```
TASK 2-MOVIE RATING PREDICTON WITH PYTHON


name:Nutan Santosh Bhilare
email id  -nutan10232@gmail.com
domain-data science
```

TASK 2-MOVIE RATING PREDICTON WITH PYTHON

name:Nutan Santosh Bhilare email id -nutan10232@gmail.com domain-data science

Importing libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import os
for dirname, _, filenames in os.walk('/content/archive (19).zip'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Importing dataset:

```
df= pd.read_csv('/content/archive (19).zip',encoding='latin1')
```

```
df.head()
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	(2019)	110 min	Comedy,	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant

Next steps: Generate code with df View recommended plots

```
df.describe()
```

	Rating
count	7919.000000
mean	5.841621
std	1.381777
min	1.100000
25%	4.900000
50%	6.000000
75%	6.800000
max	10.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      15509 non-null  object
 1   Year      14981 non-null  object
 2   Duration  7240 non-null   object
 3   Genre     13632 non-null  object
```

```
  4   Rating    7919 non-null   float64
  5   Votes     7920 non-null   object
  6   Director  14984 non-null  object
  7   Actor 1   13892 non-null  object
  8   Actor 2   13125 non-null  object
  9   Actor 3   12365 non-null  object
 dtypes: float64(1), object(9)
 memory usage: 1.2+ MB
```

```
df.shape
```

```
(15509, 10)
```

```
df.shape
```

```
(15509, 10)
```

```
df.Year.unique()
```

```
array([nan, '(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)
```

```
df.Rating.unique()
```

```
array([ nan,  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])
```

```
df.isnull().any()
```

```
Name        False
Year         True
Duration     True
Genre        True
Rating       True
Votes        True
Director     True
Actor 1      True
Actor 2      True
Actor 3      True
dtype: bool
```

```
df.duplicated().sum()
```

```
6
```

Data Exploration:

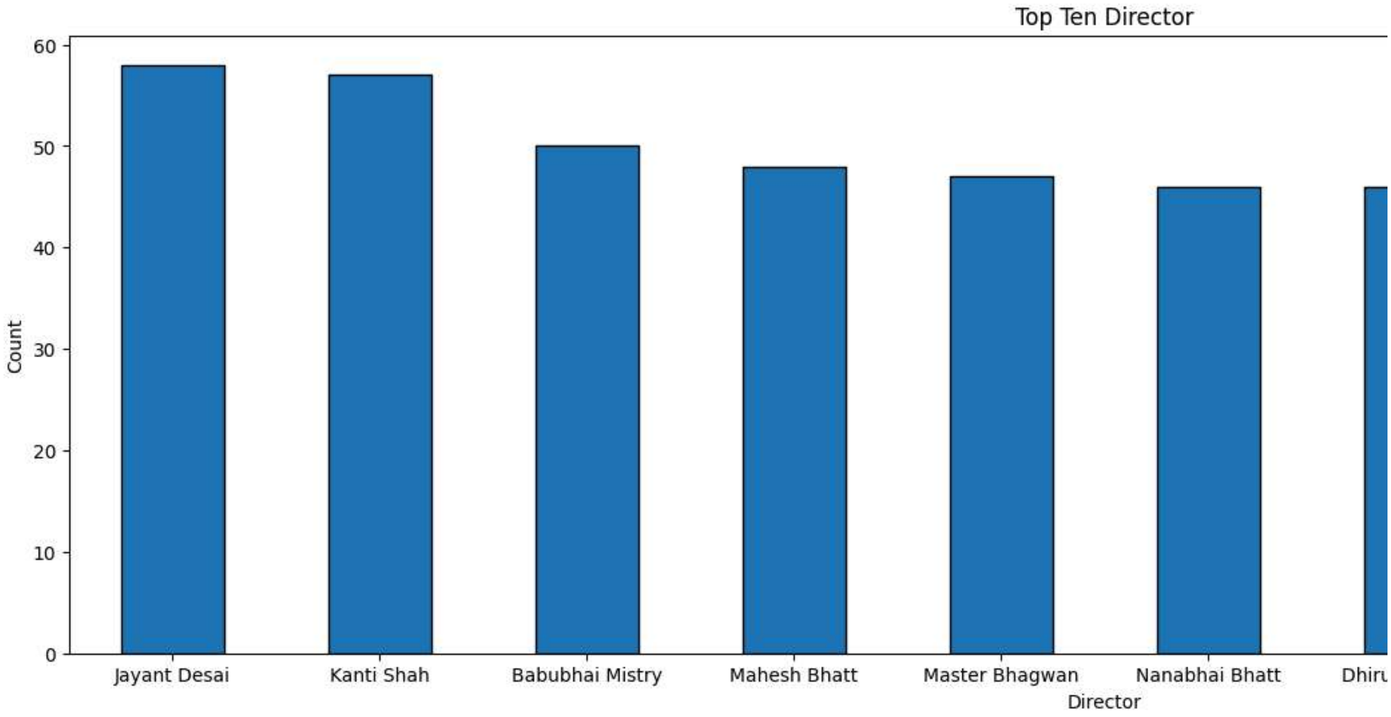
```
print('INFO:',"\n")
print(df.info(),"\n\n\n\n\n")
print('summary of the dataframe:',"\n",df.describe,"\n\n\n\n\n")
print('nunique:',"\n",df['Genre'].nunique(),"\n\n\n\n\n")
print('unique:',"\n",df['Year'].unique(),"\n\n\n\n\n")
print('Rating.unique:',"\n",df.Rating.unique(),"\n\n\n\n\n")
print('unique:',"\n",df['Duration'].unique(),"\n\n\n\n\n")
print("groupby(['Genre']':","\n",df.groupby(['Genre']).count(),"\n\n\n\n\n")
print("value_counts:","\n",df["Director"].value_counts().head(6),"\n\n\n\n\n")
print('isnull().any():',"\n",df.isnull().any(),"\n\n\n\n\n")
```

```
Action, Adventure, Crime        19    19        11      16     16        19
...                            ...   ...       ...     ...    ...       ...
Thriller, Action                 2     2         1       1      1         2
Thriller, Musical, Mystery       1     1         1       1      1         1
Thriller, Mystery                3     3         2       3      3         3
Thriller, Mystery, Family        1     1         1       1      1         1
War                              8     5         4       3      3         8

                              Actor 1  Actor 2  Actor 3
Genre
Action                           1207     1124     1005
Action, Adventure                  40       39       39
Action, Adventure, Biography        1        1        1
Action, Adventure, Comedy          42       42       42
Action, Adventure, Crime           19       19       19
...                               ...      ...      ...
Thriller, Action                    2        2        2
Thriller, Musical, Mystery          1        1        1
Thriller, Mystery                   3        3        3
Thriller, Mystery, Family           1        1        1
War                                 8        7        7

[485 rows x 9 columns]





value_counts:
 Jayant Desai       58
Kanti Shah         57
Babubhai Mistry    50
Mahesh Bhatt       48
Master Bhagwan     47
Nanabhai Bhatt     46
Name: Director, dtype: int64





isnull().any():
 Name        False
Year         True
Duration     True
Genre        True
Rating       True
Votes        True
Director     True
Actor 1      True
Actor 2      True
Actor 3      True
dtype: bool
```

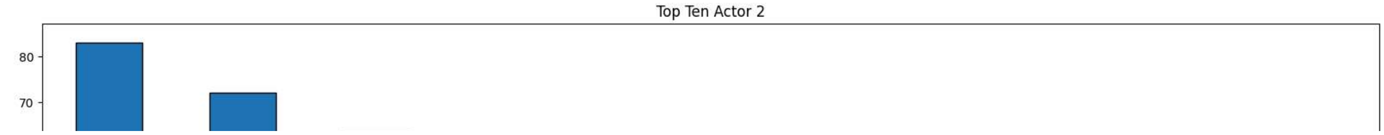
```
def TopTenPlot(column):
    global df
    df[column].value_counts().sort_values(ascending=False)[:10].plot(kind="bar", figsize=(20,6), edgecolor="k")
    plt.xticks(rotation=0)
    plt.title("Top Ten {}".format(column))
    plt.xlabel(column)
    plt.ylabel("Count")
    plt.show()
```

```
def Histogram(column):
    global df
    plt.figure(figsize=(20,6))
    plt.hist(df[column], edgecolor="k")
    plt.xticks(rotation=0)
    plt.title("Histogram of {}".format(column))
    plt.xlabel(column)
    plt.ylabel("Frequency")
    plt.show()


def Scatter(x, y, c=None):
    global df
    plt.figure(figsize=(20,6))
    plt.scatter(df[x], df[y], edgecolor="k", c=c)
    plt.xticks(rotation=0)
    plt.title("Scatter plot X:{} / Y:{}".format(x, y))
    plt.xlabel(x)
    plt.ylabel(y)
    plt.show()
```

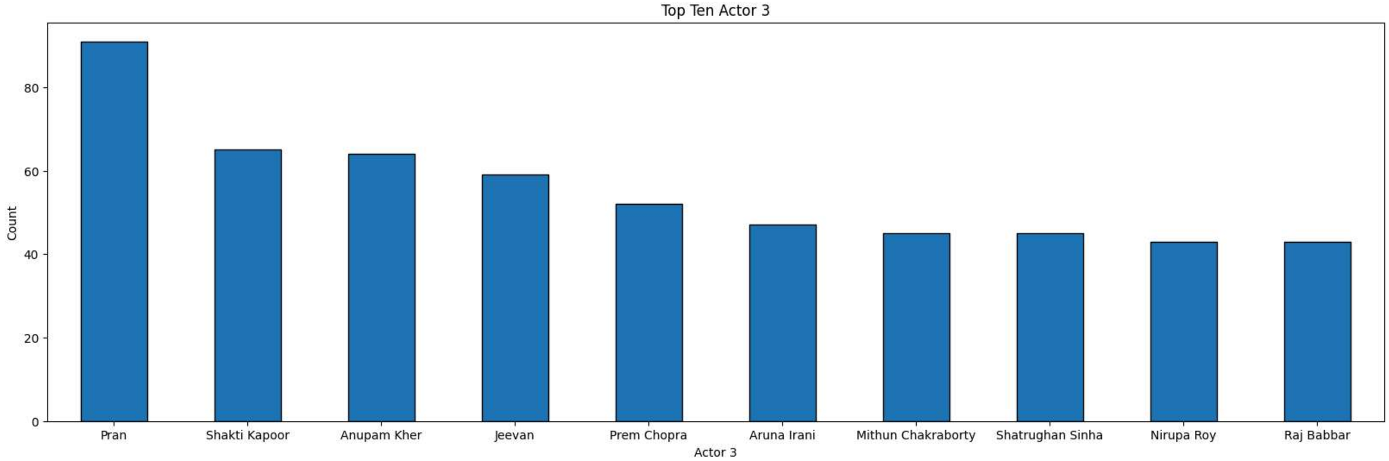
```
TopTenPlot("Director")
```



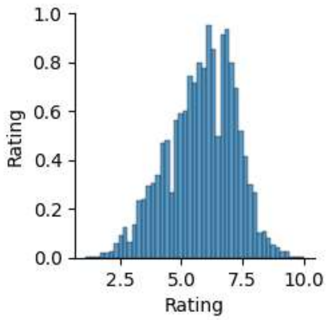
```
TopTenPlot("Actor 2")
```



```
TopTenPlot("Actor 3")
```



```
sns.pairplot(df)
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
correlation_matrix = df[numeric_columns].corr(method='spearman')
```


Rating
Rating