**PERFORM EXPLORATORY DATA ANALYSIS (EDA) WITH DATASETS LIKE EMAIL DATA SET**

**DATASET LINK: Email Spam Detection Dataset**  
You can use the classic email spam dataset from UCI.  
**Dataset URL:**  
<https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>

**LAB EXERCISE ALGORITHM (Step-by-Step EDA)**  
**Algorithm: EDA on Email (SMS) Spam Dataset**

1. Import necessary libraries
2. Load the dataset
3. Understand dataset structure (head, info, describe)
4. Check for null/missing values
5. Analyze class distribution (spam vs ham)
6. Visualize data distribution using plots
7. Perform text-specific analysis
   * Message length
   * Word frequency
   * Word clouds for spam and ham
8. Check correlations (if any)
9. Conclude insights from data

**Program:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv(r'D:\College Files\DEV\spam.csv', encoding='latin-1')

df = df[['v1', 'v2']]

df.columns = ['label', 'message']

df['label\_num'] = np.where(df['label'] == 'spam', 1, 0)

print(df.head())

print(df.info())

print(df.describe())

print("Missing values:\n", df.isnull().sum())

spam\_count = df['label'].value\_counts()

plt.figure(figsize=(6,4))

plt.bar(spam\_count.index, spam\_count.values, color=['blue', 'red'])

plt.xlabel("Message Type")

plt.ylabel("Count")

plt.title("Spam vs Ham Distribution")

plt.show()

df['message\_length'] = df['message'].apply(len)

plt.figure(figsize=(8,5))

plt.hist(df[df['label'] == 'spam']['message\_length'], bins=30, alpha=0.5, color='red', label='Spam')

plt.hist(df[df['label'] == 'ham']['message\_length'], bins=30, alpha=0.5, color='blue', label='Ham')

plt.xlabel("Message Length")

plt.ylabel("Frequency")

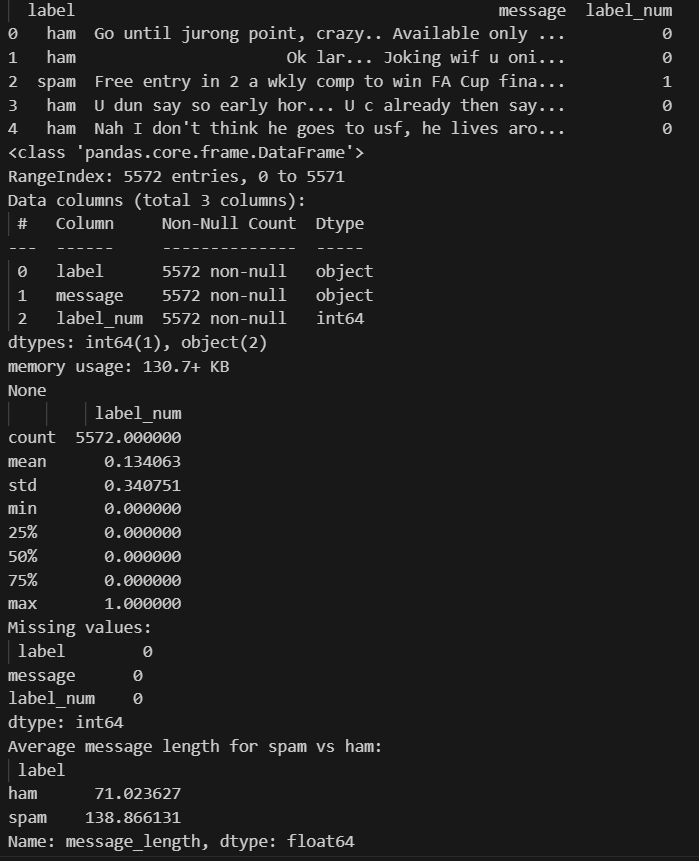
plt.title("Message Length Distribution")

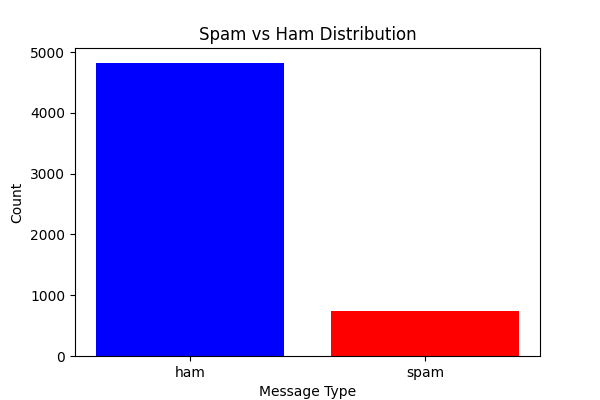
plt.legend()

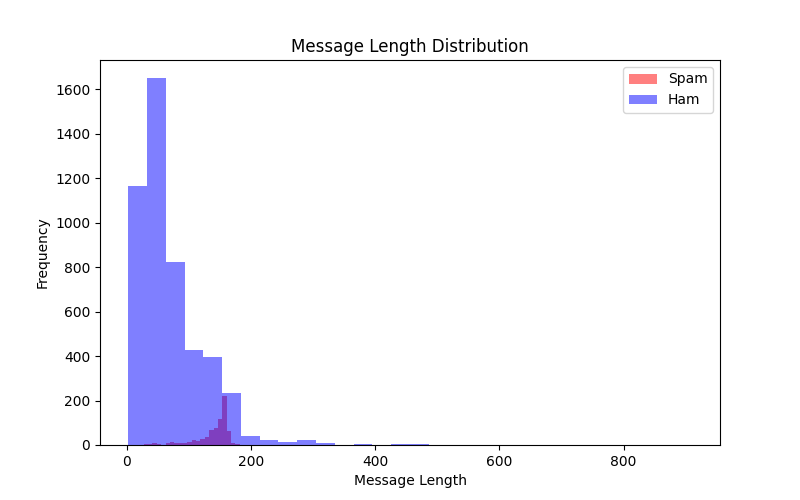
plt.show()

print("Average message length for spam vs ham:\n", df.groupby('label')['message\_length'].mean())

**Output**:







**Result:** Thus the program to implement exploratory data analysis in email datasets has been successfully executed and verified