# **EDS Theory Activity**

Name: Shreyash Lamkhade

PRN: 202401040340

Roll No: CS3-53

Batch: C33

1.Dataset Link: <a href="https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download">https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download</a>

## 2.Colab Notebook Link:

https://colab.research.google.com/drive/1AMu-2cxHQPxjTPpQayZA427VO9V6c\_4?usp=sharing

**Output:** 

```
[5] import pandas as pd
    df = pd.read_csv("Corona_text classifier1.csv")
```

#### Problem1: Display Missing values and datatypes in data??

#### Problem2: Count the frequency of each sentiment??

```
print("Sentiment counts:\n", df['Sentiment'].value_counts())

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

Problem3: Extract tweets containing the word 'mask'.

```
[11] mask_tweets = df[df['OriginalTweet'].str.contains('mask', case=False)]
print("Mask-related tweets:", mask_tweets.shape[0])
```

→ Mask-related tweets: 94

Problem4: What is Average length of tweets?

```
[12] df['TweetLength'] = df['OriginalTweet'].apply(len)
print("Avg tweet length:", df['TweetLength'].mean())
```

₹ Avg tweet length: 213.4439178515008

Problem5: Sort the dataset by tweet length in descending order.

```
[13] df_sorted = df.sort_values('TweetLength', ascending=False)
    print("Longest tweet:", df_sorted.iloc[@]['OriginalTweet'])
```

Engest tweet: Here®s a list of all @WEFoodbank Drop-Off points across the NE & all donations are greatly appreciated at this unchartered time. Thanks.?

There®s a new WishList & online Supermarket shopping can be sent for delivery during @WEFoodbank opening hours at these addresses:

#NUFC ? https://t.co/bgAyCPH4N0 https://t.co/oMI76q8Duu

Problem6: Calculate the standard deviation of tweet lengths.

```
[14] print("Std dev of lengths:", df['TweetLength'].std())

Std dev of lengths: 66.52653782951096
```

Problem7: Find the top 5 most common words in tweets.

```
[15] from collections import Counter
  words = ' '.join(df['OriginalTweet']).lower().split()
  print("Top 5 words:", Counter(words).most_common(5))

Top 5 words: [('the', 4240), ('to', 3723), ('and', 2435), ('of', 2060), ('in', 1811)]
```

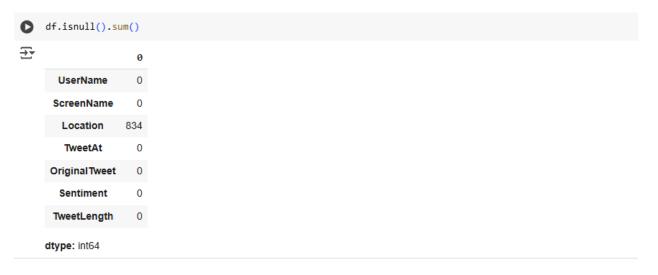
#### Problem8: Show the count of each sentiment category.

```
[16] print(df['Sentiment'].value_counts())

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

#### Problem9: Find the user with the most tweets.

### Problem10: Count the number of missing values per column.



#### Problem11: Filter tweets with more than 100 characters.

```
long_tweets = df[df['TweetLength'] > 100]
    print(long tweets)
₹
         UserName ScreenName
                                       Location TweetAt \
                                           NYC 02-03-2020
    0
                   44953
              1
    1
               2
                      44954
                                   Seattle, WA 02-03-2020
    3
               4
                     44956
                                   Chicagoland 02-03-2020
              5
                     44957 Melbourne, Victoria 03-03-2020
    4
                     44958 Los Angeles 03-03-2020
    5
              6
              . . .
                                     Israel ?? 16-03-2020
    3793
             3794
                      48746
                      48747
                                Farmington, NM 16-03-2020
    3794
             3795
             3796
                     48748
                                  Haverford, PA 16-03-2020
    3795
    3796
             3797
                      48749
                                            NaN 16-03-2020
    3797
             3798
                      48750 Arlington, Virginia 16-03-2020
                                          OriginalTweet
                                                                Sentiment \
    0
         TRENDING: New Yorkers encounter empty supermar... Extremely Negative
         When I couldn't find hand sanitizer at Fred Me...
                                                                 Positive
    1
    3
         #Panic buying hits #NewYork City as anxious sh...
                                                                 Negative
         #toiletpaper #dunnypaper #coronavirus #coronav...
    4
                                                                Neutral
                                                                Neutral
         Do you remember the last time you paid $2.99 a...
    5
    3793 Meanwhile In A Supermarket in Israel -- People...
```

Positive

```
Positive
    3793 Meanwhile In A Supermarket in Israel -- People...
    3794 Did you panic buy a lot of non-perishable item...
                                                                     Negative
→ 3795 Asst Prof of Economics @cconces was on @NBCPhi...
                                                                     Neutral
    3796 Gov need to do somethings instead of biar je r... Extremely Negative
    3797 I and @ForestandPaper members are committed to... Extremely Positive
          TweetLength
    0
                 228
                 193
    1
    3
                 318
    4
                 252
    5
                 205
                 . . .
    3793
                 127
    3794
                 213
    3795
                 185
    3796
                 174
    3797
                 254
    [3536 rows x 7 columns]
```

### Problem12: Find the most frequent location.

```
[24] df['Location'].mode()[0]
```

Tr 'Ilnitad States'

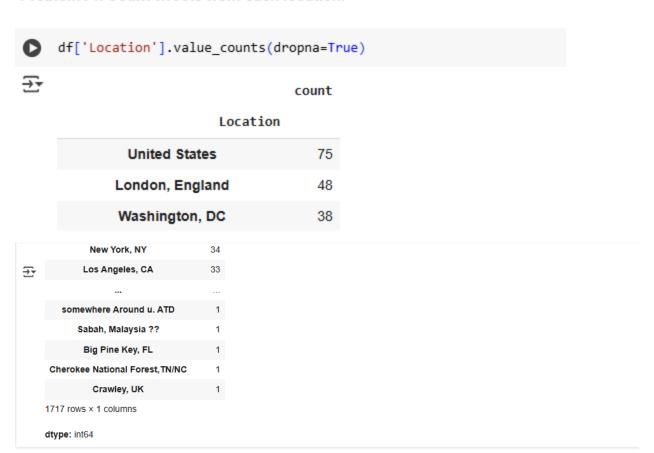
→ 'United States'

# Problem13: Calculate the percentage of negative sentiments.

```
[26] (df['Sentiment'].str.contains('Negative').mean() * 100)

To np.float64(42.99631384939442)
```

### Problem14: Count tweets from each location.



Problem15: Find users who tweeted from multiple locations.

df.groupby('UserName')['Location'].nunique().sort\_values(ascending=False).head()

		Location		
₹	UserName			
	3798	1		
	1	1		
	2	1		
	3796	1		
	3777	1		

dtype: int64

Problem16: Group by location and count sentiments.



Shenzhen, Guangdong, PR China	Positive	1
ÜT: 40.5896566,-74.4274456	Positive	1
ÜT: 40.725815,-74.00777	Negative	1
ÜT: 43.64624,-79.42516	Positive	1
ÜT: 43.661815,-79.377458	Extremely Negative	1
ÜT: 44.881667,-93.312324	Negative	1
2004 rows v 4 solumns		

2221 rows x 1 columns

dtype: int64

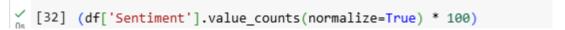
<del>\_\_\_\_\_</del>

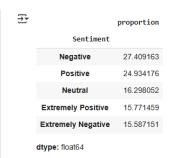
# Problem17: How many unique users are in the dataset?

```
√ [31] df['UserName'].nunique()

→ 3798
```

# Problem18: What's the sentiment distribution as percentages?





Problem19: Find tweets with URLs in them.



₹	4	5	44957	Melbourne, Victoria	03-03-2020	#toiletpaper #dunnypaper #coronavirus #coronav	Neutral	252
	5	6	44958	Los Angeles	03-03-2020	Do you remember the last time you paid \$2.99 a	Neutral	205
	3792	3793	48745	Washington D.C.	16-03-2020	@RicePolitics @MDCounties Craig, will you call	Negative	215
	3793	3794	48746	Israel ??	16-03-2020	Meanwhile In A Supermarket in Israel People	Positive	127
	3794	3795	48747	Farmington, NM	16-03-2020	Did you panic buy a lot of non-perishable item	Negative	213
	3795	3796	48748	Haverford, PA	16-03-2020	Asst Prof of Economics @cconces was on @NBCPhi	Neutral	185
	3797	3798	48750	Arlington, Virginia	16-03-2020	I and @ForestandPaper members are committed to	Extremely Positive	254
1583 rows x 7 columns								

### Problem20: What's the average word count per tweet?

```
'[39] df['WordCount'] = df['OriginalTweet'].apply(lambda x: len(x.split()))

df['WordCount'].mean()

'[39] df['WordCount'] = df['OriginalTweet'].apply(lambda x: len(x.split()))

df['WordCount'] = df['WordCount'].apply(lambda x: len(x.split()))

df['WordC
```

→ np.float64(32.909689310163245)