RATING PREDICTION FOR UPCOMING HOLLYWOOD MOVIES

Divya Sarika Amjuri

S00404824

Auburn University at Montgomery

Table Of Contents

1. Introduction

1.1 Objective

To develop a model for predicting movie ratings for upcoming movies based on the given past data.

1.2 Need of the study

Nowadays, we have so many movie streaming platforms like OTTs such as Hulu, Netflix, Disney Plus etc. Knowing a movie's rating before watching the movie will help us decide which movie to watch out of so many options.

For that, users either search for the movie in Google and or directly go to IMDb (Internet Movie database) to get details related to the movie such as rating, reviews etc. Unfortunately, IMDb allows users to provide ratings and reviews only after a movie release. If we want to select a movie for a first day show, we don't have the ratings to rely on.

To address the above problem and more, this study is undertaken to

- Decide whether to watch the movie or not before its release
- Buyers representing the theatres can use the model to decide which movie to lease and duration of that lease for optimal box-office collection
- Movie makers can use the resultant ML model to find a perfect recipe for a movie

2. Data Report

2.1 Data acquisition

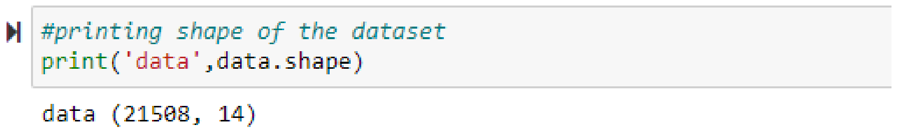
I got data from the dataset `result_dataset_directors_huge_runtime`

2.2 Visual inspection of data

This dataset has 21508 observations and 14 attributes.

- 10 attributes named `title_id`, `title_name`, `genre1`, `genre2`, `genre3`, `actor1`, `actor2`, `actor3`, `actor4`, `director_name` are of `object` type
- Rating attribute is of `float` type
- 3 attributes named `year`, `numVotes` and `runtime` are of `integer` type

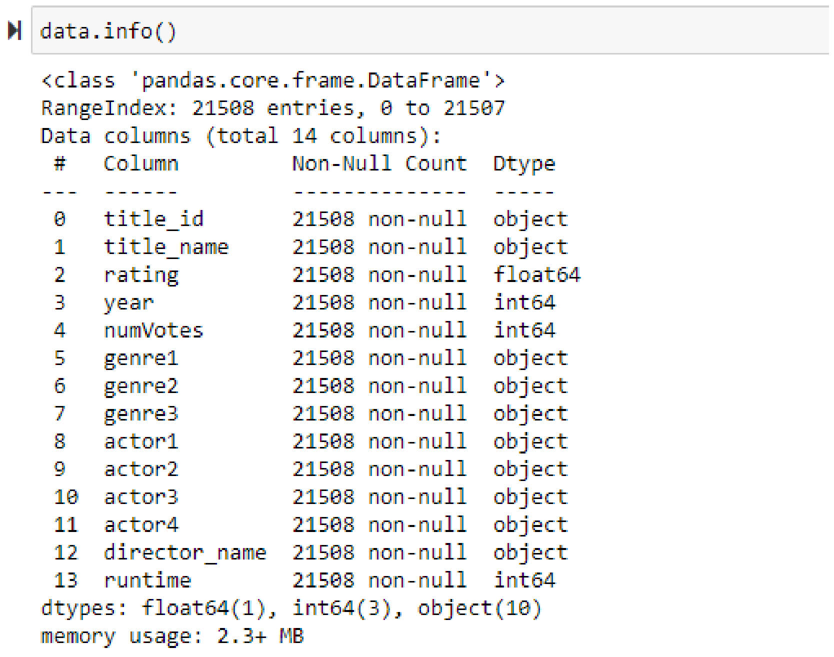
We can see the number of rows and columns from the Figure No.1 and Figure No.2 shows us the datatypes of attributes



```
#printing shape of the dataset
print('data',data.shape)
```

```
data (21508, 14)
```

Figure No.1



```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21508 entries, 0 to 21507
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   title_id       21508 non-null  object 
 1   title_name     21508 non-null  object 
 2   rating         21508 non-null  float64
 3   year           21508 non-null  int64  
 4   numVotes       21508 non-null  int64  
 5   genre1         21508 non-null  object 
 6   genre2         21508 non-null  object 
 7   genre3         21508 non-null  object 
 8   actor1         21508 non-null  object 
 9   actor2         21508 non-null  object 
 10  actor3         21508 non-null  object 
 11  actor4         21508 non-null  object 
 12  director_name  21508 non-null  object 
 13  runtime        21508 non-null  int64  
dtypes: float64(1), int64(3), object(10)
memory usage: 2.3+ MB
```

Figure No. 2

2.3 Filtering the data frame

For this problem, we are just considering full-length feature films released after 1980.

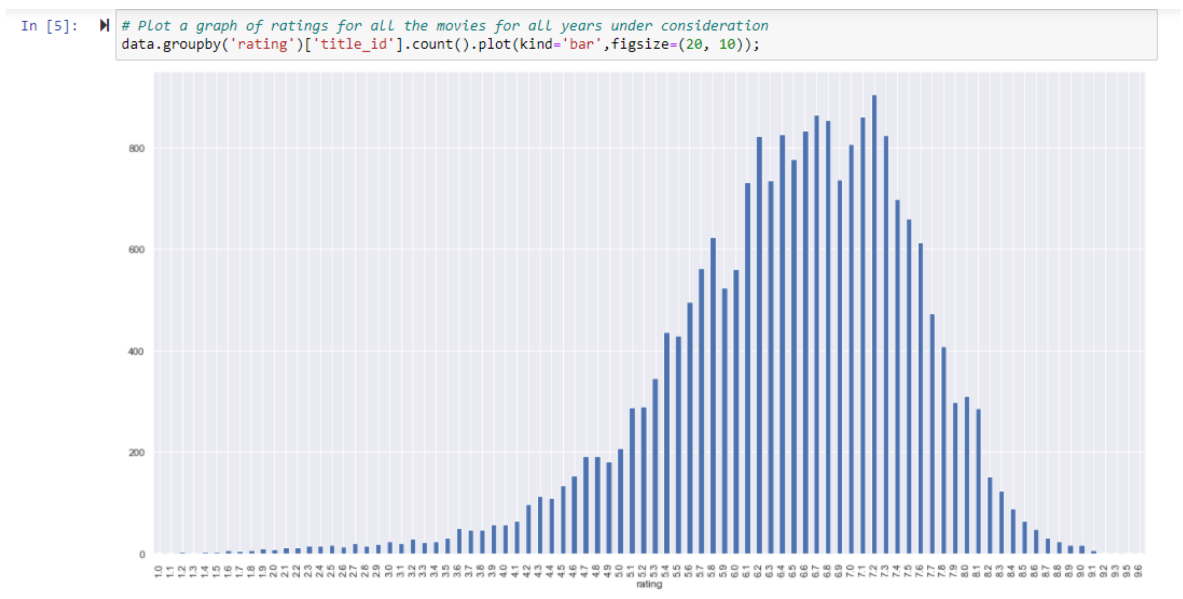
We are considering only the movies which have been released after 1980. This is because the ratings of a movie are hugely expected to be influenced by the cast and crew involved. Most of the crew involved in movie making today probably would not have any movies to their credit before 1980.

3. Data Exploration

3.1 1st Iteration

Following visualisations have been plotted

- Distribution of movies throughout the years under consideration across different ratings



```
In [5]:  # Plot a graph of ratings for all the movies for all years under consideration
         data.groupby('rating')['title_id'].count().plot(kind='bar',figsize=(20, 10));
```

Figure No. 3: Distribution of movies throughout the year

Observations

- The average rating of the movies seems to be around 7
- The movie ratings are slightly left distributed

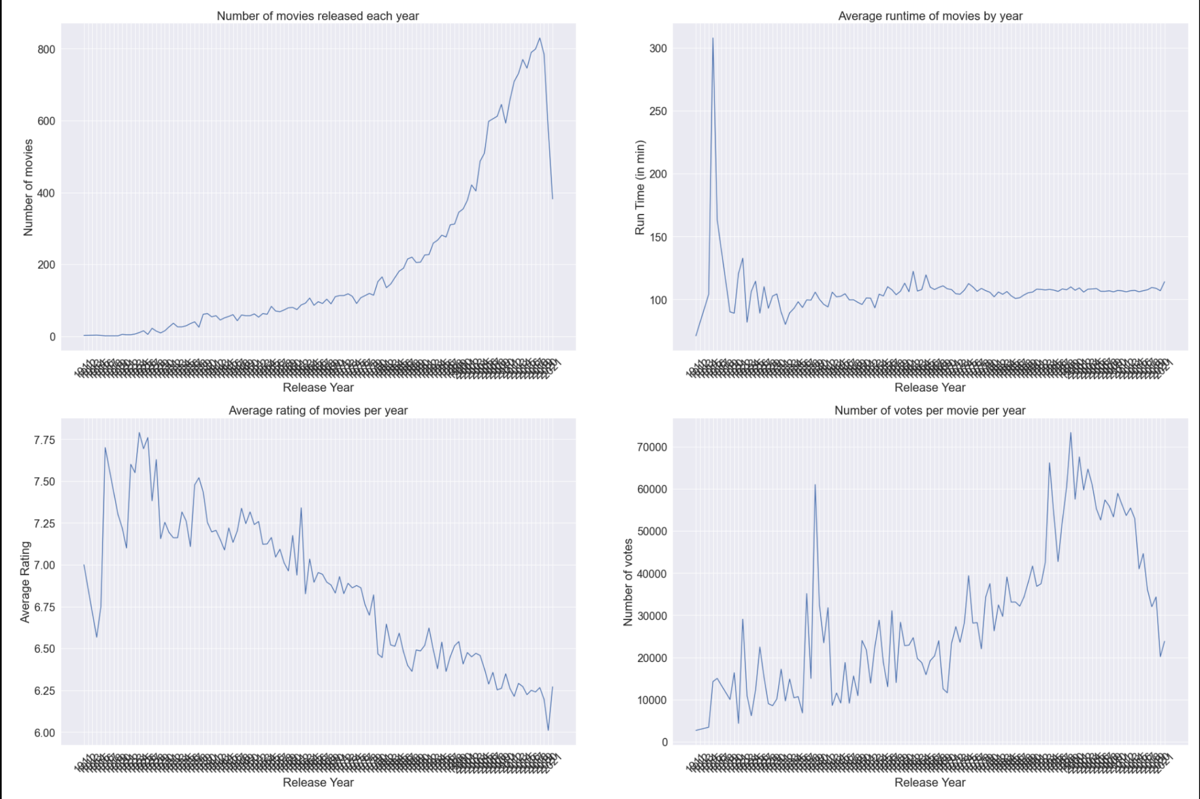


Figure No. 4: Year wise trends

Observations:

- Number of movies released each year
- Average runtime of movies
- Average rating of movies
- Number of votes per movie

In the data wrangling phase, only the movies released after 1980 were considered. Above initial visualisations revealed inconsistent trends before 1980 on multiple features. Due to that, we are considering the movies released after 1980 for the purpose of this project.

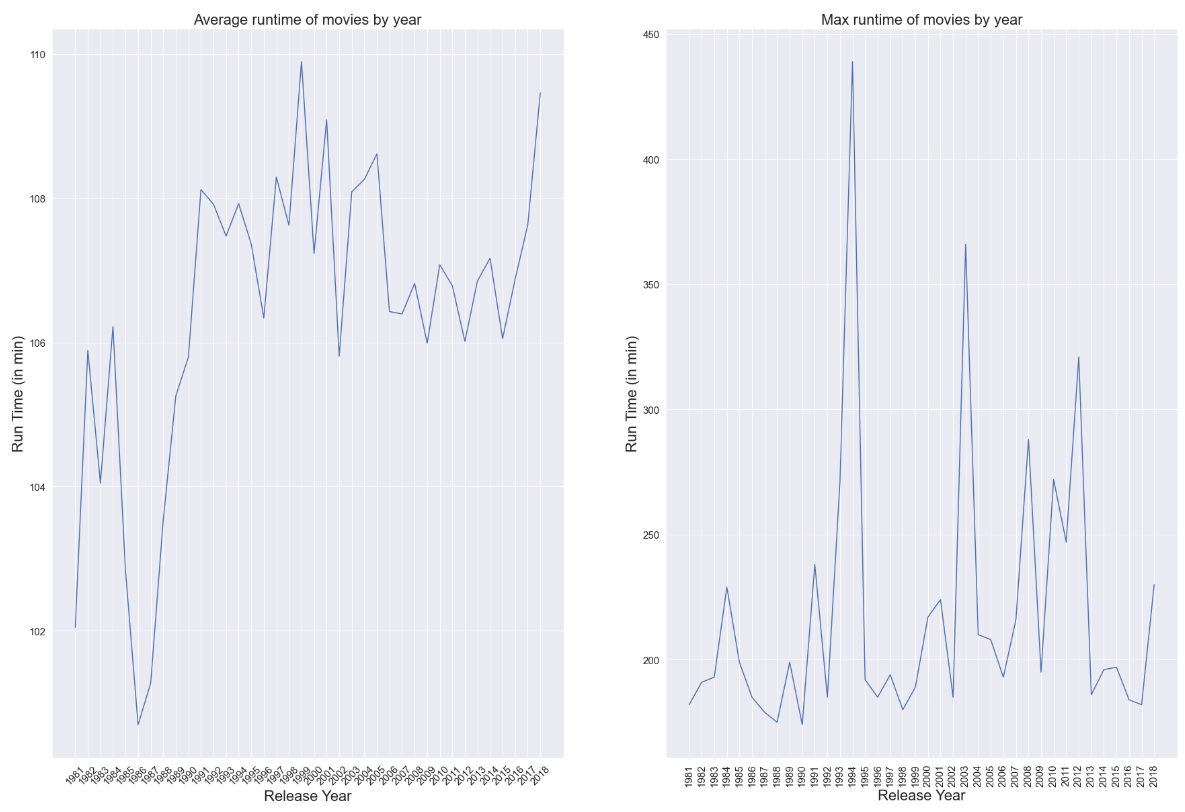
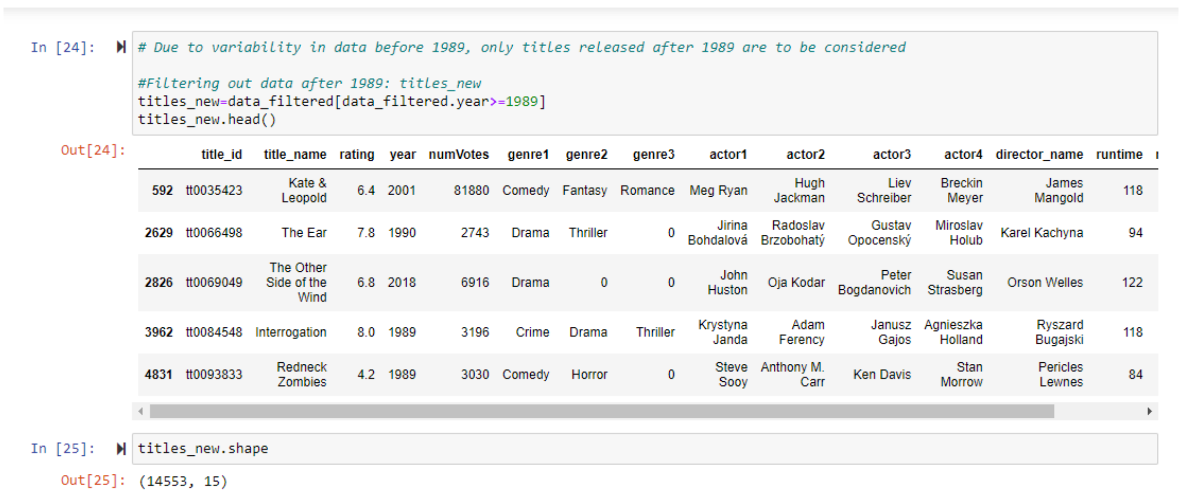


Figure No. 5: Trend of maximum runtime of movies over the years

This trend revealed the maximum runtime of one of the movies. Most feature films are between 70 and 210 minutes long (Reference: https://en.wikipedia.org/wiki/Feature_film).

The new dataset now has 14553 rows.



```
In [24]:  # Due to variability in data before 1989, only titles released after 1989 are to be considered

          #Filtering out data after 1989: titles_new
          titles_new=data_filtered[data_filtered.year>=1989]
          titles_new.head()
```

Out[24]:

	title_id	title_name	rating	year	numVotes	genre1	genre2	genre3	actor1	actor2	actor3	actor4	director_name	runtime
592	tt0035423	Kate & Leopold	6.4	2001	81880	Comedy	Fantasy	Romance	Meg Ryan	Hugh Jackman	Liev Schreiber	Breckin Meyer	James Mangold	118
2629	tt0066498	The Ear	7.8	1990	2743	Drama	Thriller	0	Jirina Bohdalová	Radoslav Brzobohatý	Gustav Opocenský	Miroslav Holub	Karel Kachyna	94
2826	tt0069049	The Other Side of the Wind	6.8	2018	6916	Drama	0	0	John Huston	Oja Kodar	Peter Bogdanovich	Susan Strasberg	Orson Welles	122
3962	tt0084548	Interrogation	8.0	1989	3196	Crime	Drama	Thriller	Krystyna Janda	Adam Ferency	Janusz Gajos	Agnieszka Holland	Ryszard Bugajski	118
4831	tt0093833	Redneck Zombies	4.2	1989	3030	Comedy	Horror	0	Steve Sooy	Anthony M. Carr	Ken Davis	Stan Morrow	Pericles Lewnes	84

```
In [25]:  titles_new.shape
Out[25]:  (14553, 15)
```

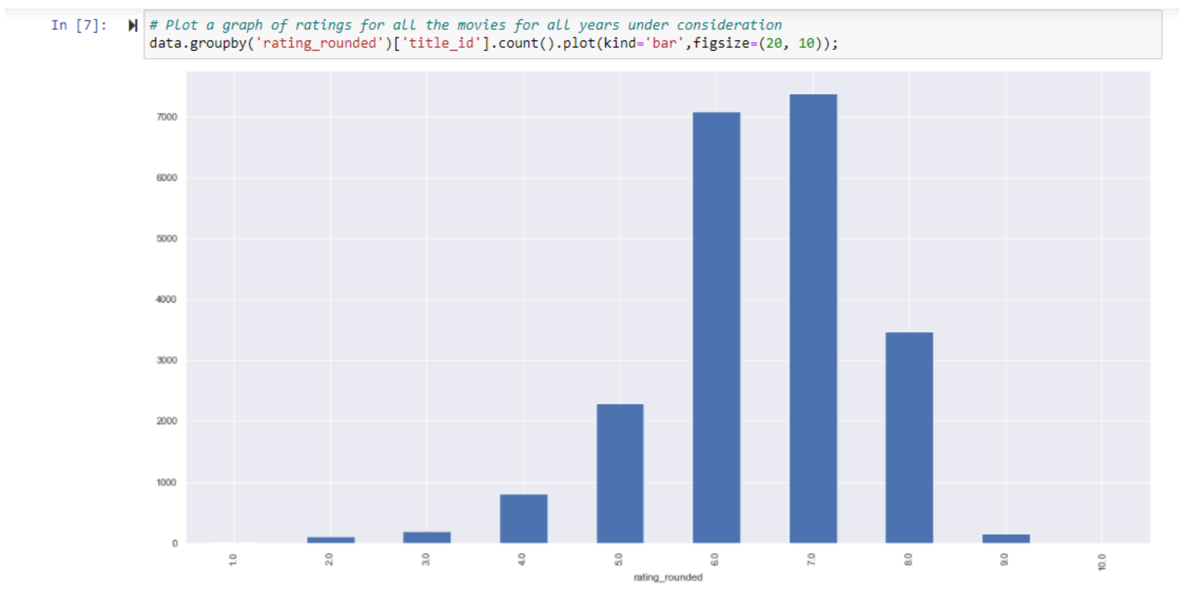
Figure No. 6: filtered dataset

3.2 2nd Iteration

The trends among different parameters after filtering the dataset.

Visualisations and observations:

- Distribution of titles among the ratings
  - The distribution is slightly left skewed
  - Mean rating is between 6 and 7



```
In [7]:  # Plot a graph of ratings for all the movies for all years under consideration
         data.groupby('rating_rounded')['title_id'].count().plot(kind='bar',figsize=(20, 10));
```

Figure No. 7: Distribution of titles among the ratings

- Year wise trends
  - Number of movies released have increased over the years
  - Average rating shows a decreasing trend

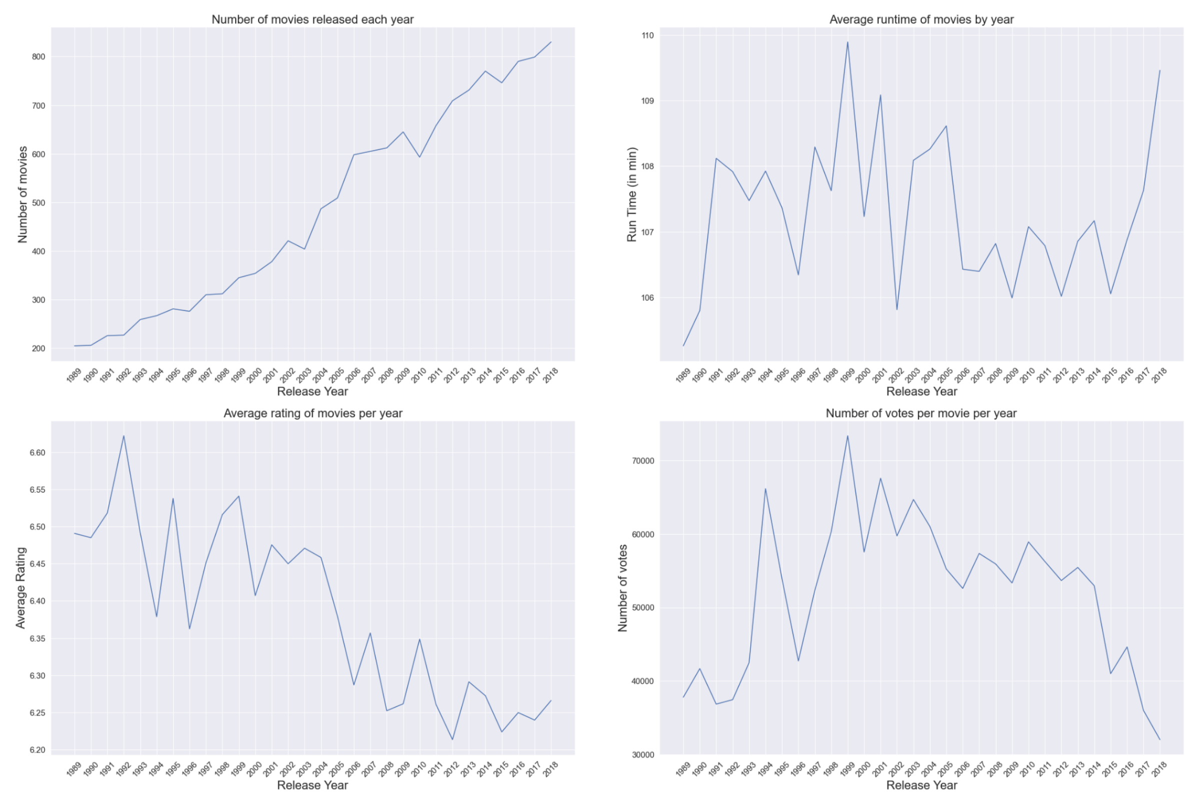


Figure No. 8 Year wise trends

We will explore the relationship between dataset columns and the movie ratings.

3.3 Length of the movie

Length of the movies is positively correlated with the ratings. This is evident from the graph below. The average runtime of movies is around 107 minutes. Movies with a runtime of more than 170 minutes have higher ratings.

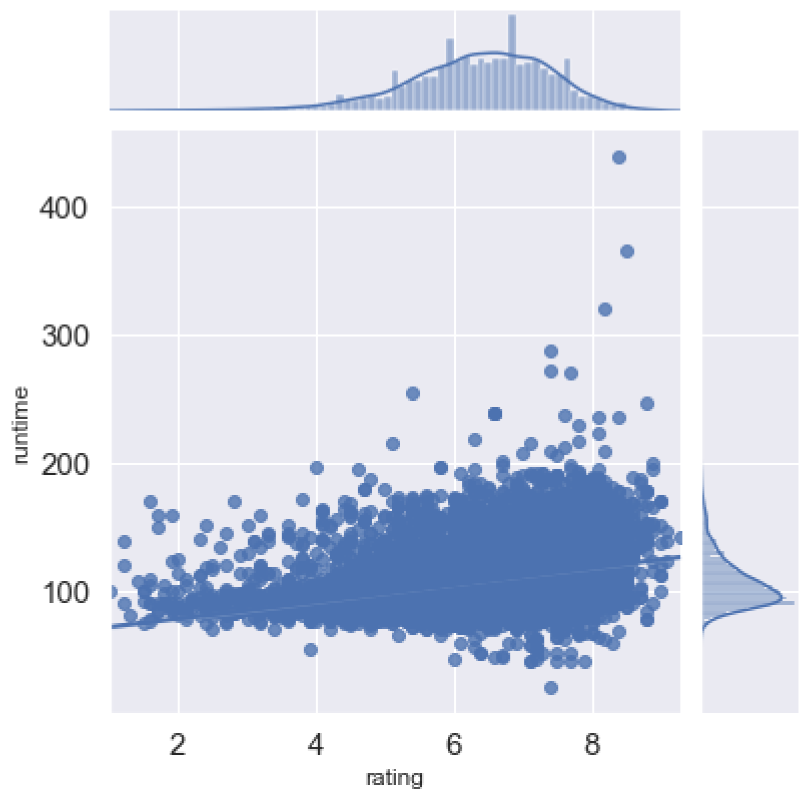


Figure No. 9 Runtime vs Rating

3.4 Number of votes

Number of votes could not be used to predict the ratings, as this will not be available to us during prediction. We have still plotted it to get insights.

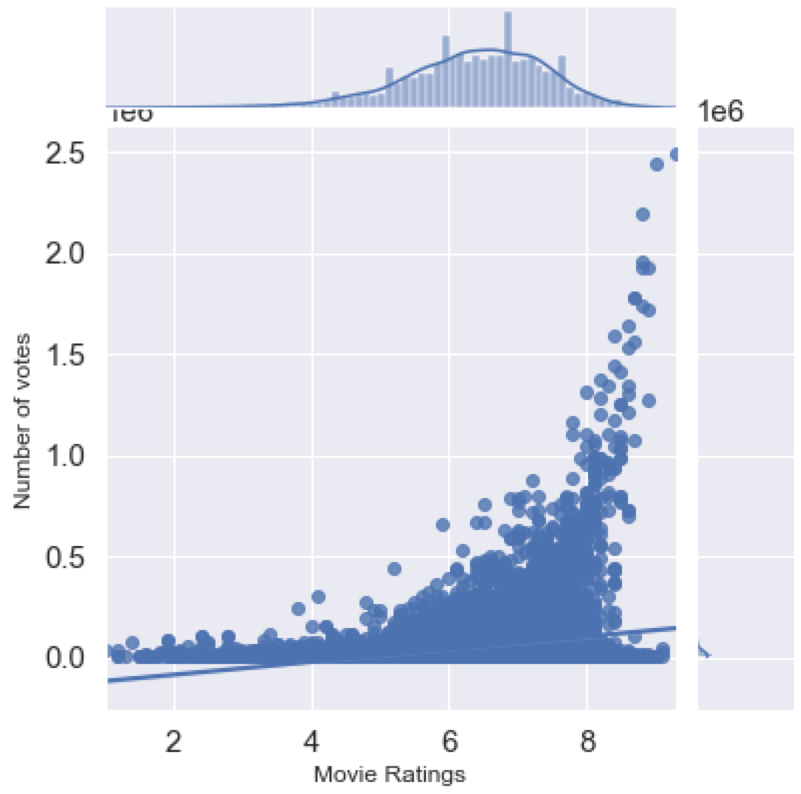


Figure No. 10 Number of votes vs Movie ratings

Observations:

- Number of votes for a movie are positively correlated to the rating.
- Most of the cases of high vote count are for highly rated movies.

From this we can conclude that people are more likely to vote for movies if they like it.

4. Feature Engineering

We have some categorical features that have to be converted to numerical variables

4.1 Genre of the movie

There are 3 different columns of genres. This is a categorical variable with 23 different categories. We convert all these categories into features by adding a column for each of them with a boolean value for each row. Let's check genres that might be applicable to a few titles. We will check for such titles and delete the columns for genre entries which are applicable to less than 100 titles.This leads to 17 genre columns.

Observations:

- Genres with higher than average rating are - history, war, animation
- So it appears that movies that are based on real life and imagination have a higher rating
- Horror movies are rated much lower than average.

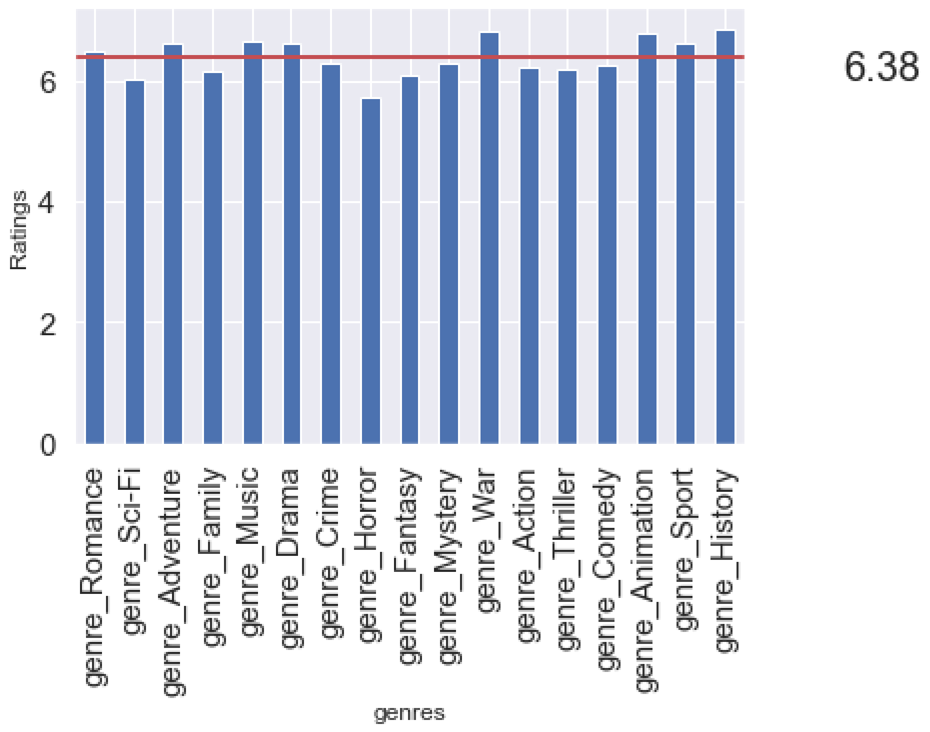


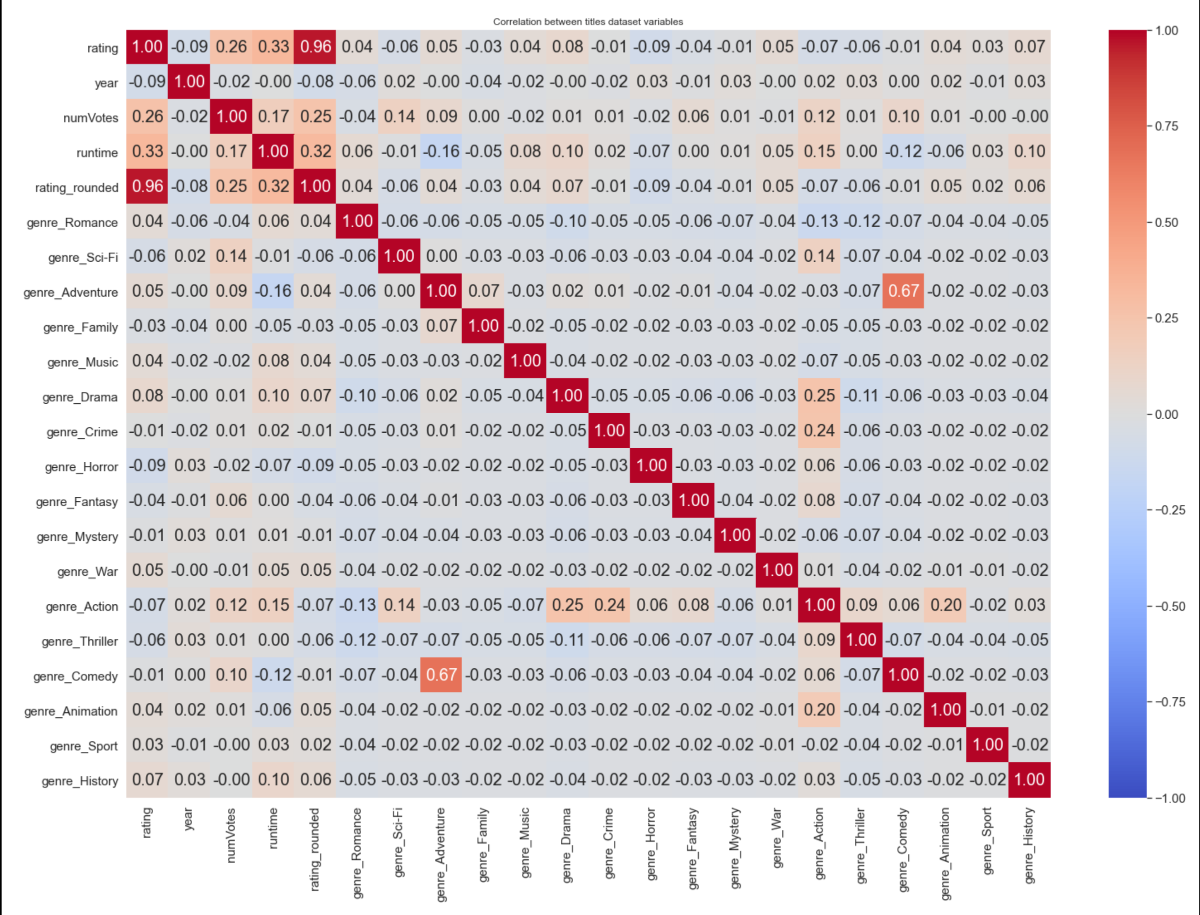
Figure No. 11 Ratings vs genres

4.2 Most frequent cast/crew

There might be some cast/crew that affects the ratings of the movies.

- It is impossible to incorporate all the crew members
- To incorporate the cast/crew, we are considering the 10 most frequent cast/crew each in the top-rated (rating 8 and above) and bottom-rated (rating 4 and below) movies
- These 20 (most frequent-10 from top rated movies, 10 from worst rated movies) cast/crew are added to the dataset as columns with boolean values.

4.3 Correlation between parameters



	rating	year	numVotes	runtime	rating_rounded	genre_Romance	genre_Sci-Fi	genre_Adventure	genre_Family	genre_Music	genre_Drama	genre_Crime	genre_Horror	genre_Fantasy	genre_Mystery	genre_War	genre_Action	genre_Thriller	genre_Comedy	genre_Animation	genre_Sport	genre_History
rating	1.00	-0.09	0.26	0.33	0.96	0.04	-0.06	0.05	-0.03	0.04	0.08	-0.01	-0.09	-0.04	-0.01	0.05	-0.07	-0.06	-0.01	0.04	0.03	0.07
year	-0.09	1.00	-0.02	-0.00	-0.08	-0.06	0.02	-0.00	-0.04	-0.02	-0.00	-0.02	0.03	-0.01	0.03	-0.00	0.02	0.03	0.00	0.02	-0.01	0.03
numVotes	0.26	-0.02	1.00	0.17	0.25	-0.04	0.14	0.09	0.00	-0.02	0.01	0.01	-0.02	0.06	0.01	-0.01	0.12	0.01	0.10	0.01	-0.00	-0.00
runtime	0.33	-0.00	0.17	1.00	0.32	0.06	-0.01	-0.16	-0.05	0.08	0.10	0.02	-0.07	0.00	0.01	0.05	0.15	0.00	-0.12	-0.06	0.03	0.10
rating_rounded	0.96	-0.08	0.25	0.32	1.00	0.04	-0.06	0.04	-0.03	0.04	0.07	-0.01	-0.09	-0.04	-0.01	0.05	-0.07	-0.06	-0.01	0.05	0.02	0.06
genre_Romance	0.04	-0.06	-0.04	0.06	0.04	1.00	-0.06	-0.06	-0.05	-0.05	-0.10	-0.05	-0.05	-0.06	-0.07	-0.04	-0.13	-0.12	-0.07	-0.04	-0.04	-0.05
genre_Sci-Fi	-0.06	0.02	0.14	-0.01	-0.06	-0.06	1.00	0.00	-0.03	-0.03	-0.06	-0.03	-0.03	-0.04	-0.04	-0.02	0.14	-0.07	-0.04	-0.02	-0.02	-0.03
genre_Adventure	0.05	-0.00	0.09	-0.16	0.04	-0.06	0.00	1.00	0.07	-0.03	0.02	0.01	-0.02	-0.01	-0.04	-0.02	-0.03	-0.07	0.67	-0.02	-0.02	-0.03
genre_Family	-0.03	-0.04	0.00	-0.05	-0.03	-0.05	-0.03	0.07	1.00	-0.02	-0.05	-0.02	-0.02	-0.03	-0.03	-0.02	-0.05	-0.05	-0.03	-0.02	-0.02	-0.02
genre_Music	0.04	-0.02	-0.02	0.08	0.04	-0.05	-0.03	-0.03	-0.02	1.00	-0.04	-0.02	-0.02	-0.03	-0.03	-0.02	-0.07	-0.05	-0.03	-0.02	-0.02	-0.02
genre_Drama	0.08	-0.00	0.01	0.10	0.07	-0.10	-0.06	0.02	-0.05	-0.04	1.00	-0.05	-0.05	-0.06	-0.06	-0.03	0.25	-0.11	-0.06	-0.03	-0.03	-0.04
genre_Crime	-0.01	-0.02	0.01	0.02	-0.01	-0.05	-0.03	0.01	-0.02	-0.02	-0.05	1.00	-0.03	-0.03	-0.03	-0.02	0.24	-0.06	-0.03	-0.02	-0.02	-0.02
genre_Horror	-0.09	0.03	-0.02	-0.07	-0.09	-0.05	-0.03	-0.02	-0.02	-0.02	-0.05	-0.03	1.00	-0.03	-0.03	-0.02	0.06	-0.06	-0.03	-0.02	-0.02	-0.02
genre_Fantasy	-0.04	-0.01	0.06	0.00	-0.04	-0.06	-0.04	-0.01	-0.03	-0.03	-0.06	-0.03	-0.03	1.00	-0.04	-0.02	0.08	-0.07	-0.04	-0.02	-0.02	-0.03
genre_Mystery	-0.01	0.03	0.01	0.01	-0.01	-0.07	-0.04	-0.04	-0.03	-0.03	-0.06	-0.03	-0.03	-0.04	1.00	-0.02	-0.06	-0.07	-0.04	-0.02	-0.02	-0.03
genre_War	0.05	-0.00	-0.01	0.05	0.05	-0.04	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02	-0.02	1.00	0.01	-0.04	-0.02	-0.01	-0.01	-0.02
genre_Action	-0.07	0.02	0.12	0.15	-0.07	-0.13	0.14	-0.03	-0.05	-0.07	0.25	0.24	0.06	0.08	-0.06	0.01	1.00	0.09	0.06	0.20	-0.02	0.03
genre_Thriller	-0.06	0.03	0.01	0.00	-0.06	-0.12	-0.07	-0.07	-0.05	-0.05	-0.11	-0.06	-0.06	-0.07	-0.07	-0.04	0.09	1.00	-0.07	-0.04	-0.04	-0.05
genre_Comedy	-0.01	0.00	0.10	-0.12	-0.01	-0.07	-0.04	0.67	-0.03	-0.03	-0.06	-0.03	-0.03	-0.04	-0.04	-0.02	0.06	-0.07	1.00	-0.02	-0.02	-0.03
genre_Animation	0.04	0.02	0.01	-0.06	0.05	-0.04	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02	-0.02	-0.01	0.20	-0.04	-0.02	1.00	-0.01	-0.02
genre_Sport	0.03	-0.01	-0.00	0.03	0.02	-0.04	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02	-0.02	-0.01	-0.02	-0.04	-0.02	-0.01	1.00	-0.02
genre_History	0.07	0.03	-0.00	0.10	0.06	-0.05	-0.03	-0.03	-0.02	-0.02	-0.04	-0.02	-0.02	-0.03	-0.03	-0.02	0.03	-0.05	-0.03	-0.02	-0.02	1.00

Figure No. 12 correlation among parameters

For each movie's category, the ratings of cast/crew in that category were aggregated to get the rating for that category. Subsequently, while building our model, we will use the aggregation function that will better serve our model.

Observations:

- The heatmap shows positive correlation between ratings and runtime
- The heatmap shows negative correlation between ratings and year, genres - Crime, fantasy, action, mystery.sci-fi, family, horror, comedy, adventure, thriller.

Although there isn't much correlation between them, the correlation exists is negative.

5. Modelling

5.1 Target and Feature Variables

The target variable ranges from 1 (lowest rating) to 10 (highest rating). As it is a continuous variable, we will be using regression algorithms.

5.2 Train Test Split

Using the entire dataset to train our model might lead to data leakage and thus affect the performance of the trained model on unseen data. To address this issue, we split our dataset into train and test datasets wherein we train our model on the train dataset, and test its performance on the test dataset. For our project, we split our dataset 70% in the train dataset and 30% in the test dataset.

5.3 Machine Learning Process and Methodology

To find the best model for our purpose, we will train our data on different algorithms, compare their performance based on the root mean square error (rmse) of the prediction and time taken to fit/predict the model, and select the best.
Algorithms employed:

- Linear Regression Regularization Model
- Ridge Regularization Model
- Lasso Ensemble Model
- Random Forest Ensemble Model
- Gradient Boost Ensemble Model
- Ada Boost

After training and evaluating our data on the above algorithms, we will also use model stacking.

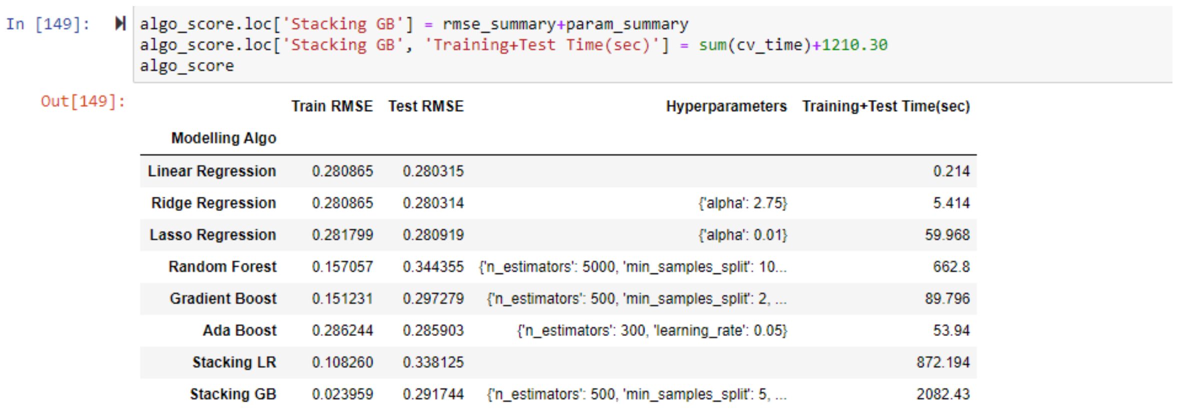
- Stack - Linear Regression
- Stack - Gradient Boost

5.4 Performance Evaluation

To compare the performance of different algorithms, we are using RMSE (Root Mean Square Error) of the prediction and time taken to fit/predict the model, and select the best. The model with lowest RMSE and time taken is desirable.

5.5 Model Comparison

We can compare the RMSE for train and test datasets for different algorithms in figure below



```
In [149]:  algo_score.loc['Stacking GB'] = rmse_summary+param_summary
           algo_score.loc['Stacking GB', 'Training+Test Time(sec)'] = sum(cv_time)+1210.30
           algo_score
Out[149]:
```

Modelling Algo	Train RMSE	Test RMSE	Hyperparameters	Training+Test Time(sec)
Linear Regression	0.280865	0.280315		0.214
Ridge Regression	0.280865	0.280314	{'alpha': 2.75}	5.414
Lasso Regression	0.281799	0.280919	{'alpha': 0.01}	59.968
Random Forest	0.157057	0.344355	{'n_estimators': 5000, 'min_samples_split': 10...	662.8
Gradient Boost	0.151231	0.297279	{'n_estimators': 500, 'min_samples_split': 2, ...	89.796
Ada Boost	0.286244	0.285903	{'n_estimators': 300, 'learning_rate': 0.05}	53.94
Stacking LR	0.108260	0.338125		872.194
Stacking GB	0.023959	0.291744	{'n_estimators': 500, 'min_samples_split': 5, ...	2082.43

Figure No. 13 Model Comparison

Stacking Gradient boost has the lowest RMSE for train dataset. But RMSE for the test dataset is very high, which indicates overfitting.

6. Summary And Conclusion

Below is the graphical representation of training and test RMSE for all algorithms used. The best performing algorithms are also overfitted. That does not concern us much. This is because we are getting low RMSE for the test dataset despite overfitting.

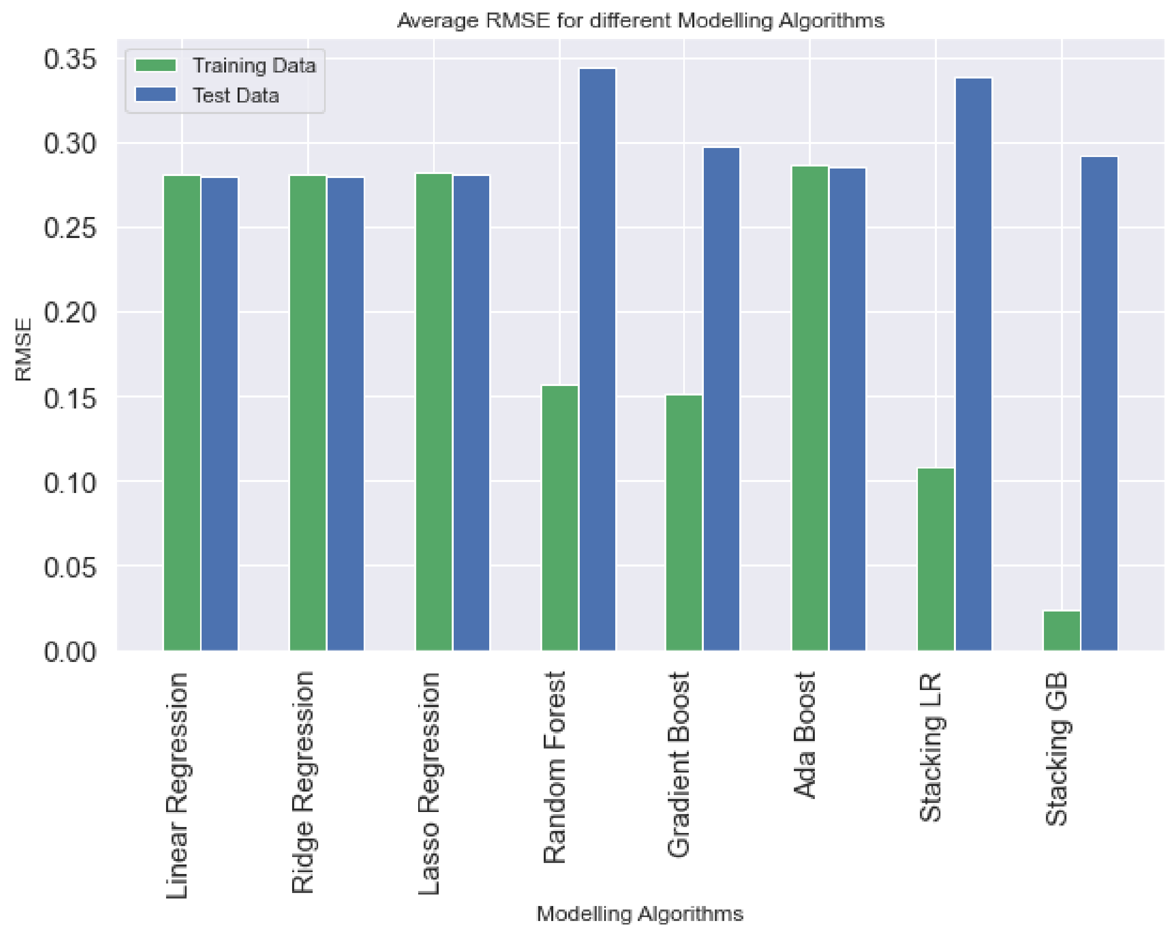


Figure No. 14 RMSE for test and train data

Random Forest and Gradient boost show the best performance on the test dataset. In order to decide one of these, we look at the time taken in fitting the model. In all the algorithms, the prediction time was very small in comparison to the fit time.

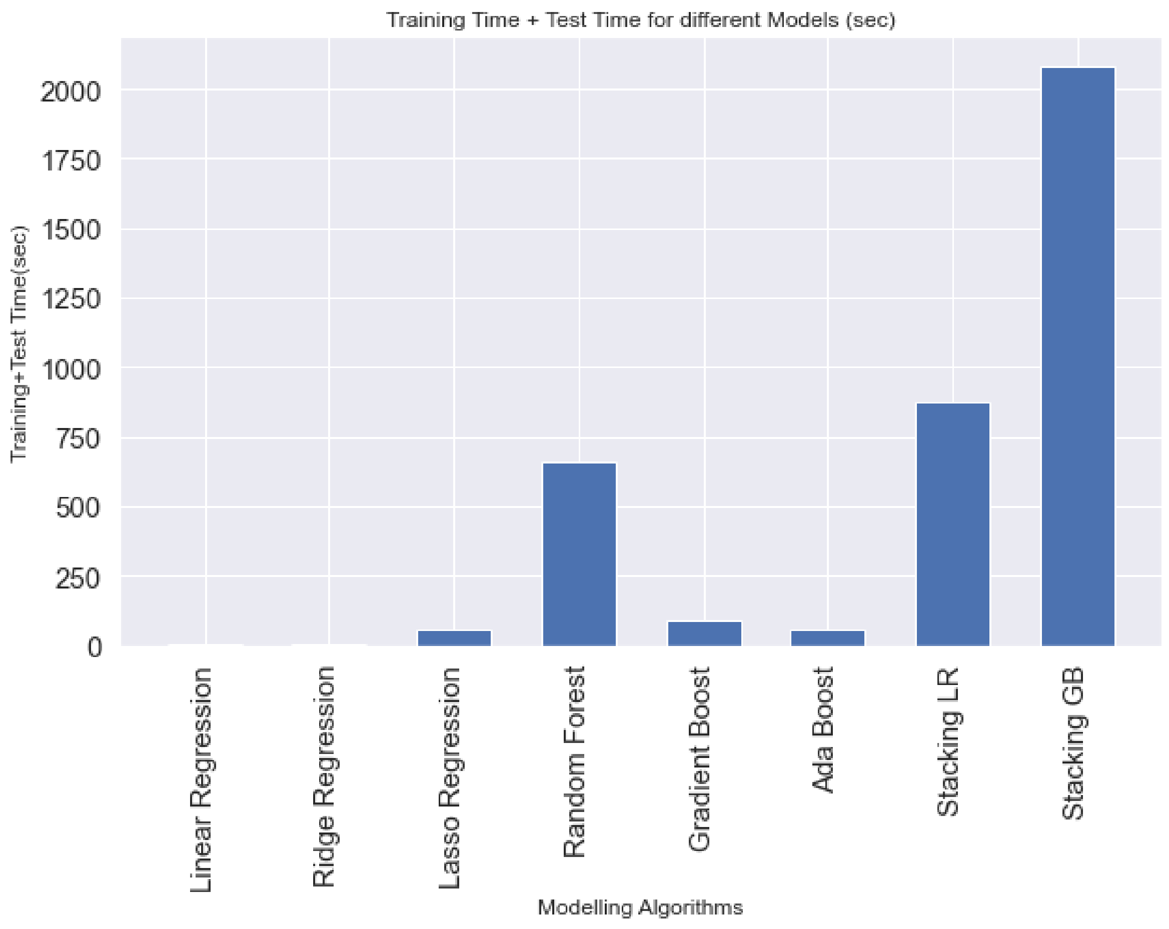


Figure No. 15 Train and Test time

As seen above, Gradient boost is taking much less time as compared to Random Forest. So we choose Gradient boost as our final algorithm. Gradient boost gives an RMSE of 0.29 on our test data. Let us take a look at the plot of actual target variable and error.

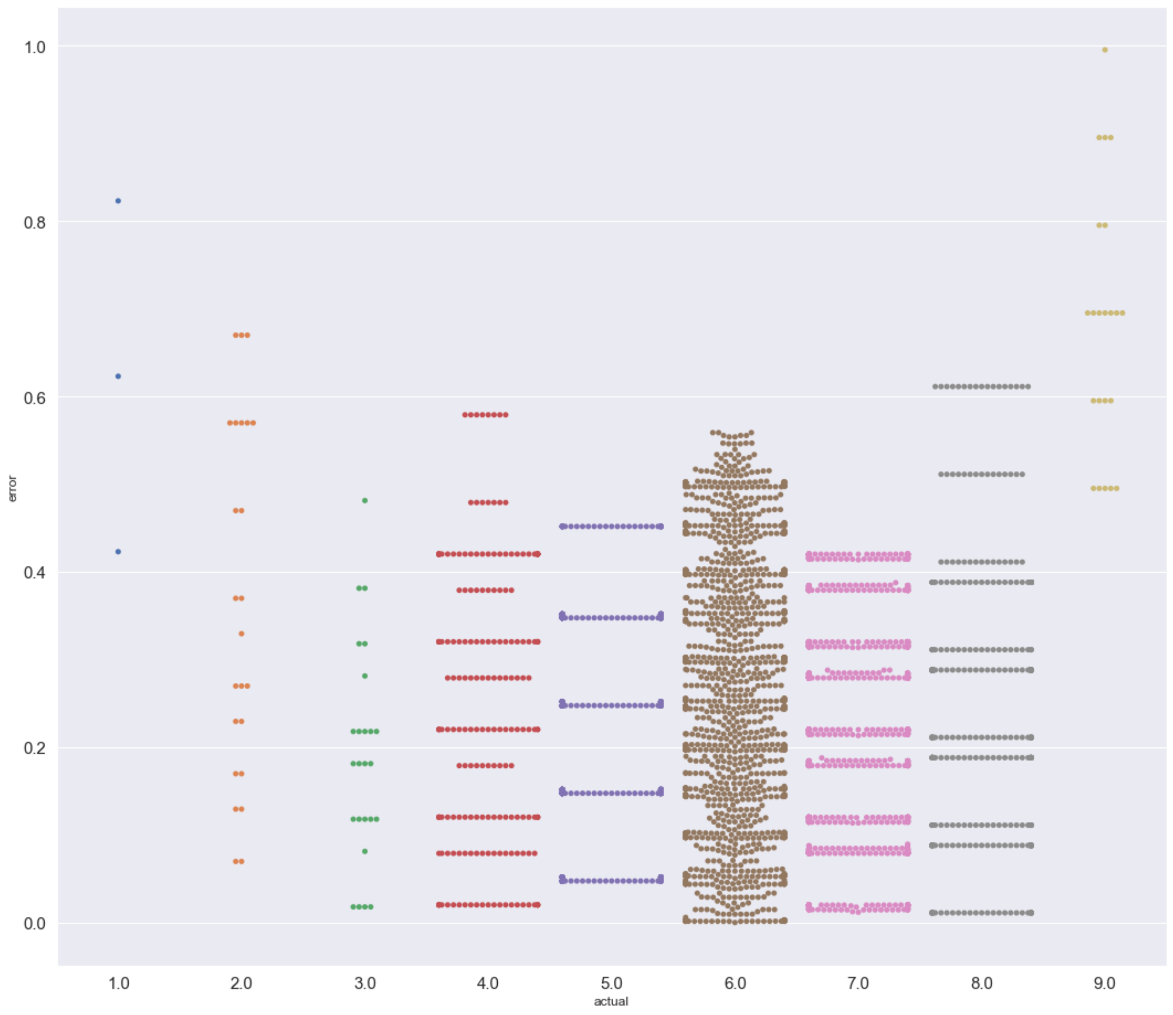


Figure No. 16 Error for ratings

We notice that error is more for ratings for which we don't have enough observations. For the rest of the cases, we get predictions close to the actual values.

7. Result

The objective of this project was to predict the IMDB rating of movies before they are released. The model we came up with gives us good results for movies rated 6, most of them within ± 1 error. For the extreme ratings though, we are getting larger errors which are mostly within ±2 error points.

8. Scope of further study

We have not considered the box office data in our project. Our model can be easily extended to predict box office collections. Also, we can use the box office collections as a feature to predict ratings.

While analysing the yearly trends, we saw a drastic change in all the trends after the year 1980. Though this might be a result of bad data, the data before 1980 can be further explored in a separate study to explore interesting insights.