Individual Final Report
Hemanth Koganti

Introduction:

Film industry is one of the top grossing industries in the world. Millions of dollars are invested in the making of each movie expecting high margin of profit. For 2019 alone, in United States 1.2 Billion cinema tickets were sold generating a revenue of approximately $11 Billion.

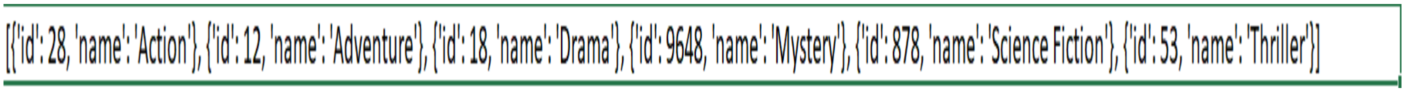
So the primary goal by undertaking this project is to investigate the influential factors affecting the success or failure of a movie which could be game changing not only for the producers but also the audience. With all this in our mind we set on a quest to answer this question: Whether is it possible to use machine learning algorithm to predict if movie will be a success (generate at least as much revenue as its budget) or a flop?

The portion of work I did was the data pre-processing and Exploratory data analysis. Exploratory data analysis was required to get valuable insights from the data. Also I had to pre-process the data as our data had some of the columns in JSON format and some of the columns had lots of missing values which needed to be dealt with before fitting it to models.

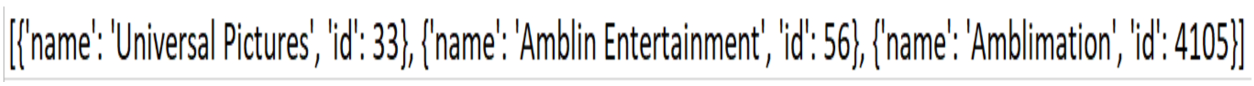
Brief explanation of my work:

Data pre-processing:

The pre-processing work I did was to **handle JSON data** in genres and production_companies columns. As each column has multiple genres and production companies respectively, we had to take the main genre and production company which are the first ones in our columns. Below pictures give us the overview of the data in genre and production_companies columns.



```
[{'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}, {'id': 18, 'name': 'Drama'}, {'id': 9648, 'name': 'Mystery'}, {'id': 878, 'name': 'Science Fiction'}, {'id': 53, 'name': 'Thriller'}]
```



```
[{'name': 'Universal Pictures', 'id': 33}, {'name': 'Amblin Entertainment', 'id': 56}, {'name': 'Amblimation', 'id': 4105}]
```

Below are the codes for the conversion of json in the mentioned columns. I have included what each line of code does in the comments :

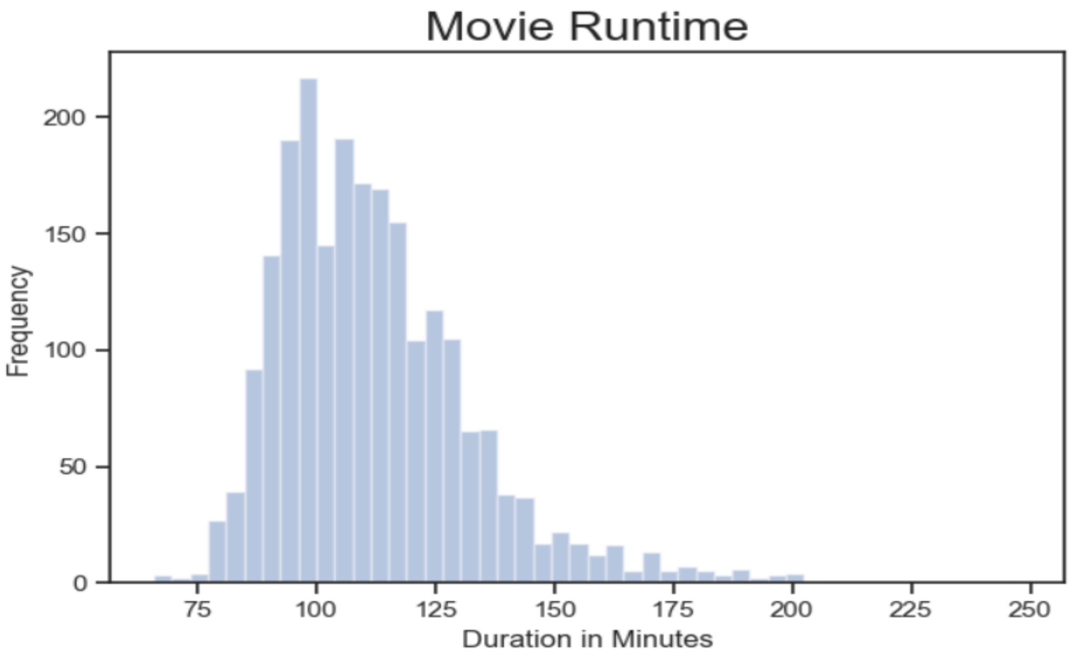
```
# converting (genre) json column to normal string column
# Replacing null values with '{}'
df_cleaned['genres'] = df_cleaned['genres'].replace(np.nan,'{}',regex = True)
# Converting Strings to Dictionaries as it have multiple genres in json format
df_cleaned['genres'] = pd.DataFrame(df_cleaned['genres'].apply(eval))
# dividing all genres in a cell into separate cols/series, concatenating it to main df & then dropping the original "genres" column from df
df_cleaned = pd.concat([df_cleaned.drop(['genres'], axis=1), df_cleaned['genres'].apply(pd.Series)], axis=1)
# Removing all columns except the major genre type for each movie
df_cleaned.drop(df_cleaned.iloc[:, 15:], inplace = True, axis = 1)
# creating separate series for "id" & "name" and concatenating it to main df
df_cleaned = pd.concat([df_cleaned.drop([0], axis=1), df_cleaned[0].apply(pd.Series)], axis=1)
df_cleaned.drop(df_cleaned.iloc[:, 14:16], inplace = True, axis = 1)     # dropping extraneous cols
df_cleaned.rename(columns = {'name' : 'Genre'}, inplace = True)   # renaming col
df_cleaned = df_cleaned[~df_cleaned['Genre'].isnull()] # removing null containing rows
```

```
# converting (production_companies) json column to normal string column
# Replacing null values with '{}'
df_cleaned['production_companies'] = df_cleaned['production_companies'].replace(np.nan,'{}',regex = True)
# Converting Strings to Dictionaries as it have multiple production companies in json format
df_cleaned['production_companies'] = pd.DataFrame(df_cleaned['production_companies'].apply(eval))
# Dividing all production companies into separate cols, concatenating these to the main df and dropping the original 'production companies' col
df_cleaned = pd.concat([df_cleaned.drop(['production_companies'], axis=1), df_cleaned['production_companies'].apply(pd.Series)], axis=1)
# Removing all production companies cols except major production company for each movie.
df_cleaned.drop(df_cleaned.iloc[:, 14:], inplace = True, axis = 1)
# creating separate series for "name" & "id" and concatenating it to main df
df_cleaned = pd.concat([df_cleaned.drop([0], axis=1), df_cleaned[0].apply(pd.Series)], axis=1)
# dropping unnecessary cols
df_cleaned.drop(df_cleaned.iloc[:, 13:15], inplace = True, axis = 1)
# renaming newly created col
df_cleaned.rename(columns = {'name' : 'Production_Company'}, inplace = True)
df_cleaned = df_cleaned[~df_cleaned['Production_Company'].isnull()]
len(df_cleaned.Production_Company.unique())
```

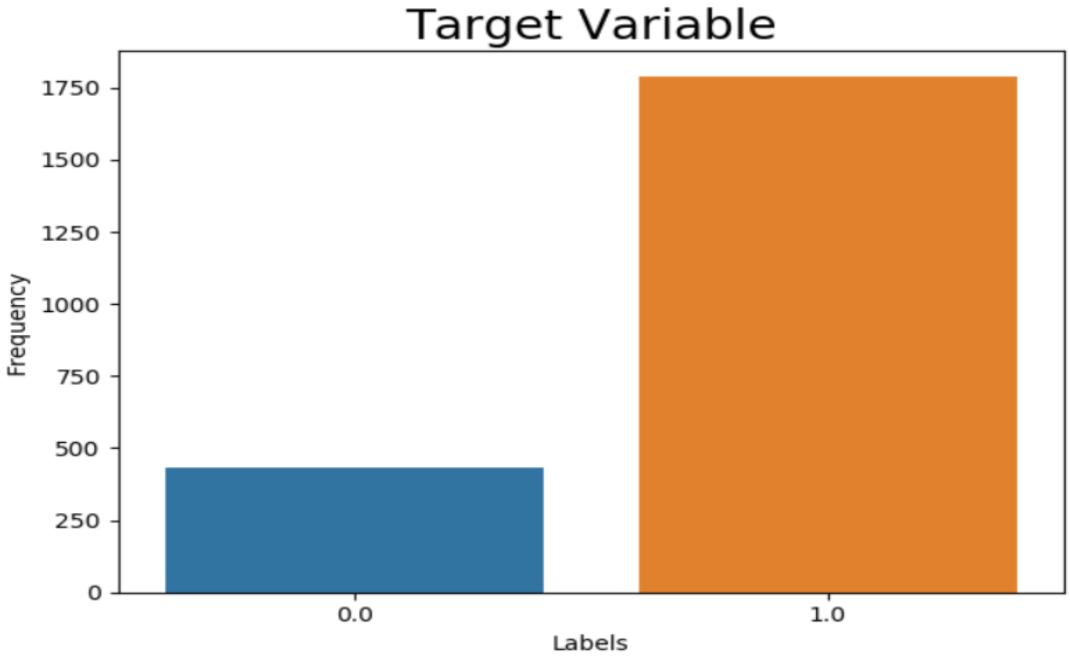
Exploratory data analysis:

Created visualizations to gain insights of the data which includes the below:

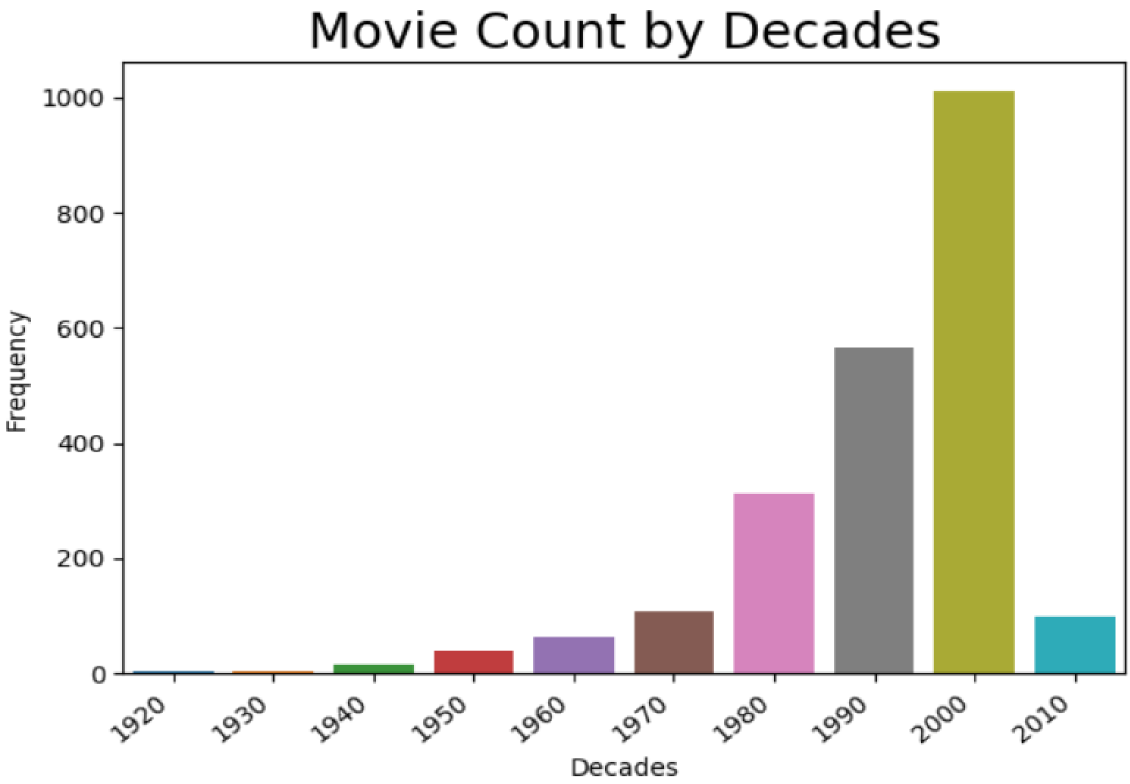
1. Histogram of runtime of movies



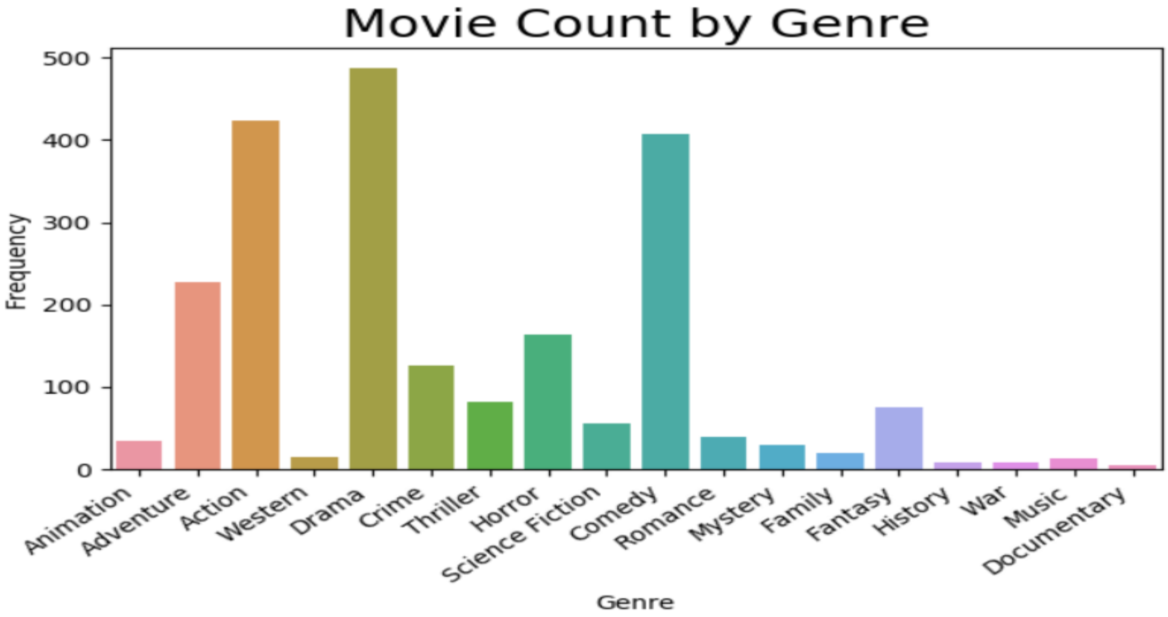
2. Bar graph depicting the target labels (successful movies and flop movies)



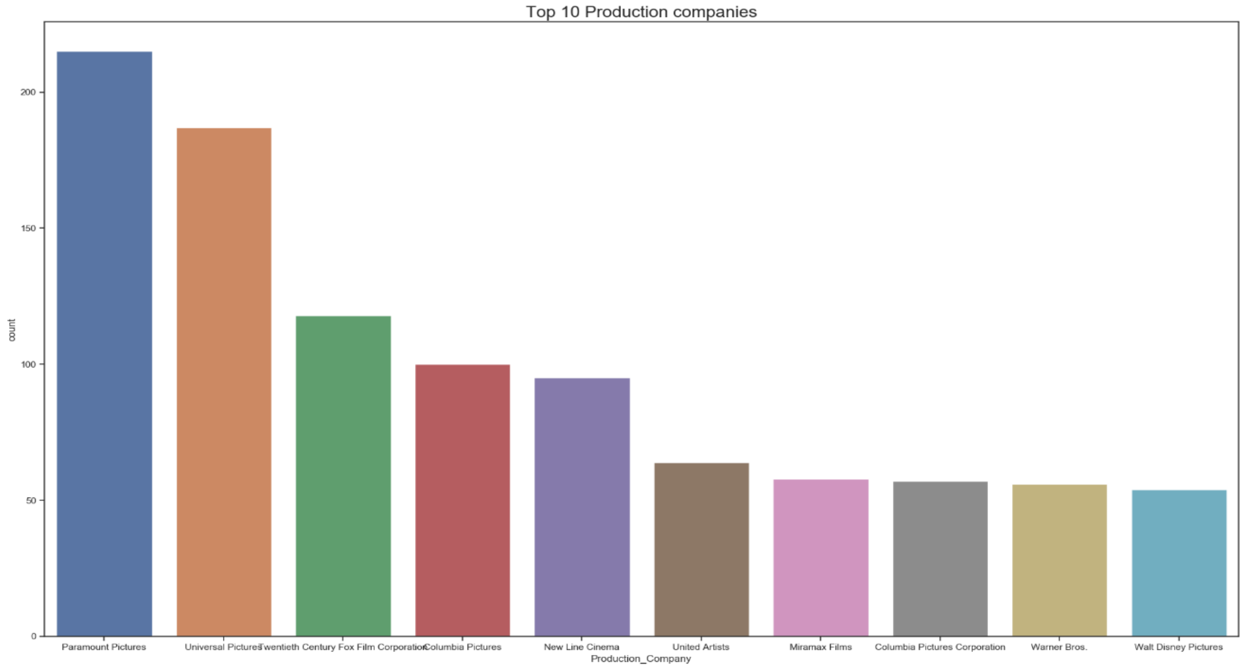
3. Bar graph showing number of movies released in each decade.



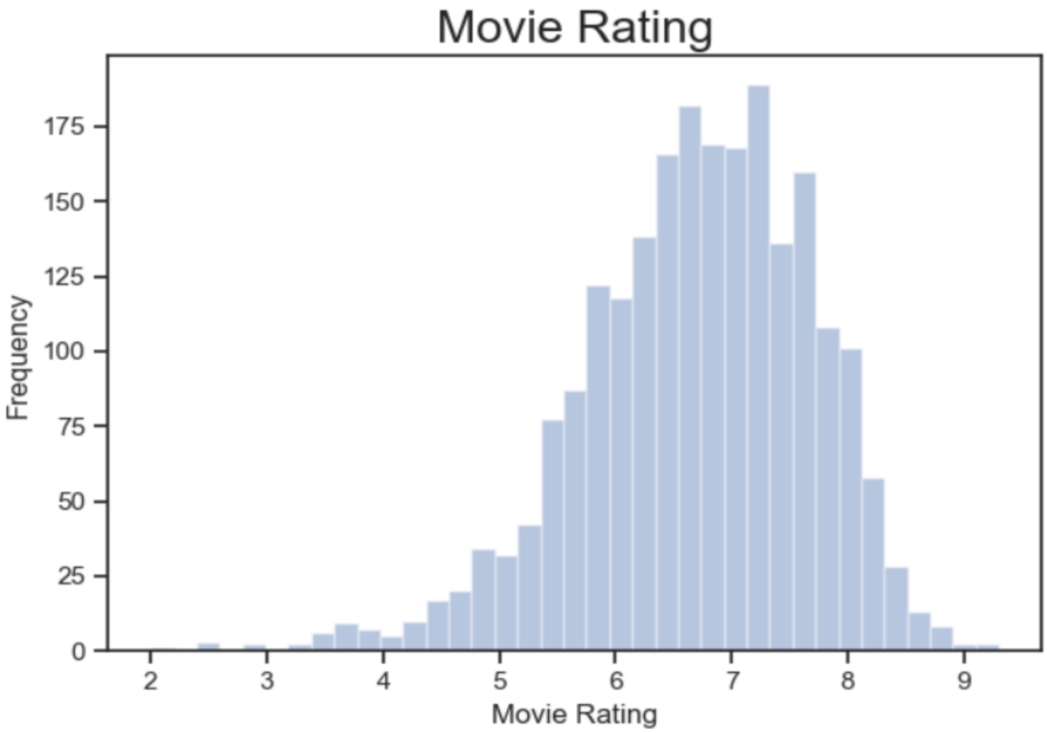
4. Bar graph showing number of movies per each Genre



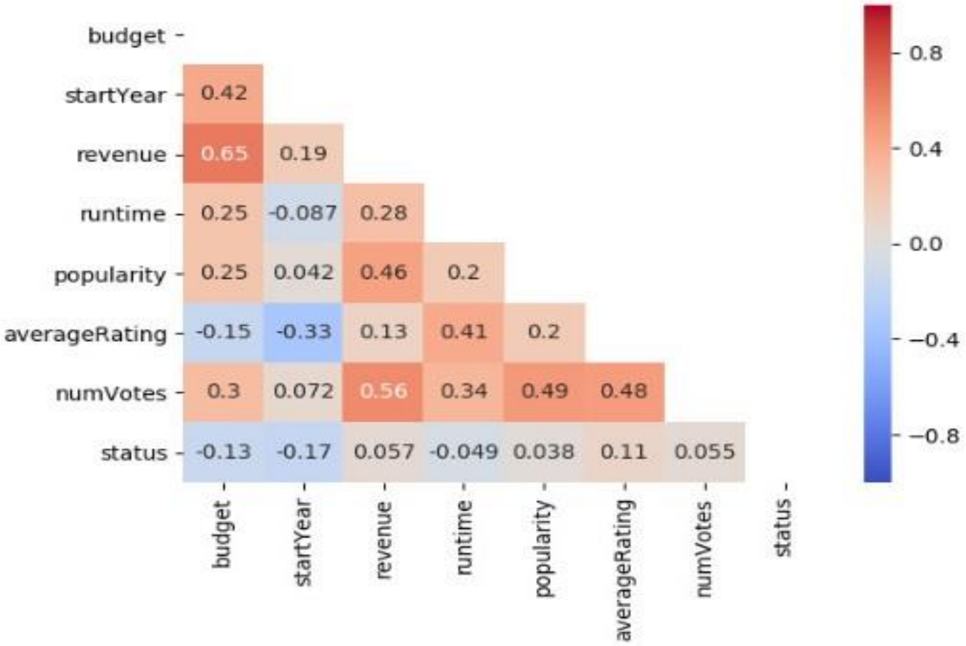
5. Bar graph to demonstrate the top 10 Production Companies.



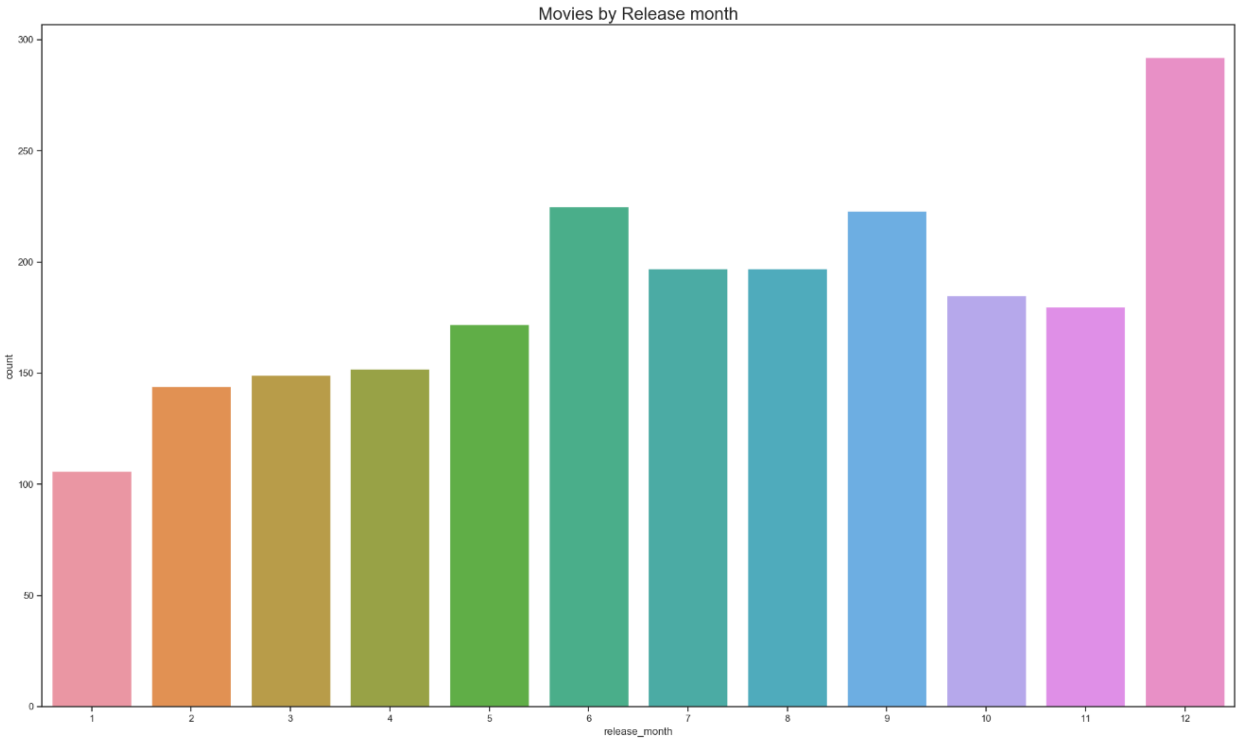
6. Histogram of movie ratings.



7. Correlation heatmap between numerical variables.



8. Bar graph illustrating number of movies released in each month.


Movies by Release month
count
release_month