EDS ACTIVITY NO.1

NAME :Lavanya .V. Uikey

DIV : CS6

ROLL NO. : CS62-36

SUBJECT : EDS

```python
import pandas as pd
import numpy as np
import os
import kagglehub # Make sure you have kagglehub installed: pip install kagglehub

# --- 1. Download the dataset ---
print("Downloading SMS Spam Collection Dataset...")
try:
    path = kagglehub.dataset_download("uciml/sms-spam-collection-dataset")
    print(f"Dataset downloaded to: {path}")

    # --- 2. Find the CSV file ---
    file_path = None
    # Look for the CSV file within the downloaded directory and its subdirectories
    # The typical file name is 'spam.csv'
    csv_filename = 'spam.csv'
    print(f"Looking for '{csv_filename}' in downloaded directory...")
    for root, dirs, files in os.walk(path):
        if csv_filename in files:
            file_path = os.path.join(root, csv_filename)
            break
```

```
OBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

 C:\Users\HP\OneDrive\Desktop> python .\vivek_eds_activity_1.py
wnloading SMS Spam Collection Dataset...
taset downloaded to: C:\Users\HP\.cache\kagglehub\datasets\uciml\sms-spam-collection-dataset\versions\1
oking for 'spam.csv' in downloaded directory...
und dataset file at: C:\Users\HP\.cache\kagglehub\datasets\uciml\sms-spam-collection-dataset\versions\1\spam.csv
taset loaded successfully with columns 'Label' and 'Message'.
ded 'Message_Length' column.
ded 'Word_Count' column.

taset Info:
lass 'pandas.core.frame.DataFrame'>
ngeIndex: 5572 entries, 0 to 5571
ta columns (total 4 columns):
   Column          Non-Null Count  Dtype
-  ------          --------------  -----
   Label           5572 non-null   object
   Message         5572 non-null   object
   Message_Length  5572 non-null   int64
   Word_Count      5572 non-null   int64
ypes: int64(2), object(2)
mory usage: 174.3+ KB
```

```
    df = pd.DataFrame() # Initialize an empty DataFrame

    if file_path:
        print(f"Found dataset file at: {file_path}")
        # --- 3. Load the dataset ---
        try:
            # This dataset is often tab-separated, and might not have headers by default
            # Let's try reading with common options or assume it's simple csv
            # Based on typical structure, it's two columns v1 (label) and v2 (text)
            # Let's try reading as CSV first, if that fails, try tab-separated
            try:
                df = pd.read_csv(file_path, encoding='latin-1')
                # Check if columns look like standard v1, v2
                if 'v1' in df.columns and 'v2' in df.columns:
                    df = df[['v1', 'v2']] # Select only relevant columns
                    df.columns = ['Label', 'Message'] # Rename columns for clarity
                    print("Dataset loaded successfully with columns 'Label' and 'Message'.")
                else:
                    # Try reading as tab-separated if CSV didn't yield expected columns
                    print("CSV read didn't yield expected columns. Trying tab separation...")
                    df = pd.read_csv(file_path, sep='\t', header=None, names=['Label', 'Message'], encoding='latin-1')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
 1   Message         5572 non-null   object
 2   Message_Length  5572 non-null   int64
 3   Word_Count      5572 non-null   int64
dtypes: int64(2), object(2)
memory usage: 174.3+ KB

Dataset Head:
  Label                                            Message  Message_Length  Word_Count
0   ham  Go until jurong point, crazy.. Available only ...             111          20
1   ham                      Ok lar... Joking wif u oni...              29           6
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...             155          28
```

```
                print("Dataset loaded successfully with columns 'Label' and 'Message' using tab separation.")

            except Exception as e_read:
                print(f"Error reading CSV/TSV file: {e_read}")


            # --- Initial Data Prep (if loading was successful) ---
            if not df.empty:
                # Add a message length column for analysis
                df['Message_Length'] = df['Message'].str.len()
                print("Added 'Message_Length' column.")

                # Add a word count column (simple split by space)
                df['Word_Count'] = df['Message'].str.split().str.len()
                print("Added 'Word_Count' column.")

                print("\nDataset Info:")
                df.info()
                print("\nDataset Head:")
                print(df.head())

        except Exception as e_load:
            print(f"An error occurred while loading or processing the dataset: {e_load}")
            df = pd.DataFrame() # Ensure df is empty if loading fails

    else:
        print(f"Error: '{csv_filename}' not found in the downloaded directory or its subdirectories.")
        df = pd.DataFrame() # Ensure df is empty if file not found

except Exception as e_download:
    print(f"An error occurred during dataset download: {e_download}")
    df = pd.DataFrame() # Ensure df is empty if download fails
```

PROBLEM STATEMENTS AND ANSWERS :

1 TO 5

```python
# Problem Statement 1: What is the total number of messages in the dataset?
total_messages = len(df)
print(f"\n1. Total number of messages: {total_messages}")

# Problem Statement 2: How many messages are labeled as 'spam'?
spam_count = df[df['Label'] == 'spam'].shape[0]
print(f"\n2. Number of spam messages: {spam_count}")

# Problem Statement 3: How many messages are labeled as 'ham'?
ham_count = df[df['Label'] == 'ham'].shape[0]
print(f"\n3. Number of ham messages: {ham_count}")

# Problem Statement 4: What is the proportion of spam messages in the dataset?
proportion_spam = spam_count / total_messages if total_messages > 0 else 0
print(f"\n4. Proportion of spam messages: {proportion_spam:.2%}")

# Problem Statement 5: What is the proportion of ham messages in the dataset?
proportion_ham = ham_count / total_messages if total_messages > 0 else 0
print(f"\n5. Proportion of ham messages: {proportion_ham:.2%}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
1   ham                      Ok lar... Joking wif u oni...             29          6
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...            155         28
3   ham  U dun say so early hor... U c already then say...             49         11
4   ham  Nah I don't think he goes to usf, he lives aro...             61         13

--- Performing Data Analysis Tasks ---

1. Total number of messages: 5572

2. Number of spam messages: 747

3. Number of ham messages: 4825

4. Proportion of spam messages: 13.41%

5. Proportion of ham messages: 86.59%
```

6 TO 8 :

```python
    # Problem Statement 6: Display the column names of the DataFrame.
    column_names = df.columns.tolist()
    print(f"\n6. Column names: {column_names}")

    # Problem Statement 7: Find the message with the longest character length.
    if 'Message_Length' in df.columns and 'Message' in df.columns:
        longest_message_index = df['Message_Length'].idxmax()
        longest_message = df.loc[longest_message_index]
        print(f"\n7. Message with the longest character length ({longest_message['Message_Length']} characters):")
        print(f"   Label: {longest_message['Label']}")
        print(f"   Message: {longest_message['Message'][:100]}...") # Print first 100 chars
    else:
        print("\n7. Required column(s) ('Message_Length' or 'Message') not found.")


    # Problem Statement 8: Find the message with the shortest character length (excluding empty messages).
    if 'Message_Length' in df.columns and 'Message' in df.columns:
        shortest_message_index = df[df['Message_Length'] > 0]['Message_Length'].idxmin()
        shortest_message = df.loc[shortest_message_index]
        print(f"\n8. Message with the shortest character length ({shortest_message['Message_Length']} characters):")
        print(f"   Label: {shortest_message['Label']}")
        print(f"   Message: {shortest_message['Message']}")
    else:
        print("\n8. Required column(s) ('Message_Length' or 'Message') not found.")


    # Problem Statement 9: Calculate the average character length of all messages.
    if 'Message_Length' in df.columns:
        average_length = df['Message_Length'].mean()
        print(f"\n9. Average message character length: {average_length:.2f}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
6. Column names: ['Label', 'Message', 'Message_Length', 'Word_Count']

7. Message with the longest character length (910 characters):
   Label: ham
   Message: For me the love should start with attraction.i should feel that I need her every time around me.she ...

8. Message with the shortest character length (2 characters):
   Label: ham
   Message: Ok
```

9 TO 11:

```
127
128
129     # Problem Statement 9: Calculate the average character length of all messages.
130     if 'Message_Length' in df.columns:
131         average_length = df['Message_Length'].mean()
132         print(f"\n9. Average message character length: {average_length:.2f}")
133     else:
134         print("\n9. 'Message_Length' column not found.")
135
136
137     # Problem Statement 10: Calculate the average character length of spam messages.
138     if 'Message_Length' in df.columns and 'Label' in df.columns:
139         spam_messages = df[df['Label'] == 'spam']
140         if not spam_messages.empty:
141             average_length_spam = spam_messages['Message_Length'].mean()
142             print(f"\n10. Average character length of spam messages: {average_length_spam:.2f}")
143         else:
144             print("\n10. No spam messages found.")
145     elif not df.empty:
146         print("\n10. Required column(s) ('Message_Length' or 'Label') not found.")
147
148
149     # Problem Statement 11: Calculate the average character length of ham messages.
150     if 'Message_Length' in df.columns and 'Label' in df.columns:
151         ham_messages = df[df['Label'] == 'ham']
152         if not ham_messages.empty:
153             average_length_ham = ham_messages['Message_Length'].mean()
154             print(f"\n11. Average character length of ham messages: {average_length_ham:.2f}")
155         else:
156             print("\n11. No ham messages found.")
157     elif not df.empty:
158         print("\n11. Required column(s) ('Message_Length' or 'Label') not found.")
159
160
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
   Label: ham
   Message: Ok

9. Average message character length: 80.12

10. Average character length of spam messages: 138.87

11. Average character length of ham messages: 71.02

12. First 5 messages containing 'free':
```

12 TO 14:

```python
# Problem Statement 12: Find and display the first 5 messages containing the word 'free'.
if 'Message' in df.columns:
    keyword = 'free'
    messages_with_keyword = df[df['Message'].str.contains(keyword, case=False, na=False)].head()
    print(f"\n12. First 5 messages containing '{keyword}':")
    if not messages_with_keyword.empty:
        for index, row in messages_with_keyword.iterrows():
            print(f"   - Label: {row['Label']}, Message: {row['Message'][:80]}...") # Print first 80 chars
    else:
        print(f"   No messages found containing '{keyword}'.")
else:
    print("\n12. 'Message' column not found.")


# Problem Statement 13: Count how many messages contain the word 'win'.
if 'Message' in df.columns:
    keyword = 'win'
    count_with_keyword = df['Message'].str.contains(keyword, case=False, na=False).sum()
    print(f"\n13. Number of messages containing '{keyword}': {count_with_keyword}")
else:
    print("\n13. 'Message' column not found.")


# Problem Statement 14: Display the value counts for the 'Label' column.
if 'Label' in df.columns:
    label_counts = df['Label'].value_counts()
    print(f"\n14. Value counts for 'Label' column:")
    print(label_counts)
else:
    print("\n14. 'Label' column not found.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
12. First 5 messages containing 'free':
   - Label: spam, Message: Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 8...
   - Label: spam, Message: FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some...
   - Label: spam, Message: Had your mobile 11 months or more? U R entitled to Update to the latest colour m...
   - Label: spam, Message: URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Tx...
   - Label: spam, Message: 07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms...

13. Number of messages containing 'win': 166


14. Value counts for 'Label' column:
Label
ham     4825
spam     747
Name: count, dtype: int64
```

15 AND 16 :

```python
    # Problem Statement 15: Find the message with the most words.
    if 'Word_Count' in df.columns and 'Message' in df.columns:
        longest_word_count_index = df['Word_Count'].idxmax()
        message_most_words = df.loc[longest_word_count_index]
        print(f"\n15. Message with the most words ({message_most_words['Word_Count']} words):")
        print(f"   Label: {message_most_words['Label']}")
        print(f"   Message: {message_most_words['Message'][:100]}...") # Print first 100 chars
    else:
        print("\n15. Required column(s) ('Word_Count' or 'Message') not found.")


    # Problem Statement 16: Display the first 10 spam messages.
    if 'Label' in df.columns and 'Message' in df.columns:
        first_10_spam = df[df['Label'] == 'spam'].head(10)
        print(f"\n16. First 10 spam messages:")
        if not first_10_spam.empty:
            for index, row in first_10_spam.iterrows():
                print(f"   - {row['Message'][:100]}...") # Print first 100 chars
        else:
            print("   No spam messages found.")
    elif not df.empty:
        print("\n16. Required column(s) ('Label' or 'Message') not found.")


    # Problem Statement 17: Display the first 10 ham messages
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
15. Message with the most words (171 words):
   Label: ham
   Message: For me the love should start with attraction.i should feel that I need her every time around me.she ...

16. First 10 spam messages:
   - Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entr...
   - FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it s...
   - WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To clai...
   - Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera f...
   - SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6day...
   - URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the word: CLAIM to...
   - XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> h...
   - England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 870...
   - Thanks for your subscription to Ringtone UK your mobile will be charged å£5/month Please confirm by ...
   - 07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free nokia mo...
   - 07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free nokia mo...
```

17 AND 18:

```
216
217     # Problem Statement 17: Display the first 10 ham messages.
218     if 'Label' in df.columns and 'Message' in df.columns:
219         first_10_ham = df[df['Label'] == 'ham'].head(10)
220         print(f"\n17. First 10 ham messages:")
221         if not first_10_ham.empty:
222             for index, row in first_10_ham.iterrows():
223                 print(f"   - {row['Message'][:100]}...") # Print first 100 chars
224         else:
225             print("   No ham messages found.")
226     elif not df.empty:
227         print("\n17. Required column(s) ('Label' or 'Message') not found.")
228
229
230     # Problem Statement 18: Calculate the standard deviation of message length.
231     if 'Message_Length' in df.columns:
232         std_dev_length = df['Message_Length'].std()
233         print(f"\n18. Standard deviation of message character length: {std_dev_length:.2f}")
234     else:
235         print("\n18. 'Message_Length' column not found.")
236
237
238     # Problem Statement 19: Calculate the average word count for spam messages.
239     if 'Word_Count' in df.columns and 'Label' in df.columns:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
17. First 10 ham messages:
   - Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a...
   - Ok lar... Joking wif u oni.....
   - U dun say so early hor... U c already then say.....
   - Nah I don't think he goes to usf, he lives around here though...
   - Even my brother is not like to speak with me. They treat me like aids patent....
   - As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertu...
   - I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enou...
   - I've been searching for the right words to thank you for this breather. I promise i wont take your h...
   - I HAVE A DATE ON SUNDAY WITH WILL!!...
   - Oh k...i'm watching here:)...

18. Standard deviation of message character length: 59.69
```

19 AND 20:

```
    # Problem Statement 19: Calculate the average word count for spam messages.
    if 'Word_Count' in df.columns and 'Label' in df.columns:
        spam_messages = df[df['Label'] == 'spam']
        if not spam_messages.empty:
            average_word_count_spam = spam_messages['Word_Count'].mean()
            print(f"\n19. Average word count for spam messages: {average_word_count_spam:.2f}")
        else:
            print("\n19. No spam messages found.")
    elif not df.empty:
        print("\n19. Required column(s) ('Word_Count' or 'Label') not found.")


    # Problem Statement 20: Count messages where the character length is greater than 200.
    if 'Message_Length' in df.columns:
        long_messages_count = df[df['Message_Length'] > 200].shape[0]
        print(f"\n20. Number of messages with character length > 200: {long_messages_count}")
    else:
        print("\n20. 'Message_Length' column not found.")

else:
    print("\nDataFrame is empty. Cannot perform analysis tasks.")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

18. Standard deviation of message character length: 59.69

19. Average word count for spam messages: 23.85

20. Number of messages with character length > 200: 112
PS C:\Users\HP\OneDrive\Desktop\vvk eds assg1>
```

THANKYOU