

June 25, 2024

### Task-3: MOVIE RATING PREDICTION WITH PYTHON

```
[213]: #Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('/content/IMDb Movies India.csv',encoding='ISO-8859-1')
df
```

```
[213]:
```

	Name	Year	Duration	Genre	\
0		NaN	NaN	Drama	
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	
2	#Homecoming	(2021)	90 min	Drama, Musical	
3	#Yaaram	(2019)	110 min	Comedy, Romance	
4	...And Once Again	(2010)	105 min	Drama	
...	...	...	...	...	
15504	Zulm Ko Jala Doonga	(1988)	NaN	Action	
15505	Zulmi	(1999)	129 min	Action, Drama	
15506	Zulmi Raj	(2005)	NaN	Action	
15507	Zulmi Shikari	(1988)	NaN	Action	
15508	Zulm-O-Sitam	(1998)	130 min	Action, Drama	

	Rating	Votes	Director	Actor 1	Actor 2	\
0	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	
1	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	
2	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	
3	4.4	35	Ovais Khan	Prateik	Ishita Raj	
4	NaN	NaN	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	
...	...	...	...	...	...	
15504	4.6	11	Mahendra Shah	Naseeruddin Shah	Sumeet Saigal	
15505	4.5	655	Kuku Kohli	Akshay Kumar	Twinkle Khanna	
15506	NaN	NaN	Kiran Thej	Sangeeta Tiwari	NaN	
15507	NaN	NaN	NaN	NaN	NaN	
15508	6.2	20	K.C. Bokadia	Dharmendra	Jaya Prada	

	Actor 3
0	Rajendra Bhatia

```

1      Arvind Jangid
2      Roy Angana
3      Siddhant Kapoor
4      Antara Mali
...
15504   Suparna Anand
15505   Aruna Irani
15506   NaN
15507   NaN
15508   Arjun Sarja

```

[15509 rows x 10 columns]

```
[214]: #get overview of the dataset
df.head()
```

```
[214]:
```

	Name	Year	Duration	Genre	\
0		NaN	NaN	Drama	
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	
2	#Homecoming	(2021)	90 min	Drama, Musical	
3	#Yaaram	(2019)	110 min	Comedy, Romance	
4	...And Once Again	(2010)	105 min	Drama	

	Rating	Votes	Director	Actor 1	Actor 2	\
0	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	
1	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	
2	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	
3	4.4	35	Ovais Khan	Prateik	Ishita Raj	
4	NaN	NaN	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	

	Actor 3
0	Rajendra Bhatia
1	Arvind Jangid
2	Roy Angana
3	Siddhant Kapoor
4	Antara Mali

```
[215]: df.tail()
```

```
[215]:
```

	Name	Year	Duration	Genre	Rating	Votes	\
15504	Zulm Ko Jala Doonga	(1988)	NaN	Action	4.6	11	
15505	Zulmi	(1999)	129 min	Action, Drama	4.5	655	
15506	Zulmi Raj	(2005)	NaN	Action	NaN	NaN	
15507	Zulmi Shikari	(1988)	NaN	Action	NaN	NaN	
15508	Zulm-O-Sitam	(1998)	130 min	Action, Drama	6.2	20	

	Director	Actor 1	Actor 2	Actor 3
--	----------	---------	---------	---------

15504	Mahendra Shah	Naseeruddin Shah	Sumeet Saigal	Suparna Anand
15505	Kuku Kohli	Akshay Kumar	Twinkle Khanna	Aruna Irani
15506	Kiran Thej	Sangeeta Tiwari	NaN	NaN
15507	NaN	NaN	NaN	NaN
15508	K.C. Bokadia	Dharmendra	Jaya Prada	Arjun Sarja

```
[216]: df.shape
```

```
[216]: (15509, 10)
```

```
[217]: #Check for missing values
df.isna().sum()
```

```
[217]: Name          0
Year          528
Duration      8269
Genre        1877
Rating       7590
Votes        7589
Director      525
Actor 1       1617
Actor 2       2384
Actor 3       3144
dtype: int64
```

```
[218]: #Drop less impact column
df.drop(['Votes'],axis=1,inplace=True)
```

```
[219]: df['Year']=df['Year'].fillna(df['Year'].mode()[0])
df['Duration']=df['Duration'].fillna(df['Duration'].mode()[0])
df['Genre']=df['Genre'].fillna(df['Genre'].mode()[0])
df['Rating']=df['Rating'].fillna(df['Rating'].mean())
df['Director']=df['Director'].fillna(df['Director'].mode()[0])
df['Actor 1']=df['Actor 1'].fillna(df['Actor 1'].mode()[0])
df['Actor 2']=df['Actor 2'].fillna(df['Actor 2'].mode()[0])
df['Actor 3']=df['Actor 3'].fillna(df['Actor 3'].mode()[0])
```

```
[220]: df.isna().sum()
```

```
[220]: Name          0
Year          0
Duration       0
Genre         0
Rating        0
Director      0
Actor 1       0
Actor 2       0
```

```
Actor 3      0
dtype: int64
```

```
[221]: #Find unique values
df['Name'].unique()
```

```
[221]: array([' ', '#Gadhvi (He thought he was Gandhi)', '#Homecoming', ...,
        'Zulmi Raj', 'Zulmi Shikari', 'Zulm-0-Sitam'], dtype=object)
```

```
[222]: df = df.dropna(subset=['Name'])
df.shape
```

```
[222]: (15509, 9)
```

```
[223]: df['Year'].unique()
```

```
[223]: array(['(2019)', '(2021)', '(2010)', '(1997)', '(2005)', '(2008)',
        '(2012)', '(2014)', '(2004)', '(2016)', '(1991)', '(1990)',
        '(2018)', '(1987)', '(1948)', '(1958)', '(2017)', '(2020)',
        '(2009)', '(2002)', '(1993)', '(1946)', '(1994)', '(2007)',
        '(2013)', '(2003)', '(1998)', '(1979)', '(1951)', '(1956)',
        '(1974)', '(2015)', '(2006)', '(1981)', '(1985)', '(2011)',
        '(2001)', '(1967)', '(1988)', '(1995)', '(1959)', '(1996)',
        '(1970)', '(1976)', '(2000)', '(1999)', '(1973)', '(1968)',
        '(1943)', '(1953)', '(1986)', '(1983)', '(1989)', '(1982)',
        '(1977)', '(1957)', '(1950)', '(1992)', '(1969)', '(1975)',
        '(1947)', '(1972)', '(1971)', '(1935)', '(1978)', '(1960)',
        '(1944)', '(1963)', '(1940)', '(1984)', '(1934)', '(1955)',
        '(1936)', '(1980)', '(1966)', '(1949)', '(1962)', '(1964)',
        '(1952)', '(1933)', '(1942)', '(1939)', '(1954)', '(1945)',
        '(1961)', '(1965)', '(1938)', '(1941)', '(1931)', '(1937)',
        '(2022)', '(1932)', '(1923)', '(1915)', '(1928)', '(1922)',
        '(1917)', '(1913)', '(1930)', '(1926)', '(1914)', '(1924)'],
        dtype=object)
```

```
[224]: # Remove parentheses and convert to integer
df['Year'] = df['Year'].str.strip('()').astype(int)
```

```
[225]: df.dtypes
```

```
[225]: Name      object
      Year      int64
      Duration object
      Genre    object
      Rating    float64
      Director object
      Actor 1   object
```

```
Actor 2      object
Actor 3      object
dtype: object
```

```
[226]: df['Duration'].unique()
```

```
[226]: array(['120 min', '109 min', '90 min', '110 min', '105 min', '147 min',
            '142 min', '59 min', '82 min', '116 min', '96 min', '161 min',
            '166 min', '102 min', '87 min', '132 min', '66 min', '146 min',
            '112 min', '168 min', '158 min', '126 min', '94 min', '138 min',
            '124 min', '144 min', '157 min', '136 min', '107 min', '113 min',
            '80 min', '122 min', '149 min', '148 min', '130 min', '121 min',
            '188 min', '115 min', '103 min', '114 min', '170 min', '100 min',
            '99 min', '140 min', '128 min', '93 min', '125 min', '145 min',
            '75 min', '111 min', '134 min', '85 min', '104 min', '92 min',
            '137 min', '127 min', '150 min', '119 min', '135 min', '86 min',
            '76 min', '70 min', '72 min', '151 min', '95 min', '52 min',
            '89 min', '143 min', '177 min', '117 min', '123 min', '154 min',
            '88 min', '175 min', '153 min', '78 min', '139 min', '133 min',
            '101 min', '180 min', '60 min', '46 min', '164 min', '162 min',
            '171 min', '160 min', '152 min', '62 min', '163 min', '165 min',
            '141 min', '210 min', '129 min', '156 min', '240 min', '172 min',
            '155 min', '118 min', '167 min', '106 min', '193 min', '57 min',
            '108 min', '45 min', '195 min', '174 min', '81 min', '178 min',
            '58 min', '184 min', '97 min', '98 min', '131 min', '176 min',
            '169 min', '77 min', '91 min', '84 min', '173 min', '74 min',
            '67 min', '181 min', '300 min', '79 min', '65 min', '48 min',
            '183 min', '159 min', '83 min', '68 min', '49 min', '201 min',
            '64 min', '186 min', '50 min', '69 min', '207 min', '55 min',
            '61 min', '185 min', '187 min', '216 min', '63 min', '54 min',
            '198 min', '51 min', '71 min', '73 min', '218 min', '191 min',
            '321 min', '199 min', '53 min', '56 min', '179 min', '47 min',
            '206 min', '190 min', '211 min', '247 min', '213 min', '223 min',
            '2 min', '189 min', '224 min', '202 min', '255 min', '197 min',
            '182 min', '214 min', '208 min', '21 min', '200 min', '192 min',
            '37 min', '261 min', '238 min', '204 min', '235 min', '298 min',
            '217 min', '250 min'], dtype=object)
```

```
[227]: #Replace min from each unique values and convert integer
df['Duration']=df['Duration'].str.replace(' min',' ').astype(int)
```

```
[228]: df['Genre'].unique()
```

```
[228]: array(['Drama', 'Drama, Musical', 'Comedy, Romance',
            'Comedy, Drama, Musical', 'Drama, Romance, War', 'Documentary',
            'Horror, Mystery, Thriller', 'Action, Crime, Thriller', 'Horror',
            'Horror, Romance, Thriller', 'Comedy, Drama, Romance', 'Thriller',
```

'Comedy, Drama', 'Comedy, Drama, Fantasy', 'Comedy, Drama, Family',  
 'Crime, Drama, Mystery', 'Horror, Thriller', 'Biography',  
 'Comedy, Horror', 'Action', 'Drama, Horror, Mystery', 'Comedy',  
 'Action, Thriller', 'Drama, History', 'Drama, History, Sport',  
 'Horror, Mystery, Romance', 'Horror, Mystery',  
 'Drama, Horror, Romance', 'Action, Drama, History',  
 'Action, Drama, War', 'Comedy, Family',  
 'Adventure, Horror, Mystery', 'Action, Sci-Fi',  
 'Crime, Mystery, Thriller', 'War', 'Sport',  
 'Biography, Drama, History', 'Horror, Romance', 'Crime, Drama',  
 'Drama, Romance', 'Adventure, Drama', 'Comedy, Mystery, Thriller',  
 'Action, Crime, Drama', 'Crime, Thriller',  
 'Horror, Sci-Fi, Thriller', 'Crime, Drama, Thriller',  
 'Drama, Mystery, Thriller', 'Drama, Sport',  
 'Drama, Family, Musical', 'Action, Comedy', 'Comedy, Thriller',  
 'Action, Adventure, Fantasy', 'Drama, Romance, Thriller',  
 'Action, Drama', 'Drama, Horror, Musical',  
 'Action, Biography, Drama', 'Adventure, Comedy, Drama', 'Mystery',  
 'Action, Fantasy, Mystery', 'Adventure, Drama, Mystery',  
 'Mystery, Thriller', 'Adventure', 'Drama, Musical, Thriller',  
 'Comedy, Crime, Drama', 'Musical, Romance', 'Documentary, Music',  
 'Documentary, History, Music', 'Drama, Fantasy, Mystery',  
 'Drama, Family, Sport', 'Drama, Thriller',  
 'Documentary, Biography', 'Action, Adventure, Comedy', 'Romance',  
 'Comedy, Drama, Music', 'Comedy, Horror, Mystery', 'Musical',  
 'Musical, Romance, Drama', 'Family, Romance',  
 'Action, Sci-Fi, Thriller', 'Action, Drama, Romance',  
 'Mystery, Romance', 'Fantasy', 'Family', 'Drama, Family',  
 'Action, Comedy, Drama', 'Action, Drama, Thriller',  
 'Drama, Horror, Thriller', 'Drama, Musical, Romance',  
 'Comedy, Sci-Fi', 'Action, Romance', 'Action, Crime',  
 'Action, Drama, Crime', 'Drama, Family, Music',  
 'Action, Mystery, Thriller', 'Action, Drama, Family',  
 'Action, Mystery', 'Drama, History, Romance',  
 'Crime, Drama, Music', 'Sci-Fi', 'Animation',  
 'Crime, Mystery, Romance', 'Action, Adventure, Romance',  
 'Music, Romance', 'Action, Comedy, Crime',  
 'Comedy, Family, Fantasy', 'Romance, Drama',  
 'Drama, Family, Romance', 'Romance, Drama, Family',  
 'Musical, Romance, Thriller', 'Family, Musical, Romance',  
 'Action, Drama, Fantasy', 'Family, Drama', 'Crime, Drama, Romance',  
 'Musical, Drama, Romance', 'Drama, Music, Musical',  
 'Drama, Mystery', 'Adventure, Comedy, Romance',  
 'Crime, Drama, Horror', 'Family, Music, Musical',  
 'Action, Musical, Thriller', 'Action, Romance, Thriller',  
 'Romance, Thriller', 'Drama, Music', 'Crime, Drama, Musical',  
 'Action, Crime, Mystery', 'Action, Adventure, Thriller',

'Comedy, Romance, Sci-Fi', 'Crime', 'Action, Drama, Mystery',  
 'Action, Comedy, Thriller', 'Biography, Drama',  
 'Action, Comedy, Fantasy', 'Drama, Family, Horror',  
 'Action, Adventure, Family', 'Documentary, Biography, Musical',  
 'Action, Drama, Musical', 'Adventure, Thriller', 'Crime, Mystery',  
 'Drama, Crime', 'Drama, Fantasy, Romance',  
 'Comedy, Romance, Thriller', 'Musical, Comedy, Drama',  
 'Biography, History, War', 'Action, Comedy, Romance',  
 'Drama, History, Musical', 'Action, Crime, Horror',  
 'Adventure, Fantasy', 'Adventure, Drama, Fantasy',  
 'Adventure, Fantasy, Romance', 'Action, Adventure, Drama',  
 'Action, Adventure', 'Comedy, Crime', 'Crime, Drama, Fantasy',  
 'Adventure, Drama, Romance', 'History', 'Drama, Fantasy, Thriller',  
 'Musical, Fantasy', 'Documentary, Thriller',  
 'Mystery, Romance, Musical', 'Family, Drama, Romance',  
 'History, Musical, Romance', 'Musical, Drama, Crime',  
 'Adventure, Crime, Romance', 'Musical, Thriller, Mystery',  
 'Drama, Comedy', 'Biography, Drama, Romance', 'Biography, Music',  
 'Biography, Drama, Music', 'Drama, Sci-Fi',  
 'Drama, Family, Thriller', 'Comedy, Musical, Romance',  
 'Drama, Family, Comedy', 'Action, Thriller, Romance',  
 'Animation, Adventure', 'Action, Crime, Musical',  
 'Action, Crime, Romance', 'Animation, Action, Adventure',  
 'Action, Drama, Sport', 'Comedy, History', 'Documentary, History',  
 'Drama, Comedy, Family', 'Action, Adventure, Crime',  
 'Documentary, Biography, Music', 'Comedy, Musical',  
 'Biography, Crime, Thriller', 'Adventure, Mystery, Thriller',  
 'Biography, Drama, Sport', 'Action, Comedy, Musical',  
 'Mystery, Romance, Thriller', 'Action, Adventure, Musical',  
 'Crime, Musical, Mystery', 'Action, Thriller, Crime',  
 'Adventure, Comedy, Crime', 'Comedy, Horror, Musical',  
 'Adventure, Family', 'Family, Thriller', 'Drama, Action, Crime',  
 'Drama, War', 'Action, Drama, Adventure',  
 'Adventure, Fantasy, History', 'Fantasy, Musical',  
 'Comedy, Drama, Thriller', 'Drama, Fantasy', 'Musical, Drama',  
 'Action, Drama, Horror', 'Biography, Crime, Drama',  
 'Action, Drama, Music', 'Adventure, Drama, Family',  
 'Drama, Romance, Musical', 'Comedy, Musical, Drama',  
 'Adventure, Comedy, Musical', 'Crime, Drama, Family',  
 'Thriller, Musical, Mystery', 'Documentary, Adventure, Crime',  
 'Drama, Action, Horror', 'Adventure, Crime, Drama',  
 'Documentary, Biography, Sport', 'Crime, Fantasy, Mystery',  
 'Documentary, Biography, Drama', 'Action, Fantasy, Thriller',  
 'Adventure, Drama, History', 'Animation, Drama, History',  
 'Comedy, Horror, Thriller', 'Drama, Family, History',  
 'Animation, History', 'Biography, Drama, Musical', 'Music',  
 'Family, Comedy', 'Adventure, Mystery', 'Family, Fantasy',

'Documentary, History, News', 'Drama, Mystery, Romance',  
 'Comedy, Fantasy', 'Action, Crime, Family',  
 'Drama, Musical, Mystery', 'Action, Thriller, Mystery',  
 'Drama, Family, Fantasy', 'Action, Family',  
 'Action, Adventure, Mystery', 'Horror, Fantasy', 'Comedy, Action',  
 'Adventure, Romance', 'Drama, Adventure',  
 'Animation, Drama, Romance', 'Comedy, Crime, Romance',  
 'Adventure, Comedy', 'Comedy, Drama, Sport',  
 'Documentary, Crime, History', 'Musical, Mystery, Drama',  
 'Adventure, Drama, Sci-Fi', 'Action, Romance, Western',  
 'Comedy, Fantasy, Romance', 'Animation, Action, Comedy',  
 'Drama, Fantasy, Sci-Fi', 'Drama, Horror', 'Family, Drama, Comedy',  
 'Action, Adventure, History', 'Comedy, Family, Romance',  
 'Biography, History', 'Animation, Family',  
 'Drama, Fantasy, History', 'Animation, Adventure, Fantasy',  
 'Adventure, Comedy, Family', 'Drama, History, War',  
 'Animation, Drama, Fantasy', 'Action, Musical, Romance',  
 'Crime, Action, Drama', 'Comedy, Romance, Musical',  
 'Fantasy, Drama', 'Musical, Action, Crime', 'Documentary, Drama',  
 'Action, Horror, Thriller', 'Action, Horror, Sci-Fi',  
 'Mystery, Sci-Fi, Thriller', 'Biography, Family',  
 'Drama, Action, Comedy', 'Drama, Music, Romance',  
 'Action, Biography, Crime', 'Adventure, Drama, Musical',  
 'Family, Music, Romance', 'Fantasy, Mystery, Romance',  
 'Drama, Crime, Family', 'Drama, Family, Action',  
 'Romance, Comedy, Drama', 'Animation, Adventure, Comedy',  
 'Sci-Fi, Thriller', 'Romance, Family, Drama',  
 'Action, Family, Thriller', 'Adventure, Crime, Thriller',  
 'Drama, Romance, Sport', 'Comedy, Crime, Mystery',  
 'Adventure, Comedy, Mystery', 'Action, Fantasy', 'Comedy, Mystery',  
 'Animation, Adventure, Family', 'Adventure, Drama, Music',  
 'Biography, Drama, War', 'Documentary, Comedy, Drama',  
 'Musical, Drama, Family', 'Animation, Comedy, Drama',  
 'Fantasy, Musical, Drama', 'Adventure, Crime, Mystery',  
 'Comedy, Drama, Mystery', 'Documentary, News',  
 'Drama, Musical, Family', 'Action, Romance, Drama',  
 'Comedy, Crime, Thriller', 'Action, Musical', 'Action, History',  
 'Action, Comedy, Mystery', 'Drama, Family, Mystery',  
 'Adventure, Drama, Thriller', 'Documentary, Reality-TV',  
 'Action, Fantasy, Horror', 'Drama, History, Thriller',  
 'Documentary, Family', 'Documentary, Biography, Family',  
 'Comedy, Sport', 'Animation, Comedy, Family',  
 'Crime, Romance, Thriller', 'Comedy, Musical, Action',  
 'Action, Mystery, Sci-Fi', 'Comedy, Crime, Musical',  
 'Drama, Adventure, Action', 'History, Romance', 'Reality-TV',  
 'Fantasy, History', 'Family, Drama, Thriller',  
 'Musical, Mystery, Thriller', 'Musical, Comedy, Romance',



'Musical, Action, Drama', 'Action, Musical, War',  
 'Romance, Comedy', 'Horror, Crime, Thriller',  
 'Crime, Drama, History', 'Comedy, Drama, Horror',  
 'Crime, Horror, Thriller', 'Animation, Comedy',  
 'Romance, Action, Crime', 'Musical, Thriller',  
 'Action, Romance, Comedy', 'Comedy, Family, Musical',  
 'Horror, Drama, Mystery', 'Thriller, Mystery, Family',  
 'Comedy, Drama, Sci-Fi', 'Documentary, Adventure',  
 'Documentary, Biography, Crime', 'Musical, Action',  
 'Musical, Mystery', 'Action, Crime, Sci-Fi',  
 'Action, Horror, Mystery', 'Fantasy, Horror',  
 'Adventure, Family, Fantasy', 'Fantasy, Sci-Fi', 'Comedy, War',  
 'Romance, Action, Drama', 'Musical, Family, Romance',  
 'Romance, Drama, Action', 'Family, Comedy, Drama',  
 'Comedy, Music, Romance', 'Comedy, Family, Sci-Fi',  
 'Action, Drama, Western', 'Adventure, Romance, Thriller',  
 'Biography, Comedy, Drama', 'Action, Mystery, Romance',  
 'Romance, Sport', 'Crime, Romance', 'Action, Thriller, Western',  
 'Crime, Musical, Romance', 'Romance, Thriller, Mystery',  
 'Drama, Crime, Mystery', 'Biography, Drama, Family',  
 'Action, Family, Mystery', 'Comedy, Mystery, Romance',  
 'Drama, Thriller, Action', 'Documentary, Short',  
 'Documentary, Western', 'Musical, Family, Drama',  
 'Action, Family, Musical', 'Animation, Family, Musical',  
 'Drama, Fantasy, Horror', 'Action, Adventure, Sci-Fi',  
 'Drama, Action, Musical', 'Drama, Musical, Sport',  
 'Action, Comedy, Horror', 'Drama, Fantasy, Musical',  
 'Action, Fantasy, Musical', 'Animation, Action', 'Comedy, Music',  
 'Documentary, Drama, Romance', 'Drama, Music, Thriller',  
 'Fantasy, Musical, Mystery', 'Drama, Fantasy, War', 'Action, War',  
 'Action, Adventure, War', 'Horror, Musical',  
 'Fantasy, Mystery, Thriller', 'Adventure, Biography, Drama',  
 'Family, Romance, Sci-Fi', 'Drama, Romance, Family',  
 'Animation, Adventure, Drama', 'Family, Romance, Drama',  
 'Animation, Action, Sci-Fi', 'Adventure, Comedy, Fantasy',  
 'Comedy, Crime, Family', 'Horror, Musical, Thriller',  
 'Biography, Drama, Thriller', 'Drama, Western',  
 'Romance, Sci-Fi, Thriller', 'Comedy, Musical, Family',  
 'Comedy, Horror, Romance', 'Thriller, Action',  
 'Fantasy, Thriller, Action', 'Fantasy, Romance',  
 'Action, Drama, Comedy', 'Family, Fantasy, Romance',  
 'Comedy, Crime, Horror', 'Horror, Mystery, Sci-Fi',  
 'Animation, Action, Drama', 'Family, Mystery',  
 'Adventure, Biography, History', 'Fantasy, Horror, Mystery',  
 'Family, Musical', 'Drama, Family, Adventure',  
 'Crime, Horror, Mystery', 'Documentary, Drama, Fantasy',  
 'Action, Adventure, Biography', 'Biography, History, Thriller',

```
'Action, Family, Drama', 'Documentary, Drama, Sport',
'Thriller, Mystery', 'Musical, Drama, Comedy',
'Documentary, History, War', 'Adventure, Horror, Thriller',
'Action, Adventure, Horror', 'Action, Crime, War',
'Adventure, Musical, Romance', 'Action, Fantasy, Sci-Fi',
'Drama, Comedy, Action', 'Documentary, Sport',
'Documentary, Adventure, Music', 'Drama, Action, Family',
'Adventure, History, Thriller', 'Adventure, Horror, Romance',
'Adventure, Crime, Horror', 'Mystery, Musical, Romance',
'Action, Crime, History', 'Documentary, Musical',
'Adventure, Fantasy, Musical', 'Documentary, Family, History',
'Documentary, Drama, Family', 'Drama, Mystery, Sci-Fi',
'Animation, Drama, Musical', 'Drama, History, Mystery',
'Drama, Sport, Thriller', 'Action, Crime, Fantasy',
'Comedy, Musical, Mystery', 'Romance, Musical, Action',
'Musical, Drama, Fantasy', 'Animation, Family, History',
'Action, Drama, News', 'Romance, Musical, Comedy',
'Adventure, Fantasy, Horror', 'Adventure, History',
'Comedy, Drama, History', 'Mystery, Sci-Fi',
'Action, Thriller, War', 'Documentary, Drama, News',
'Documentary, Crime, Mystery', 'Adventure, Horror',
'Animation, Drama, Adventure', 'Crime, Horror, Romance',
'Documentary, Adventure, Drama', 'Documentary, Biography, History',
'Fantasy, Horror, Romance', 'Comedy, Fantasy, Musical',
'Crime, Musical, Thriller', 'Documentary, War',
'Action, Comedy, War', 'Crime, Drama, Sport',
'Musical, Adventure, Drama', 'Horror, Romance, Sci-Fi',
'Musical, Mystery, Romance', 'Romance, Musical, Drama',
'Adventure, Fantasy, Sci-Fi'], dtype=object)
```

```
[229]: df.columns
```

```
[229]: Index(['Name', 'Year', 'Duration', 'Genre', 'Rating', 'Director', 'Actor 1',
        'Actor 2', 'Actor 3'],
        dtype='object')
```

```
[230]: df['Director'].unique()
```

```
[230]: array(['J.S. Randhawa', 'Gaurav Bakshi', 'Soumyajit Majumdar', ...,
        'Mozes Singh', 'Ved Prakash', 'Kiran Thej'], dtype=object)
```

```
[231]: df['Actor 1'].unique()
```

```
[231]: array(['Manmauji', 'Rasika Dugal', 'Sayani Gupta', ..., 'Meghan Jadhav',
        'Roohi Berde', 'Sangeeta Tiwari'], dtype=object)
```

```
[232]: df['Actor 2'].unique()
```

```
[232]: array(['Birbal', 'Vivek Ghamande', 'Plabita Borthakur', ...,
            'Devan Sanjeev', 'Prince Daniel', 'Sarah Jane Dias'], dtype=object)
```

```
[233]: df['Actor 3'].unique()
```

```
[233]: array(['Rajendra Bhatia', 'Arvind Jangid', 'Roy Angana', ...,
            'Shatakshi Gupta', 'Valerie Agha', 'Suparna Anand'], dtype=object)
```

```
[234]: df['Rating'].unique()
```

```
[234]: array([[ 5.84162142,  7.         ,  4.4         ,  4.7         ,  7.4         ,
                5.6         ,  4.         ,  6.2         ,  5.9         ,  6.5         ,
                5.7         ,  6.3         ,  7.2         ,  6.6         ,  7.3         ,
                7.1         ,  6.9         ,  3.5         ,  5.         ,  4.5         ,
                6.4         ,  4.1         ,  4.8         ,  8.1         ,  5.5         ,
                6.8         ,  6.1         ,  7.7         ,  5.1         ,  7.6         ,
                3.1         ,  3.3         ,  7.8         ,  8.4         ,  5.2         ,
                4.3         ,  5.8         ,  4.6         ,  7.5         ,  6.7         ,
                3.6         ,  3.9         ,  5.4         ,  4.2         ,  5.3         ,
                3.4         ,  3.         ,  8.         ,  6.         ,  3.8         ,
                7.9         ,  2.7         ,  4.9         ,  2.4         ,  3.7         ,
                3.2         ,  2.5         ,  2.8         ,  2.6         ,  2.9         ,
                8.2         ,  8.7         ,  8.3         ,  9.3         ,  8.8         ,
                2.1         ,  2.3         ,  8.5         ,  8.6         ,  9.         ,
                9.6         ,  1.7         ,  9.1         ,  2.         ,  1.4         ,
                8.9         ,  1.9         ,  9.4         ,  9.7         ,  1.8         ,
                9.2         ,  1.6         , 10.         ,  2.2         ,  1.1         ]])
```

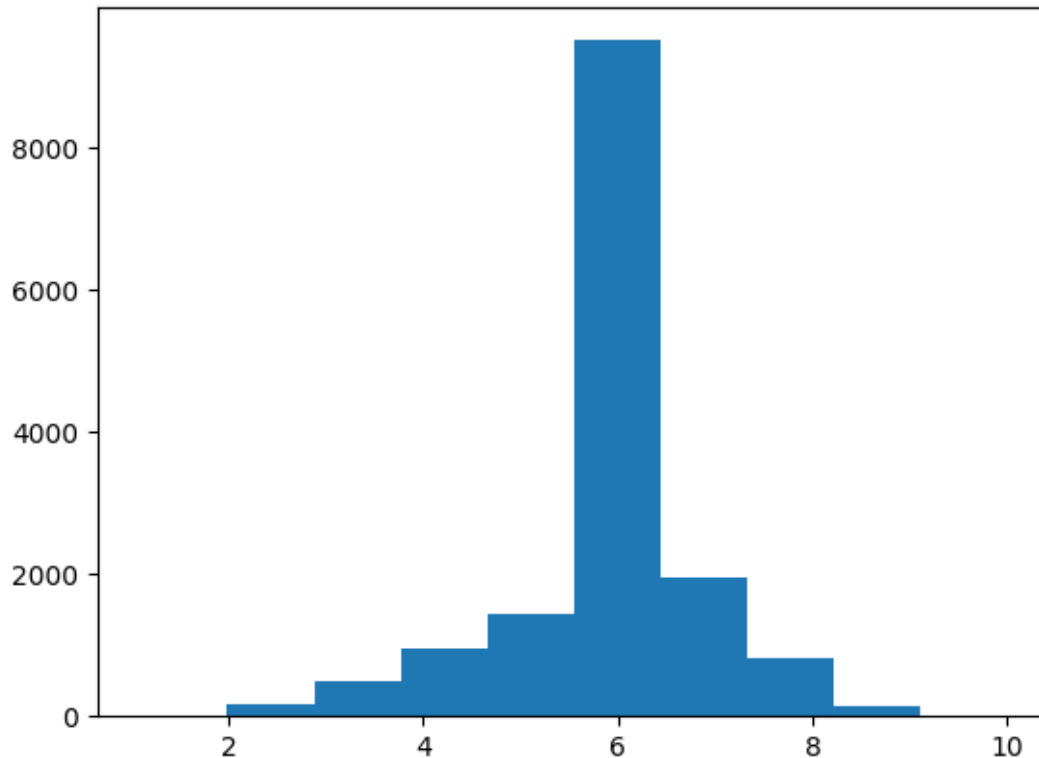
```
[235]: #Encoding the categorical data
from sklearn.preprocessing import LabelEncoder
encode=LabelEncoder()
cols=['Name', 'Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3']
for i in cols:
    df[i]=encode.fit_transform(df[i])
```

```
[236]: df.dtypes
```

```
[236]: Name          int64
      Year          int64
      Duration      int64
      Genre         int64
      Rating        float64
      Director      int64
      Actor 1       int64
      Actor 2       int64
      Actor 3       int64
      dtype: object
```

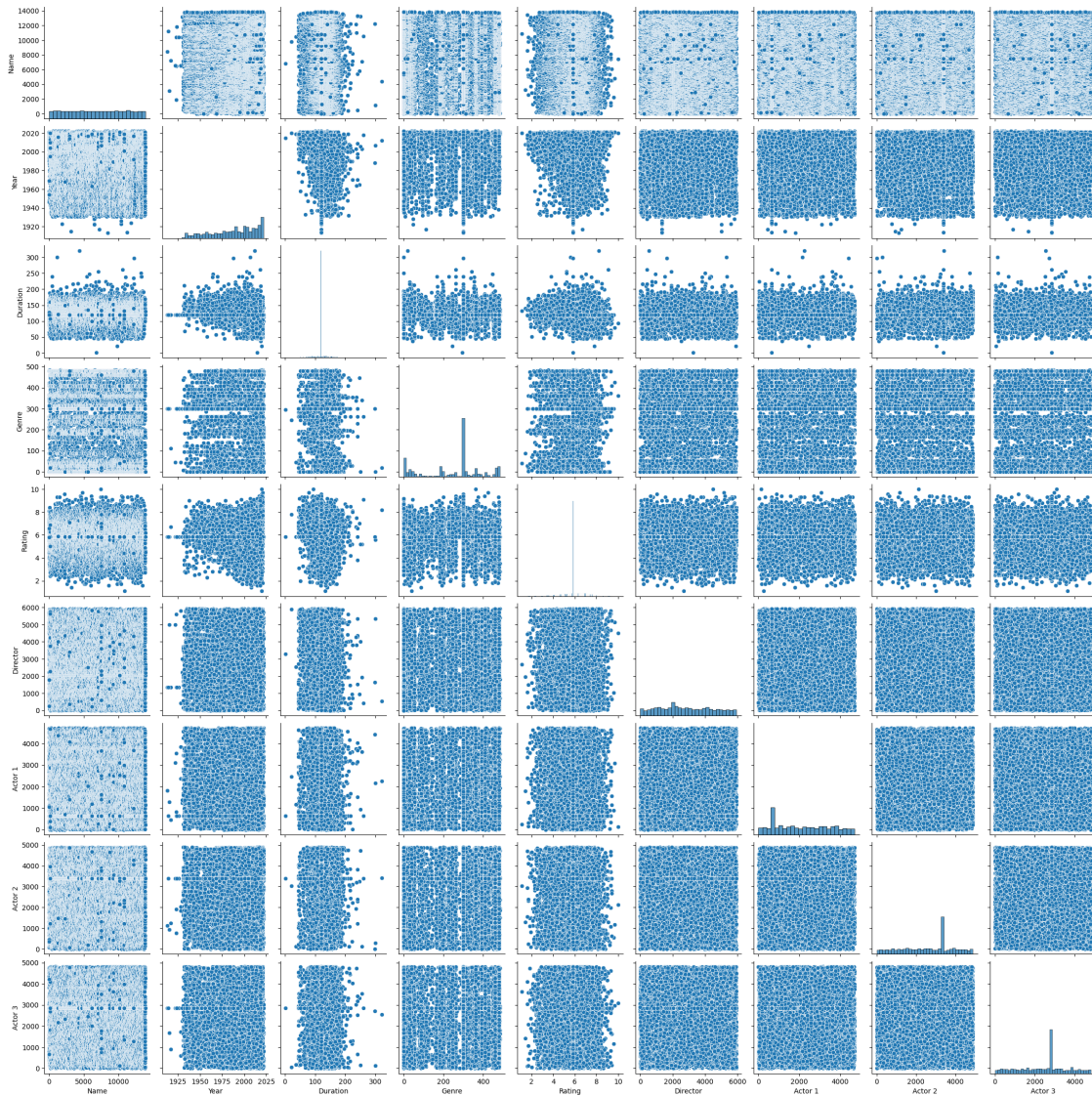
```
[237]: #Plot histogram of target column  
plt.hist(df['Rating'])
```

```
[237]: (array([ 20., 159., 493., 947., 1439., 9508., 1954., 817., 153.,  
          19.]),  
       array([ 1.1 ,  1.99,  2.88,  3.77,  4.66,  5.55,  6.44,  7.33,  8.22,  
          9.11, 10.  ]),  
       <BarContainer object of 10 artists>)
```



```
[238]: #Pair plot for numerical columns  
sns.pairplot(data=df)
```

```
[238]: <seaborn.axisgrid.PairGrid at 0x79382df8faf0>
```



```
[239]: #Pearson Correlation
corre=df.corr()
corre
```

```
[239]:
```

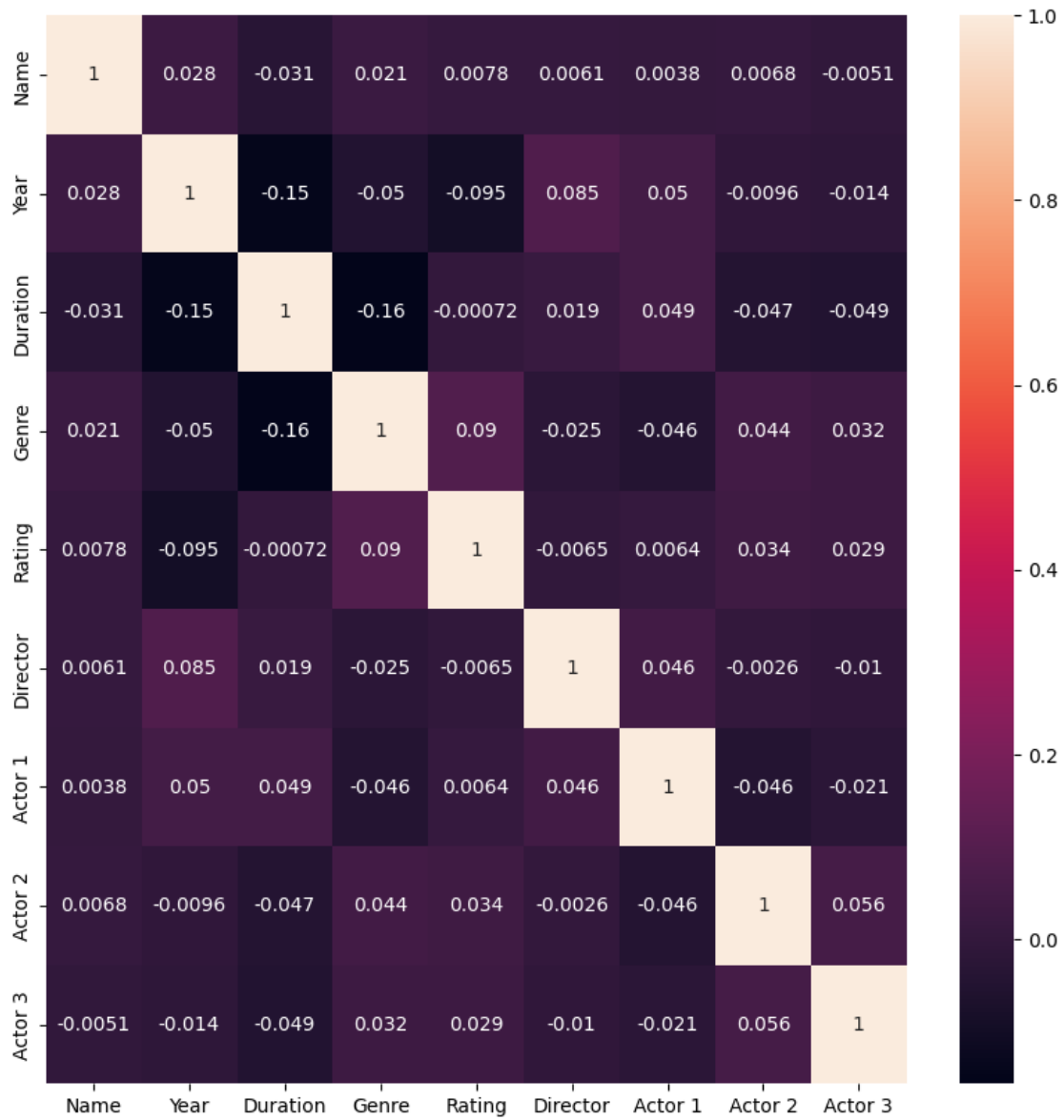
	Name	Year	Duration	Genre	Rating	Director	\
Name	1.000000	0.027765	-0.031178	0.020936	0.007825	0.006100	
Year	0.027765	1.000000	-0.146810	-0.050268	-0.095034	0.085407	
Duration	-0.031178	-0.146810	1.000000	-0.156592	-0.000717	0.019402	
Genre	0.020936	-0.050268	-0.156592	1.000000	0.089609	-0.024915	
Rating	0.007825	-0.095034	-0.000717	0.089609	1.000000	-0.006505	
Director	0.006100	0.085407	0.019402	-0.024915	-0.006505	1.000000	
Actor 1	0.003782	0.049890	0.048575	-0.045798	0.006368	0.046344	
Actor 2	0.006816	-0.009639	-0.046864	0.043876	0.034354	-0.002620	

Actor 3 -0.005118 -0.013917 -0.048712 0.032303 0.028900 -0.010193

	Actor 1	Actor 2	Actor 3
Name	0.003782	0.006816	-0.005118
Year	0.049890	-0.009639	-0.013917
Duration	0.048575	-0.046864	-0.048712
Genre	-0.045798	0.043876	0.032303
Rating	0.006368	0.034354	0.028900
Director	0.046344	-0.002620	-0.010193
Actor 1	1.000000	-0.045737	-0.021483
Actor 2	-0.045737	1.000000	0.055704
Actor 3	-0.021483	0.055704	1.000000

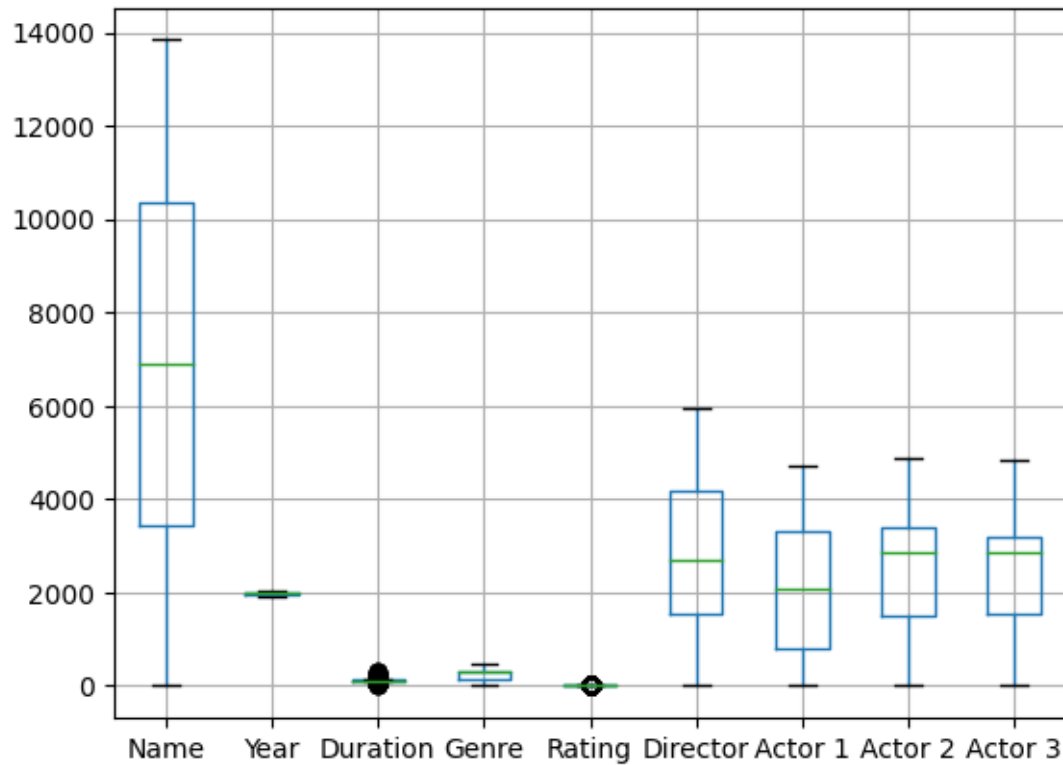
```
[240]: #heatmap
plt.figure(figsize=(10,10))
sns.heatmap(corre,annot=True)
```

[240]: <Axes: >



```
[241]: df.boxplot()
```

```
[241]: <Axes: >
```

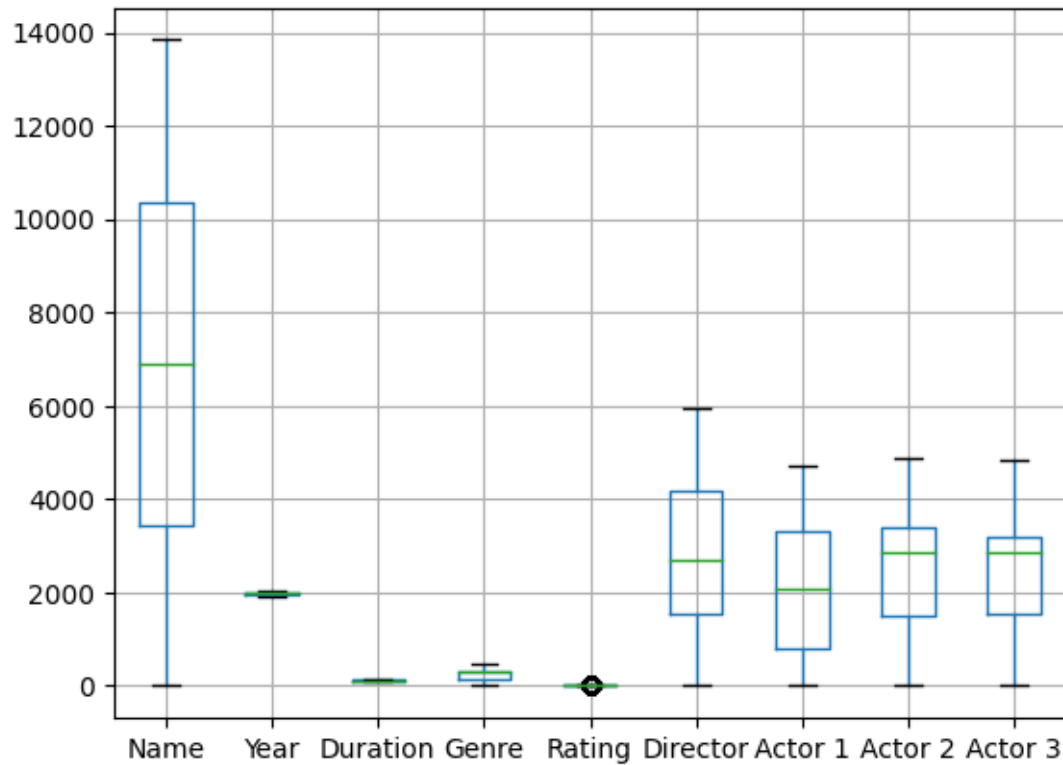


```
[242]: #Outlier Removal
columns=df.columns[df.columns!='Rating']
def out_rem(dfe,col):
    for i in col:
        q1=dfe[i].quantile(0.25)
        q3=dfe[i].quantile(0.75)
        IQR=q3-q1
        upper=q3+(1.5*IQR)
        lower=q1-(1.5*IQR)
        df[i]=df[i].clip(upper,lower)
    out_rem(df,columns)
```

```
[243]: df.boxplot()
```

```
[243]: <Axes: >
```





```
[244]: #x and y separation
x=df.drop(['Rating'],axis=1)
x
```

```
[244]:
```

	Name	Year	Duration	Genre	Director	Actor 1	Actor 2	Actor 3
0	0	2019	120.0	299	1926	2250	800	3108
1	1	2019	109.0	299	1548	3280	4790	527
2	2	2021	106.5	351	5123	3713	2866	3450
3	3	2019	110.0	228	3319	2917	1504	4020
4	7	2010	106.5	299	385	3112	3462	405
...	...	...	...	...	...	...	...	...
15504	13832	1988	120.0	0	2690	2586	4299	4262
15505	13834	1999	129.0	40	2499	227	4532	519
15506	13835	2005	120.0	0	2424	3609	3402	2847
15507	13836	1988	120.0	0	2034	636	3402	2847
15508	13833	1998	130.0	40	2195	1139	1589	490

[15509 rows x 8 columns]

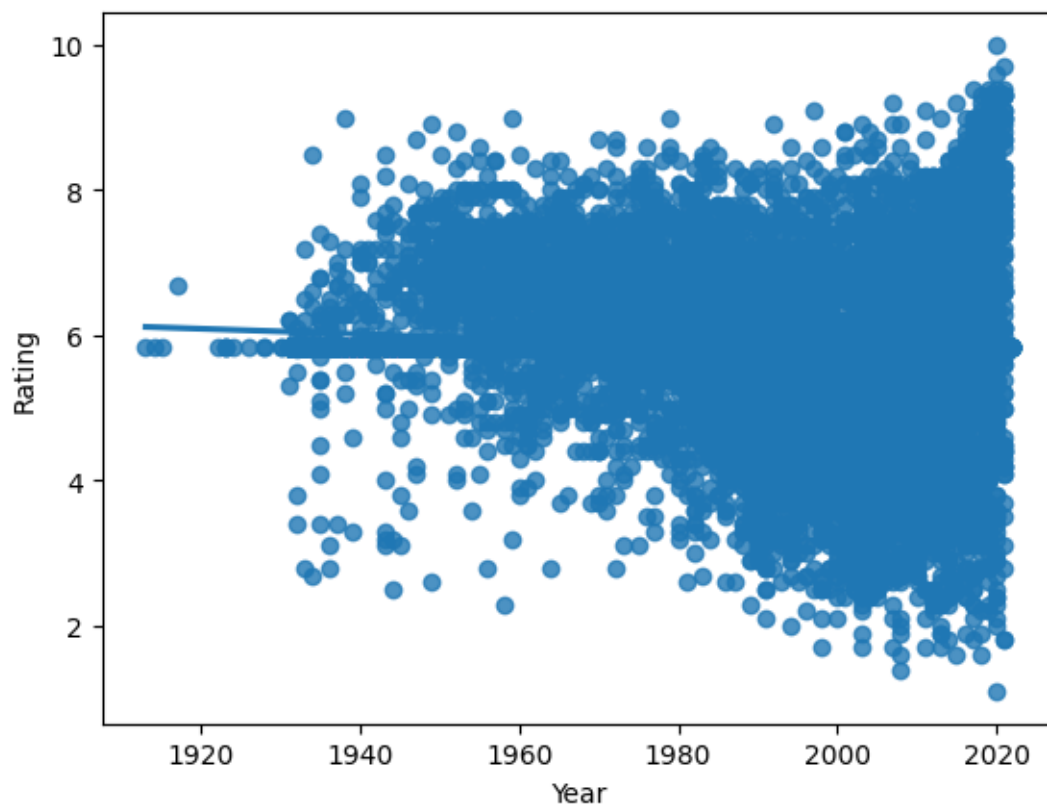
```
[245]: y=df['Rating']
y
```

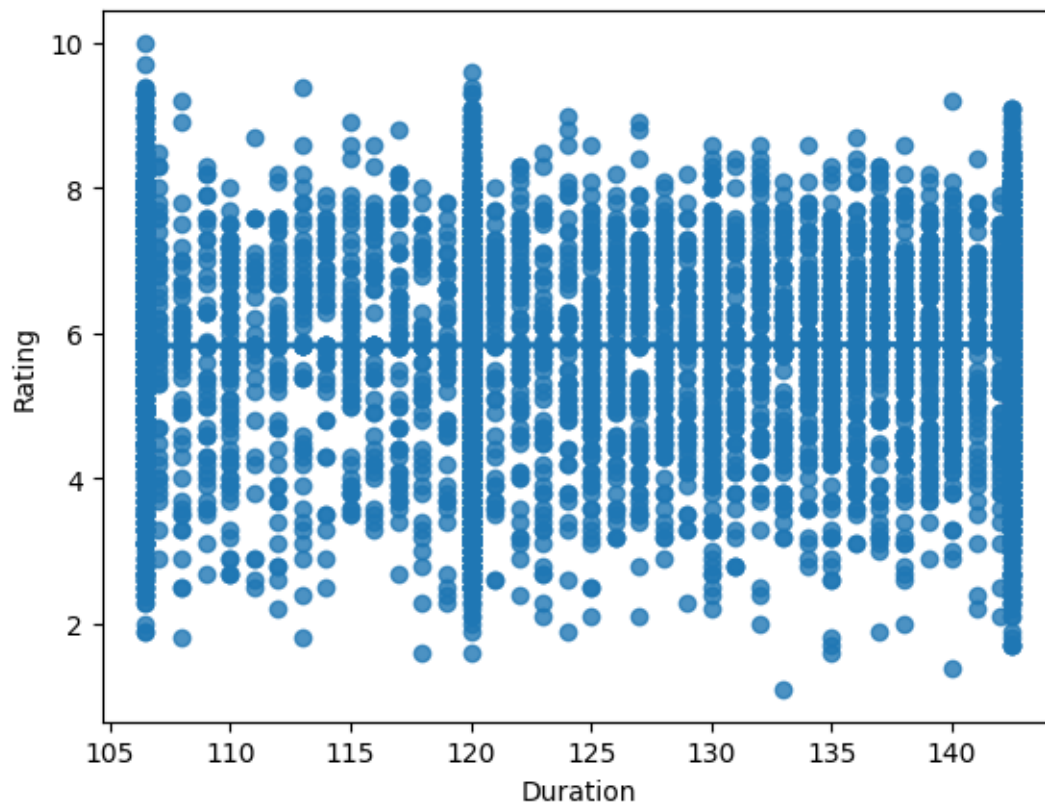
```
[245]: 0      5.841621
      1      7.000000
      2      5.841621
      3      4.400000
      4      5.841621
      ...
      15504    4.600000
      15505    4.500000
      15506    5.841621
      15507    5.841621
      15508    6.200000
      Name: Rating, Length: 15509, dtype: float64
```

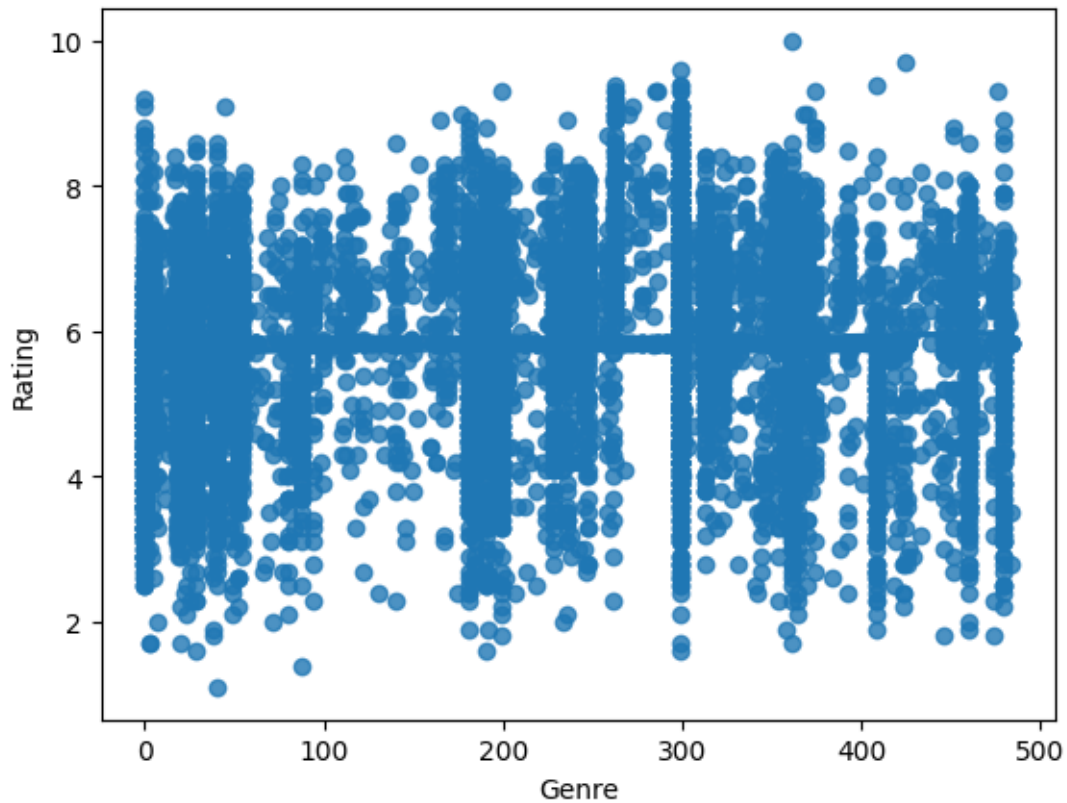
```
[246]: df.columns
```

```
[246]: Index(['Name', 'Year', 'Duration', 'Genre', 'Rating', 'Director', 'Actor 1',
        'Actor 2', 'Actor 3'],
        dtype='object')
```

```
[247]: # Draw Regression plot
column=['Year', 'Duration', 'Genre']
for i in column :
    sns.regplot(x=i,y=y,data=df)
    plt.xlabel(i)
    plt.ylabel('Rating')
    plt.show()
```







```
[248]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪30,random_state=42)
len(x_train)
```

[248]: 10856

```
[249]: #Model creation
from sklearn.linear_model import LinearRegression
mod1=LinearRegression()
from sklearn.ensemble import RandomForestRegressor
mod2=RandomForestRegressor(n_estimators=10,random_state=42)
from sklearn.metrics import mean_absolute_percentage_error,r2_score
lst=[mod1,mod2]
```

```
[256]: for i in lst:
    i.fit(x_train,y_train)
    print("model is",i)
    y_pred=i.predict(x_test)
    #print(y_pred)
    print("Percentage error",mean_absolute_percentage_error(y_test,y_pred))
```

```

model is LinearRegression()
Percentage error 0.1257267904060852
model is RandomForestRegressor(n_estimators=10, random_state=42)
Percentage error 0.12487925994201257

```

```

[251]: df1=pd.DataFrame({'Actualvalue':y_test,'Predictedvalue':y_pred,'Difference':
    ↳y_test-y_pred})
df1

```

```

[251]:

```

	Actualvalue	Predictedvalue	Difference
11115	5.841621	5.841621	0.000000
2921	6.800000	5.824162	0.975838
3463	5.841621	5.957459	-0.115838
2495	5.841621	5.841621	0.000000
15263	5.841621	5.841621	0.000000
...	...	...	...
11391	5.500000	5.689135	-0.189135
14647	5.841621	5.499135	0.342486
7238	5.841621	5.841621	0.000000
8747	5.841621	5.957459	-0.115838
12012	5.800000	5.224162	0.575838

[4653 rows x 3 columns]

```

[252]: print("slope is")
list(zip(x,mod1.coef_))

```

slope is

```

[252]: [('Name', 3.129636311826346e-06),
      ('Year', -0.003343296404757914),
      ('Duration', 0.0023265885513322096),
      ('Genre', 0.0006231679059181003),
      ('Director', -1.9895636170930847e-07),
      ('Actor 1', 8.327184417413337e-06),
      ('Actor 2', 2.178630595479577e-05),
      ('Actor 3', 2.3172974124414685e-05)]

```

```

[253]: print("Contant is",mod1.intercept_)

```

Contant is 11.900786860688742