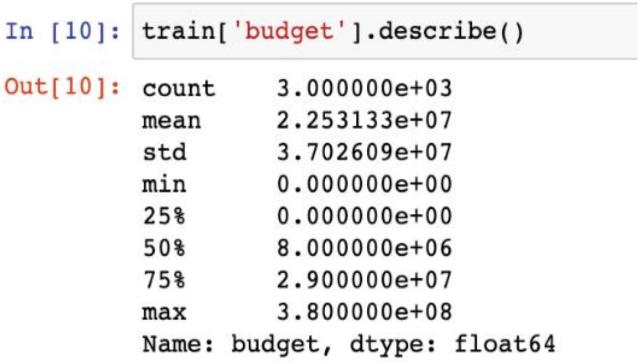
Figure 1 Training Dataset information



```
In [9]: train['revenue'].describe()

Out[9]: count    3.000000e+03
        mean     6.672585e+07
        std      1.375323e+08
        min      1.000000e+00
        25%      2.379808e+06
        50%      1.680707e+07
        75%      6.891920e+07
        max      1.519558e+09
        Name: revenue, dtype: float64
```

Figure 2 Training Dataset Revenue statistical variables

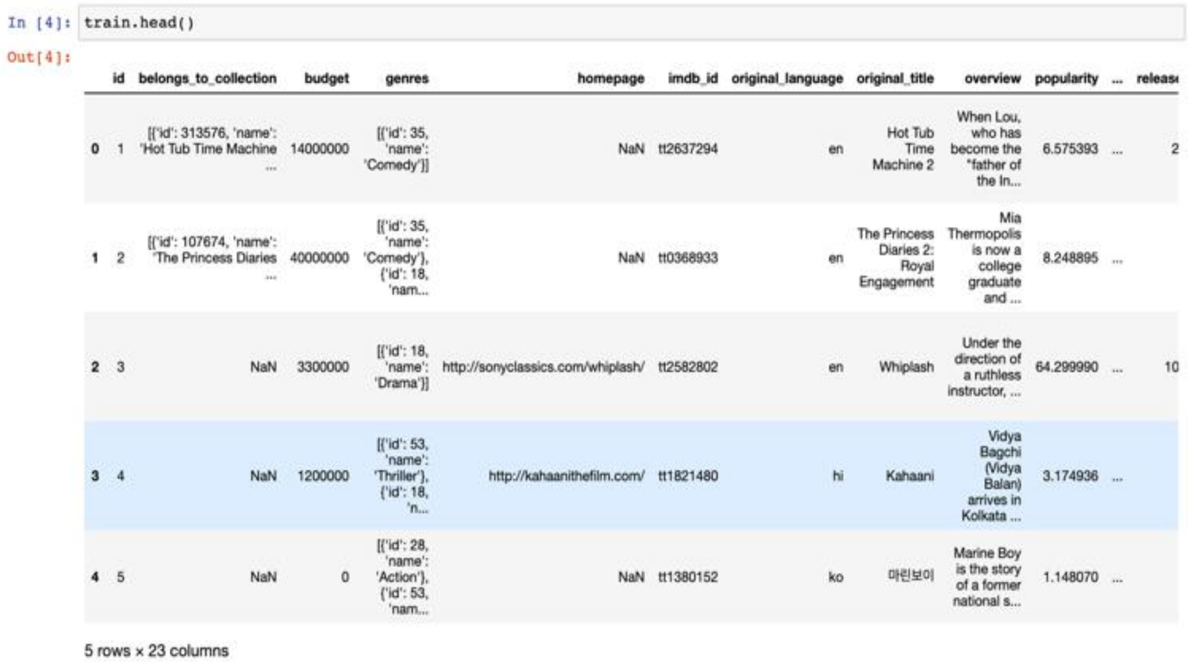


```
In [10]: train['budget'].describe()

Out[10]: count    3.000000e+03
         mean     2.253133e+07
         std      3.702609e+07
         min      0.000000e+00
         25%      0.000000e+00
         50%      8.000000e+06
         75%      2.900000e+07
         max      3.800000e+08
         Name: budget, dtype: float64
```

Figure 3 Training Dataset Revenue statistical variables

From the above it can be understood that the Mean of the revenue is around 6 Million. And the Mean of the budget is around 2.5 Million.



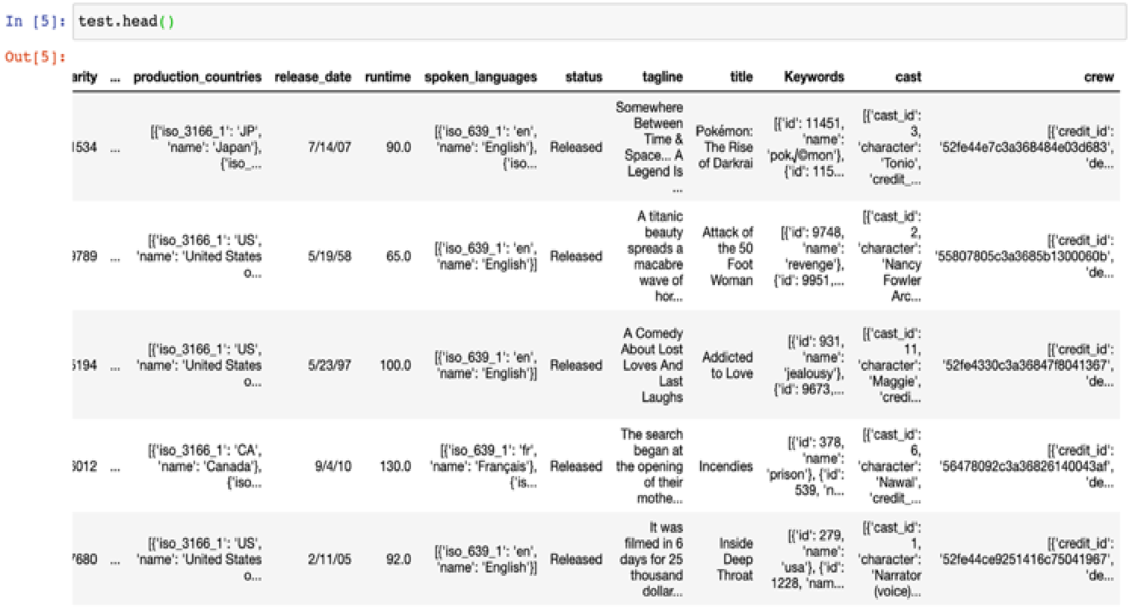
```
In [4]: train.head()

Out[4]:
```

	id	belongs_to_collection	budget	genres	homepage	imdb_id	original_language	original_title	overview	popularity	...	releas
0	1	[{'id': 313576, 'name': 'Hot Tub Time Machine ...	14000000	[{'id': 35, 'name': 'Comedy'}]	NaN	tt2637294	en	Hot Tub Time Machine 2	When Lou, who has become the "father of the In...	6.575393	...	2
1	2	[{'id': 107674, 'name': 'The Princess Diaries ...	40000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	tt0368933	en	The Princess Diaries 2: Royal Engagement	Mia Thermopolis is now a college graduate and ...	8.248895	...	
2	3	NaN	3300000	[{'id': 18, 'name': 'Drama'}]	http://sonyclassics.com/whiplash/	tt2582802	en	Whiplash	Under the direction of a ruthless instructor, ...	64.299990	...	10
3	4	NaN	1200000	[{'id': 53, 'name': 'Thriller'}, {'id': 18, 'n...	http://kahaanithefilm.com/	tt1821480	hi	Kahaani	Vidya Bagchi (Vidya Balan) arrives in Kolkata ...	3.174936	...	
4	5	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 53, 'nam...	NaN	tt1380152	ko	마린보이	Marine Boy is the story of a former national s...	1.148070	...	

5 rows × 23 columns

Figure 4 Training Dataset



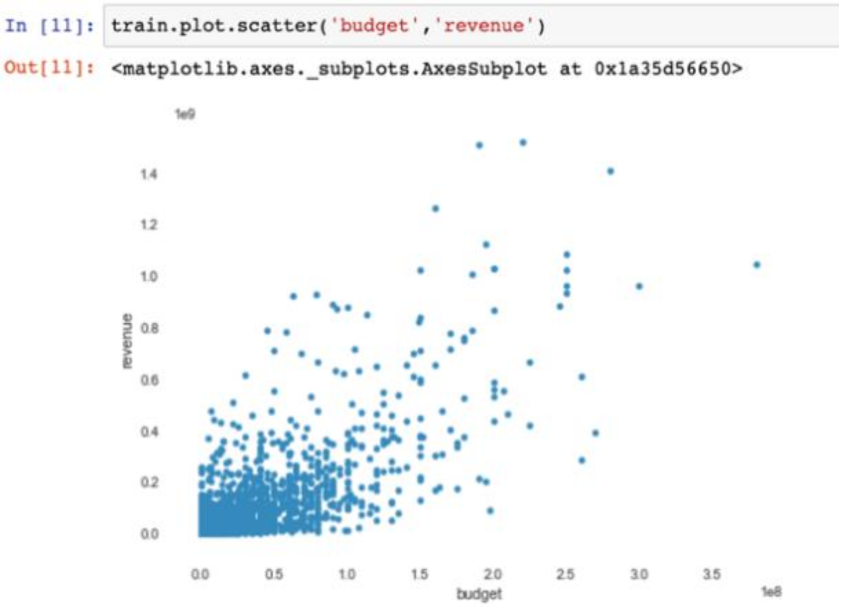
```
In [5]: test.head()
Out[5]:
```

arity	...	production_countries	release_date	runtime	spoken_languages	status	tagline	title	Keywords	cast	crew
1534	...	[{'iso_3166_1': 'JP', 'name': 'Japan'}, {'iso_...	7/14/07	90.0	[{'iso_639_1': 'en', 'name': 'English'}, {'iso...	Released	Somewhere Between Time & Space... A Legend Is ...	Pokémon: The Rise of Darkrai	[{'id': 11451, 'name': 'pok√©mon'}, {'id': 115...	[{'cast_id': 3, 'character': 'Tonio', 'credit_...	[{'credit_id': '52fe44e7c3a368484e03d683', 'de...
3789	...	[{'iso_3166_1': 'US', 'name': 'United States o...	5/19/58	65.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	A titanic beauty spreads a macabre wave of hor...	Attack of the 50 Foot Woman	[{'id': 9748, 'name': 'revenge'}, {'id': 9951,...	[{'cast_id': 2, 'character': 'Nancy Fowler Arc...	[{'credit_id': '55807805c3a3685b1300060b', 'de...
5194	...	[{'iso_3166_1': 'US', 'name': 'United States o...	5/23/97	100.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	A Comedy About Lost Loves And Last Laughs	Addicted to Love	[{'id': 931, 'name': 'jealousy'}, {'id': 9673,...	[{'cast_id': 11, 'character': 'Maggie', 'credi...	[{'credit_id': '52fe4330c3a36847f8041367', 'de...
3012	...	[{'iso_3166_1': 'CA', 'name': 'Canada'}, {'iso...	9/4/10	130.0	[{'iso_639_1': 'fr', 'name': 'Français'}, {'is...	Released	The search began at the opening of their mothe...	Incendies	[{'id': 378, 'name': 'prison'}, {'id': 539, 'n...	[{'cast_id': 6, 'character': 'Nawal', 'credit_...	[{'credit_id': '56478092c3a36826140043af', 'de...
7680	...	[{'iso_3166_1': 'US', 'name': 'United States o...	2/11/05	92.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	It was filmed in 6 days for 25 thousand dollar...	Inside Deep Throat	[{'id': 279, 'name': 'usa'}, {'id': 1228, 'nam...	[{'cast_id': 1, 'character': 'Narrator (voice)...	[{'credit_id': '52fe44ce9251416c75041967', 'de...

Figure 5 Test Dataset

Revenue vs. Budget

The scatter plot for Revenue vs Budget recapitulates that if the Budget increases then the Revenue also increases. If the Budget of the movie is high it means it might have star actors or the production costs.



```
In [11]: train.plot.scatter('budget','revenue')
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1a35d56650>
```

Figure 6 Revenue vs. Budget

Revenue vs. has_homepage

If the movie has a homepage then it might be a high Budget film as per the catplot the movies having homepage had more revenues than the movies without having a home page and in other cases, the home page doesn't matter.

Since the use of homepage is an ambiguous attribute, we drop the attribute homepage else it will yield false results.

Revenue vs Collection

Collection for an instance is a movie belonging to a particular series, and since this is also an ambiguous attribute we drop collection attribute else it may yield false predictions.

Revenue vs Language

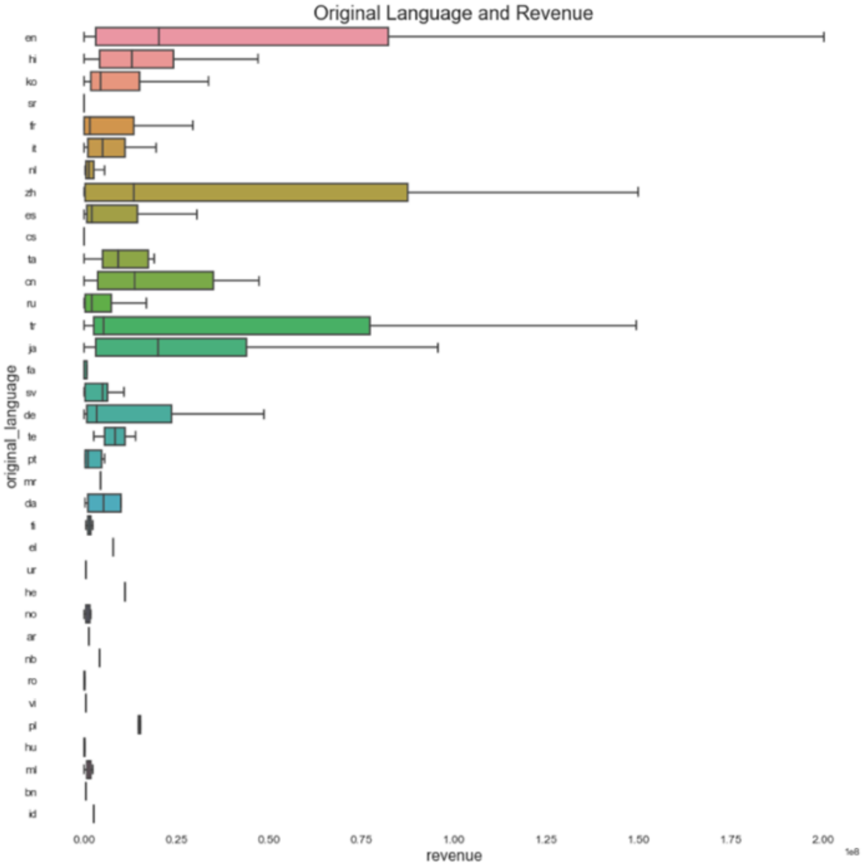


Figure 7 Revenue vs. Original Language

According to this Box plot, ‘en’ that is English yields high revenues. Since English is the most spoken language in the world. English is the most Profitable language for the Movie.

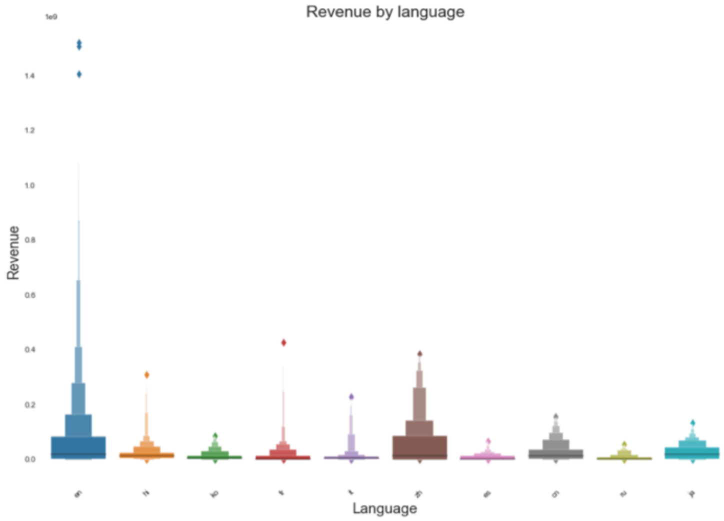


Figure 8 Revenue vs. Language

Revenue vs. Genres

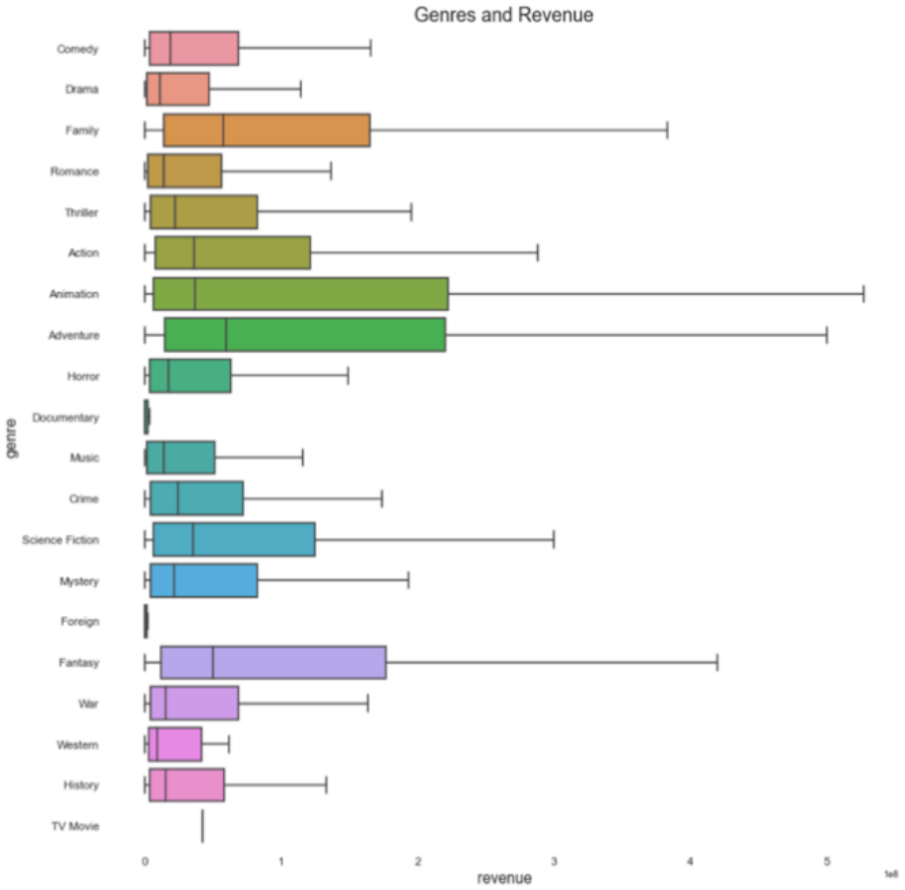


Figure 9 Revenue vs. Genres

Genre is a major factor to be considered to predict the movie's revenue. As the below Box plot shows the Adventure Genre has the highest revenues compared to other Genres, Secondly Animated movies. The Genre will have high correlation with the Budget and as well as the Movie's Revenue.

Revenue vs. different Number of Genres in the Film

It can be seen from the below plot that the movies having Four Genres has more revenue than the others. Because if the movie only contains one Genre people might not like it which may result in the less revenue and even if the movie has more number of movies it may result in less revenue because people could not absorb all the genres or it may reduce the quality of the film by just focusing on the Increasing the genres.

Revenue vs. Number of Production Companies

This Number of production in some cases represents that the movie is of high budget, so the more number of companies are involved to produce the film, So the number of production companies is indirectly linked to the Budget. As per the plot obtained, we can see the three production companies are having very high revenue compared to others.

Revenue vs. Number of Production Countries

The number of production countries is an ambiguous factor and correctly could not have impact on the movie's revenue. So we drop the production countries column from our DataFrame.

Revenue vs. Overview

Mapping overview present to 1 and nulls to 0. This variable is unnecessary variable and can be dropped, because this does not affect the movies revenue much. So we drop the Overview column from our DataFrame – Train.

Revenue vs. Cast Members

Cast members is one of the most important Factors for the Revenue prediction. The cast members include the actors in the movie. As the actors will have their own individual popularity if the number of cast members or various actors from the various language will yield much revenue. So cast members become one of the major factor. For a movie the more crew members are required than the cast members.

Revenue vs. Crew Members

Crew members is one of the most important Factors for the Revenue prediction. The crew constitutes the people who are hired by a production company, for the purpose of producing a movie which includes every person who is not visible on the screen but works for the film. In case of High Budget movies the will require more number of people as a crew. So crew members become one of the major factor. For a movie the more crew members are required than the cast members. So, the revenue gets high when the crew numbers increased respectively.

Correlation between Variables

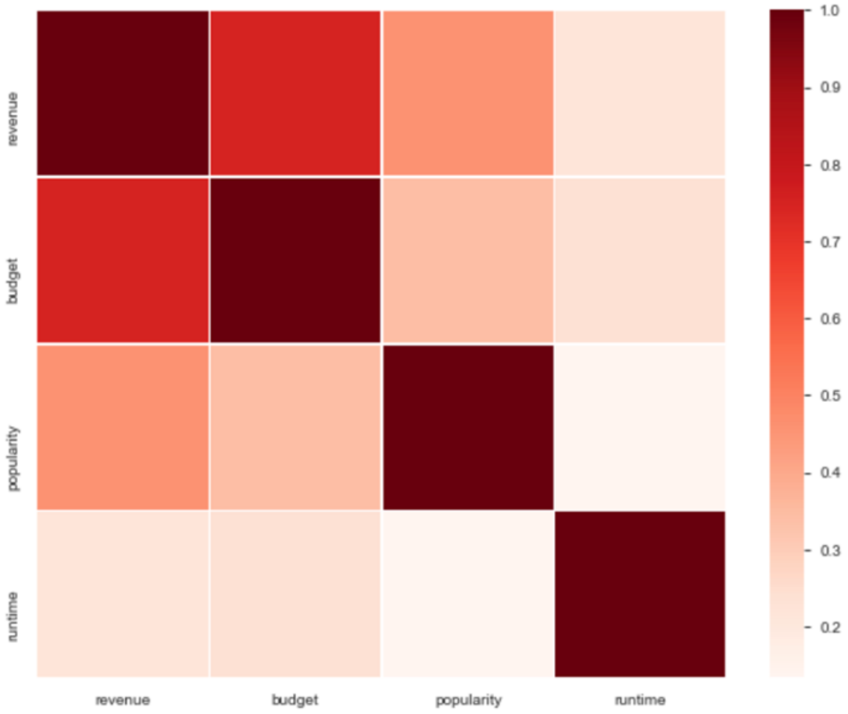


Figure 10 Correlation between Variables

The following shows the relation between the left variables after dropping in our DataFrame. Higher the intensity of the colour, higher the correlation between the variables.

From the below correlation plot, the Budget and Revenue has the Highest correlation and it shows that the value of the Revenue mainly depends on the Budget of the Movie. Secondly, the popularity has the second correlation between Revenue and the Popularity. And, then the runtime is the least correlation in our given variables.

Revenue vs. Release_date

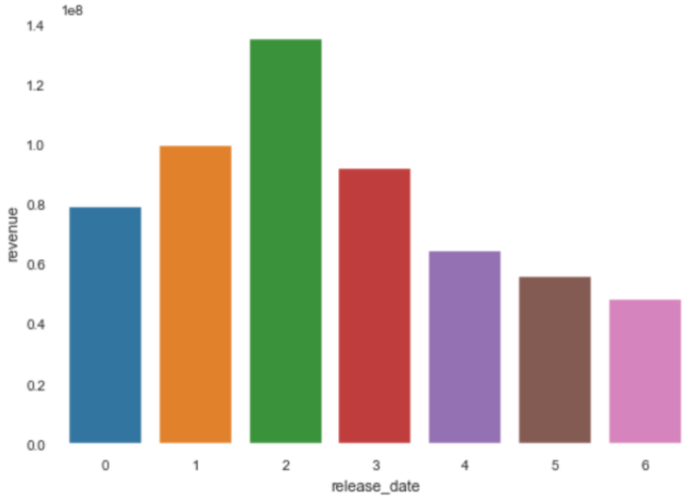


Figure 11 Revenue vs. Release_date

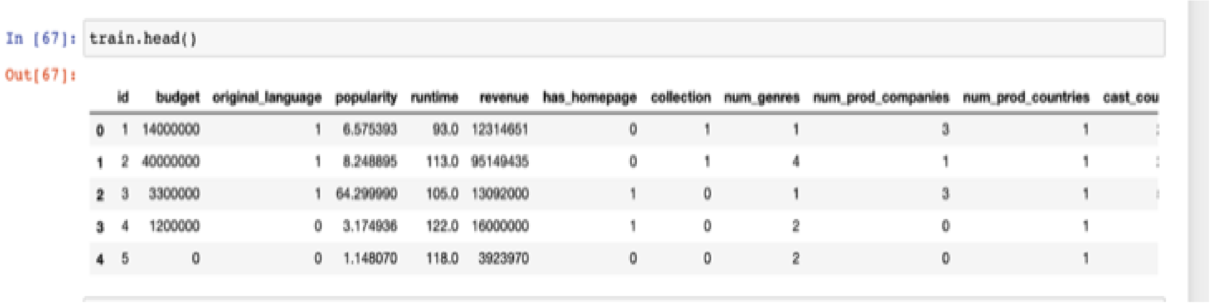
From the below bar plot, 0 represents Sunday and 6 represents Saturday respectively. We observe that the movies released on Tuesday had got high revenue and the revenue has less as we got towards the weekend.

Revenue vs. Tagline

Since Tagline is a column given, which might be an ambiguous factor and may lead to the false results. So we drop the tagline column in order so that no false factors are remained.

5. THE FINAL DATASET

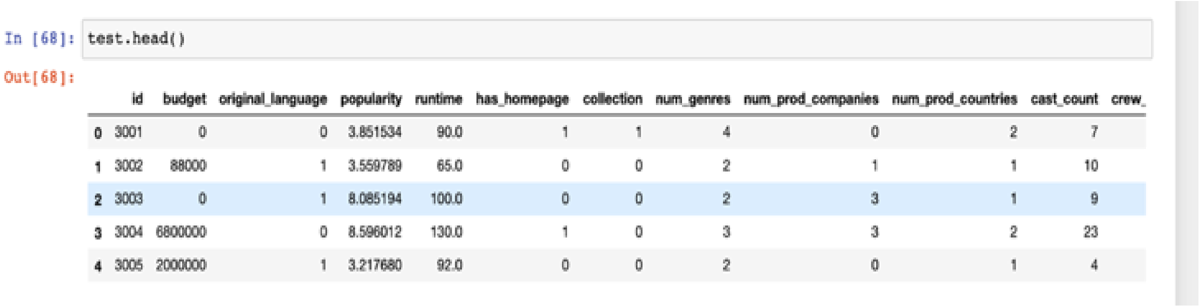
The below are the final variables left in our DataFrame after Dropping all the unnecessary variables.



```
In [67]: train.head()
Out[67]:
```

	id	budget	original_language	popularity	runtime	revenue	has_homepage	collection	num_genres	num_prod_companies	num_prod_countries	cast_cou
0	1	14000000	1	6.575393	93.0	12314651	0	1	1	3	1	
1	2	40000000	1	8.248895	113.0	95149435	0	1	4	1	1	
2	3	3300000	1	64.299990	105.0	13092000	1	0	1	3	1	
3	4	1200000	0	3.174936	122.0	16000000	1	0	2	0	1	
4	5	0	0	1.148070	118.0	3923970	0	0	2	0	1	

Figure 12 Train set



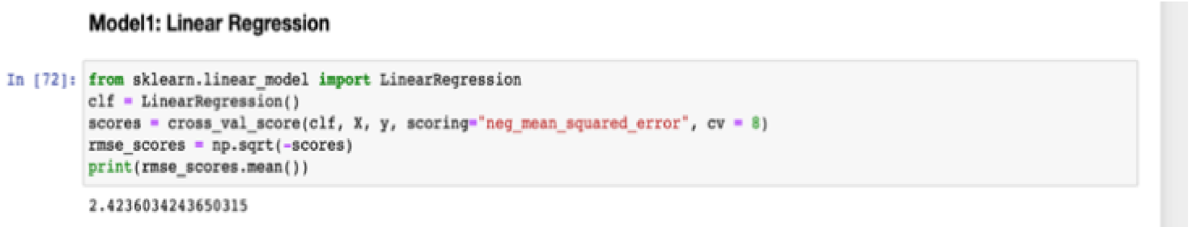
```
In [68]: test.head()
Out[68]:
```

	id	budget	original_language	popularity	runtime	has_homepage	collection	num_genres	num_prod_companies	num_prod_countries	cast_count	crew_
0	3001	0	0	3.851534	90.0	1	1	4	0	2	7	
1	3002	88000	1	3.559789	65.0	0	0	2	1	1	10	
2	3003	0	1	8.085194	100.0	0	0	2	3	1	9	
3	3004	6800000	0	8.596012	130.0	1	0	3	3	2	23	
4	3005	2000000	1	3.217680	92.0	0	0	2	0	1	4	

Figure 13 Test set

6. TRAINING THE MODEL

Linear Regression



Model1: Linear Regression

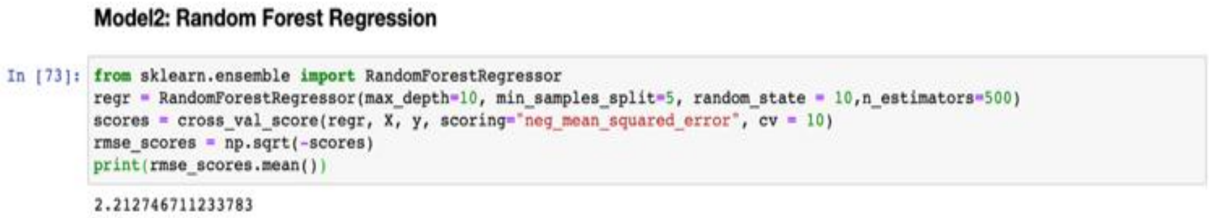
```
In [72]: from sklearn.linear_model import LinearRegression
         clf = LinearRegression()
         scores = cross_val_score(clf, X, y, scoring="neg_mean_squared_error", cv = 8)
         rmse_scores = np.sqrt(-scores)
         print(rmse_scores.mean())

         2.4236034243650315
```

Figure 14 Linear Regression

Root Mean Square value of the Linear Regression obtained for our DataSet is 2.4236

Random Forest Regression



Model2: Random Forest Regression

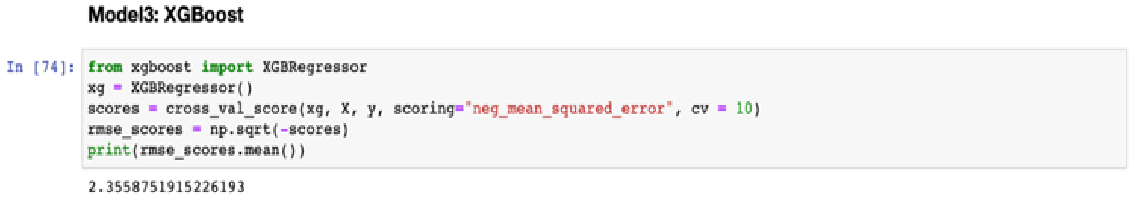
```
In [73]: from sklearn.ensemble import RandomForestRegressor
         regr = RandomForestRegressor(max_depth=10, min_samples_split=5, random_state = 10,n_estimators=500)
         scores = cross_val_score(regr, X, y, scoring="neg_mean_squared_error", cv = 10)
         rmse_scores = np.sqrt(-scores)
         print(rmse_scores.mean())

         2.212746711233783
```

Figure 15 Random Forest Regression

Root Mean Square value of the Random Forest Regression obtained for our DataSet is 2.2127.

XGBoost



Model3: XGBoost

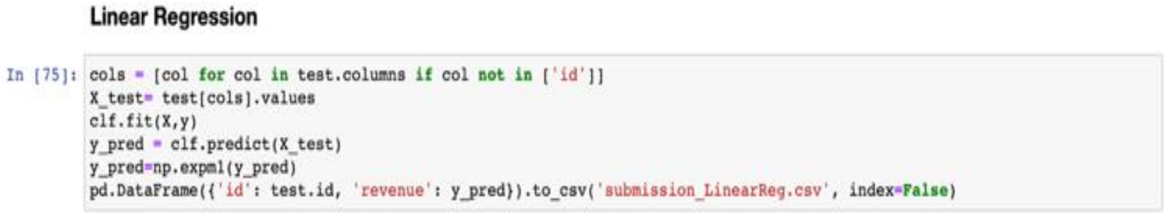
```
In [74]: from xgboost import XGBRegressor
         xg = XGBRegressor()
         scores = cross_val_score(xg, X, y, scoring="neg_mean_squared_error", cv = 10)
         rmse_scores = np.sqrt(-scores)
         print(rmse_scores.mean())

         2.3558751915226193
```

Figure 16 XGBoost

Root Mean Square value of the Random Forest Regression obtained for our Dataset is 2.3558.

7. TESTING



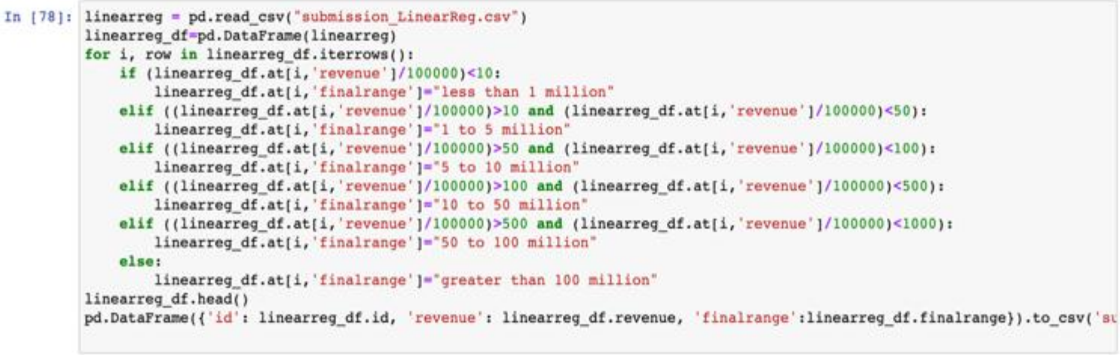
Linear Regression

```
In [75]: cols = [col for col in test.columns if col not in ['id']]
         X_test= test[cols].values
         clf.fit(X,y)
         y_pred = clf.predict(X_test)
         y_pred=np.expm1(y_pred)
         pd.DataFrame({'id': test.id, 'revenue': y_pred}).to_csv('submission_LinearReg.csv', index=False)
```

Figure 17 Linear Regression Prediction

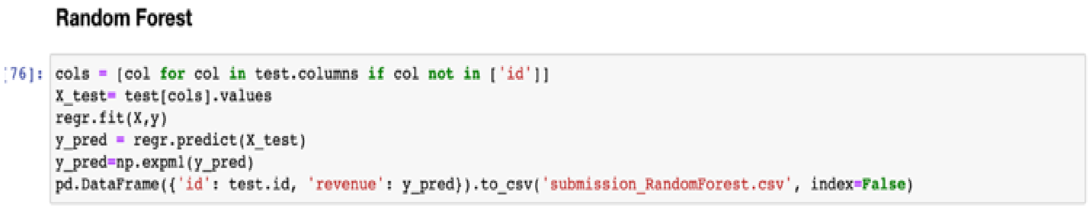
From the above we obtained a CSV file containing the MovieId and the revenue. So we also wanted to include in which range the Movie Falls in. This will have more accuracy compared to that of discrete values. So we did the following code execution and obtained the ranges for each and every Movie. The ranges are as follows:

- Less than 1 Million
- 1 to 5 Million
- 5 to 10 Million
- 10 to 50 Million
- 50 to 100 Million
- Greater than 100 Million



```
In [78]: linearreg = pd.read_csv("submission_LinearReg.csv")
         linearreg_df=pd.DataFrame(linearreg)
         for i, row in linearreg_df.iterrows():
             if (linearreg_df.at[i,'revenue']/100000)<10:
                 linearreg_df.at[i,'finalrange']="less than 1 million"
             elif ((linearreg_df.at[i,'revenue']/100000)>10 and (linearreg_df.at[i,'revenue']/100000)<50):
                 linearreg_df.at[i,'finalrange']="1 to 5 million"
             elif ((linearreg_df.at[i,'revenue']/100000)>50 and (linearreg_df.at[i,'revenue']/100000)<100):
                 linearreg_df.at[i,'finalrange']="5 to 10 million"
             elif ((linearreg_df.at[i,'revenue']/100000)>100 and (linearreg_df.at[i,'revenue']/100000)<500):
                 linearreg_df.at[i,'finalrange']="10 to 50 million"
             elif ((linearreg_df.at[i,'revenue']/100000)>500 and (linearreg_df.at[i,'revenue']/100000)<1000):
                 linearreg_df.at[i,'finalrange']="50 to 100 million"
             else:
                 linearreg_df.at[i,'finalrange']="greater than 100 million"
         linearreg_df.head()
         pd.DataFrame({'id': linearreg_df.id, 'revenue': linearreg_df.revenue, 'finalrange':linearreg_df.finalrange}).to_csv('su
```

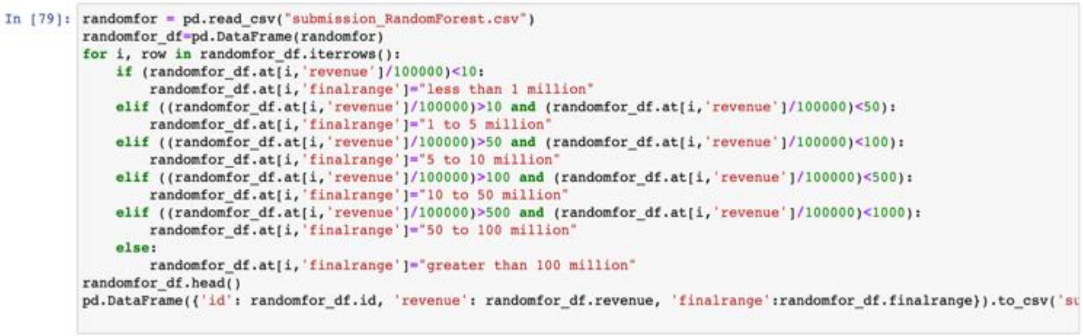
Figure 18 Revenue in Millions



Random Forest

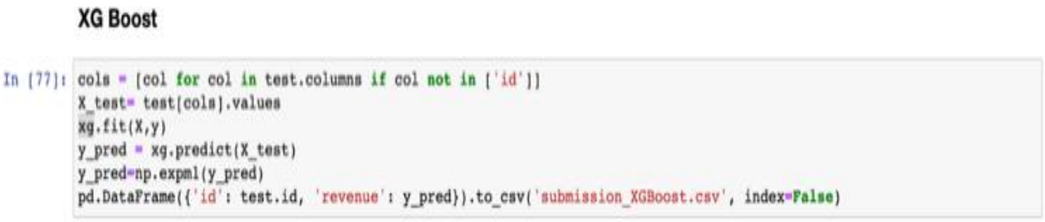
```
[76]: cols = [col for col in test.columns if col not in ['id']]
      X_test= test[cols].values
      regr.fit(X,y)
      y_pred = regr.predict(X_test)
      y_pred=np.expm1(y_pred)
      pd.DataFrame({'id': test.id, 'revenue': y_pred}).to_csv('submission_RandomForest.csv', index=False)
```

Figure 19 Random Forest Prediction



```
In [79]: randomfor = pd.read_csv("submission_RandomForest.csv")
         randomfor_df=pd.DataFrame(randomfor)
         for i, row in randomfor_df.iterrows():
             if (randomfor_df.at[i,'revenue']/100000)<10:
                 randomfor_df.at[i,'finalrange']="less than 1 million"
             elif ((randomfor_df.at[i,'revenue']/100000)>10 and (randomfor_df.at[i,'revenue']/100000)<50):
                 randomfor_df.at[i,'finalrange']="1 to 5 million"
             elif ((randomfor_df.at[i,'revenue']/100000)>50 and (randomfor_df.at[i,'revenue']/100000)<100):
                 randomfor_df.at[i,'finalrange']="5 to 10 million"
             elif ((randomfor_df.at[i,'revenue']/100000)>100 and (randomfor_df.at[i,'revenue']/100000)<500):
                 randomfor_df.at[i,'finalrange']="10 to 50 million"
             elif ((randomfor_df.at[i,'revenue']/100000)>500 and (randomfor_df.at[i,'revenue']/100000)<1000):
                 randomfor_df.at[i,'finalrange']="50 to 100 million"
             else:
                 randomfor_df.at[i,'finalrange']="greater than 100 million"
         randomfor_df.head()
         pd.DataFrame({'id': randomfor_df.id, 'revenue': randomfor_df.revenue, 'finalrange':randomfor_df.finalrange}).to_csv('su
```

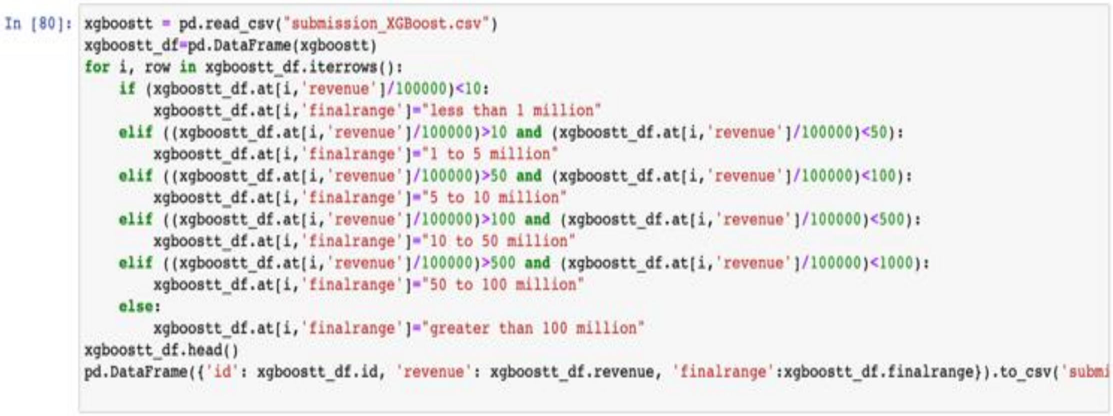
Figure 20 Revenue in Millions



XG Boost

```
In [77]: cols = [col for col in test.columns if col not in ['id']]
         X_test= test[cols].values
         xg.fit(X,y)
         y_pred = xg.predict(X_test)
         y_pred=np.expm1(y_pred)
         pd.DataFrame({'id': test.id, 'revenue': y_pred}).to_csv('submission_XGBoost.csv', index=False)
```

Figure 21 XG Boost Prediction



```
In [80]: xgboostt = pd.read_csv("submission_XGBoost.csv")
         xgboostt_df=pd.DataFrame(xgboostt)
         for i, row in xgboostt_df.iterrows():
             if (xgboostt_df.at[i,'revenue']/100000)<10:
                 xgboostt_df.at[i,'finalrange']="less than 1 million"
             elif ((xgboostt_df.at[i,'revenue']/100000)>10 and (xgboostt_df.at[i,'revenue']/100000)<50):
                 xgboostt_df.at[i,'finalrange']="1 to 5 million"
             elif ((xgboostt_df.at[i,'revenue']/100000)>50 and (xgboostt_df.at[i,'revenue']/100000)<100):
                 xgboostt_df.at[i,'finalrange']="5 to 10 million"
             elif ((xgboostt_df.at[i,'revenue']/100000)>100 and (xgboostt_df.at[i,'revenue']/100000)<500):
                 xgboostt_df.at[i,'finalrange']="10 to 50 million"
             elif ((xgboostt_df.at[i,'revenue']/100000)>500 and (xgboostt_df.at[i,'revenue']/100000)<1000):
                 xgboostt_df.at[i,'finalrange']="50 to 100 million"
             else:
                 xgboostt_df.at[i,'finalrange']="greater than 100 million"
         xgboostt_df.head()
         pd.DataFrame({'id': xgboostt_df.id, 'revenue': xgboostt_df.revenue, 'finalrange':xgboostt_df.finalrange}).to_csv('submi
```

Figure 22 Revenue in Millions

8. RESULTS

ID 3010 - Toy Story 2

Revenue Collected - 497.4 Million

Model	Predicted Revenue	Predicted Range
Linear Model	357817471.1728	Greater than 100 million

Random Forest	325920667.4186	Greater than 100 million
XGBoost	530433300.0	Greater than 100 million

ID 3045: Captain America: The First Avenger

Revenue Collected – 370 Million

Model	Predicted Revenue	Predicted Range
Linear Model	742777120.7819	Greater than 100 million
Random Forest	433173030.8184	Greater than 100 million
XGBoost	260447170	Greater than 100 million

ID 3069 – Rango

Revenue Collected – 245.7 Million

Model	Predicted Revenue	Predicted Range
Linear Model	37264518.44930320	10 to 50 million
Random Forest	277902660	Greater than 100 million
XGBoost	199690897.1021	Greater than 100

ID 3595: When a Stranger Calls

Revenue Collected – 67 Million

Model	Predicted Revenue	Predicted Range
Linear Model	32861522.65513290	10 to 50 million
Random Forest	16614550.0482	10 to 50 million
GBoost	24534494	10 to 50 million

9. CONCLUSION

The film industry is an unpredictable business either the producer gain higher profits or they might get into huge loss, so it is difficult for a human to predict that the movie box-office prior to the release. Although it is very important for production studios to be able to predict the movie box office revenues before they are released, the prediction of box office revenue is still classified as an art rather than a science because most experts predict revenue based on their own rules of thumb, hunches, and their experience. This project helps the production studios to predict box office revenues that can be used to decide for planning the production and the movie distribution stages.

This process gave outputs with less Root mean square Error for Random Forest is less when compared to others and Random forest gave better results than the Linear Regression and the XGBoost Algorithms.

10. REFERENCES

[1] Matt Vitelli mvitelli. Predicting Box Office Revenue for Movies. *Semantic scholar* (2015)

[2] Zhang, W., Skiena, S. Improving movie gross prediction through news analysis. *In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (2009)

[3] Márton Mestyán, Taha Yasseri, János Kertész. Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data. *In: Plus Journal* (2013)

[4] Javaria Ahmad, Prakash Duraisamy, Amr Yousef, Bill Buckles. Movie success prediction using data mining. *IEEE 2017 8th International Conference.*

[5] Prashant Rajput, Priyanka Sapkal, and Shefali Sinha. Box Office Revenue Prediction Using Dual Sentiment Analysis. *International Journal of Machine Learning and Computing*, Vol. 7, No. 4, August 2017

[6] Ahmada Azuraliza, Abu Bakar bMohd, Ridzwan Yaakubb. Movie Revenue Prediction Based on Purchase Intention Mining Using YouTube Trailer Reviews. *Information Processing & Management* (2020)

[7] Antara Upadhyay, Nivedita Kamath, Shalin Shanghavi, Tanisha Mandvikar, Pranali Wagh. Movie Success Prediction Using Data Mining. *IJEDR* 2018

[8] Taegu Kim, Jungsik Hong and Pilsung Kang. Box Office Forecasting considering Competitive Environment and Word-of-Mouth in Social Networks: A Case Study of Korean Film Market. *Computational Intelligence and Neuroscience*, 2017.

[9] Krishnamoorthy A., Vijayarajan V., Sapthagiri R. (2019) Automated Shopping Experience Using Real-Time IoT. In: Satapathy S., Bhateja V., Somanah R., Yang XS., Senkerik R. (eds) *Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, vol 862. Springer, Singapore.* https://doi.org/10.1007/978-981-13-3329-3_20

[10] Arundeep Kaur, AP Gurpinder Kaur. Predicting Movie Success: Review of Existing Literature. *International Journal of Advanced Research in Computer Science and Software Engineering* (2013).

[11] Ms. Mary Margarat Valentine, Ms. Veena Kulkarni, Dr.R.R. Sedamkar. A Model for Predicting Movie's Performance using Online Rating and Revenue. *International Journal of Scientific & Engineering Research*, Volume 4, Issue 9, September-2013 277.

[12] Ajay Siva Santosh Reddy, Pratik Kasat, Abhiyash Jain. Box-Office Opening Prediction of Movies based on Hype Analysis through Data Mining. *International Journal of Computer Applications* (0975 – 8887) Volume 56– No.1, October 2012.