IMDB ANALYSIS (PYTHON PROJECT)

Importing all the required libraries and packages

```
import numpy as np                # linear algebra
import pandas as pd               # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
```

Youtube Video Link : https://youtu.be/Asm2_qgBK48

```
import warnings
warnings.filterwarnings('ignore')
```

To read the dataset to be used

```
data = pd.read_csv('/content/imdb_analysis.zip')        #for importing the imdb analysis dataset
```

Display the first 5 top rows

```
data.head()
```

1. Display Top 10 Rows of The Dataset

```
data.head(10)
```

2. Check Last 10 Rows of The Dataset

```
data.tail(10)
```

3. Find Shape of Our Dataset (Number of Rows And Number of Columns)

```
data.shape          # 1000 rows and 12 columns
```

To print the number of rows and columns

print('Number of Rows',data.shape[0]) # axis=0 means row function

print('Number of Columns',data.shape[1]) #axis=1 means column function

4. Getting Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

data.info()

5. Check Null Values In The Dataset

To check does the dataset has any null or missing values?

print("Any null or missing values present in our dataset?"," ",data.isnull().values.any())

To check if any values are present in the dataset(particularly in which column)

data.isnull().any()

To check the no of null values in the dataset for each particular column

data.isnull().sum() # .sum() is used so that we get the no. of nulls in the particular columns

Heatmap showing the rate of null values presence in the dataset

plt.figure(figsize=(10,5))

sns.heatmap(data.isnull())

plt.show()

Barplot for the visualisation of the null values present in the dataset

Plt.figure(figsize=(9,5))

```
plt.title("NULL VALUES IN THE DATASET")
plt.bar(data.columns,data.isnull().sum(),color="green")
plt.ylabel("No of null values")
plt.xlabel("Columns")
plt.xticks(rotation=78)
plt.grid()
plt.show()
```

To check the percentage of the missing or the null values in the dataset

```
per_missing=data.isnull().sum()/len(data)*100
print("THE PERCENTAG OF THE MISSING VALUES ARE:")
per_missing
```

6. Drop All The Missing Values

```
data = data.dropna(axis=0)          # .dropna() is used to drop the row which has null values....axis=0
```

Heatmap to check for any null or missing values in the dataset

```
sns.heatmap(data.isnull())
plt.show()
```

7. Check For Duplicate Data

```
dup_data=data.duplicated().any()
print("Are there any duplicated values in data?",dup_data)
data.head(10)
```

8. Get Overall Statistics About The DataFrame

```
data.describe()
```

To display the column names

```
print(data.columns)
```

9. Display Title of The Movie Having Runtime >= 180 Minutes

```
data[data['Runtime (Minutes)']>=180]['Title']
```

Or

```
data[data["Runtime (Minutes)"]>=180].Title
```

10. In Which Year There Was The Highest Voting?

```
data[data['Votes']==(data["Votes"].max())].Year
```

To see the record where the voting is the highest

```
data.sort_values(by='Votes',ascending=False).head(1)
```

Barplot to see the highest vote is in which year

```
plt.bar('Year','Votes',data=data)
plt.title("Votes By Year")
plt.grid()
plt.show()
```

11. In Which Year There Was The Highest Revenue?

To find the year to which the revenue is the highest

```
data[data['Revenue (Millions)']==(data['Revenue (Millions)'].max())].Year
```

To find the record of the year to which the revenue was the highest

```
data.sort_values(by='Revenue (Millions)',ascending=False).head(1)
```

Barplot to see the year in which the revenue was the highest

```
plt.bar('Year','Revenue (Millions)',data=data,color='violet')
plt.title("Revenue By Year")
plt.show()
```

12. Find The Average Rating For Each Director

```
data.groupby('Director')['Rating'].mean().sort_values(ascending=False)
```

13. Display Top 10 Lengthy Movies Title

Displaying the records of the top 10 lengthy movies

```
data.sort_values(by="Runtime (Minutes)",ascending=False).head(10)
```

To find the top 10 lengthy movie titles

```
le=data.nlargest(10,'Runtime (Minutes)')[['Title','Runtime (Minutes)']]\
.set_index("Title")
le
```

Or

```
t=data.sort_values(by="Runtime (Minutes)",ascending=False)[['Title','Runtime (Minutes)']].head(10)
t
```

To make a barplot of the top 10 lengthy movie titles

```
plt.figure(figsize=(10,5))
sns.barplot(y='Runtime (Minutes)',x=le.index,data=le,palette="Blues")
plt.title('Top 5 Lengtly Movies')
plt.xticks(rotation=78)
plt.show()
```

14. Display Number of Movies Per Year
To find the number of movies per year

```
data['Year'].value_counts()
```

Visualisation of the number of movies per year

```
sns.countplot(x='Year',data=data,color="indigo")
plt.title("Number of Movies Per Year")
```

15. Find Most Popular Movie Title (Higest Revenue)

```
data.columns
data[data['Revenue (Millions)'].max() == data['Revenue (Millions)']]['Title']
```

Barplot to show the most popular movie titles based upon highest revenue earned

```
tr=data.sort_values(by="Revenue (Millions)",ascending =False)
tr1=tr['Title'].head()
t=tr['Revenue (Millions)'].head()
plt.barh(tr1,t,data=data,color=['violet','yellow','red','green','darkviolet'])
plt.title("Most Popular Movie Title")
plt.show()
```

16. Display Top 10 Highest Rated Movie Titles And its Directors

```
top_10=data.nlargest(10,'Rating')[['Title','Rating','Director']].set_index('Title')
top_10
```

Barplot to visualize the data

```
sns.barplot(x=top_10['Rating'],y=top_10.index,palette="plasma")
plt.title("Display Top 10 Highest Rated Movie Titles")
plt.show()
```

17. Display Top 10 Highest Revenue Movie Titles

data.columns

To get the records of the top 10 movies with highest revenue

t=data.sort_values(by='Revenue (Millions)',ascending=False)

t.head(10)

To get the titles of the top 10 movies with the highest revenue

t["Title"].head(10)

top_10 = data.nlargest(10,'Revenue (Millions)')[['Title','Director','Revenue (Millions)']].set_index('Title')

top_10

Barplot to show the top 10 movies with highest revenue

sns.barplot(x=top_10['Revenue (Millions)'],y=top_10.index,palette='viridis')

plt.title("Display Top 10 Highest Revenue Movie Titles")

plt.show()

18. Find Average Rating of Movies Year-wise

data.columns

data1=data.groupby('Year')['Rating'].mean()

data1

or

data1.sort_values(ascending=False)

Barplot to show the Year Vs Average Rating of the movies per year

data1.plot(kind='bar',figsize=(10,5),color="orange")

```
plt.grid()
plt.title("Average Rating of Movies Year-wise")
plt.show()
```

19. Does Rating Affect The Revenue?

```
sns.scatterplot(x='Rating',y='Revenue (Millions)',data=data,color="magenta")
```

Answer : Yes

20. Classify Movies Based on Ratings [Good,Better and Best]

```
data.columns
def rating(rating):
    if rating>=7.0:
        return 'Excellent'
    elif rating>=6.0:
        return 'Good'
    else:
        return 'Average'

data['rating_cat']=data['Rating'].apply(rating)    #.apply(rating) is used to apply the function 'rating' on the dataset

data.head()
```

21. Count Number of Action Movies

```
count=0
gen=input("Enter the Genre:")
for value in data['Genre']:
  if gen in value:
    count+=1
```

```
print("The number of Action movies:",count)
```

OR

```
len(data[data['Genre'].str.contains('action',case=False)])
```

22. Find the unique values from the Genre

To split all the elements by using ' , '

```
list1=[]
for values in data['Genre']:
  list1.append(values.split(','))
list1
```

To split the list elements in the list as single elements

```
one_list=[]
for item in list1:
  for items in item:
    one_list.append(items)
one_list
```

```
len(one_list)
```

To store all the elements once in a list and then check if the elements exist in the list or not.....if no then the element is unique

```
uni_list=[]
for item in one_list:
  if item not in uni_list:
    uni_list.append(item)
uni_list
```

To find the length of the unique list

```
len(uni_list)
```

23. How many films were made in each genre?

```
one_list
from collections import Counter
Counter(one_list)
```

#24. How many actors are present in the films mentioned in the dataset

```
list1=[]
for val in data['Actors']:
  list1.append(val.split(','))
list1


onr_l=[]
for vall in list1:
  for item in vall:
    onr_l.append(item)
onr_l


len(onr_l)


unni_a=[]
for items in onr_l:
  if items not in unni_a:
    unni_a.append(items)
unni_a


len(unni_a)
```

24. How many films did each actor do?

```
from collections import Counter

Counter(unni_a)
```

#** ----------Here ends my project---------- **
