- Name : Pratik subhashchandra bharti
- Division : ET1
- Roll No : ET1-77
- PRN : 202401070106

♊ **TOPIC : SMS SPAM COLLECTION DATASET**

❖ PROBLEM STATEMENT:

1. Correlation between Message Length and Spam: Is there any noticeable correlation (even if not statistically significant just by observation) between the length of a message and whether it's spam?.

```
main.py > ...
correlation = df[['message_length', 'label_num']].corr().iloc[0,
1]
print(f"Correlation between message length and spam (numerical
label): {correlation:.2f}")
print("\nObservation: While there's a slight positive correlation,
it's not very strong, suggesting message length alone isn't a
perfect predictor of spam. However, the average lengths in problem
statement 2 do show a difference.")
```

2. Distribution of Message Lengths: Create bins of message lengths (e.g., 0-20

characters, 21-50 characters, etc.) and find the distribution of spam and ham messages across these bins.

```
main.py > ...
python bins = [0, 20, 50, 100, 150, 200, float('inf')]
labels = ['0-20', '21-50', '51-100', '101-150', '151-200', '>200'] df['length_bins'] =
pd.cut(df['message_length'], bins=bins, labels=labels, right=False) print("Distribution of
message lengths:") print(df.groupby(['label', 'length_bins']).size().unstack(fill_value=0))
```

3. Messages Containing a Specific Symbol: How many spam messages contain the symbol "$"?

```
main.py > ...
contains_dollar_spam = df[(df['label'] == 'spam') &
(df['message'].str.contains(r'\$'))].shape[0]
print("Number of spam messages containing '$':",
contains_dollar_spam)
```

4. Messages Starting with a Specific Word: How many messages start with the word "Free"?

```
Welcome    main.py 1
main.py > ...
starts_with_free = df['message'].str.startswith('Free ')
print("Number of messages starting with 'Free ':",
starts_with_free.sum())
```

5. Average Number of Words per Message: What is the average number of words in a spam message versus a ham message?

```python
df['word_count'] = df['message'].apply(lambda x: len(x.split()))
print("Average number of words in spam messages:", df[df['label']
== 'spam']['word_count'].mean())
print("Average number of words in ham messages:", df[df['label']
== 'ham']['word_count'].mean())
```

6. Presence of Uppercase Letters (Spam vs. Ham): What is the proportion of spam messages containing uppercase letters compared to ham messages containing uppercase letters?

```python
spam_has_upper = df[df['label'] ==
'spam']['message'].str.contains(r'[A-Z]').sum()
ham_has_upper = df[df['label'] ==
'ham']['message'].str.contains(r'[A-Z]').sum()
print(f"Proportion of spam with uppercase: {spam_has_upper /
total_spam:.2f}")
print(f"Proportion of ham with uppercase: {ham_has_upper /
total_ham:.2f}")
```

7. Presence of Uppercase Letters: How many messages contain at least one uppercase letter?

```python
contains_uppercase = df['message'].str.contains(r'[A-Z]')
print("Number of messages containing at least one uppercase letter:", contains_uppercase.sum())
```

8. Presence of Digits (Spam vs. Ham): What is the proportion of spam messages containing digits compared to ham messages containing digits?

```python
spam_has_digit = df[df['label'] == 'spam']['message'].str.contains(r'\d').sum()
ham_has_digit = df[df['label'] == 'ham']['message'].str.contains(r'\d').sum()
total_spam = df['label'].value_counts()['spam']
total_ham = df['label'].value_counts()['ham']
print(f"Proportion of spam with digits: {spam_has_digit / total_spam:.2f}")
print(f"Proportion of ham with digits: {ham_has_digit / total_ham:.2f}")
```

9. Presence of Digits: How many messages contain at least one digit?

```python
contains_digit = df['message'].str.contains(r'\d')
print("Number of messages containing at least one digit:",
contains_digit.sum())
```

10. Unique Word Count (Ham): How many unique words are there in the ham messages?

```python
print("Number of unique words (ham):", len(word_counts_ham))
```

11. Unique Word Count (Spam): How many unique words are there in the spam messages?

```python
print("Number of unique words (spam):", len(word_counts_spam))
```

12Unique Word Count (Overall): How many unique words are there in the entire dataset?

```python
print("Number of unique words (overall):",
len(word_counts_overall))
```

13. Shortest Message: What is the shortest message in the dataset, and is it spam or ham?

```python
shortest_msg = df.loc[df['message_length'].idxmin()]
print("Shortest message:\n", shortest_msg['message'])
print("Label of the shortest message:", shortest_msg['label'])
```

14. Longest Message: What is the longest message in the dataset, and is it spam or ham?

```python
longest_msg = df.loc[df['message_length'].idxmax()]
print("Longest message:\n", longest_msg['message'])
print("Label of the longest message:", longest_msg['label'])
```

15. Percentage of Spam: What percentage of the total messages are classified as spam?

```python
spam_percentage = (df['label'].value_counts(normalize=True) *
100)['spam']
print(f"Percentage of spam messages: {spam_percentage:.2f}%")
```

16. Most Frequent Words (Ham): What are the top 10 most frequently occurring words specifically in ham messages?

```python
ham_words = ' '.join(df[df['label'] ==
'ham']['message']).lower().split()
word_counts_ham = Counter(ham_words)
print("Top 10 most frequent words (ham):",
word_counts_ham.most_common(10))
```

17. Most Frequent Words (Spam): What are the top 10 most frequently occurring words specifically in spam messages?

```python
spam_words = ' '.join(df[df['label'] ==
'spam']['message']).lower().split()
word_counts_spam = Counter(spam_words)
print("Top 10 most frequent words (spam):",
word_counts_spam.most_common(10))
```

18. Most Frequent Words (Overall): What are the top 10 most frequently occurring words in

the entire dataset?

```python
all_words = ' '.join(df['message']).lower().split()
word_counts_overall = Counter(all_words)
print("Top 10 most frequent words (overall):",
word_counts_overall.most_common(10))
```

19. Message Length Analysis: What is the average length (in characters) of spam

messages versus ham messages?

```python
df['message_length'] = df['message'].apply(len)
print("Average length of spam messages:", df[df['label'] ==
'spam']['message_length'].mean())
print("Average length of ham messages:", df[df['label'] ==
'ham']['message_length'].mean())
```

20. Basic Data Exploration: How many spam and ham (non-spam) messages are there in

the dataset?

```python
print("Number of spam messages:",
df['label'].value_counts()['spam'])
print("Number of ham messages:",
df['label'].value_counts()['ham'])
```