ESSENTIAL OF DATA SCIENCE

Theory Activity No. 1

Name – Sameet Anil Pisal

Div – CS8

Roll No. – CS8-14

PRN – 202401120018

- 20 problem statements for Kaggel Text Classification Dataset using Numpy and Pandas.
- Kaggle Link - https://www.kaggle.com/datasets/newra008/movie-review-and-rating

10 Problem Statements Using NumPy:

1. **Find the mean of "Rating" across all movies.**
2. **Find the median value of "Rating".**
3. **Find the maximum "Rating" given to any movie.**
4. **Find the minimum "Rating" given.**
5. **Find the standard deviation of "Rating".**
6. **Count how many movies were released after the year 2010.**
7. **Calculate the percentage of movies with "Rating" greater**

than 8.

8. **Find the average number of characters in "Movie" names.**
9. **Find the 75th percentile of "Rating".**
10. **Find how many movies have exactly a 10/10 rating.**

- **Solution:-**

```
import numpy as np
import pandas as pd

# Load dataset
df = pd.read_csv('your_dataset.csv')

# Q1
print("Q1. Mean of Rating:", np.mean(df['Rating']))

# Q2
print("Q2. Median of Rating:", np.median(df['Rating']))

# Q3
print("Q3. Maximum Rating:", np.max(df['Rating']))

# Q4
print("Q4. Minimum Rating:", np.min(df['Rating']))

# Q5
print("Q5. Standard Deviation of Rating:", np.std(df['Rating']))

# Q6
print("Q6. Movies released after 2010:", np.sum(df['Year'] > 2010))

# Q7
```

```
print("Q7. Percentage of movies with Rating > 8:", (np.sum(df['Rating'] > 8) / len(df)) * 100)

# Q8
print("Q8. Average length of Movie names:", np.mean(df['Movie'].apply(len)))

# Q9
print("Q9. 75th percentile of Rating:", np.percentile(df['Rating'], 75))

# Q10
print("Q10. Movies with exactly 10 Rating:", np.sum(df['Rating'] == 10))
```

```
PS C:\Users\samee\Desktop\EDS> python -u "c:\Users\samee\Desktop\EDS\main.py"
Q1. Mean of Rating: 3.041
Q2. Median of Rating: 3.0
Q3. Maximum Rating: 5
Q4. Minimum Rating: 1
Q5. Standard Deviation of Rating: 1.4245416806818956
Q6. Number of movies released after 2010: 1000
Q7. Percentage of movies with Rating > 8: 0.0
Q8. Average length of Movie names: 14.15
Q9. 75th percentile of Rating: 4.0
Q10. Movies with exactly 10 Rating: 0
PS C:\Users\samee\Desktop\EDS>
```

📊 10 Problem Statements Using Pandas:

1. **Find the total number of unique "Genres".**
2. **Find the number of movies released in the earliest year.**
3. **List top 5 most frequent genres.**
4. **Find the number of movies with missing "Review" values.**
5. **Replace missing "Review" values with "No Review Provided".**
6. **Find the number of unique movie names.**
7. **Find the top 3 most common words used in "Review" texts.**
8. **Count the number of reviews mentioning the word**

"excellent" or "amazing".

9. **Find the average rating for each genre separately.**
10. **Find the total number of movies released each year.**

- **Solution:-**

```
# Q11
print("Q11. Number of unique Genres:", df['Genres'].nunique())

# Q12
print("Q12. Movies released in the earliest year:", df[df['Year']
== df['Year'].min()].shape[0])

# Q13
print("Q13. Top 5 frequent Genres:\n",
df['Genres'].value_counts().head(5))

# Q14
print("Q14. Number of missing Reviews:",
df['Review'].isnull().sum())

# Q15
df['Review'].fillna('No Review Provided', inplace=True)
print("Q15. Missing Reviews replaced with 'No Review
Provided'.")

# Q16
print("Q16. Number of unique Movies:", df['Movie'].nunique())

# Q17
print("Q17. Top 3 most common words in Reviews:\n",
pd.Series('
```

'.join(df['Review'].dropna()).lower().split()).value_counts().head(3))

# Q18
print("Q18. Number of reviews mentioning 'excellent' or 'amazing':", df['Review'].str.contains('excellent|amazing', case=False, na=False).sum())

# Q19
print("Q19. Average Rating for each Genre:\n", df.groupby('Genres')['Rating'].mean())

# Q20
print("Q20. Total number of movies released each year:\n", df['Year'].value_counts())

```
Q11. Number of unique Genres: 10
Q12. Movies released in the earliest year: 250
Q13. Top 5 frequent Genres:
 Genres
Action/Adventure    250
Action/Sci-fi       200
Action/Fantasy      150
Thriller/Drama      100
Action/Comedy        50
Open file in editor (ctrl + click)
iews: 0
c:\Users\samee\Desktop\EDS\main.py:17: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
hained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are se
tting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the original object.


  df['Review'].fillna('No Review Provided', inplace=True)
Q15. Missing Reviews replaced with 'No Review Provided'.
Q16. Number of unique Movies: 19
Q17. Top 3 most common words in Reviews:
 the    5431
and    2877
a      2405
Name: count, dtype: int64
Q18. Number of reviews mentioning 'excellent' or 'amazing': 126
Q19. Average Rating for each Genre:
 Genres
 Action/Horror      3.000
Action/Adventure    3.028
Action/Comedy       3.000
```

```
  df['Review'].fillna('No Review Provided', inplace=True)
Q15. Missing Reviews replaced with 'No Review Provided'.
Q16. Number of unique Movies: 19
Q17. Top 3 most common words in Reviews:
 the    5431
and    2877
a      2405
Name: count, dtype: int64
Q18. Number of reviews mentioning 'excellent' or 'amazing': 126
Q19. Average Rating for each Genre:
 Genres
 Action/Horror      3.000
Action/Adventure    3.028
Action/Comedy       3.000
Action/Fantasy      3.000
Action/Sci-fi       3.000
Horror/Thriller     3.000
Sci-fi/Adventure    3.000
Sci-fi/Romance      3.000
Thriller/Drama      3.340
Thriller/Mystery    3.000
Name: Rating, dtype: float64
Q20. Total number of movies released each year:
 Year
2021    450
2016    250
2020    100
2017     50
2019     50
2018     50
2022     50
Name: count, dtype: int64
PS C:\Users\samee\Desktop\EDS>
```