Theory Activity No. 1

Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.

Dataset :- Kaggle Text Classification Dataset

Name : Mahir Mulani

Roll No : ET1-54

PRN : 20240107013O

Batch : ET13

Dataset link

question

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#1) Display the first 10 rows and last 10 rows of the dataset.
print("First 10 rows:")
print(df.head(10))
```

[1] ✓ 0.3s Python

Output

```
First 10 rows:
   UserName  ScreenName              Location     TweetAt  \
0         1       44953                   NYC  02-03-2020
1         2       44954          Seattle, WA  02-03-2020
2         3       44955                   NaN  02-03-2020
3         4       44956          Chicagoland  02-03-2020
4         5       44957  Melbourne, Victoria  03-03-2020
5         6       44958          Los Angeles  03-03-2020
6         7       44959                   NaN  03-03-2020
7         8       44960  Geneva, Switzerland  03-03-2020
8         9       44961                   NaN  04-03-2020
9        10       44962       Dublin, Ireland  04-03-2020
```

question

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#2) Find how many missing values are there in each column.
missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

[2] ✓ 0.0s Python

Output

```
Missing values in each column:
 UserName            0
ScreenName           0
Location           834
TweetAt              0
OriginalTweet        0
Sentiment            0
dtype: int64
```

question

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#3)Fill missing Location with "Unknown"
df['Location']= df['Location'].fillna('Unknown')
print(df['Location'])
```

[7] ✓ 0.0s Python

Output

```
0                       NYC
1              Seattle, WA
2                  Unknown
3              Chicagoland
4       Melbourne, Victoria
               ...
3793             Israel ??
3794         Farmington, NM
3795          Haverford, PA
3796                Unknown
3797    Arlington, Virginia
Name: Location, Length: 3798, dtype: object
```

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#4)How many unique locations are there in the dataset?
unique_locations = df['Location'].nunique()
print("Number of unique locations:", unique_locations)
```

[9] ✓ 0.0s Python

```
Number of unique locations: 1717
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#5)     Top 5 most common locations in the dataset.
topLocations  = df['Location'].value_counts().head(5)
print("Top 5 most common locations in the dataset:", topLocations)
```

[10] ✓ 0.0s Python

```
Top 5 most common locations in the dataset: Location
United States      75
London, England    48
Washington, DC     38
New York, NY       34
Los Angeles, CA    33
Name: count, dtype: int64
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#6) Convert TweetAt to datetime, extract month and year, and create a new column
df['TweetAt'] = pd.to_datetime(df['TweetAt'],dayfirst=True)
print(df['TweetAt'])
```

[14] ✓ 0.0s Python

question

```
0       2020-03-02
1       2020-03-02
2       2020-03-02
3       2020-03-02
4       2020-03-03
           ...
3793    2020-03-16
3794    2020-03-16
3795    2020-03-16
3796    2020-03-16
3797    2020-03-16
Name: TweetAt, Length: 3798, dtype: datetime64[ns]
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#7)Find how many tweets are Positive, Negative, Neutral, etc.
sum = df['Sentiment'].value_counts()
print(sum)
```

[16] ✓ 0.0s Python

question

```
Sentiment
Negative              1041
Positive               947
Neutral                619
Extremely Positive     599
Extremely Negative     592
Name: count, dtype: int64
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")


#8) Find the earliest and latest Tweet dates.
earliest = df['TweetAt'].min()
latest = df['TweetAt'].max()
print(f"Earliest: {earliest}, Latest: {latest}")
```

[17] ✓ 0.0s Python

```
Earliest: 02-03-2020, Latest: 16-03-2020
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")
#9)Find the number of tweets posted each day.
tweets_per_day = df['TweetAt'].value_counts().sort_index()
print(tweets_per_day)
```

[18] ✓ 0.0s Python

question

```
TweetAt
02-03-2020       4
03-03-2020       4
04-03-2020       8
05-03-2020       6
06-03-2020       2
07-03-2020       7
08-03-2020       9
09-03-2020      16
10-03-2020      54
11-03-2020     165
12-03-2020     685
13-03-2020    1233
14-03-2020     614
15-03-2020     519
16-03-2020     472
Name: count, dtype: int64
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#10)  Find the most common location of users.
most_common_location = df['Location'].mode()[0]
print(f"Most Common Location: {most_common_location}")
```

[19] ✓ 0.0s Python

```
Most Common Location: United States
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#11) Find the tweet with the maximum number of characters.

max_length_tweet = df.loc[df['OriginalTweet'].str.len().idxmax(), 'OriginalTweet']
print(max_length_tweet)
```

[20] ✓ 0.0s Python

question

```
In a Calgary grocery store lineup, I said to my wife, "this #coronavirus thing feels like Christmas to me".

Why? She asked.?
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#12) Find average number of characters per tweet.

average_characters = df['OriginalTweet'].str.len().mean()
print(f"Average Characters per Tweet: {average_characters}")
```

[21] ✓ 0.0s Python

```
Average Characters per Tweet: 213.4439178515008
```

question

Output

question

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#13) List all tweets containing the word 'coronavirus'

coronavirus_tweets = df[df['OriginalTweet'].str.contains('coronavirus', case=False)]

print(coronavirus_tweets[['OriginalTweet']])
```

[22] ✓ 0.0s Python

Output

```
                                          OriginalTweet
0       TRENDING: New Yorkers encounter empty supermar...
1       When I couldn't find hand sanitizer at Fred Me...
2       Find out how you can protect yourself and love...
3       #Panic buying hits #NewYork City as anxious sh...
4       #toiletpaper #dunnypaper #coronavirus #coronav...
...                                                   ...
3779  Stuck inside?  How about getting some reading ...
3786  We've noticed a shift in consumer #research in...
3787  Its funny seeing all these people fight and pa...
3793  Meanwhile In A Supermarket in Israel -- People...
3795  Asst Prof of Economics @cconces was on @NBCPhi...

[1698 rows x 1 columns]
```

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#14) How many tweets are Extremely Positive or Extremely Negative?
extreme_sentiment_count = df[df['Sentiment'].isin(['Extremely Positive', 'Extremely Negative'])].shape[0]

print(f"Extreme Sentiment Tweets: {extreme_sentiment_count}")
```

[23] ✓ 0.0s Python

question

```
Extreme Sentiment Tweets: 1191
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#15) Group tweets by Sentiment and find the average tweet length.
df['TweetLength'] = df['OriginalTweet'].str.len()
avg_length_by_sentiment = df.groupby('Sentiment')['TweetLength'].mean()
print(avg_length_by_sentiment)
```

[24] ✓ 0.0s Python

question

```
Sentiment
Extremely Negative    234.214527
Extremely Positive    240.268781
Negative              212.871278
Neutral               167.849758
Positive              213.923970
Name: TweetLength, dtype: float64
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#16) Create a new column indicating if the tweet contains the word 'toiletpaper'.

df['ContainsToiletPaper'] = df['OriginalTweet'].str.contains('toiletpaper', case=False)
print(df[['OriginalTweet', 'ContainsToiletPaper']].head())
```

[25] ✓ 0.0s Python

question

```
                                       OriginalTweet  ContainsToiletPaper
0  TRENDING: New Yorkers encounter empty supermar...               False
1  When I couldn't find hand sanitizer at Fred Me...               False
2  Find out how you can protect yourself and love...               False
3  #Panic buying hits #NewYork City as anxious sh...               False
4  #toiletpaper #dunnypaper #coronavirus #coronav...                True
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

# 17) Using NumPy, find how many tweets have length greater than 150 characters.

long_tweets = np.sum(df['OriginalTweet'].str.len() > 150)\

print(f"Tweets longer than 150 characters: {long_tweets}")
```

[26] ✓ 0.0s Python

```
Tweets longer than 150 characters: 3016
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#18) Find the 5 most frequent words in all tweets.

from collections import Counter
all_words = ' '.join(df['OriginalTweet']).lower().split()
common_words = Counter(all_words).most_common(5)
print(common_words)
```

[27] ✓ 0.0s Python

```
[('the', 4240), ('to', 3723), ('and', 2435), ('of', 2060), ('in', 1811)]
```

question

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#19) Sort the dataset based on Tweet length (longest first).

df['TweetLength'] = df['OriginalTweet'].str.len()

df_sorted = df.sort_values(by='TweetLength', ascending=False)
print(df_sorted[['OriginalTweet', 'TweetLength']].head())
```

[29] ✓ 0.0s Python

question

```
                                          OriginalTweet  TweetLength
2576  Here□s a list of all @WEFoodbank Drop-Off poin...          342
2400  In a Calgary grocery store lineup, I said to m...          342
236   So, according to this dude if you eat Rama mar...          334
2531  HELLO EVERYONE ????\r\r\nWe made &amp; sell hi...          331
2061  As people stock up on food &amp; hygiene produ...          331
```

Output

```python
# 20 problem statements solved using pandas abd numpy

import pandas as pd
import numpy as np

df = pd.read_csv("Corona_NLP_test.csv")

#20) Find total number of tweets per sentiment per day.

tweets_sentiment_day = df.groupby(['TweetAt', 'Sentiment']).size().unstack(fill_value=0)
print(tweets_sentiment_day)
```

[30] ✓ 0.0s Python

question

```
Sentiment    Extremely Negative  Extremely Positive  Negative  Neutral  \
TweetAt
02-03-2020                    1                   1         1        0
03-03-2020                    0                   0         0        3
04-03-2020                    4                   2         0        0
05-03-2020                    0                   3         1        1
06-03-2020                    0                   2         0        0
07-03-2020                    1                   0         4        0
08-03-2020                    3                   2         2        2
09-03-2020                    5                   3         3        2
10-03-2020                    6                   7        19        7
11-03-2020                   16                  27        42       32
12-03-2020                  101                 112       217      100
13-03-2020                  193                 181       340      205
14-03-2020                  103                  83       174       92
15-03-2020                   87                  88       124       95
16-03-2020                   72                  88       114       80
```

Output