

# EDS Theory Activity

**Name:** Shreyash Lamkhade

**PRN:** 202401040340

**Roll No:** CS3-53

**Batch:** C33

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

**2.Colab Notebook Link:**

[https://colab.research.google.com/drive/1AMu-2cxHQPxiTPpQayZA427VO9V6c\\_4?usp=sharing](https://colab.research.google.com/drive/1AMu-2cxHQPxiTPpQayZA427VO9V6c_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??

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3798 entries, 0 to 3797
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   UserName        3798 non-null   int64
 1   ScreenName      3798 non-null   int64
 2   Location        2964 non-null   object
 3   TweetAt        3798 non-null   object
 4   OriginalTweet   3798 non-null   object
 5   Sentiment       3798 non-null   object
dtypes: int64(2), object(4)
memory usage: 178.2+ KB
None
```

#### 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   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[0]['OriginalTweet'])
```

```
Longest 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	592
Name: count, dtype: int64	

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

```
[17] if 'UserName' in df.columns:
      print("Most active user:", df['UserName'].value_counts().idxmax())
```

⇒ Most active user: 3798

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

▶ df.isnull().sum()

⇒

	0
UserName	0
ScreenName	0
Location	834
TweetAt	0
OriginalTweet	0
Sentiment	0
TweetLength	0

dtype: int64

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

```

▶ long_tweets = df[df['TweetLength'] > 100]
print(long_tweets)

```

```

⇒
   UserName  ScreenName      Location  TweetAt \
0          1      44953          NYC  02-03-2020
1          2      44954  Seattle, WA  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
...      ...      ...      ...      ...
3793      3794      48746    Israel ??  16-03-2020
3794      3795      48747  Farmington, NM  16-03-2020
3795      3796      48748  Haverford, PA  16-03-2020
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
1  When I couldn't find hand sanitizer at Fred Me...      Positive
3  #Panic buying hits #NewYork City as anxious sh...      Negative
4  #toiletpaper #dunnypaper #coronavirus #coronav...      Neutral
5  Do you remember the last time you paid $2.99 a...      Neutral
...      ...      ...
3793  Meanwhile In A Supermarket in Israel -- People...      Positive

```

```

3793 Meanwhile In A Supermarket in Israel -- People... Positive
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
1             193
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]
```

```
'United States'
```

 'United States'


**Problem13: Calculate the percentage of negative sentiments.**

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


 np.float64(42.99631384939442)

**Problem14: Count tweets from each location.**

```
df['Location'].value_counts(dropna=True)
```



Location		count
United States		75
London, England		48
Washington, DC		38




New York, NY	34
Los Angeles, CA	33
...	...
somewhere Around u. ATD	1
Sabah, Malaysia ??	1
Big Pine Key, FL	1
Cherokee National Forest, TN/NC	1
Crawley, UK	1

1717 rows x 1 columns

dtype: int64

**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.**

✓  
0s [30] `df.groupby(['Location', 'Sentiment']).size()`

 0

Location		Sentiment	
		Negative	1
Black Country, West midlands		Negative	1
England.		Positive	1
Pleiades & SantaFe		Extremely Positive	1



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

2221 rows × 1 columns

dtype: int64

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



```
[31] df['UserName'].nunique()
```



3798

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



```
[32] (df['Sentiment'].value_counts(normalize=True) * 100)
```



proportion	
Sentiment	
Negative	27.409163
Positive	24.934176
Neutral	16.298052
Extremely Positive	15.771459
Extremely Negative	15.587151

dtype: float64

**Problem19: Find tweets with URLs in them.**



```
df[df['OriginalTweet'].str.contains('http')]
```



	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment	TweetLength
0	1	44953	NYC	02-03-2020	TRENDING: New Yorkers encounter empty supermar...	Extremely Negative	228
1	2	44954	Seattle, WA	02-03-2020	When I couldn't find hand sanitizer at Fred Me...	Positive	193
3	4	44956	Chicagoland	02-03-2020	#Panic buying hits #NewYork City as anxious sh...	Negative	318



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()

np.float64(32.909689310163245)
```