Name- Jamdade Aarti Sanjay
Roll No - ETI-12
Class- ETI
PRN- 2024010700043

MIT | Academy of Engineering
(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

I will assume a sample dataset structured like this:

id	text	label
1	"I love this product!"	Positive
2	"Worst experience ever."	Negative
3	"It was okay, not great."	Neutral
4	"Amazing service!"	positive
5.	"Terrible support team."	negative

1. Display the first 10 rows of the dataset.

```
df.head(10)
```

Output:

	id	text	label
0	1	I love this product!	positive
1	2	Worst experience ever.	negative
2	3	It was okay, not great.	neutral
3	4	Amazing service.	positive
4	5	Terrible support team.	negative

2. Check if there are any missing (null) values in the dataset.

```
import pandas as pd
df.isnull().sum()
```

Output:

```
id       0
text     0
label    0
dtype: int64
```

3. How many unique labels are there?
```
df['label'].nunique()
```
Output - 3

4. What is the count of each label?
```
df['label'].value_counts()
```
Output -
```
positive    2
negative    2
neutral     1
Name: label, dtype: int64
```

5. Find the text entry with the longest text.
```
df['text_length'] = df['text'].apply(len)
df.loc[df['text_length'].idxmax()]
```
Output-
```
id
text           It was okay, but not great.
label                           neutral
text_length                          24
Name: 2, dtype: object
```

6. Find the average text length.
```
np.mean(df['text_length'])
```
Output - 21.2

7. Find the average word count across all texts
```
np.mean(df['word_count'])
```
output - 3.6

8. Add a new column for the number of words in each text.

```
df['word_count'] = df['text'].apply(lambda x: len(x.split()))
df.head()
```

Output -

id	text	label	text_length	word_count
0 1	I love this product!	positive	21	4
1 2	Worst experience ever.	negative	24	3
2 3	It was okay, not great	neutral	24	6
3 4	Amazing service!	positive	16	2
4 5	Terrible support team.	negative	24	3

9. Show all text entries labeled as 'positive'.

```
df[df['label'] == 'positive']
```

Output -

id	text	label
0 1	I love this product!	positive
3 4	Amazing service!	positive

10. Get only the id and label columns.

```
df[['id','label']]
```

Output-

	id	label
0	1	positive
1	2	negative
2	3	mixed
3	4	positive
4	5	negative

11. Import the dataset and convert the text_length column into a Numpy array.

```
import numpy as np
length_array = np.array(df['text_length'])
length_array
```

Output-

```
array([21, 24, 24, 16, 24])
```

12. Find the maximum text length.

```
np.max(length_array)
```

Output = 24

13. Find the minimum text length.

```
np.min(length_array)
```

Output - 16

14. Calculate the ratio of text longer than average length.

```
length_array = np.array(df['text_length'])
average_length = np.mean(length_array)
ratio = np.sum(length_array > average_
        length)/length_array.size
ratio
output -0.6
```

15. Create a new Numpy array where lengths > average are labeled 1, else 0.

```
labels_array = np.where(length_array >
average_length,1,0)
labels_array
Output - array([0,1,1,0,1])
```

16. Find the IDs of the texts whose length is within one standard deviation of the mean

```
mean = np.mean(length_array)
std = np.std(length_array)
ids_within_std = df['id']
[np.abs(length_array - mean) <= std]
ids_within_std.values
Output - array([1,2,3,5])
```

17. Using broadcasting, cal. the absolute differen. between each text's length and every other text's length.

```
diff_matrix = np.abs(length_array.reshape(-1,1)
                - length_array)
diff_matrix
```

Output -

```
array([
    [0, 3, 3, 5, 3],
    [3, 0, 0, 8, 0],
    [3, 0, 0, 8, 0],
    [5, 8, 8, 0, 8],
    [3, 0, 0, 8, 0]
])
```

18. Vectorize a function to classify text lengths.

```
def classify_length(x):
    if x<20:
        return 'short'
    elif x<=24:
        return 'medium'
    else:
        return 'long'

vectorized_classify = np.vectorize(classify_
    length)
vectorized_classify(length_array)
```

Output -

```
array(['medium', 'medium', 'medium',
 'short', 'medium'], dtype='<U6')
```

19. Find how many unique text lengths exist using numpy.

```
np.unique(length_array).size
```

Output - 3

20. Normalize all values in the combined array between 0 and 1 using Min-Max scaling.

```
min_vals = combined_array.min(axis=0)
max_vals = combined_array.max(axis=0)
normalized_array = (combined_array -
  min_vals) / (max_vals - min_vals)
normalized_array
```

Output -

```
array([[0.625, 0.5 ],
       [1.   , 0.25],
       [1.   , 1.  ],
       [0.   , 0.  ]])
```