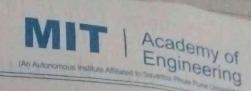
	ROLL NO - ETI-12 Class- ETI Sozyoto 10043 Twill assume a sample dataset structured like this:
	id text I "I love this product!" Positive 2 "Worst experience ever". Negative 3 "It was okay, not great." Neutral 4 "Amazing service!" positive 5. "Terrible support team." negative
4x213	Display the first 10 rows of the dataset. df. head (10) Output: text
	o I Jove this product! positive
-	hlotst experience ever. negative 2 3 It was okay, not great. neutral
	Amazing service. positive Terrible support team.
2.	negative Check if there are any missing (nwl) Values in the dataset. df. isnull(). sum() output: id o label o



3.	Han many unique labele and
	How many unique labels are there? df ['label']. nunique() Output - 3
	Output - 3
	TO SHALL THE PROBLEM TO THE FEW PARTY.

- 4. What is the count of each label?

 df['rabel']. value_counts()

 Output positive 2

 negative 2

 newtral |

 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].idxmaxO

 Output
 id

text It was okay, but not great.

label neutral

text length 22

Name: 2, dtype: object

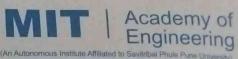
- 6. Find the average text length.

 np. mean (df['text_length])

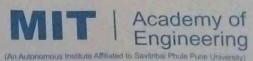
 Output- 21.2
- 7. Find the average toord count gaross
 all texts

 np. mean(df['word count'])

 output 3.6



	(An Autonomous Institute Affiliated to Savitribal Phule Pune University)
8.	Add a new column for the number of words
	in each text.
	df['word_count'] = df['text'] apply
	Clambdax: len(x.split())
	df. head()
1057	Output -
	id text
-0-	label text length word count
	o I I love this product!
	positive 21 4 12 Worst experience ever.
	neartive 24 3
	negative 24 3 23 It was oray, not great
TOH	neutral 24 6 3 4 Amazing service!
	2
	7 Terrible Support team.
3	negative 24 3
3	
	Show all text entries labeled as positive!
9.	df [df ['label'] == 'positive']
	output - label
	o I Tlove this product!
	11:00
	positive 8 4 Amazing service!
	positive
43(1)	POSTTIVO



	(An Autonomous Institute Affiliated to Savitribai Phule Pune University)
THE PERSON	
10.	Get only the id and label columns. df[dT'id','label']] Output- id label O I positive 1 2 negative 2 3 mixed 3 4 positive 4 5 negative
	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])
171129	Find the maximum text length. np. max (tength-array) Output = 24
13.	Find the minimum text length. np. min Clength-array) Output - 16
14.	Calculate the ratio of text longer than average length.

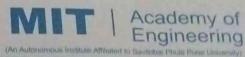


length_array = np. array (df['text_length]) average_length = np. mean (length_array) tatio = np. sum Clength_array > average length/ length_array. size. output -0.6 15. Create a new Numpy array where lengths? average are labeled 1, else o. labels array = np. where clength_array? avera average length, 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.

9

mean = np. mean Clength array) std = np. std Clength_array)
ids_cwithin_std = df ['id'] Enp. abs Clength_array-mean X = std]
ids_within_std. values Output - array ([1,2,3,5]

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



diff matrix = np. abs Clength_array reshaped, n
- rength_array) diff moutrix Outputarray ([[0,3,3,5,3], 3,0101810], [3,0,0,8,0], [5,8,8,0,8], [3,0,0,8,6] Vectorize a function to classify text lengths. def classify length (x): return 'short' 61if X<=571; return 'medium' else: return 'long' vectorized = classify = np. vectorize (classify) vectorized classify Clength_array) array (['medium', 'medium', 'medium', 'short', Imedium'], dtype = '<u6')

16	Find how many unique text lengths exist using numpy. np unique (length array) size Output -3	1
20.	Normalize all values in the combined array between o and I using Min-Max scaling. Min vals = combined_array min (axisso) max vals = combined_array niax (axisso) normalized_array = (combined array = min vals) / (max vals = min vals) normalized_array Output = ([[0.625 , 0.5] [1 , 0.25] [1 , 0.25] [1 , 0.3]	