Name - Mrunali Bhoyar

Roll No. - CS3-69

DATASET - Enron Email Dataset

Uploading the file of dataset:

```
from google.colab import files
uploaded = files.upload()
```

Choose Files mail_data.csv

- **mail_data.csv**(text/csv) - 485702 bytes, last modified: 4/28/2025 - 100% done

Saving mail_data.csv to mail_data.csv

```
!pip install pandas numpy
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

Importing Libraries and Load Dataset:

```
import pandas as pd
import numpy as np

# Load your CSV
df = pd.read_csv('/content/mail_data.csv')
df.head()
```

| | Category | Message |
|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |

Next steps: View recommended plots New interactive sheet

Structure of the dataset:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

**20 problem statements and their corresponding NumPy and Pandas-based solutions:**

[Basic Exploration]

1. Total number of emails in the dataset.

```
len(df)
```

```
5572
```

2. Finding the number of unique categories exist.

```
df['Category'].nunique()
```

```
2
```

3. Number of emails belonging to each category.

```
df['Category'].value_counts()
```

| Category | count |
|---|---|
| ham | 4825 |
| spam | 747 |

dtype: int64

4. Percentage of emails belonging to each category.

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

| Category | proportion |
|---|---|
| ham | 86.593683 |
| spam | 13.406317 |

**dtype:** float64

5. Finding number of missing values in the dataset.

```
df.isnull().sum()
```

| | 0 |
|---|---|
| Category | 0 |
| Message | 0 |

**dtype:** int64

[Text Preprocessing]

6. Finding the average length of an email message.

```
df['message_length'] = df['Message'].astype(str).apply(len)
df['message_length'].mean()
```

```
np.float64(80.36898779612348)
```

7. Maximum and Minimum length of messages.

```
df['message_length'].max(), df['message_length'].min()
```

```
(910, 2)
```

8. The distribution of message lengths.

```
df['message_length'].describe()
```

| | message_length |
|---|---|
| **count** | 5572.000000 |
| **mean** | 80.368988 |
| **std** | 59.926946 |
| **min** | 2.000000 |
| **25%** | 35.750000 |
| **50%** | 61.000000 |
| **75%** | 122.000000 |
| **max** | 910.000000 |

**dtype:** float64

9. Email message which is the longest?

```
df.loc[df['message_length'].idxmax()]['Message']
```

```
'For me the love should start with attraction.i should feel that I need her every time around me.she should be the first thing which co
mes in my thoughts.I would start the day and end it with her.she should be there every time I dream.love will be then when my every bre
ath has her name.my life should happen around her.my life will be named to her.I would cry for her.will give all my happiness and take
all her sorrows.I will be ready to fight with anyone for her.I will be in love when I will be doing the craziest things for her.love wi
ll be when I don't have to proove anyone that my girl is the most beautiful lady on the whole planet.I will always be singing praises f
or her.love will be when I start up making chicken curry and end up makiing sambar.life will be the most beautiful then will get every
```

10.Email message that is the shortest.

```
df.loc[df['message_length'].idxmin()]['Message']
```

```
'Ok'
```

[Text Analysis]

11.Messages that contain the word "meeting".

```
df['contains_meeting'] = df['Message'].astype(str).str.contains('meeting', case=False)
df['contains_meeting'].sum()
```

```
np.int64(43)
```

12. Top 10 most common words in all emails.

```
from collections import Counter

words = ' '.join(df['Message'].dropna().astype(str)).lower().split()
word_counts = Counter(words)
word_counts.most_common(10)
```

```
[('to', 2234),
 ('i', 2217),
 ('you', 1921),
 ('a', 1433),
 ('the', 1326),
 ('u', 996),
 ('and', 968),
 ('is', 868),
 ('in', 857),
 ('my', 755)]
```

13.Number of emails that contain the word "urgent".

```
df['Message'].astype(str).str.contains("urgent", case=False).sum()
```

```
np.int64(68)
```

14. Messages that are blank or empty.

```
df['Message'].isna().sum() + (df['Message'].astype(str).str.strip() == '').sum()
```

```
np.int64(0)
```

15.Average number of words per message.

```
df['word_count'] = df['Message'].astype(str).apply(lambda x: len(x.split()))
df['word_count'].mean()
```

```
np.float64(15.584170854271356)
```

[ NumPy Statistical Calculation]

16.Standard deviation of word count in messages using NumPy.

```
np.std(df['word_count'])
```

```
11.405574793618133
```

17. Median word count of the messages.

```
np.median(df['word_count'])
```

```
np.float64(12.0)
```

18. Number of messages having word count greater than the average.

```
df[df['word_count'] > df['word_count'].mean()].shape[0]
```

```
2245
```

[Labeling & Categorization]

19.Adding a new column: Label as "Long" if word count > 20, else "Short".

```
df['length_label'] = np.where(df['word_count'] > 20, 'Long', 'Short')
df[['Message', 'length_label']].head()
```

| | Message | length_label |
|---|---|---|
| 0 | Go until jurong point, crazy.. Available only ... | Short |
| 1 | Ok lar... Joking wif u oni... | Short |
| 2 | Free entry in 2 a wkly comp to win FA Cup fina... | Long |
| 3 | U dun say so early hor... U c already then say... | Short |
| 4 | Nah I don't think he goes to usf, he lives aro... | Short |

20.Proportion of long vs short messages.

```
df['length_label'].value_counts(normalize=True)
```

| length_label | proportion |
|---|---|
| Short | 0.704953 |
| Long | 0.295047 |

dtype: float64