34. **Data Analysis using pandas of python having imdb\_movie\_data.**

**Aim :**

To write a program to analyze the IMDb movie dataset using Python's pandas library.

**Algorithm :**

**Step 1** - Start the process.

**Step 2**  - **Import necessary libraries,** Import pandas for data handling. Import pandas

library as pd.

**Step 3**  - Load the dataset using pd.read\_csv() and store it in movies\_df.

**Step 4** - Display top rows using head() and bottom rows using tail().

**Step 5** - Display dataset structure using shape and columns.

**Step 6** - Find mean revenue using.

**Step 7** - Filter movies , Released after 2005 and before 2010.With rating ≥ 8.0 With

revenue less than 25th quantile.

**Step 8** - Display top rows of updated DataFrame.

**Step 9** - Select rows using .loc[] and .iloc[] and end the program.

**Step 10**  End the program.

**Program :**

from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

movies\_df = pd.read\_csv('/content/sample\_data/movies/IMDB-Movie-Data.csv')

print(movies\_df.head(5))

print(movies\_df.tail(10))

print(movies\_df.info())

print(movies\_df.shape)

print(movies\_df.columns)

movies\_df.rename(columns={

'Runtime (Minutes)': 'Runtime',

'Revenue (Millions)': 'Revenue\_millions'

}, inplace=True)

print(movies\_df.isnull().sum()) # null count per column

revenue = movies\_df['Revenue\_millions']

revenue\_mean = revenue.mean()

print("Revenue mean:", revenue\_mean)

movies\_df['Revenue\_millions'].fillna(revenue\_mean, inplace=True)

print(movies\_df.describe())

print(movies\_df['Genre'].describe())

print(movies\_df['Genre'].value\_counts().head(20))

subset = movies\_df[['Genre', 'Rating']]

print(subset.head())

prom = movies\_df[movies\_df['Title'] == 'Prometheus']

print(prom)

print(movies\_df.iloc[0])

flt = movies\_df[

(movies\_df['Year'] >= 2005) &

(movies\_df['Year'] <= 2010) &

(movies\_df['Rating'] >= 8.0) &

(movies\_df['Revenue\_millions'] < movies\_df['Revenue\_millions'].quantile(0.85))

]

print(flt)

def rating\_function(x):

if x >= 8.0:

return 'good'

elif (x >= 7.0) & (x < 8.0):

return 'better'

else:

return 'bad'

movies\_df['Rating\_category'] = movies\_df['Rating'].apply(rating\_function)

print(movies\_df.head(5))

good\_rev = movies\_df[

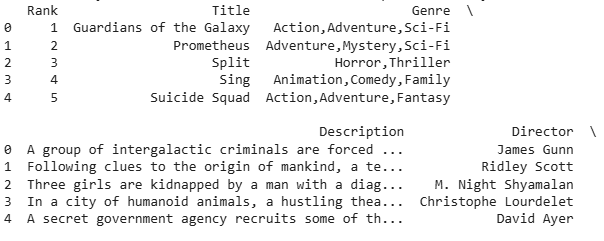
(movies\_df['Rating\_category'] == 'good') &

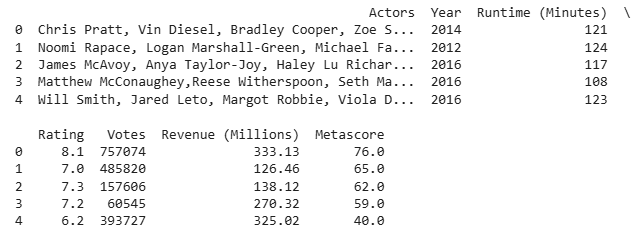
(movies\_df['Revenue\_millions'] > 100)

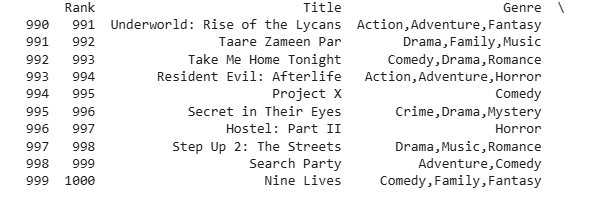
]

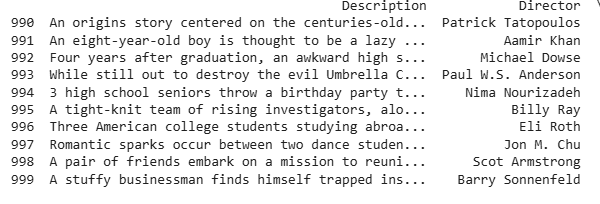
print(good\_rev)

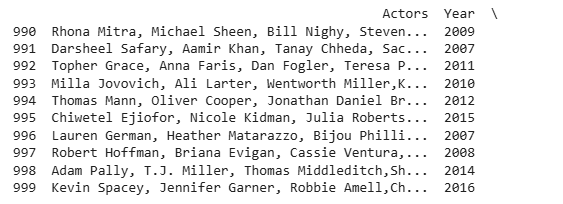
**Output :**

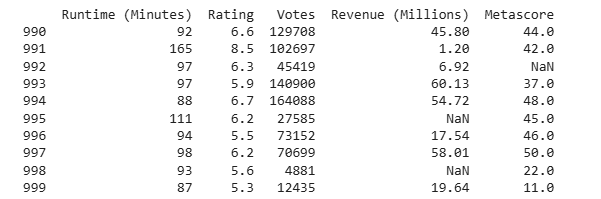


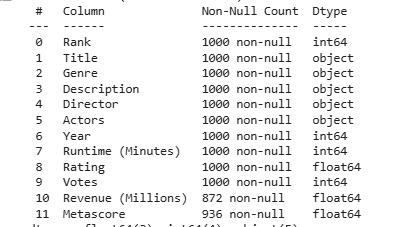


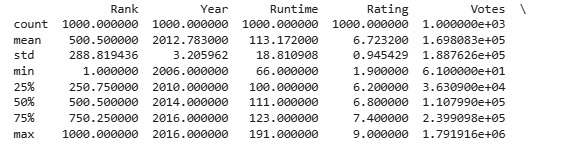


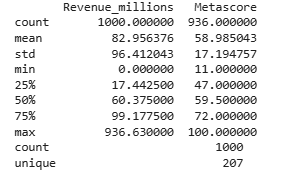


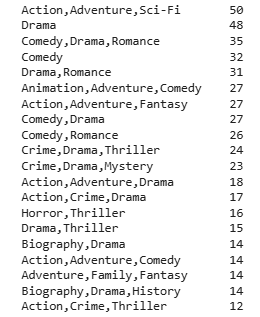


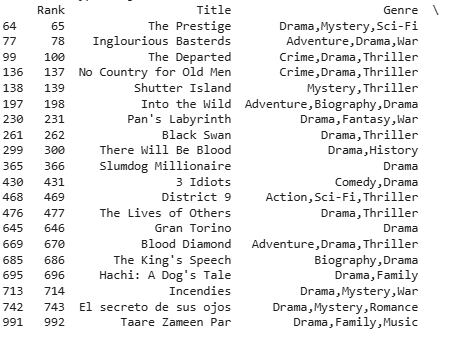


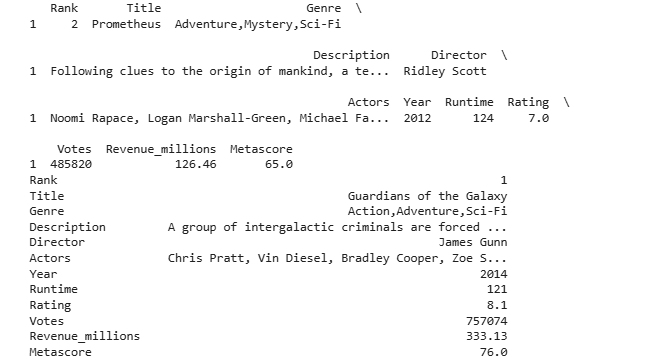


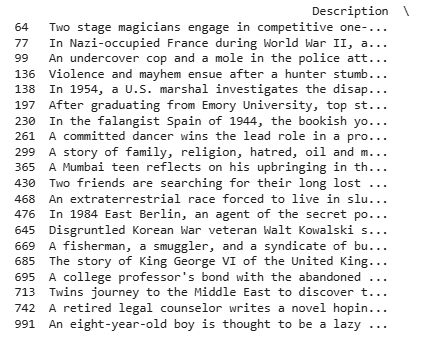


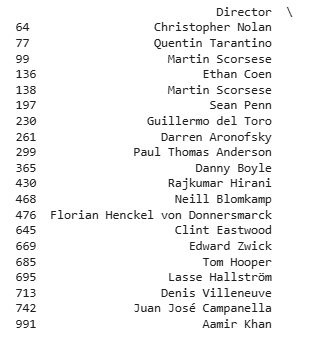
****

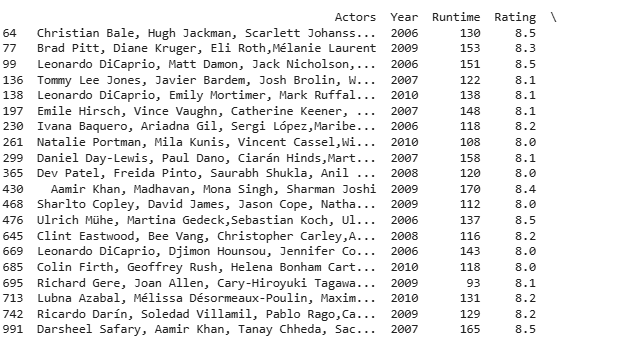
****

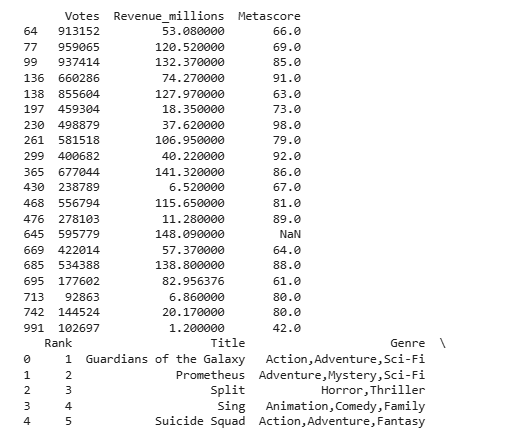
****

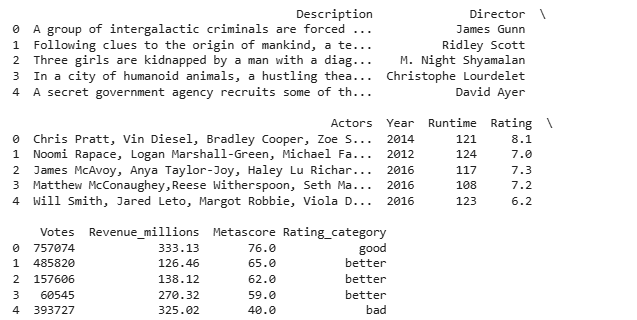
****

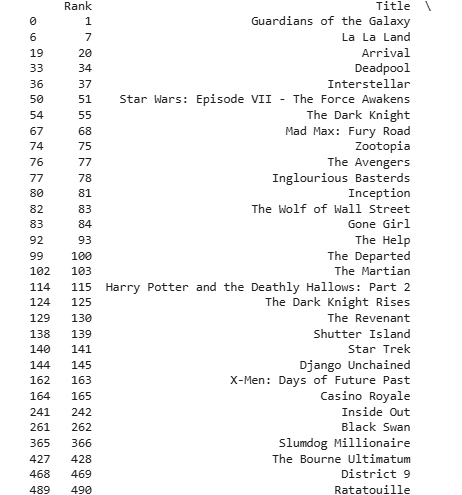
****

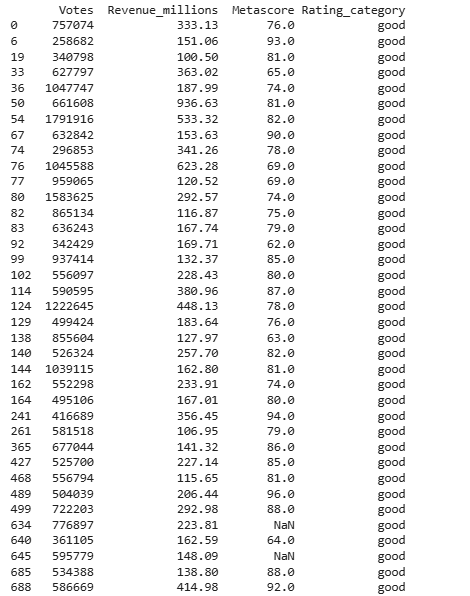
****

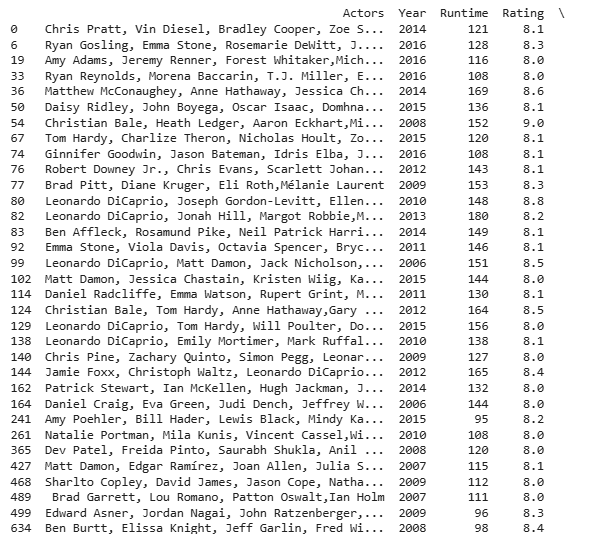
****

****

****

****

****

****

**Result :**

Thus, our program has been successfully saved and executed.

**33. Features of Numpy ,Mean,Median,mode and correlation coefficient using Numpy of Python.**

**Aim :**

To write a Python program using **NumPy** to demonstrate the features of NumPy and to compute **Mean, Median, Mode, and Correlation Coefficient** of given data.

**Algorithm :**

**Step 1** - Start the process.

**Step 2 -** Import the necessary libraries. Import numpy as np for numerical operations

and stats from scipy for mode calculation.

**Step 3** -Create a NumPy array with sample numeric data.

**Step 4** -Compute the Mean of the array using np.mean().

**Step 5** - Compute the Median of the array using np.median().

**Step 6** - Compute the Mode of the array using stats.mode().

**Step 7** -Define two numeric arrays x and y for correlation analysis.

**Step 8** -Compute the Correlation Coefficient using np.corrcoef(x, y).

**Step 9** -Display the results of Mean, Median, Mode, and Correlation Coefficient.

**Step 10** -End the program.

**Program :**

import numpy as np

from scipy import stats

data = np.array([10, 20, 20, 40, 50, 50, 50, 70, 90])

mean\_val = np.mean(data)

print("Mean:", mean\_val)

median\_val = np.median(data)

print("Median:", median\_val)

mode\_val = stats.mode(data, keepdims=True)

print("Mode:", mode\_val.mode[0], " (Count:", mode\_val.count[0], ")")

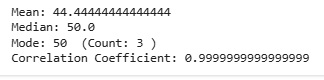
x = np.array([1, 2, 3, 4, 5])

y = np.array([2, 4, 6, 8, 10])

corr\_matrix = np.corrcoef(x, y)

print("Correlation Coefficient:", corr\_matrix[0, 1])

**Output :**

****

**Result :**

Thus, our program has been successfully saved and executed.