Study and Analysis of Various Movies and TV-Series Data extracted from iMDB

Digital Assignment 2

Applied Statistical Method
Subject Code: MAT5007
Slot: E1

Under the guidance of: Prof Mythili G Y

By
Himadri Sankar Chatterjee (19MCA0148)
Masters in Computer Application

Abstract

Every year thousands of new movies and TV-Shows are being released. The entertainment industry is growing at a very fast pace. It plays a major role in impacting the economy of the whole world at large. With sites like iMDb (Internet Movie Database), providing valuable information, it not only helps people in deciding their next choices of shows, but also provides a wide platform to perform analysis on the trends that take over the industry from time-to-time. All the people working in this industry have invested a large amount in understanding and predicting the success of various movies and tv-series. These valuable insights, largely impact the upcoming movie releases and their performance. Various research has been conducted to achieve high accuracy in the given task. This paper works on acquiring necessary information on various movies and tv-shows from online sites and perform various basic visualization and analysis on the acquired data. We also try to fit a machine learning model to predict the iMDB Movie_Score from some given information about a movie.

Introduction

The entertainment industry is mostly governed by the impact of movies and tv-shows that form an evergreen component of our daily lives. The entertainment industry is a multi-billion-dollar industry that portrays the transforming face of the culture and events that affects our day-to-day lives. These movies/tv-series are classified based on their genre, that form a major factor for choice of people. These popularity of movies of all the genre forms a cycle of ups and downs that depend on the present conditions of the society. Some of the major genre in this field are comedy, action, thriller, animation, romance, sci-fi, mystery and others. The genre of choice of a mass gives a faint idea of the things that people long to see in the wide screens or on the tv, from the comfort of their home. To understand such details in the existing trends, people have invested a lot of time and effort in collecting and analyzing data collected from various resources.

Out of all the existing free online resource site, that provides a lot of information about the movies and the tv-series, like Rotten Tomatoes, Metacritic and iMDB, the Internet Movie Database (or, iMDB) remains the number one choice of all the people. The iMDB is an excellent source of finding all relevant information regarding a particular show. It consists of data ranging from language, year of release, cast, crew, duration, technical specs, budget, revenue, user ratings and reviews, critic ratings, etc. With such large datasets, its difficult to manually figure out trends and perform analysis on the data. iMDB already have an opensource dataset of its own, that is there in Kaggle to perform analysis on. However, we decided to extract various other information to work on. In our work, we have extracted data on various attributes of both movies and tv-shows that might prove useful in performing analysis and prediction. In the later part, we shall concentrate our focus in working with data only related to movies. Our collected dataset, contains information about 5027 movies and tv-series all together and 12 attributes for each of the show.

Literature Review

Various work has already been performed on datasets on Movies, one of which has been the topic of competition organized by iMDB themselves after the publicized their data for research.

Apart from visualizing the data based on their attributes, various machine learning models has been trained and tested to perfectly fit the corresponding task of prediction on the data. Originally, these works were done on the iMDB 5000 Movie Dataset, that included properties like director name, number of critic reviews, name of the actors and actresses and their corresponding Facebook likes, plot keywords and the language of release. Researchers have performed various analysis in establishing relation between the various attributes of this dataset and infer meaningful conclusions from their study.

In one of the papers by Dauenhauer et al [1], analysis was performed on the movie dataset based on their respective genre using various visualization tools like Many Eyes and Google Fusion Tables. Their work concluded that movie popularity and movie profitability are not necessarily linked to each other. The kind of movies to be produced next depends mostly on the profits of the previous movies. Moreover, they also concluded that movies addressing current events are more likely to attract a large audience.

In another paper by Vanitha et al [2], analysis was done on movie data collected from Kaggle. They performed research especially on Indian movies and concluded that the release of movies is affected by the season trend and also on various major media events happening all around the world.

Latif and Afzal [3], in their paper applied machine learning techniques on predicting the popularity of the movies from the data they have extracted from the iMDB site. Their Logistic Regression model and Simple Logistic model performed very well on the dataset, compared to the other traditional machine learning models that were applied for the task. The achieved around 84% accuracy.

Various efforts have been made to apply data mining techniques on the data extracted from iMDB website, but unfortunately, it has been proved to be very difficult. Extensive cleaning and integration of the collected data with sometime manual interventions to arrange them, proved to be very ineffective and time consuming.

Methodology

This section defines the general methods applied on the collected data to perform the task of analysis and prediction. Our primary source of information has naturally been iMDB. We decided on some basic properties about the movie that might affect the success, henceforth the rating of the movie or tv-show. We decided to stick with the first 5000 suggestion from the iMDB site for the list of shows to be considered.

The dataset we collected from referring the online site have to cleaned. Not all the attributes carry meaningful values for both Movies and TV-Shows. So, while the common attributes that contained sensible values have been used to study the complete data together, we had to separate the records related to movies, to perform specified task on them.

Most of the data collected have turned out to be either NULL or absurd based on incomplete or false information as has been posted online. Thus, cleaning the data, initially has been a huge task. We revisited each movie/tv-show site containing respective information and had to recheck and update the data to establish meaning. All the data collected has been stored in a comma separated file, that is easily viewed through Excel and is easier to work on with.

We have used Python and some of the basic data analysis and manipulation libraries like Pandas, Numpy, Matplotlib for carrying out the task. The work has been catalogued in an iPython notebook for easy referencing and understandability of the analysis.

Movie Dataset Description

The movie dataset, that we worked on has been specifically curated to our needs. The data have been referred from the iMDB site as of September, 2019. With continuous updates, its hard to provide an exact analysis of the data, but the relative difference in our inferences are quite low. The dataset consisted of the following attributes:

1. **Name**: This includes the name of the movie or the tv-series being referred.
2. **Year**: This denotes the release year of the movies. Since, some of the series may have been completed while some are ongoing, we have decided to avoid allowing such discrepancies and keep the value NULL for such shows.
3. **Type**: This indicates whether the show under consideration is a 'Movie' or a 'TV Series'
4. **Genre**: This is a list containing the genre addressed by the specific show.
5. **MovieScore**: This denotes the value of the iMDB rating for the corresponding show.
6. **Metascore**: This include the value of the rating provided by another famous online site for movie reviews, the Metacritic. This site does not rate the tv series, so their corresponding values have been assigned 0.
7. **Duration**: This denotes the total runtime of each movie and average runtime of each episode of the tv series.
8. **MovieColor**: This denotes the color of the movie/tv-series when it was originally released.
9. **MovieLanguage**: This denotes the language of the movie/tv-series when it was originally released.
10. **MovieWorldwideGross**: This denotes the total earning of the movie across the world. For tv-series, the corresponding field has been assigned the value 0.
11. **MovieURL**: This includes the link to the page on iMDB containing all the information related to the corresponding show.
12. **Total_Votes**: This include the number of people who rated the show on iMDB.

Here is what the first 4 rows of the datset looks like:

	Name	Year	Type	Genre	MovieScore	Metascore	Duration	MovieColor	MovieLanguage	MovieWorldwideGross	MovieURL	Total_Votes
0	The Lion King	2019	Movie	['Animation', 'Adventure', 'Drama', 'Family', ...	7.1	55	118	Color	English	1564549294	https://www.imdb.com/title/tt6105098/	95301
1	Once Upon a Time... in Hollywood	2019	Movie	['Comedy', 'Drama']	8.0	83	161	Color	English	283722549	https://www.imdb.com/title/tt7131622/	164018
2	Joker	2019	Movie	['Crime', 'Drama', 'Thriller']	9.6	75	122	Color	English	0	https://www.imdb.com/title/tt7286456/	7756
3	Stranger Things	-	TV Series	['Drama', 'Fantasy', 'Horror', 'Mystery', 'Sci...	8.8	0	51	Color	English	0	https://www.imdb.com/title/tt4574334/	654652

Each of the attribute have been checked and updated to remove NULL values and other discrepancies.
Not all of the information will be helpful in analyzing the data together. The required combination of data is extracted from the dataset as and when required.
The shape of the dataset is (5027 X 12), indicating 502 shows (including both movies and tv-series) and 12 attributes corresponding to each show.

Movie Dataset Visualization and Analysis

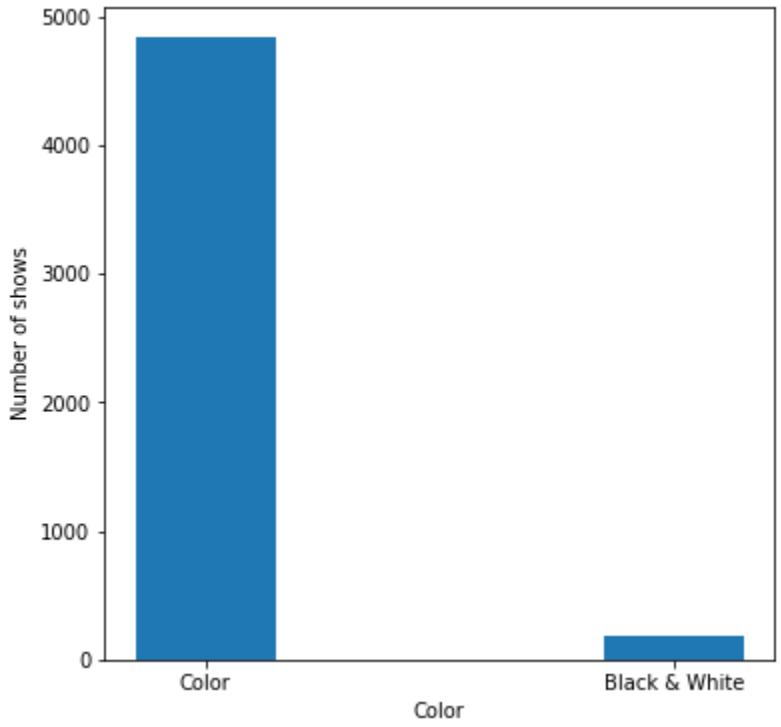
With the complete data at hand, we performed some basic visualizations to understand the data:

Aim: The distribution of 'Color' vs 'Black & White' movies in our dataset

Syntax:

```
colorCount = data.MovieColor.value_counts()
plt.figure(1, figsize=(6, 6))
plt.bar([0, 1], colorCount, width=0.3)
plt.xticks([0,1], ["Color", "Black & White"])
plt.xlabel('Color')
plt.ylabel('Number of shows')
plt.show()
```

Result:



Since we have the attribute of MovieColor, we check the distribution of the movies across the two categories. From the visalization, it its clear, that the majority of the top 500 shows are dominated by ‘Color’ movies, while an almost negligible number of ‘Black & White’ movies still qualify to remain in the list.

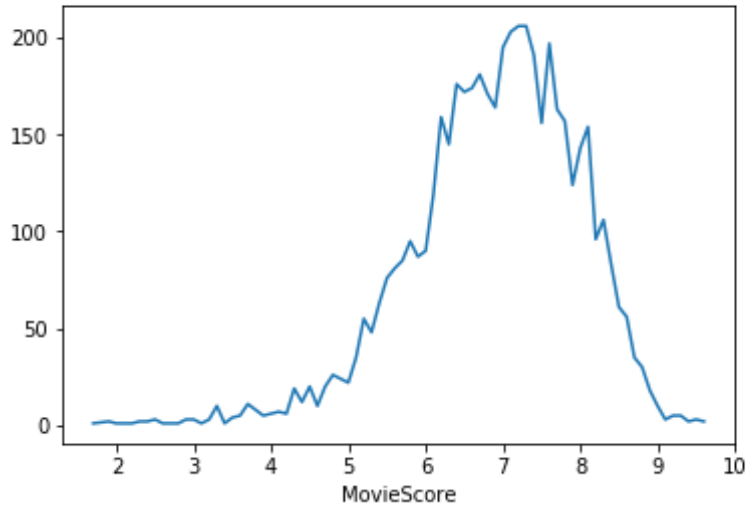
Aim: To visualize the distribution of iMDB Scores across all the shows.

Syntax:

```
dataScoreGroup = data.groupby(['MovieScore'])['Name'].count()
dataScoreGroup.plot()
```

Result:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d15391ac50>
```



Thus, the common distribution of score lie in the range of 6 to 8. None of the move has ever achieved a perfect sore of 10. The highest being 9.6 out of 10.

Aim: To extract information on the top 30 shows based on their iMDB rating

Syntax:

```
topRated10 = data.sort_values('MovieScore', ascending = False)[['Name', 'Type',
'MovieScore']][:10]
```

Result:

2	Joker	Movie	9.6
1077	Pew News	TV Series	9.6
3238	Critical Role	TV Series	9.5
2001	Family of Thakurganj	Movie	9.5
40	Breaking Bad	TV Series	9.5
4510	Wu-Tang: An American Saga	TV Series	9.4
16	Game of Thrones	TV Series	9.4
146	The Wire	TV Series	9.3
94	Rick and Morty	TV Series	9.3
642	Ardaas Karaan	Movie	9.3

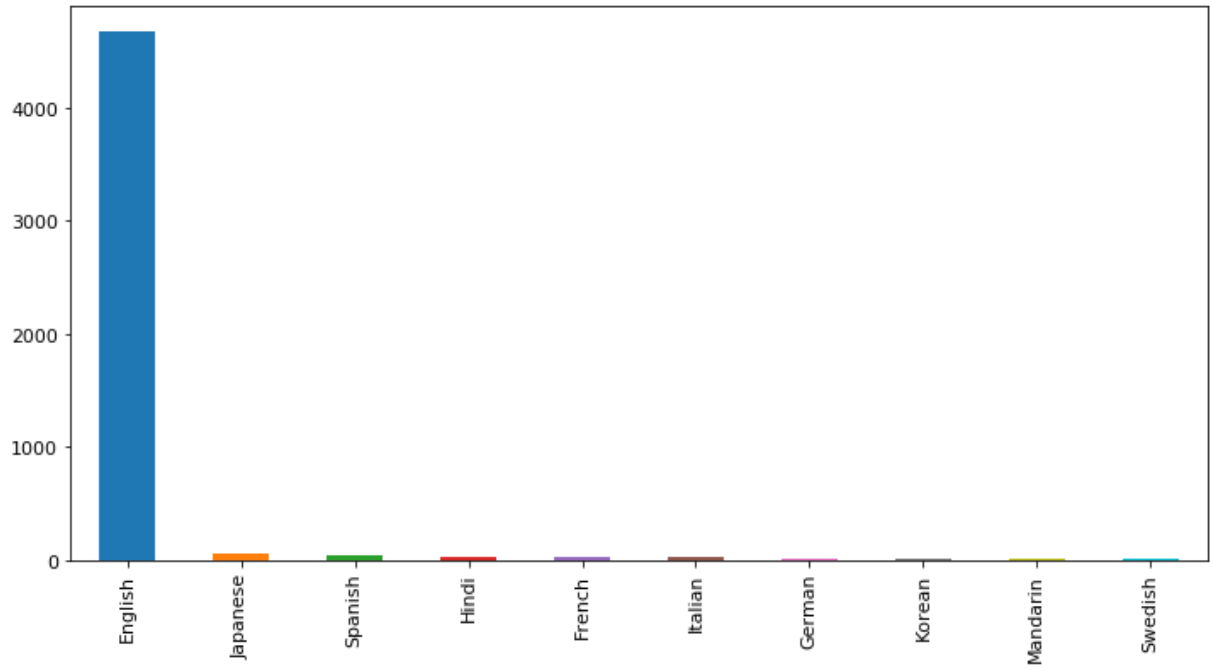
The top 10 movies and tv-series are displayed, with the highest rating of “9.6” out of a maximum 10, being achieved by one recently released movie Joker and a popular tv-series Pew News.

Aim: To plot the no of movies released based on the top 10 languages preferred.

Syntax:

```
languageList = data.MovieLanguage.value_counts()
plt.figure(1, figsize=(12, 6))
languageList[:10].plot(kind = "bar")
plt.show()
```

Result:



Majority of the movies and tv-series in our dataset are originally released in English. Following it in the list are Japanese, Spanish, Hindi and French, making it to the top 5 languages.

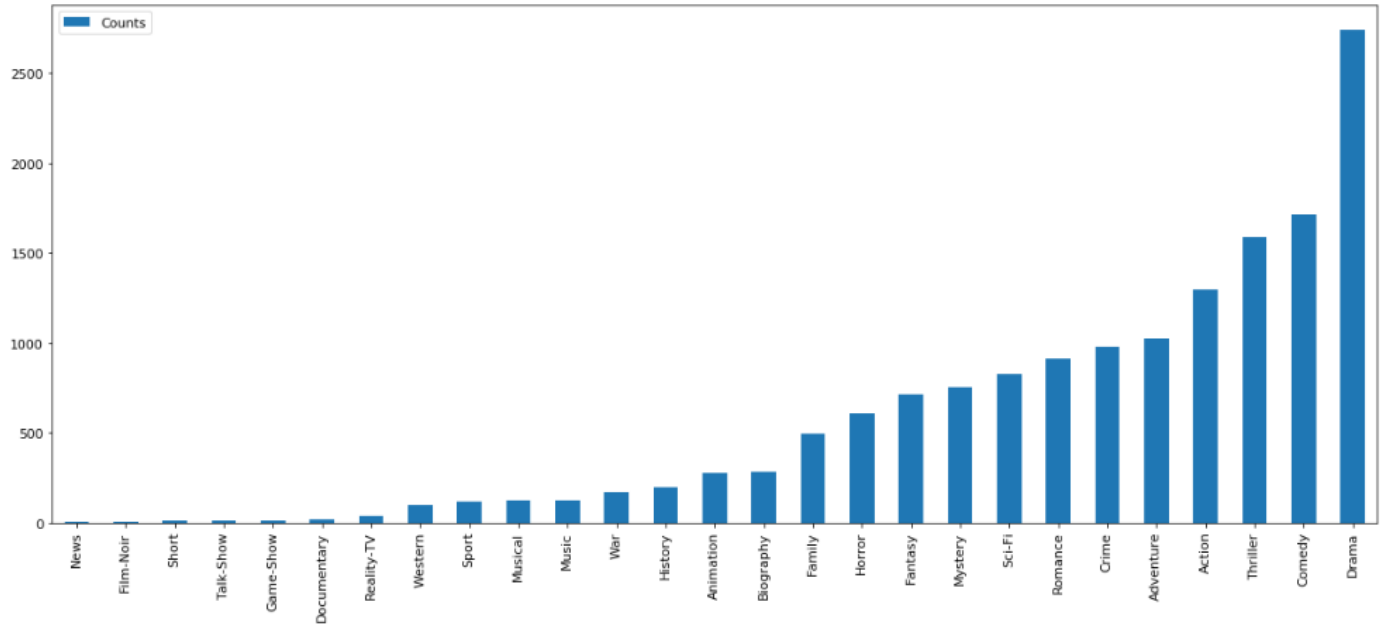
Aim: To plot the distribution of movies/tv-series with respect to their corresponding genre.

Syntax:

```
genreData = data['Genre'].str.split(',')
genre = []
genreDict = {}
for glist in genreData:
   for item in glist:
      item = re.findall(r'\'(.*?)\'', item)[0]
      genre.append(item)
      if item not in genreDict:
         genreDict[item] = 1
      else:
         genreDict[item] = genreDict[item]+1

Gen = pd.DataFrame.from_dict(genreDict, orient = 'index')
Gen.columns = ['Counts']
Gen = Gen.sort_values('Counts', ascending = True)
Gen.plot(kind = 'bar', figsize=(20, 8))
plt.show()
```

Result:



The number of shows belonging to the 'Drama' genre, is very high compared to its neighbors, Comedy, Thriller and Action.

Aim: Boxplot Diagram to study relation between the iMDB Score of shows and their Genre.

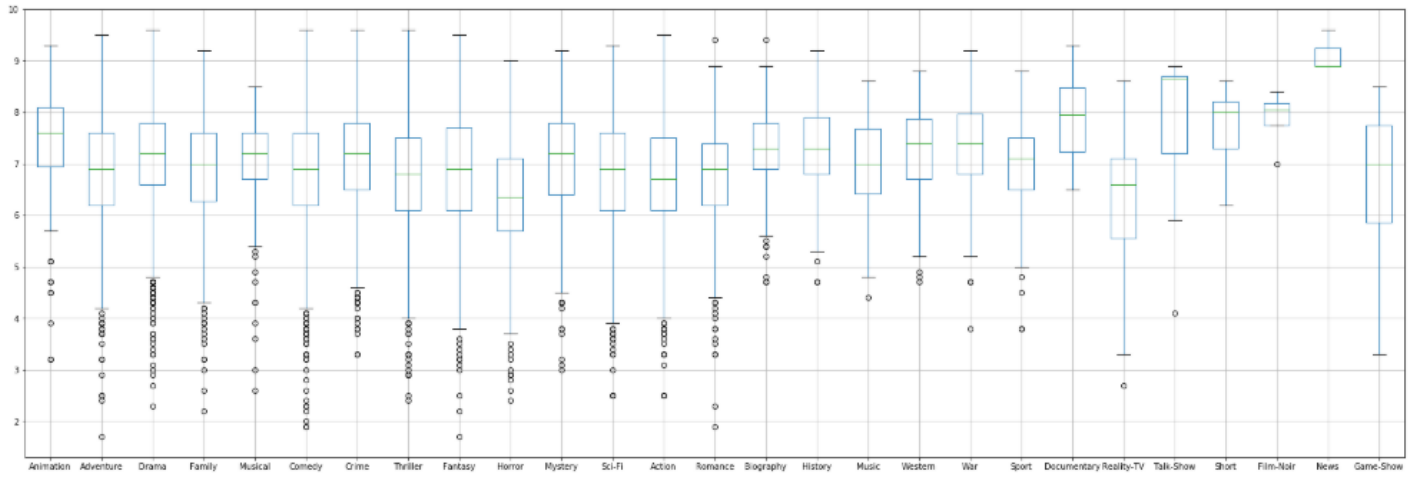
Syntax:

```
genreScoreDict = {}
for item in genre:
   if item not in genreScoreDict:
      genreScoreDict[item] = []

for record in data.iterrows():
   #print(record[1][3])
   glist = record[1][3].split(',')
   glist = [re.findall(r'\'(.*?)\'', x)[0] for x in glist]
   for item in glist:
      genreScoreDict[item].append(record[1][4])

GenScore = pd.DataFrame.from_dict(genreScoreDict, orient = 'index')
plt.figure(1, figsize=(30, 10))
GenScore.transpose().boxplot()
plt.show()
```

Result:



The boxplot diagram gives an idea about the distribution of the iMDB Score with respect to each Genre based on a five number summary (minimum, first quartile(Q1), median, third quartile(Q3), maximum).

A short description of the data analyzed, provides the more information. This includes details for the first 10 genre from the genre list being considered.

	Animation	Adventure	Drama	Family	Musical	Comedy	Crime	Thriller	Fantasy	
count	279.000	1026.000	2743.000	496.000	123.000	1714.000	976.000	1587.000	712.000	608.000
mean	7.46523	6.821248	7.131863	6.82258	6.96179	6.839323	7.07469	6.751229	6.83862	6.34901
std	0.97004	1.078977	0.960920	1.08615	1.05009	1.066425	1.00695	1.063544	1.15954	1.08567
min	3.20000	1.700000	2.300000	2.2000	2.6000	1.900000	3.3000	2.400000	1.7000	2.4000
25%	6.95000	6.200000	6.600000	6.2750	6.7000	6.200000	6.5000	6.100000	6.1000	5.7000
50%	7.60000	6.900000	7.200000	7.0000	7.2000	6.900000	7.2000	6.800000	6.9000	6.3500
75%	8.10000	7.600000	7.800000	7.6000	7.6000	7.600000	7.8000	7.500000	7.7000	7.1000

	Animation	Adventure	Drama	Family	Musical	Comedy	Crime	Thriller	Fantasy	
max	9.30000	9.500000	9.600000	9.20000	8.50000	9.600000	9.60000	9.600000	9.50000	9.00000

We now focus on the data related to each movie in the dataset. We extracted the necessary information from the dataset and cleaned the data to remove discrepancies.

The first 4 rows of the newly extracted movie dataset is shown below:

	Name	Year	Type	Genre	MovieScore	Metascore	Duration	MovieColor	MovieLanguage	MovieWorldwideGross	MovieURL	Total_Votes
0	The Lion King	2019	Movie	['Animation', 'Adventure', 'Drama', 'Family', ...	7.1	55	118	Color	English	1564549294	https://www.imdb.com/title/tt6105098/	95301
1	Once Upon a Time... in Hollywood	2019	Movie	['Comedy', 'Drama']	8.0	83	161	Color	English	283722549	https://www.imdb.com/title/tt7131622/	164018
9	Spider-Man: Far from Home	2019	Movie	['Action', 'Adventure', 'Sci-Fi']	7.9	69	129	Color	English	1122182596	https://www.imdb.com/title/tt6320628/	159116
11	Avengers: Endgame	2019	Movie	['Action', 'Adventure', 'Sci-Fi']	8.6	78	181	Color	English	2796255402	https://www.imdb.com/title/tt4154796/	544683

Aim: To understand the various properties of each quantified attribute of the movie dataset.

Syntax:

```
print(movieData.describe())
```

Result:

Feature	MovieScore	Metascore	Duration	MovieWorldwideGross	Total_Votes
count	3149.0000	3149.0000	3149.0000	3.149000e+03	3.149000e+03
mean	6.7450	59.0035	112.5379	1.495425e+08	1.670515e+05
std	0.9445	17.8371	20.13	2.273137e+08	2.033072e+05
min	1.9000	5.0000	64.0000	5.760000e+02	8.100000e+01
25%	6.2000	46 .0000	98.0000	1.898874e+07	4.690900e+04
50%	6.8000	59. 0000	109.0000	6.791866e+07	1.034160e+05
75%	7.4000	72.0 000	123.0000	1.839361e+08	2.086010e+05
max	9.3000	100.0 00	321.0000	2.796255e+09	2.130344e+06

The describe() methods provides all the necessary statistic regarding each quantified attribute of the movie dataset. The relation between these attributes can be understood through knowing the correlation between the data.

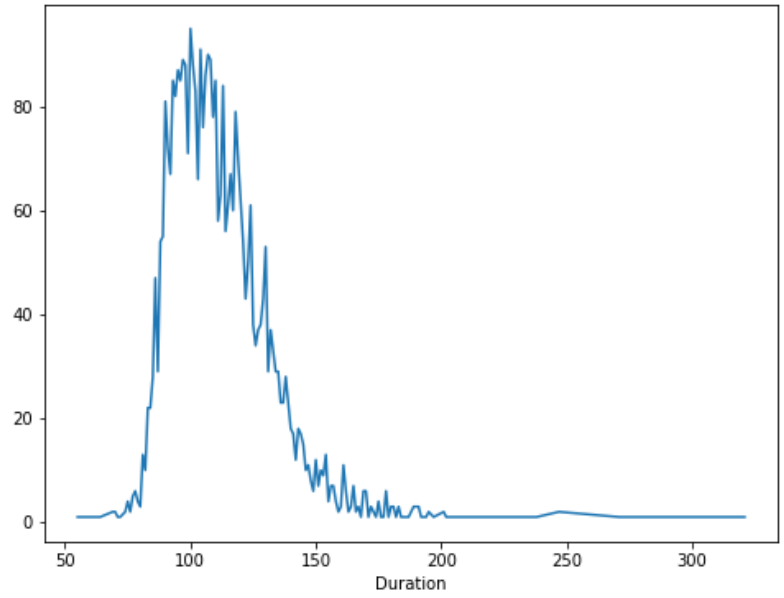
Aim: Plotting the distribution of movies with respect to their Duration, or total runtime.

Syntax:

```
movieDurationGroup =
movieData.groupby(['Duration'])['Name'].count().sort_index(axis=0, ascending = True)
print(movieDurationGroup)
movieDurationGroup.plot(figsize = (8, 6))
```

Result:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d15610ca90>
```



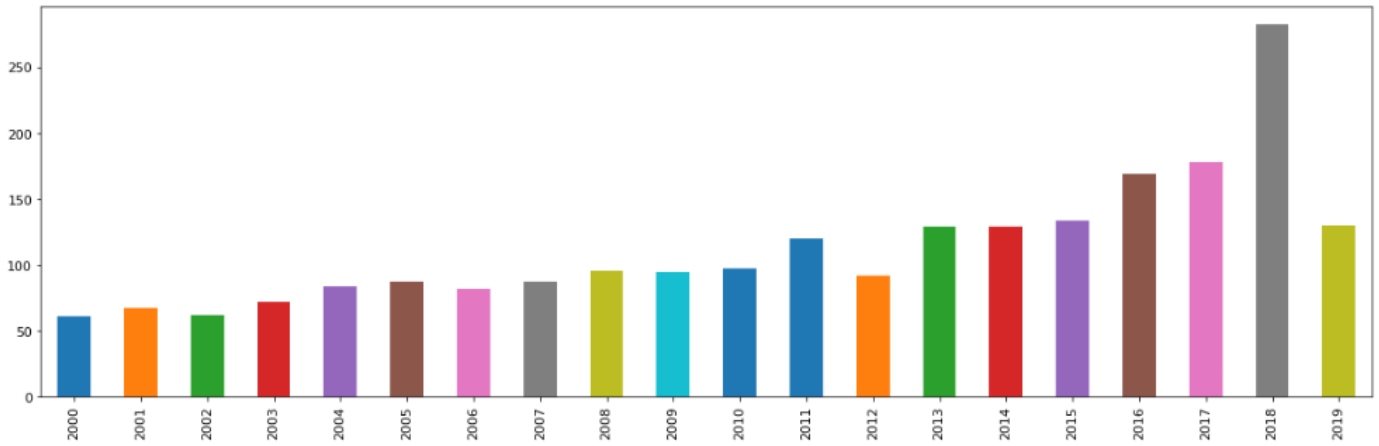
The general duration of a movie ranges from 90 to 120 minutes. The movie with minimum duration was of 64 minutes and the movie with maximum duration ran for 321 minutes.

Aim: Plotting the number of movies release per year for the last 20 years.

Syntax:

```
year = movieData.Year.value_counts().sort_index(axis = 0, ascending = True)
plt.figure(1, figsize=(20, 6))
year[:-20].plot(kind = 'bar')
plt.show()
```

Result:

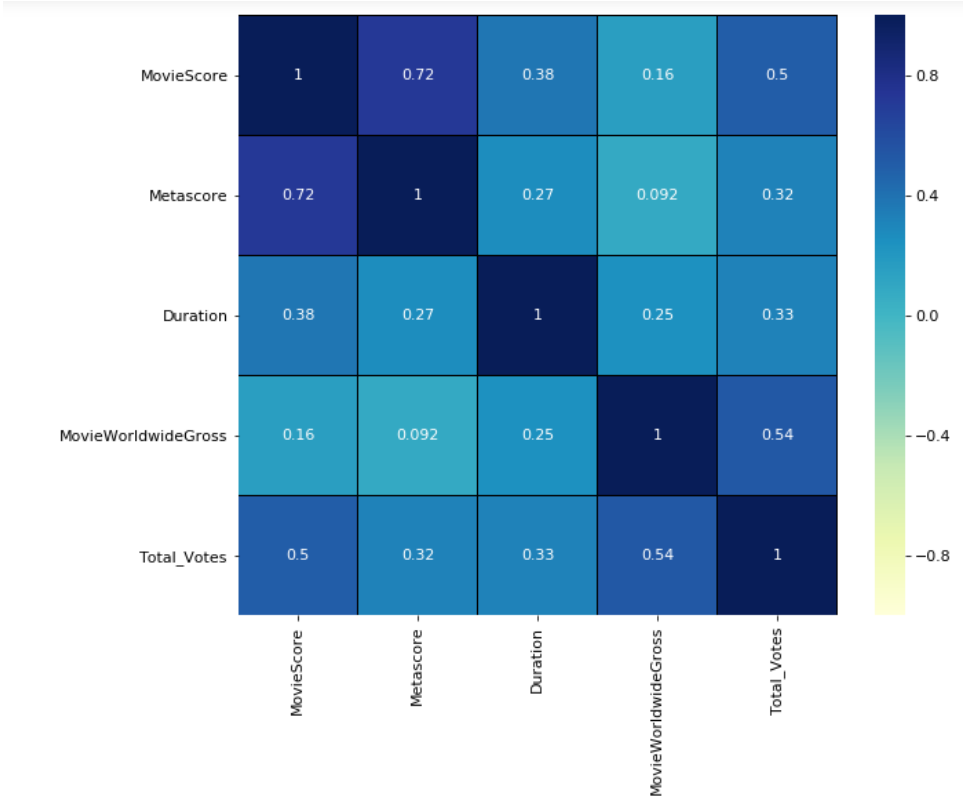


Aim: Plotting the Pearson's Correlation Coefficient between the attributes of the movie dataset.

Syntax:

```
plt.figure(1, figsize=(12,10))
sns.heatmap(movieData.corr(), linewidths = 0.5, vmax = 1, vmin = -1, annot = True, cmap='YlGnBu', square = True, linecolor = 'Black')
```

Result:



This diagram shows the Pearson's Correlation Coefficient between the attributes of thee data. From the above heatmap, we can figure out that there exists.

i. a strong correlation between the iMDB Score for the movie and the Metacritic Score.
ii. a moderate correlation between the iMDB Score and the Total Votes
iii. a very weak correlation between the Metacritic Score and the Worldwide Gross of the movie.
iv. A little fair correlation between the Duration of the movie with both the iMDB Score and the Metacritic Score.

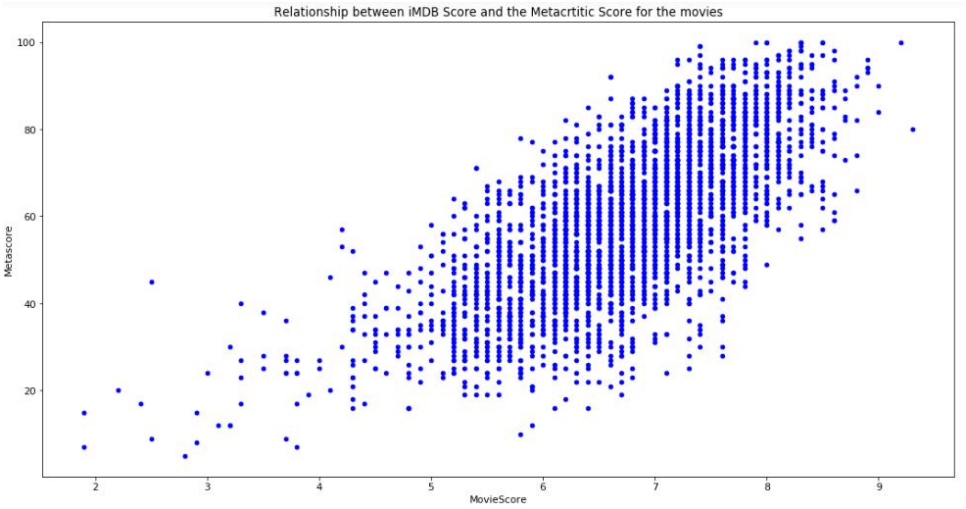
Few of the Scatter Plot diagrams helps us understand the relation between the attributes.

Aim: Scatter Plot diagram between iMDB Score and Metacritic Score.

Syntax:

```
plt.rcParams['font.size'] = 11
plt.rcParams['figure.figsize'] = (15.0, 8.0)
movieData.plot(kind = 'scatter', x='MovieScore', y='Metascore', color='Blue')
plt.title("Relationship between iMDB Score and the Metacrtitic Score for the movies")
plt.tight_layout()
```

Result:



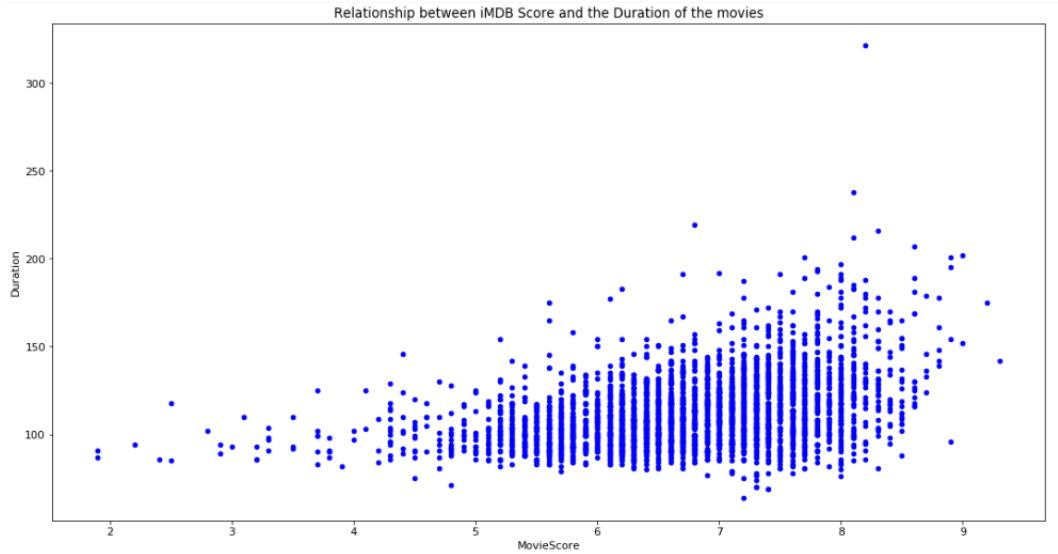
Aim: Scatter Plot between iMDB Score and Duration of the Movies.

Syntax:

```
plt.rcParams['font.size'] = 11
plt.rcParams['figure.figsize'] = (15.0, 8.0)
```

```
movieData.plot(kind = 'scatter', x='MovieScore', y='Duration', color='Blue')
plt.title("Relationship between iMDB Score and the Duration of the movies")
plt.tight_layout()
```

Result:

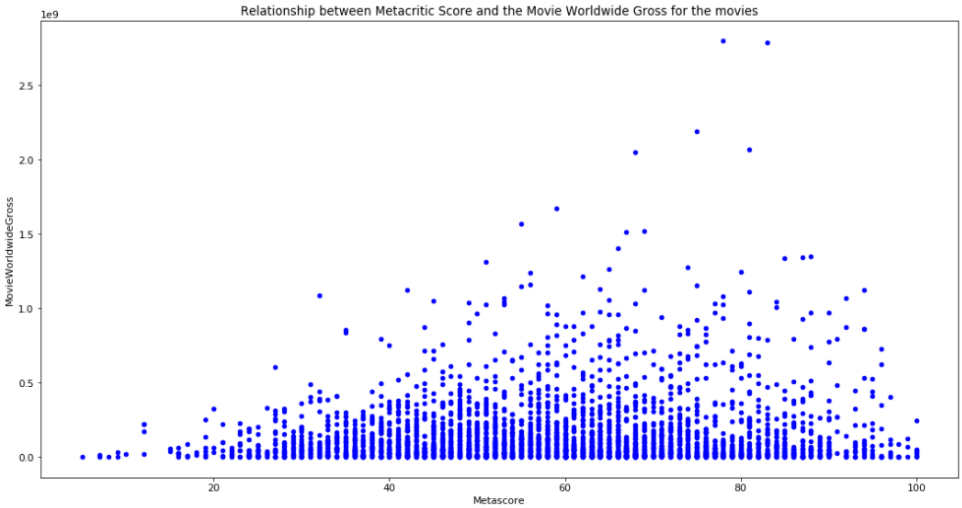


Aim: Scatter Plot between Metacritic Score and the Worldwide Gross of the movie.

Syntax:

```
plt.rcParams['font.size'] = 1
plt.rcParams['figure.figsize'] = (15.0, 6.0)
movieData.plot(kind = 'scatter', x='Metascore', y='MovieWorldwideGross', color='Blue')
plt.title("Relationship between the Metacritic Score and the Duration of the movies")
plt.tight_layout()
```

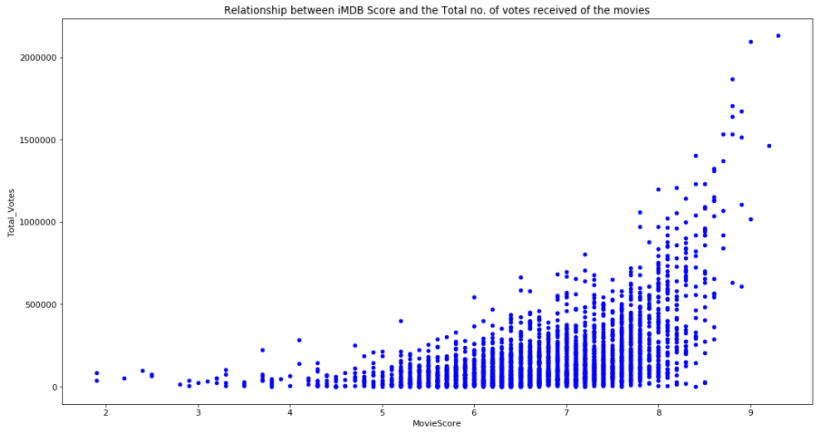
Result:



Aim: Scatter Plot between iMDB Score and the Total number of Votes received.
Syntax:

```
plt.rcParams['font.size'] = 11
plt.rcParams['figure.figsize'] = (15.0, 8.0)
movieData.plot(kind = 'scatter', x='MovieScore', y='Total_Votes', color='Blue')
plt.title("Relationship between iMDB Score and the Total no. of votes received of the movies")
plt.tight_layout()
```

Result:



From the above Scatter Plot, we can obtain a fair amount of idea about the relation between the attributes, that the dataset hold.

We try to train a Machine Learning model to predict the iMDB Score of a movie given the Metacritic Score, Movie Worldwide Gross, Duration of the movie and the Total number of Votes received by the movie.

Aim: Implement a simple Linear Regression model to predict iMDB Score of a movie, when the other attributes are given.

Syntax:

```
#Libraries required for Machine Learning on selected data related to only movies
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LinearRegression, Perceptron, SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing, utils
```

```
#Applying Logistic Regression to predict iMDB Score from the given Metacritic Score,
Duration of the Movie, Total Worldwide Gross
#and the Total Votes received by the movie.

#Preparing the data for training and testing
features = ['Metascore', 'Duration', 'MovieWorldwideGross', 'Total_Votes']
target = ['MovieScore']

train, test = train_test_split(movieData, test_size = 0.3)

X_train = train[features]
Y_train = train[target]
X_test = test[features]
Y_test = test[target]

print(X_train.shape)
print(Y_train.shape)
print(X_test.shape)
print(Y_test.shape)

#Prediction using Linear Regression
logreg = LinearRegression()
logreg.fit(X_train, Y_train)
Y_pred = logreg.predict(X_test)
acc_log = round(logreg.score(X_test, Y_test) * 100, 2)
acc_log
```

Result:

Shape of the Train and Test Set:

```
(2204, 4)
(2204, 1)
(945, 4)
(945, 1)
```

Mean Accuracy of the Linear Regression Model over the Test Data:

60.7

Conclusion and Future Work:

Out of various other traditional machine learning models, that we have tried to fit to the data, the simple Linear Regression Model performed comparatively well, achieving an accuracy of only 60.4% over the test dataset.

The data we collected is still not scaled or normalized to be trained on a general neural network architecture. The data on the Movie Worldwide Gross plays negligible effect on determining the iMDB Score for the particular movie. Although, the Metacritic Score of a movie is highly correlated to the iMDB Score of that movie, bearing a strong positive correlation.

Compared to other datasets, this dataset contained very less information and thus might have skipped other influential factors that highly affect the iMDB Score. In the future work, we wish to collect more valuable information on all the movies, normalize the data for effective training and possible deploy a deep learning model that can predict the scores with high accuracy.

References:

1. Janette Dauenhauer, Joneta Hockett, Joanne Mammarelli, and Michael Yarem, Information Analysis of Movie Genre.
   Source: http://cluster.ischool.drexel.edu/~cchen/courses/INFO633/13-14/mammarelli.pdf

2. V. Vanitha, V. P. Sumathi, V. Soundariya, An Exploratory Data Analysis of Movie Review Dataset.
   Source: https://www.ijrte.org/wp-content/uploads/papers/v7i4s/E2008017519.pdf

3. Muhammad Hassan Latif and Hammad Afzal, Prediction of Movies popularity Using Machine Learning Techniques.
   Source: http://paper.ijcsns.org/07_book/201608/20160820.pdf

4. Saraee, MH, White, S and Eccleston, J, A data mining approach to analysis and prediction of movie ratings. University of Salford.
   Sourcce: http://usir.salford.ac.uk/18838/1/Wessex_movie.pdf:public