
SC1015

Mini Project

IMDB Top 2000 Movies Dataset

Chin Wei Hao U2322704F

Chin Hui Qi, Cheryl U2321555A

Christopher Lim Wai Ming U2322618D

Lab Group FCE2, Team 2





Table of contents

01

Our Motivation

You can describe the
topic of the section here

02

Exploratory Data Analysis

You can describe the
topic of the section here

03

Data Preparation

You can describe the
topic of the section here

04

Machine Learning

You can describe the
topic of the section here

05

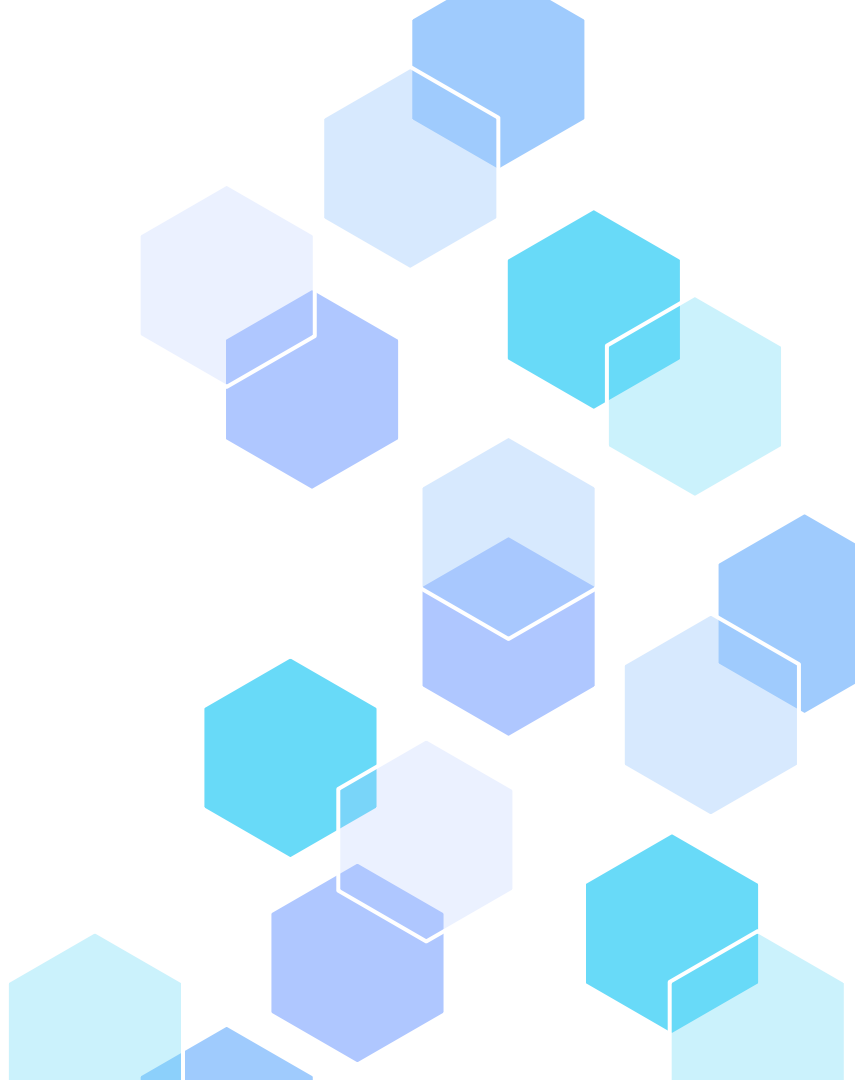
Outcome

You can describe the
topic of the section here

01

Introduction

IMDB Rating



Practical Motivation

Objective:

Determine the influence of various factors (such as Metascore, Votes, Duration, Gross value) on the IMDb ratings of the top 2000 movies.

Primary Data Points in our dataset:

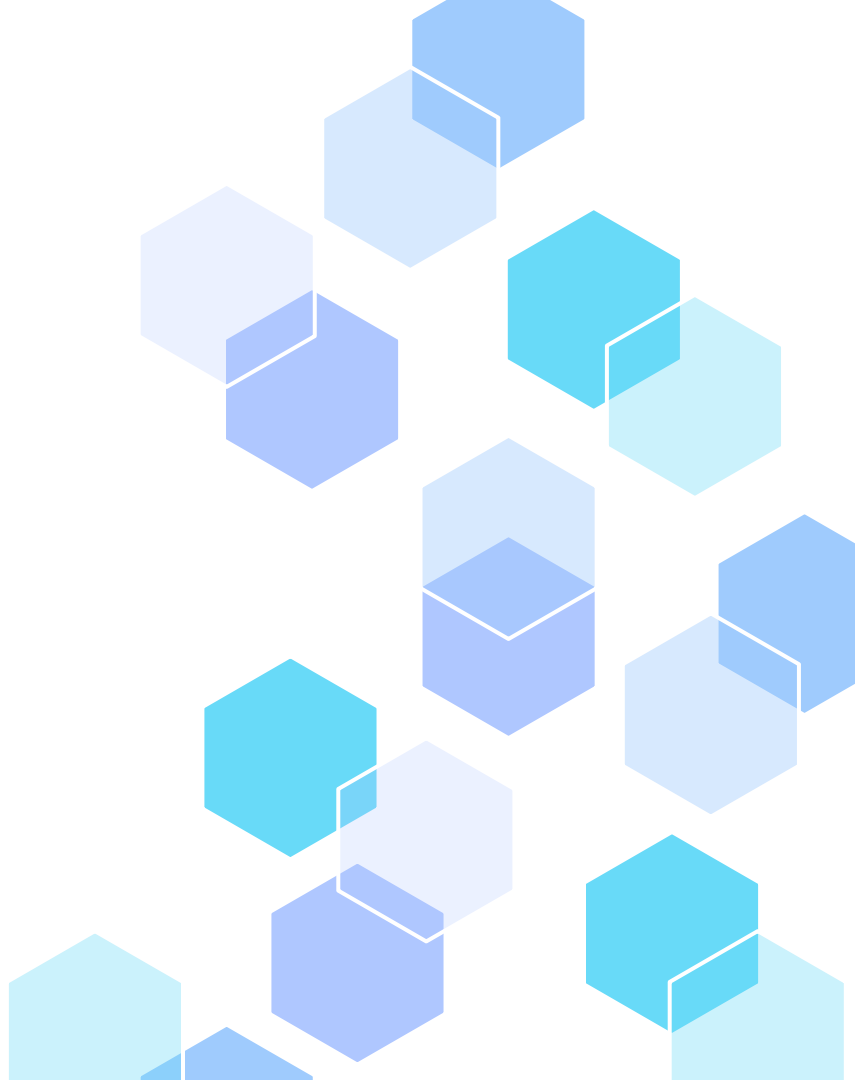
Duration, IMDb ratings, Gross Value, Metascore, Votes

Other Data Points in our dataset:

Name of movie, Release Year, Genre, Director, Cast

02

Exploratory Data Analysis



Initial Data Insights

```
[2]: # Import datasets
data = pd.read_csv('imdb_top_2000_movies.csv')
data.head()
```

```
[2]:
```

	Movie Name	Release Year	Duration	IMDB Rating	Metascore	Votes	Genre	Director	Cast	Gross
0	The Godfather	1972	175	9.2	100.0	2,002,655	Crime, Drama	Francis Ford Coppola	Marlon Brando	\$134.97M
1	The Godfather Part II	1974	202	9.0	90.0	1,358,608	Crime, Drama	Francis Ford Coppola	Al Pacino	\$57.30M
2	Ordinary People	1980	124	7.7	86.0	56,476	Drama	Robert Redford	Donald Sutherland	\$54.80M
3	Lawrence of Arabia	1962	218	8.3	100.0	313,044	Adventure, Biography, Drama	David Lean	Peter O'Toole	\$44.82M
4	Straw Dogs	1971	113	7.4	73.0	64,331	Crime, Drama, Thriller	Sam Peckinpah	Dustin Hoffman	NaN

Preliminary exploration

```
[3]: df = pd.DataFrame(data)  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

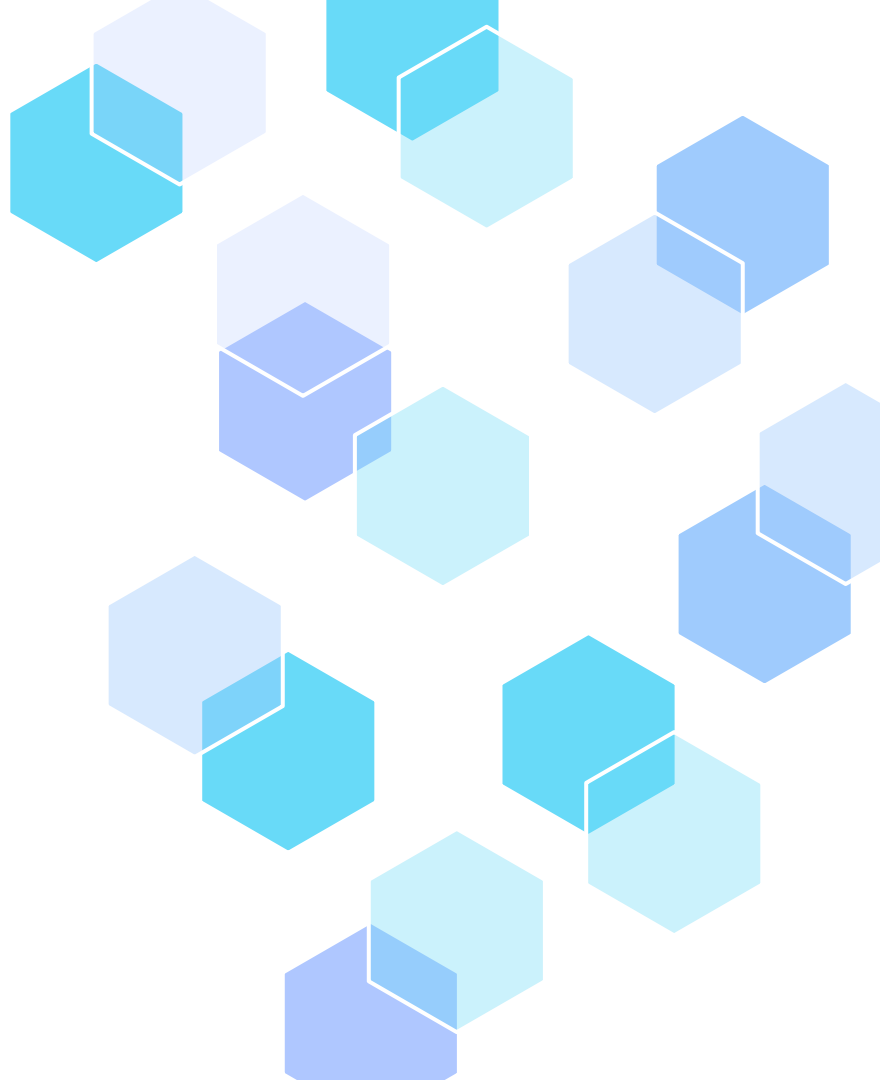
```
RangeIndex: 2000 entries, 0 to 1999
```

```
Data columns (total 10 columns):
```

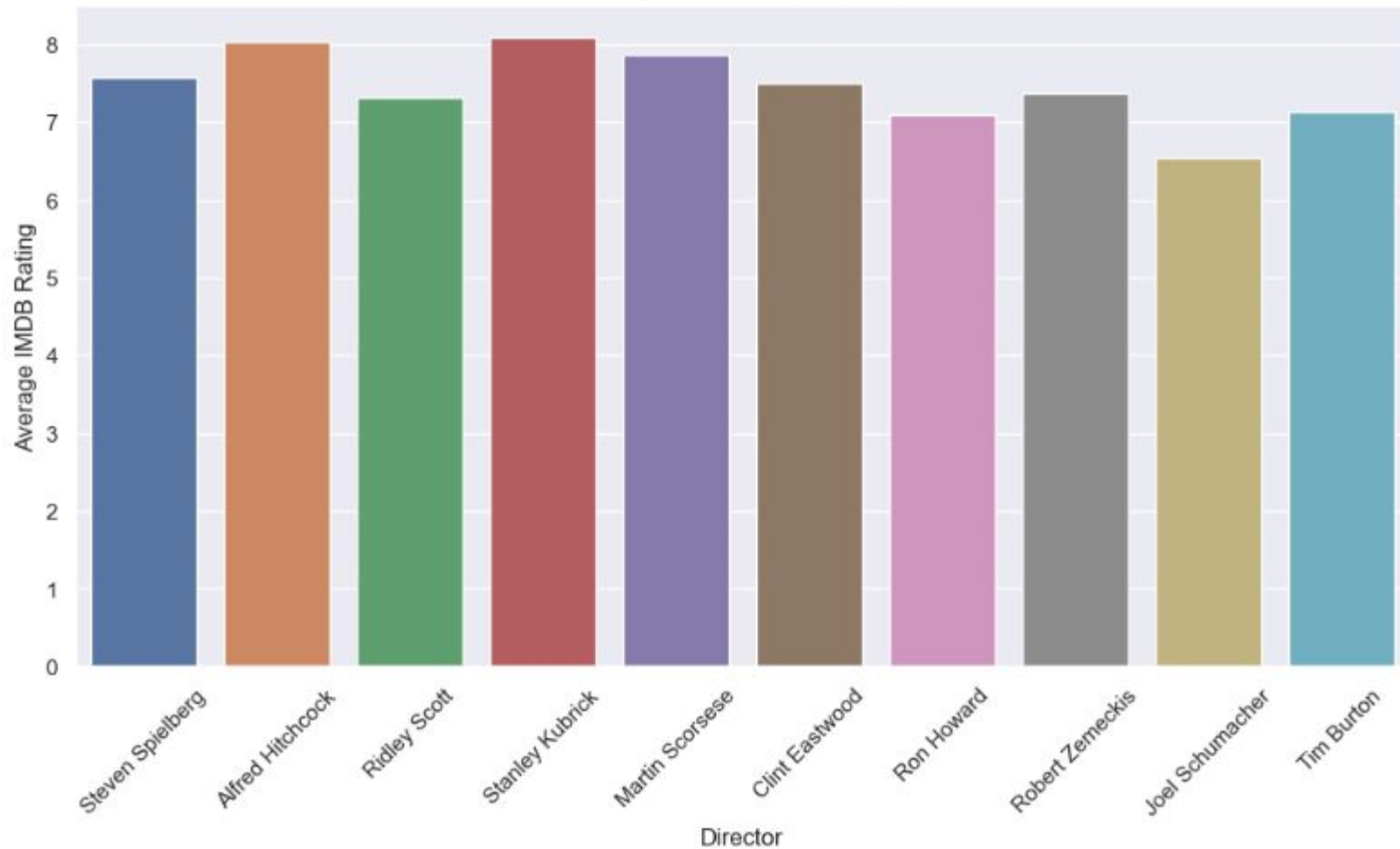
#	Column	Non-Null Count	Dtype
0	Movie Name	2000 non-null	object
1	Release Year	2000 non-null	object
2	Duration	2000 non-null	int64
3	IMDB Rating	2000 non-null	float64
4	Metascore	1919 non-null	float64
5	Votes	2000 non-null	object
6	Genre	2000 non-null	object
7	Director	2000 non-null	object
8	Cast	2000 non-null	object
9	Gross	1903 non-null	object

```
dtypes: float64(2), int64(1), object(7)
```

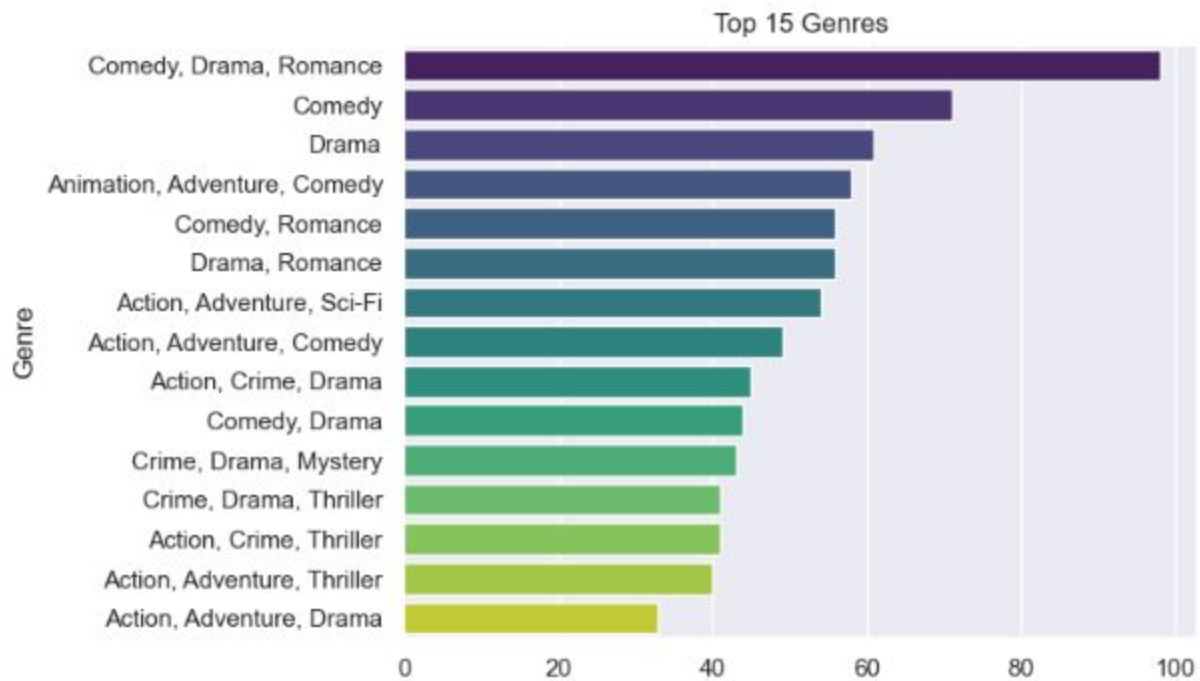
```
memory usage: 156.4+ KB
```

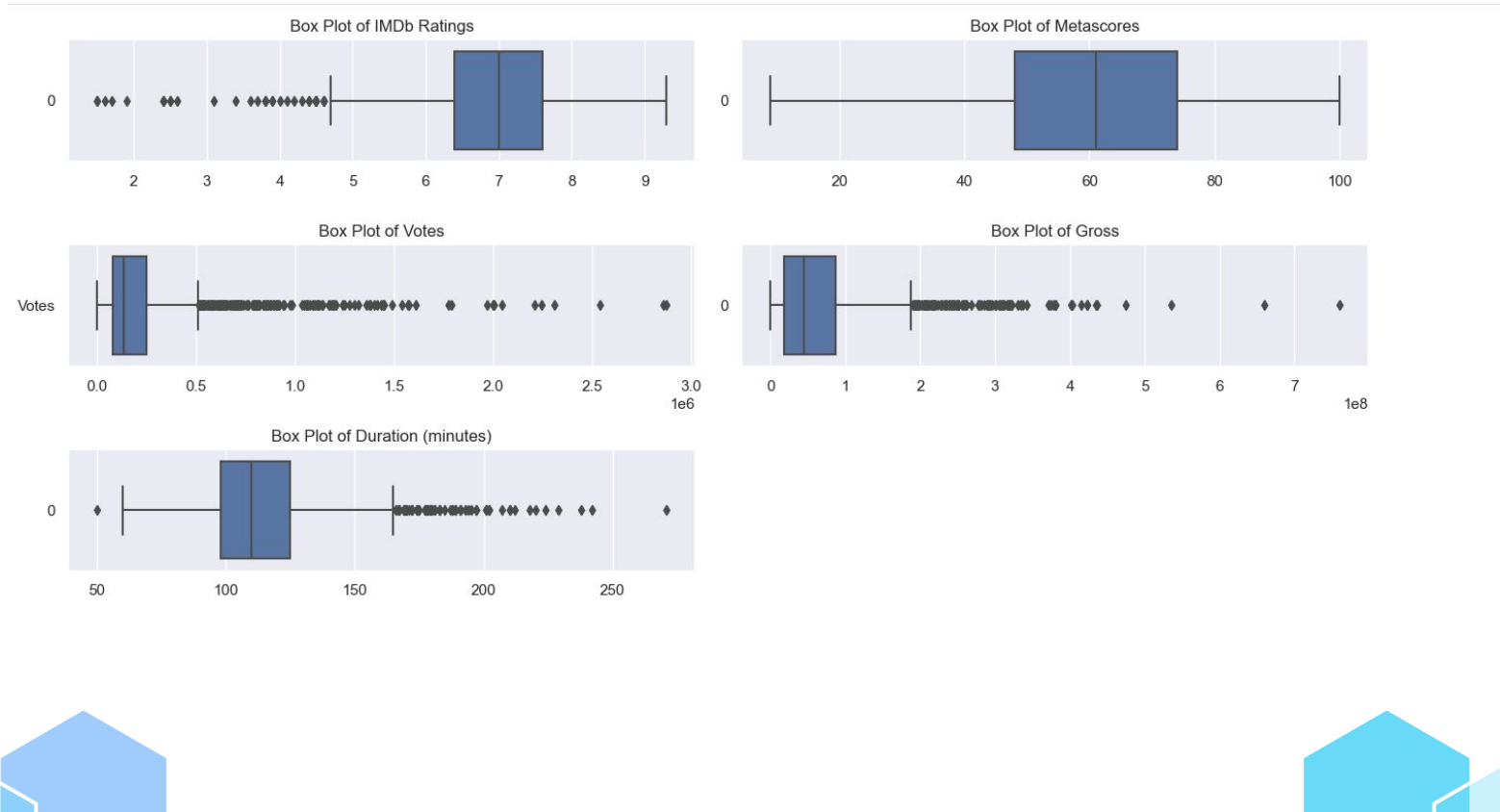


IMDB Ratings by Director (Top 10)



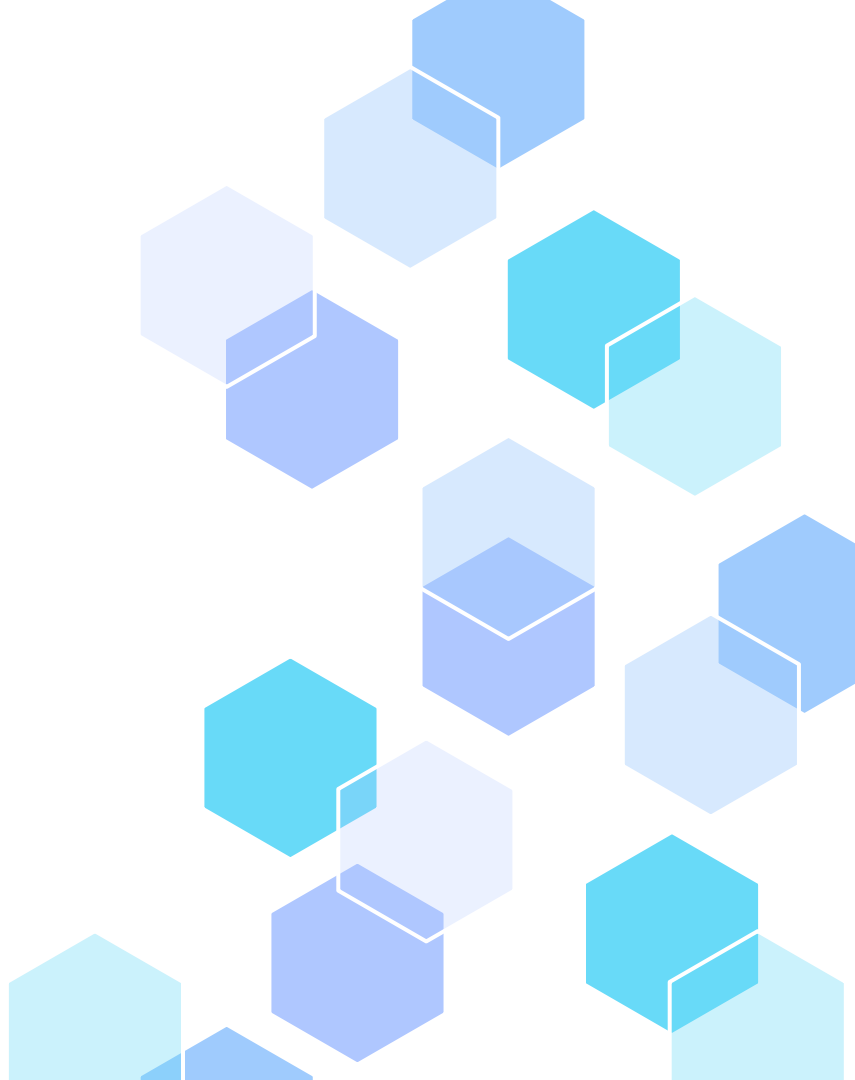
Text(0.5, 1.0, 'Top 15 Genres')





03

Data preparation & cleaning



Data Preparation

```
[3]: # Remove all commas convert object to float for Votes
data['Votes'] = data['Votes'].str.replace(',', '')
data['Votes'] = data['Votes'].astype(float)

# Remove $ sign and M symbol and convert to float for Gross
data['Gross'] = data['Gross'].str.replace('$', '')
data['Gross'] = data['Gross'].str.replace('M', '')
data['Gross'] = data['Gross'].astype(float)
data['Gross'] = data['Gross'] * (10 ** 6)
# Convert Release Year from object to float
data['Release Year'] = pd.to_numeric(data['Release Year'], errors='coerce')
df = pd.DataFrame(data)
```

[4]:

	Movie Name	Release Year	Duration	IMDB Rating	Metascore	Votes	Genre	Director	Cast	Gross
0	The Godfather	1972.0	175	9.2	100.0	2002655.0	Crime, Drama	Francis Ford Coppola	Marlon Brando	134970000.0
1	The Godfather Part II	1974.0	202	9.0	90.0	1358608.0	Crime, Drama	Francis Ford Coppola	Al Pacino	57300000.0
2	Ordinary People	1980.0	124	7.7	86.0	56476.0	Drama	Robert Redford	Donald Sutherland	54800000.0
3	Lawrence of Arabia	1962.0	218	8.3	100.0	313044.0	Adventure, Biography, Drama	David Lean	Peter O'Toole	44820000.0
4	Straw Dogs	1971.0	113	7.4	73.0	64331.0	Crime, Drama, Thriller	Sam Peckinpah	Dustin Hoffman	NaN
...
1995	The Young Victoria	2009.0	105	7.2	64.0	66235.0	Biography, Drama, History	Jean-Marc Vallée	Emily Blunt	11000000.0
1996	Tooth Fairy	NaN	101	5.0	36.0	49527.0	Comedy, Family, Fantasy	Michael Lembeck	Dwayne Johnson	60020000.0
1997	The Informant	2009.0	108	6.5	66.0	67318.0	Biography, Comedy, Crime	Steven Soderbergh	Matt Damon	33310000.0
1998	Youth in Revolt	2009.0	90	6.4	63.0	75956.0	Comedy, Drama, Romance	Miguel Arteta	Michael Cera	15280000.0
1999	Quarantine	2008.0	89	6.0	53.0	77075.0	Horror, Sci-Fi, Thriller	John Erick Dowdle	Jennifer Carpenter	31690000.0

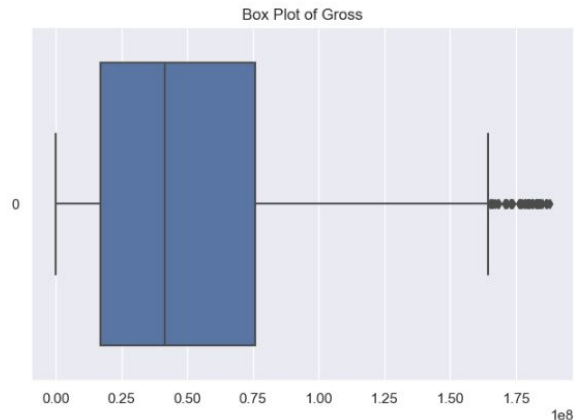
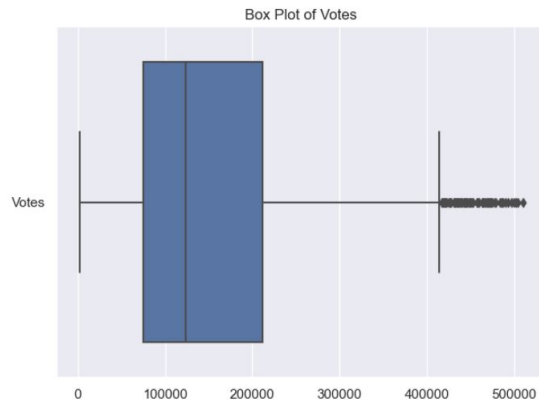
2000 rows × 10 columns

Keep	Remove
1. Duration, 2. IMDb ratings 3. Gross Value 4. Metascore 5. Votes 6. Release Year	1. Name of movie 2. Genre 3. Director 4. Cast

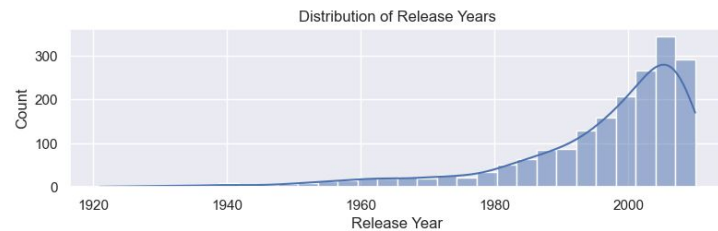
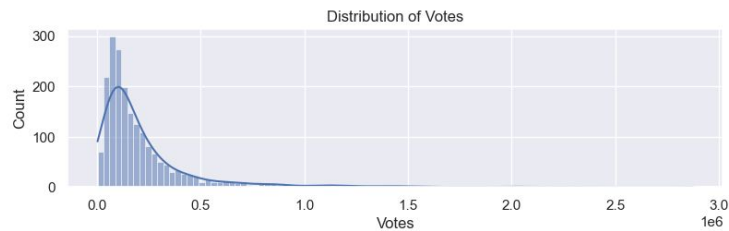
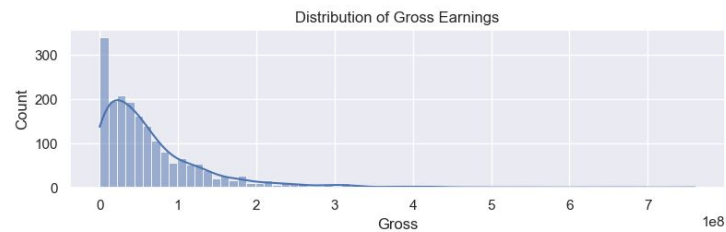
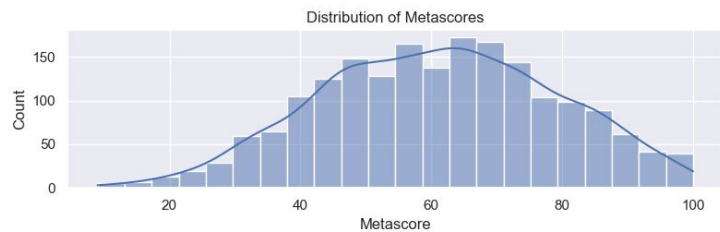
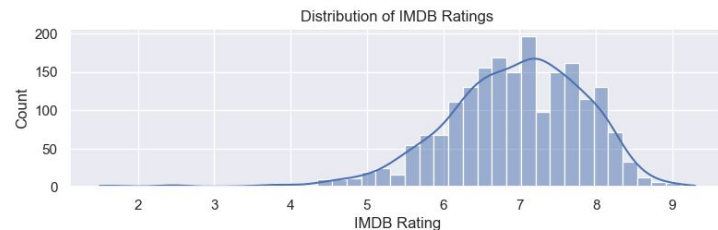
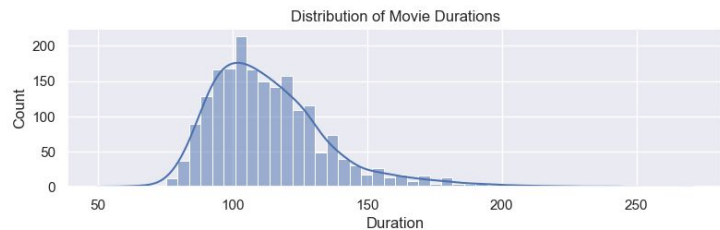
Cleaning data

```
[6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2000 entries, 0 to 1999  
Data columns (total 10 columns):  
#   Column          Non-Null Count  Dtype    
---  ---            -  
0   Movie Name      2000 non-null   object   
1   Release Year    1921 non-null   float64  
2   Duration        2000 non-null   int64    
3   IMDB Rating     2000 non-null   float64  
4   Metascore       1919 non-null   float64  
5   Votes           2000 non-null   float64  
6   Genre           2000 non-null   object   
7   Director        2000 non-null   object   
8   Cast            2000 non-null   object   
9   Gross           1903 non-null   float64  
dtypes: float64(5), int64(1), object(4)  
memory usage: 156.4+ KB
```



Analysis of data

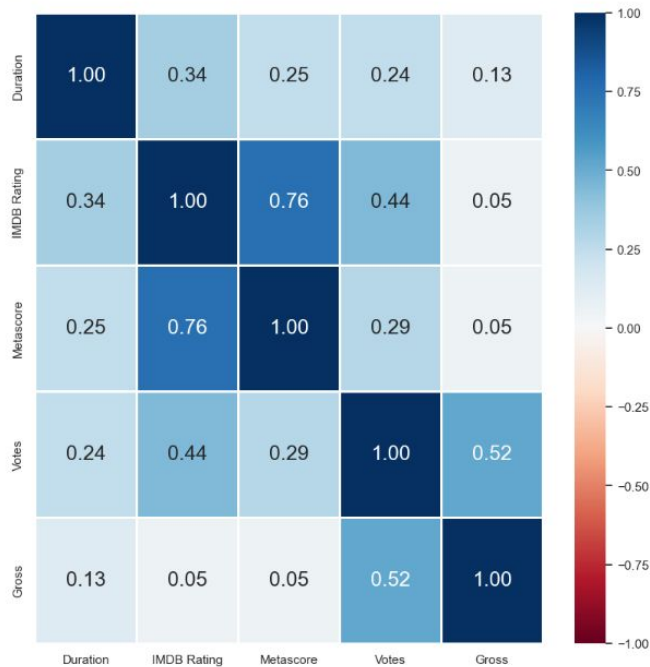


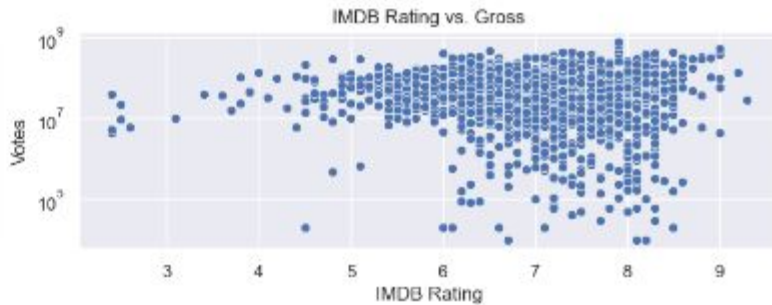
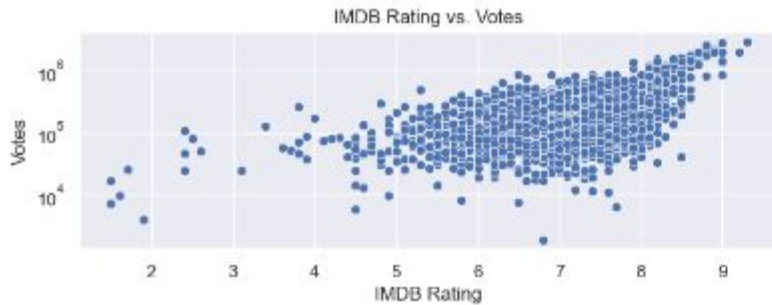
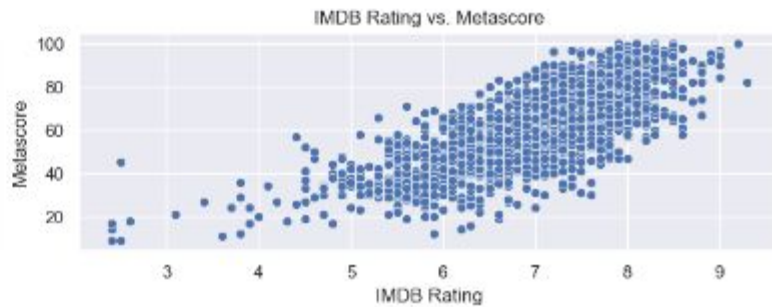
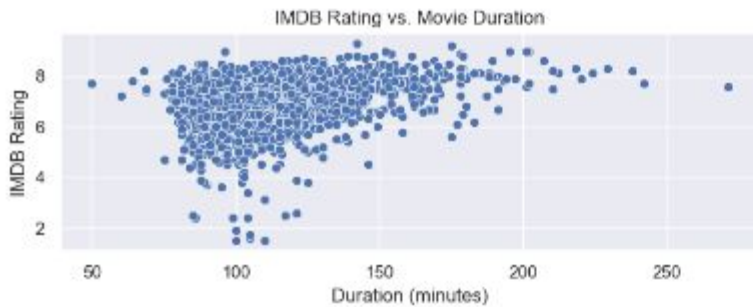
Analysis of data

Heatmap

```
[9]: corrDF=pd.DataFrame(df[['Duration', 'IMDB Rating', 'Metascore', 'Votes', 'Gross']])  
f = plt.figure(figsize=(10, 10))  
sb.heatmap(corrDF.corr(), vmin = -1, vmax = 1, linewidths = 1,  
          annot = True, fmt = ".2f", annot_kws = {"size": 10}, cmap = "RdBu")
```

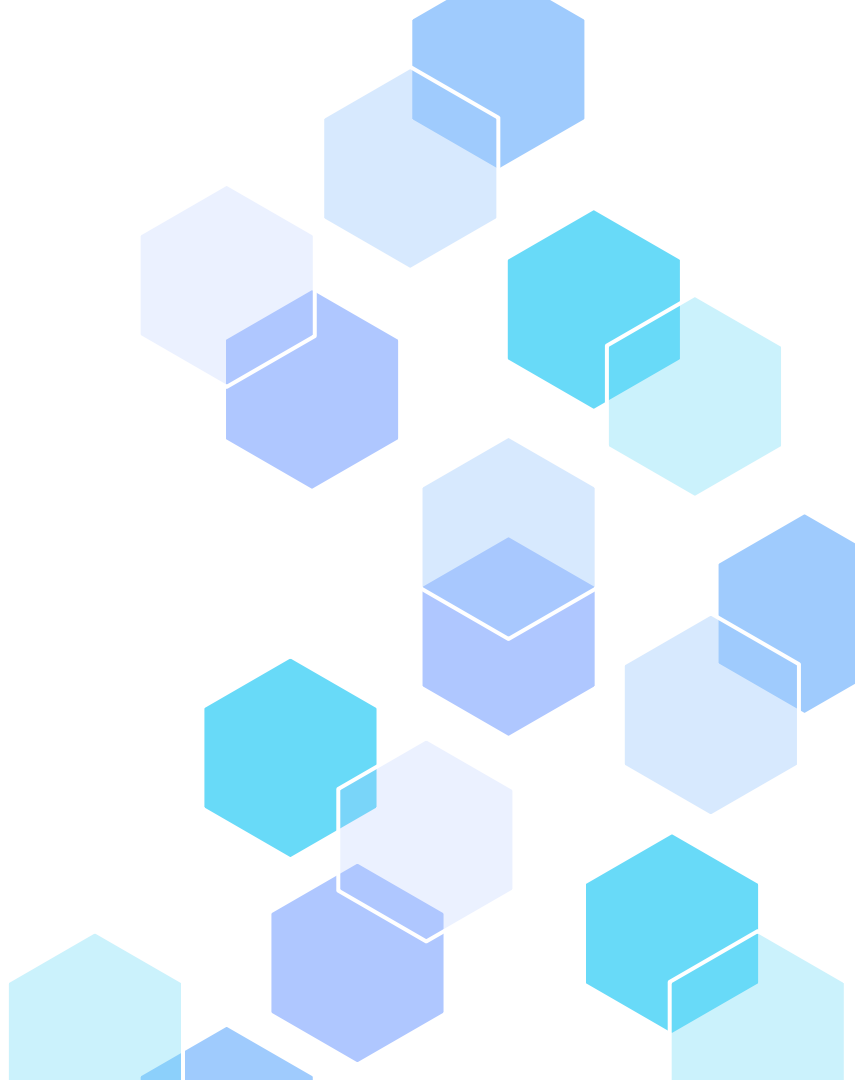
[9]: <Axes: >



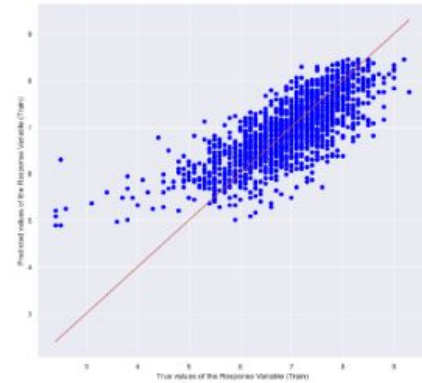


04 Machine Learning

Regression, Decision Tree, Random Forest



Regression

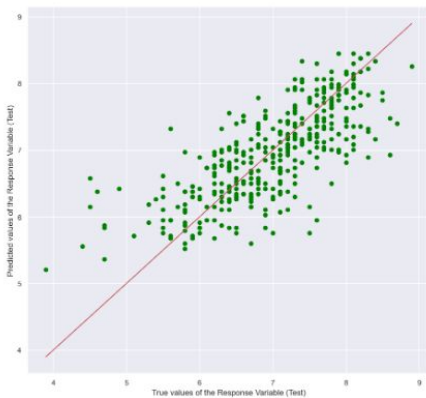
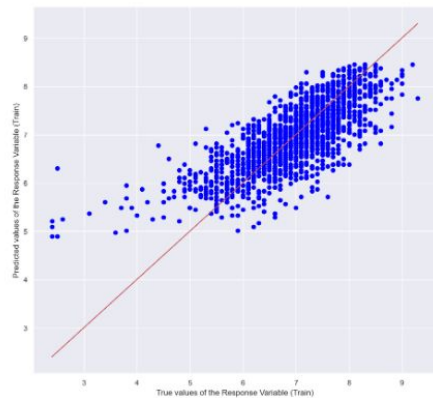


IMDB Rating vs Metascore

Intercept of Regression : b = [4.5455173]
Coefficients of Regression : a = [[0.03909264]]

Goodness of Fit of Model : Train Dataset
Explained Variance (R^2) : 0.5733895815463212
Mean Squared Error (MSE) : 0.36699191732906405

Goodness of Fit of Model : Test Dataset
Explained Variance (R^2) : 0.563041155223023
Mean Squared Error (MSE) : 0.31406405115077946

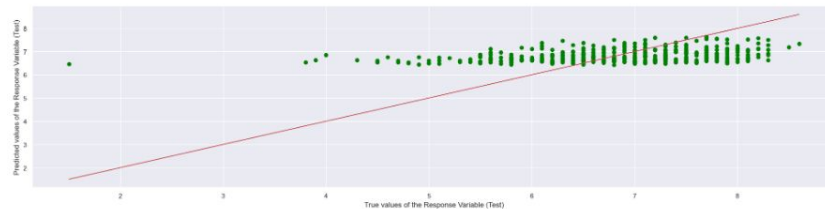
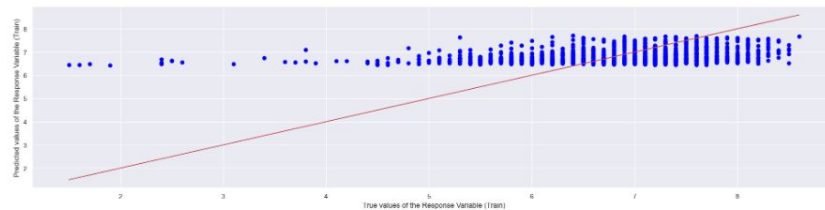


IMDB Rating vs Votes

Intercept of Regression : b = [6.42455104]
Coefficients of Regression : a = [[2.50708159e-06]]

Goodness of Fit of Model : Train Dataset
Explained Variance (R^2) : 0.087389065080938515
Mean Squared Error (MSE) : 0.7869578513104797

Goodness of Fit of Model : Test Dataset
Explained Variance (R^2) : 0.11139791960741419
Mean Squared Error (MSE) : 0.7025562930518372

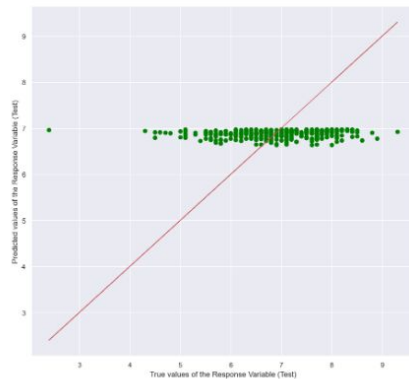
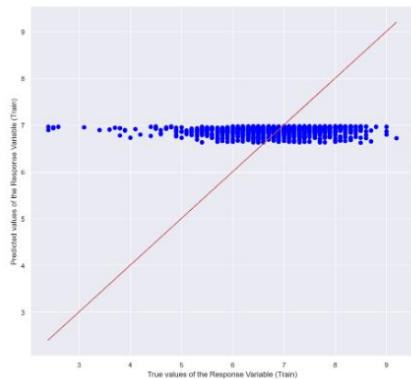


IMDB Rating vs Gross

Intercept of Regression : b = [6.97735058]
Coefficients of Regression : a = [[-1.8426805e-09]]

Goodness of Fit of Model : Train Dataset
Explained Variance (R²) : 0.008315503020466797
Mean Squared Error (MSE) : 0.814371021646203

Goodness of Fit of Model : Test Dataset
Explained Variance (R²) : 0.004015663672795378
Mean Squared Error (MSE) : 0.8098473878865355

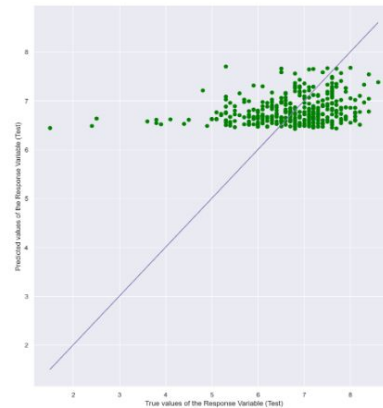
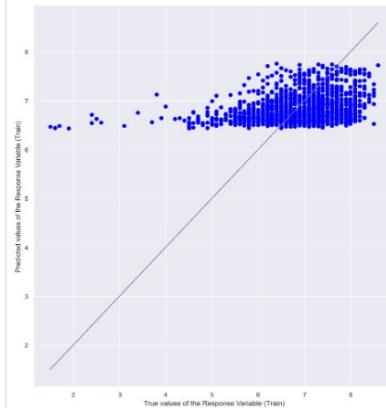


IMDB Rating vs Duration

Intercept of Regression : b = [6.4246543]
Coefficients of Regression : a = [[2.631455e-06]]

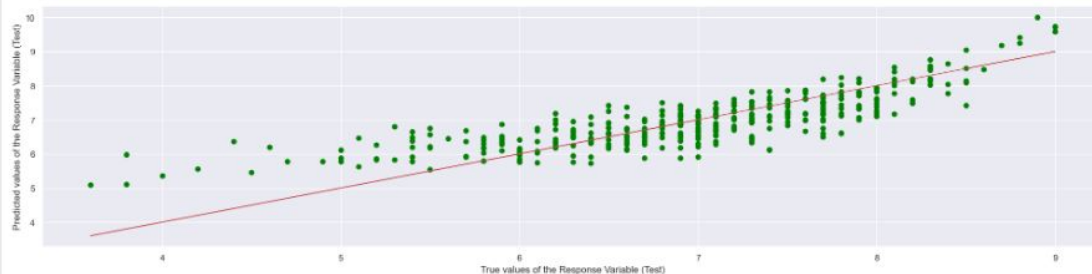
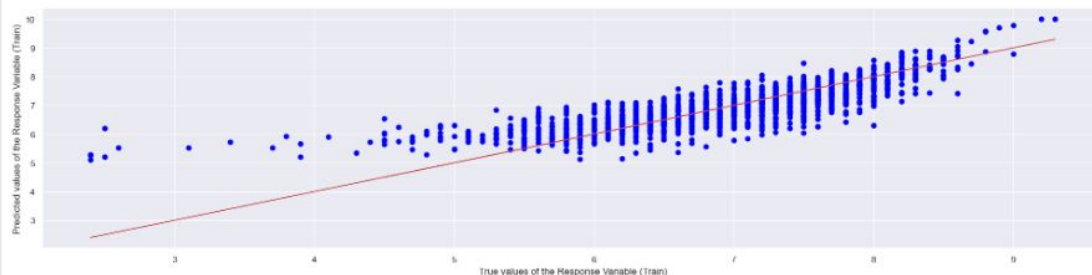
Goodness of Fit of Model : Train Dataset
Explained Variance (R²) : 0.09849251217694732
Mean Squared Error (MSE) : 0.7450839952763629
Root Mean Squared Error (RMSE) : 0.8631824808673789

Goodness of Fit of Model : Test Dataset
Explained Variance (R²) : 0.06118727426560344
Mean Squared Error (MSE) : 0.8499283539214618
Root Mean Squared Error (RMSE) : 0.9219155893689301

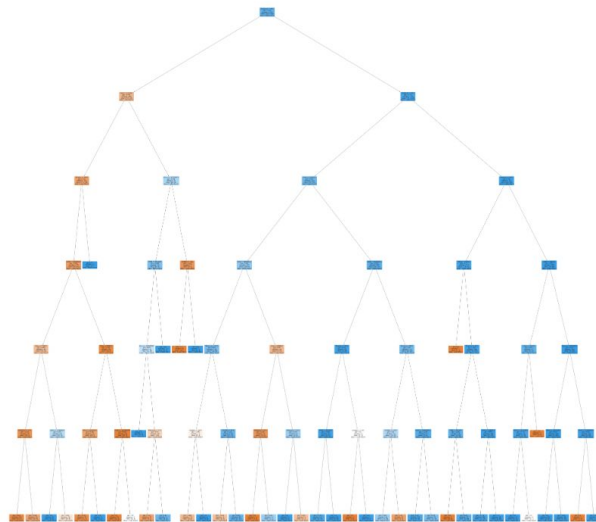


IMDB Rating vs Duration+Gross+Votes+MetaScore

Intercept of Regression	: b = [4.21918997]
Coefficients of Regression	: a = [[1.07726340e-06 -1.95981609e-09 5.63590533e-03 3.20035084e-02]]
Goodness of Fit of Model	Train Dataset
Explained Variance (R ²)	: 0.6653574052063997
Mean Squared Error (MSE)	: 0.27248534538253527
Goodness of Fit of Model	Test Dataset
Explained Variance (R ²)	: 0.6646272068009653
Mean Squared Error (MSE)	: 0.2893297167187535



Decision Tree

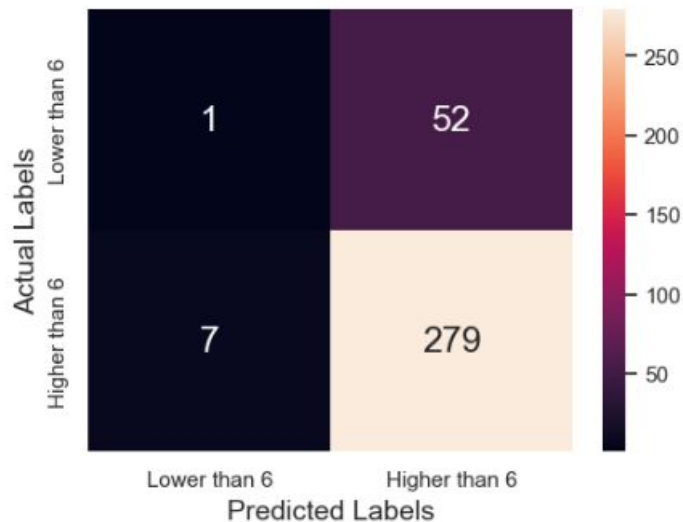
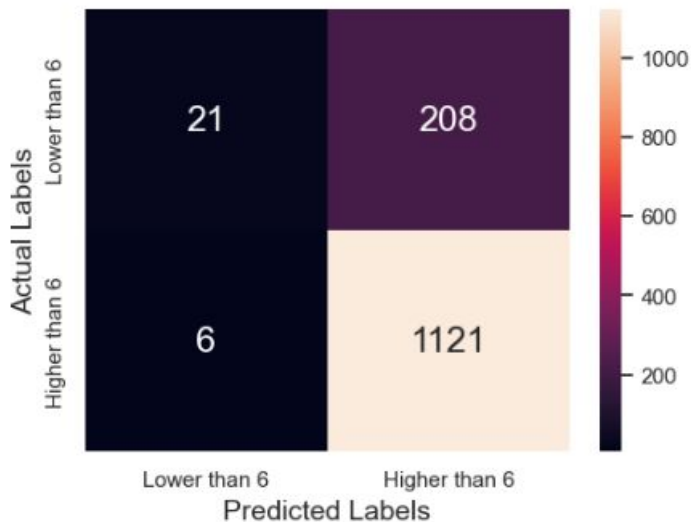


IMDB Rating vs Votes

Goodness of Fit of Model Train Dataset
Classification Accuracy : 0.8421828908554573
TPR for train : 0.9946761313220941
FPR for train : 0.9082969432314411

Goodness of Fit of Model Test Dataset
Classification Accuracy : 0.8259587020648967
TPR for train : 0.9755244755244755
FPR for train : 0.9811320754716981

Text(624.5227272727271, 0.5, 'Actual Labels')

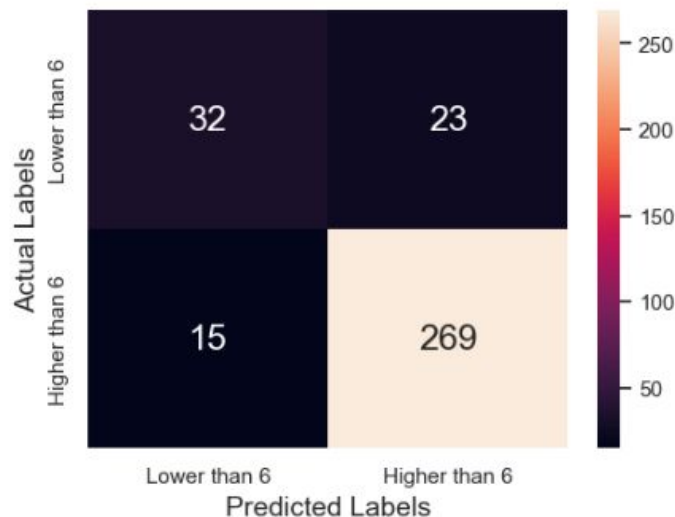


IMDB Rating vs Metascore

```
Goodness of Fit of Model      Train Dataset
Classification Accuracy      : 0.883480825958702
TPR for train   : 0.9654561558901683
FPR for train   : 0.5242290748898678
```

```
Goodness of Fit of Model      Test Dataset
Classification Accuracy      : 0.887905604719764
TPR for train   : 0.9471830985915493
FPR for train   : 0.418181818181815
```

```
Text(624.5227272727271, 0.5, 'Actual Labels')
```

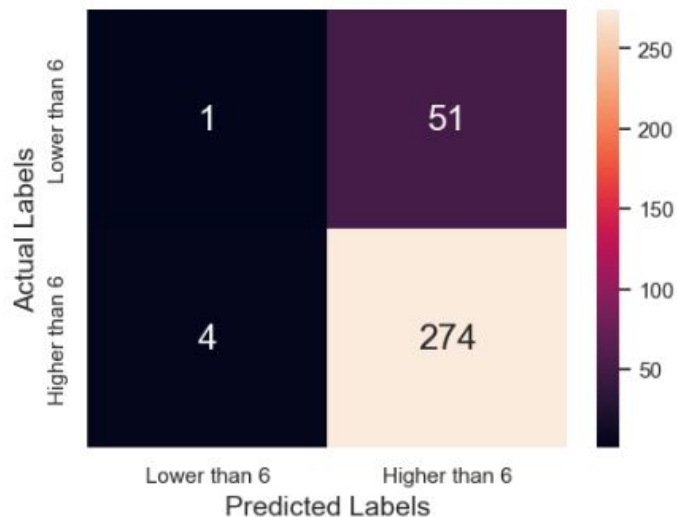
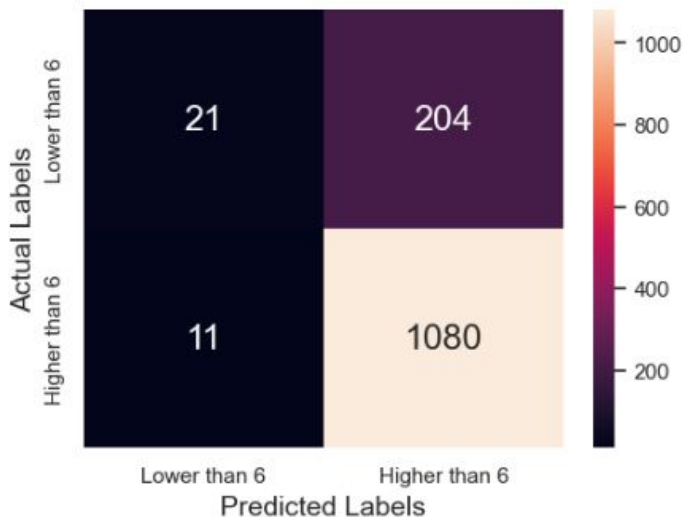


IMDB Rating vs Gross

Goodness of Fit of Model Train Dataset
Classification Accuracy : 0.8366261398176292
TPR for train : 0.9899175068744271
FPR for train : 0.9066666666666666

Goodness of Fit of Model Test Dataset
Classification Accuracy : 0.8333333333333334
TPR for train : 0.9856115107913669
FPR for train : 0.9807692307692307

Text(624.5227272727271, 0.5, 'Actual Labels')

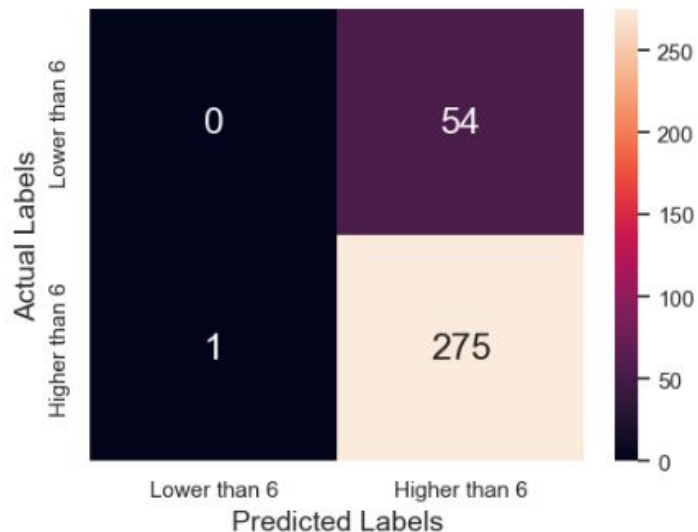
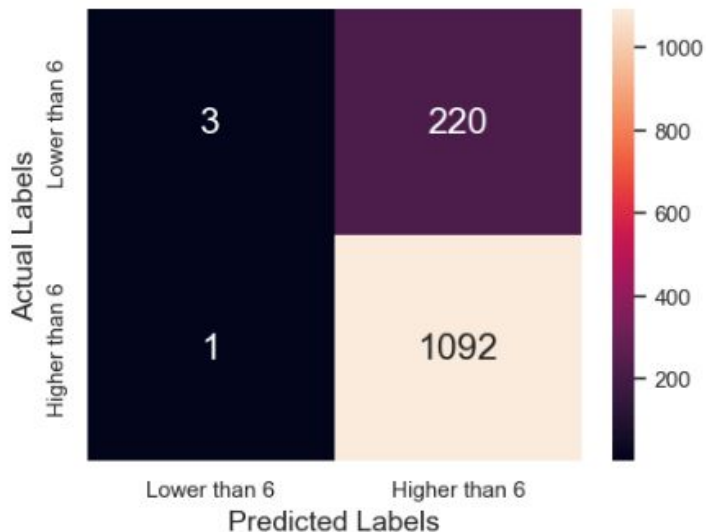


IMDB Rating vs Duration

Goodness of Fit of Model Train Dataset
Classification Accuracy : 0.8320668693009119
TPR for train : 0.9990850869167429
FPR for train : 0.9865470852017937

Goodness of Fit of Model Test Dataset
Classification Accuracy : 0.8333333333333334
TPR for train : 0.9963768115942029
FPR for train : 1.0

Text(624.5227272727271, 0.5, 'Actual Labels')

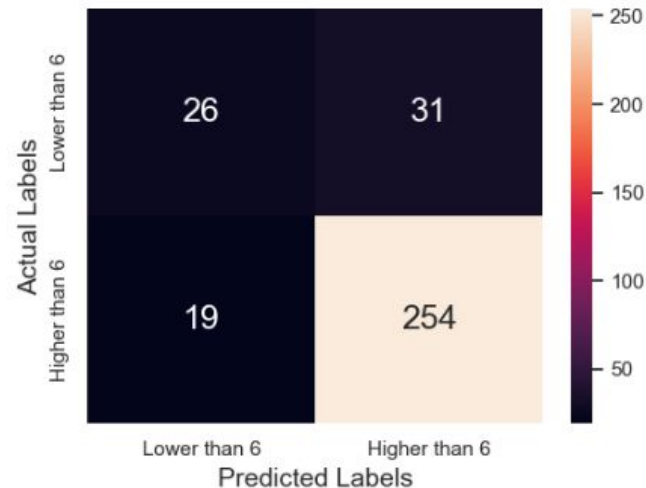


IMDB Rating vs Duration+Gross+Votes+MetaScore

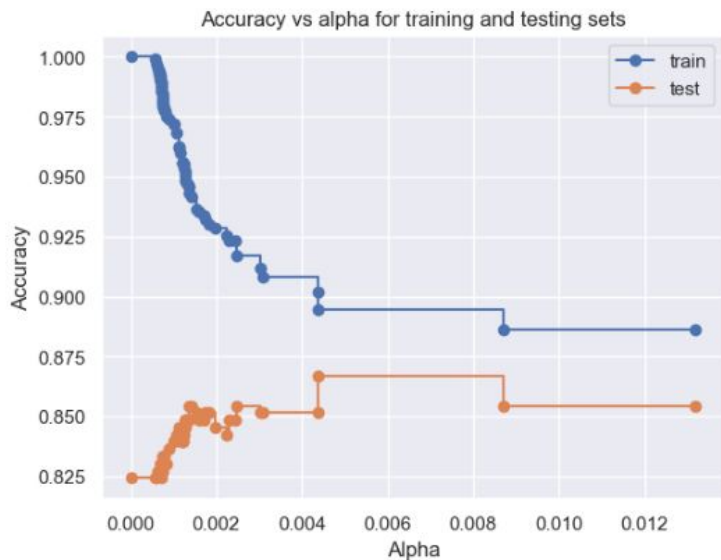
```
Goodness of Fit of Model      Train Dataset  
Classification Accuracy      : 0.9354103343465046  
TPR for train   : 0.9762773722627737  
FPR for train   : 0.2681818181818182
```

```
Goodness of Fit of Model      Test Dataset  
Classification Accuracy       : 0.8484848484848485  
TPR for train   : 0.9304029304029304  
FPR for train   : 0.543859649122807
```

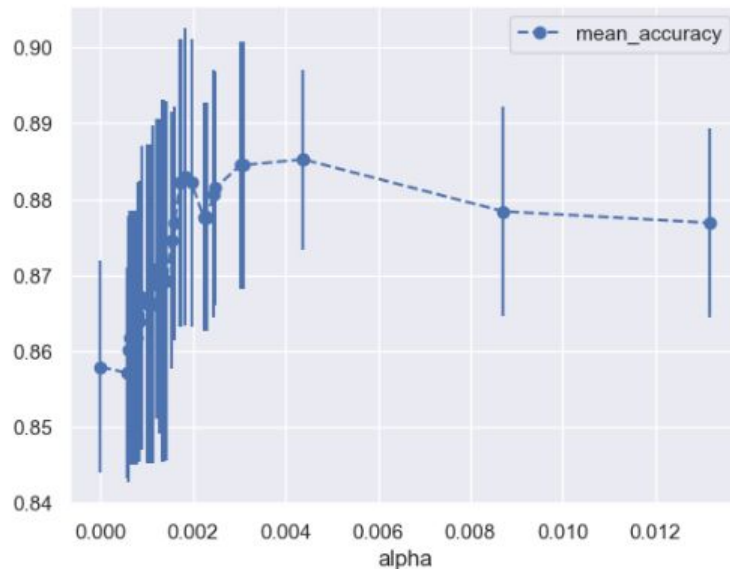
```
Text(624.5227272727271, 0.5, 'Actual Labels')
```



Visualizing alpha

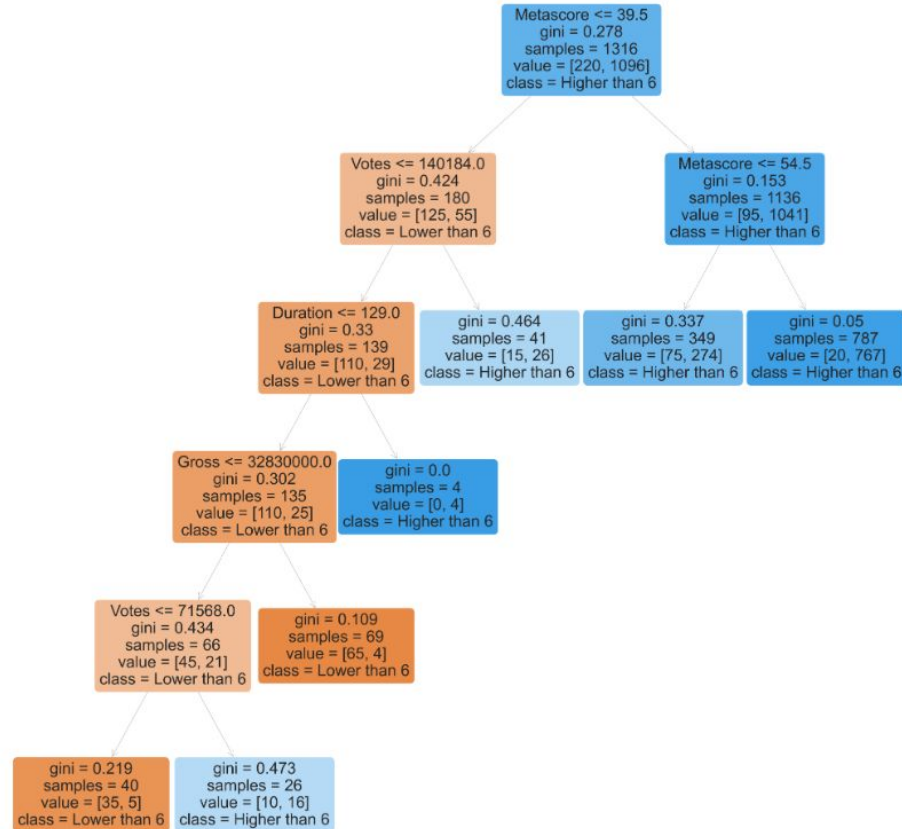


	alpha	mean_accuracy	std
55	0.002228	0.877662	0.015077
56	0.002293	0.877662	0.015077
57	0.002453	0.880709	0.016333
58	0.002486	0.881470	0.015380
59	0.003018	0.884509	0.016335
60	0.003096	0.884509	0.016335
61	0.004362	0.885260	0.011857
62	0.004383	0.885260	0.011857
63	0.008712	0.878416	0.013840



	alpha	mean_accuracy	std
61	0.004362	0.88526	0.011857

Cost Complexity Pruning

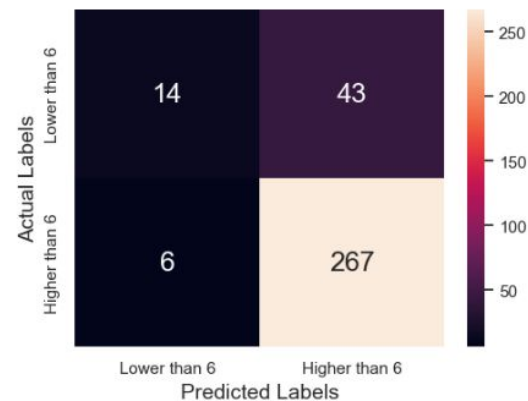


Cost Complexity Pruning

Goodness of Fit of Model Train Dataset
Classification Accuracy : 0.9019756838905775
TPR for train : 0.9917883211678832
FPR for train : 0.5454545454545454

Goodness of Fit of Model Test Dataset
Classification Accuracy : 0.8515151515151516
TPR for train : 0.978021978021978
FPR for train : 0.7543859649122807

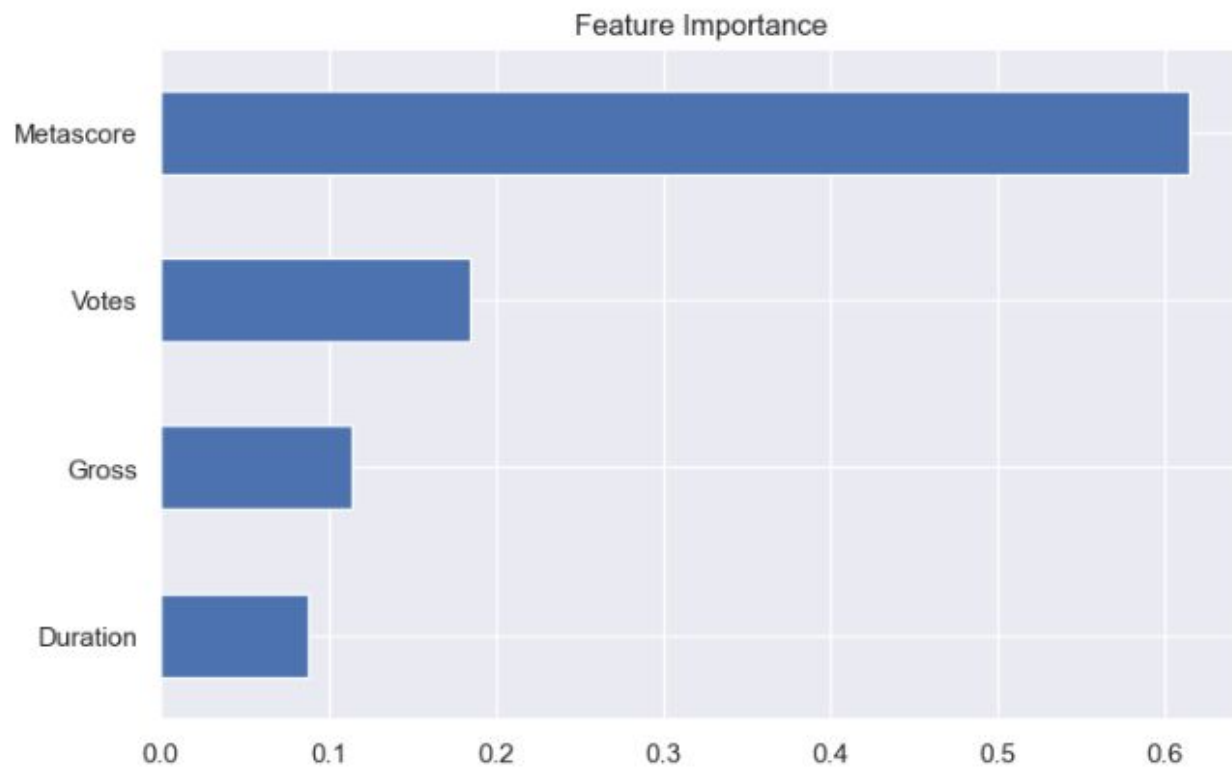
Text(624.5227272727271, 0.5, 'Actual Labels')

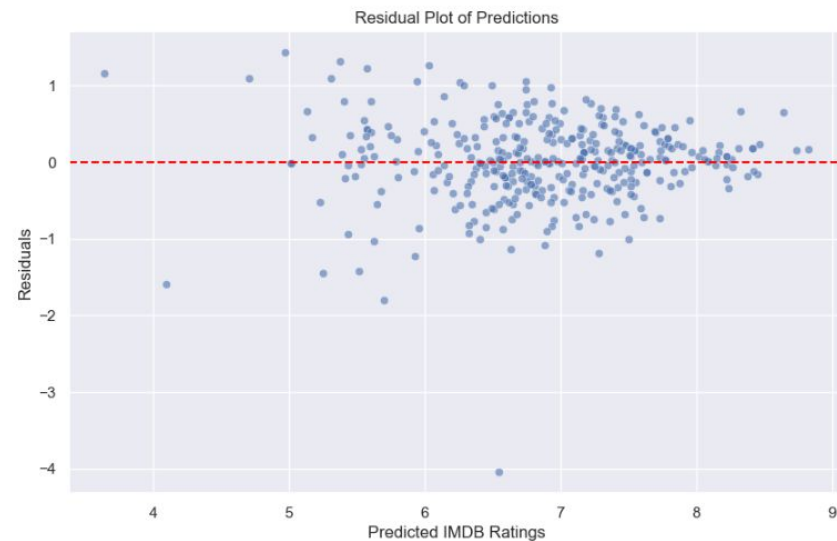
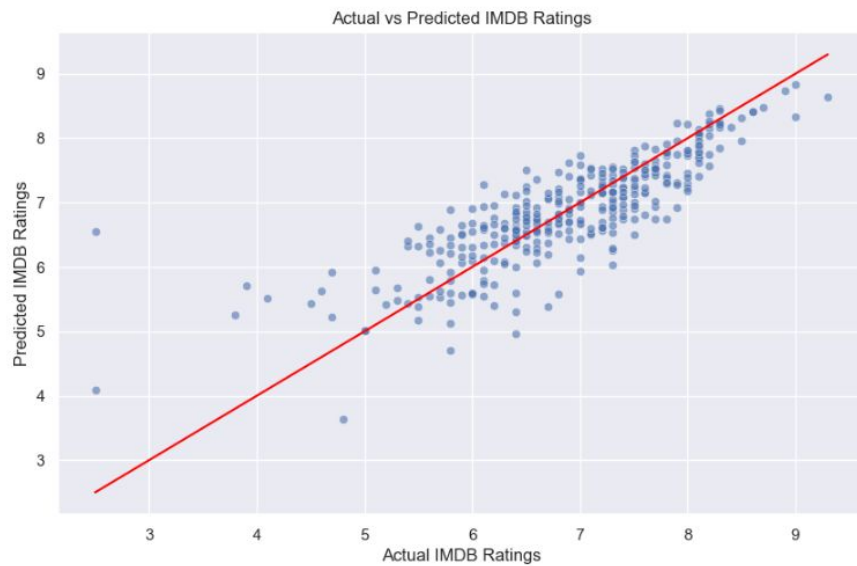


Random Forest

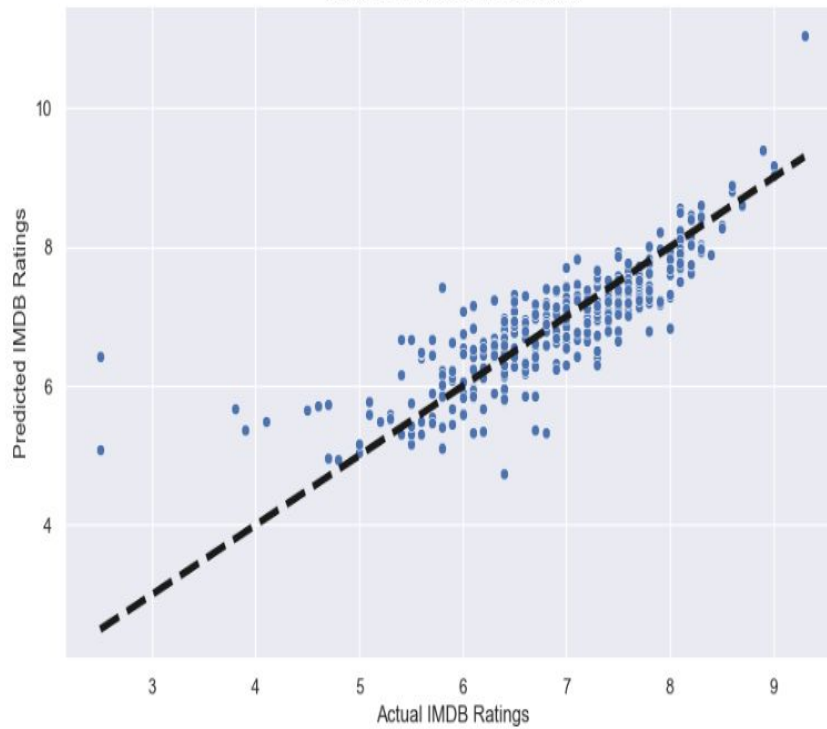
Mean Squared Error: 0.2891176221590909

R-squared: 0.6898316378080778

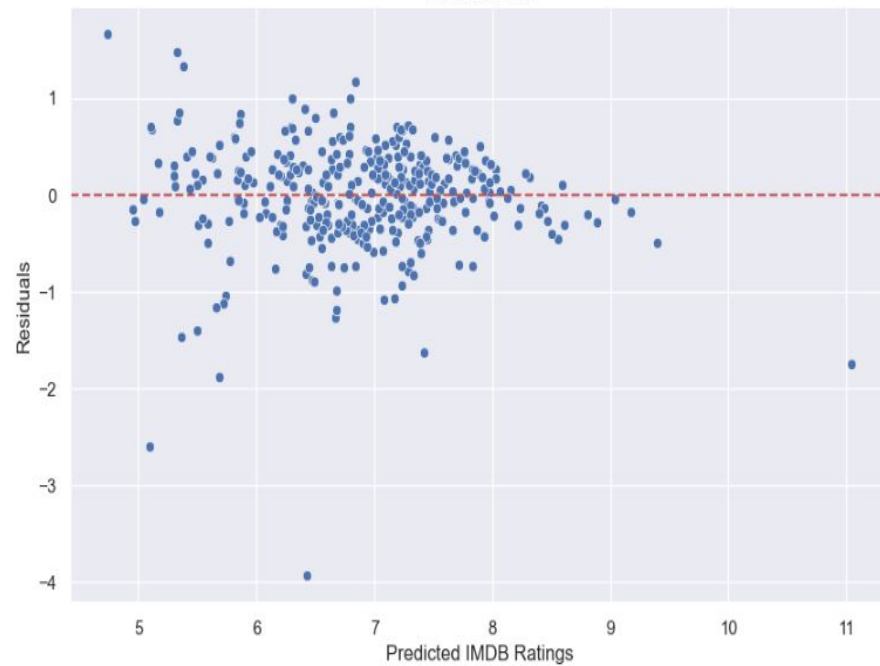




Actual vs Predicted IMDB Ratings



Residual Plot



Data Insights/Conclusion

1. All in all, we found that Metascore has the highest correlation with IMDB Ratings
2. Dataset is skewed towards higher IMDB Ratings due to nature of our dataset.
3. Model created using Decision tree has a high False Positive Rate
4. Model for Logistic Regression created using this data can only predict IMDB rating above 6 due to the dataset constraints.
5. Multivariate linear regression produced the most accurate results in predicting the ratings of the movie as compared to using univariate regression, with Metascore being the most important variable.