DATA EXLORATION AND VISUALIZATION

EXPERIMENT – 2

EXPLORATORY DATA ANALYSIS USING E-MAIL DATA SET

PROGRAM:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from wordcloud import WordCloud

import nltk

from nltk.corpus import stopwords

import string

nltk.download('stopwords')

df = pd.read\_csv("C:\\Users\\ROHITH\\Downloads\\spam (1).csv", encoding='latin-1')

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

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

print(df.head())

print(df.info())

print(df.describe())

print(df.isnull().sum())

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

sns.countplot(data=df, x='label', palette='Set2')

plt.title('Distribution of Spam vs Ham')

plt.show()

print(df['label'].value\_counts(normalize=True))df['message\_length'] = df['message'].apply(len)

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

sns.histplot(data=df, x='message\_length', hue='label', bins=50, kde=True, palette='Set1')

plt.title('Distribution of Message Lengths')

plt.xlabel('Message Length')

plt.ylabel('Count')

plt.show()

def preprocess\_text(text):

text = text.lower()

text = ''.join([ch for ch in text if ch not in string.punctuation])

words = text.split()

words = [w for w in words if w not in stopwords.words('english')]

return words

df['tokens'] = df['message'].apply(preprocess\_text)

spam\_words = df[df['label']=='spam']['tokens'].sum()

ham\_words = df[df['label']=='ham']['tokens'].sum()

spam\_wc = WordCloud(width=600, height=400, background\_color='black').generate(' '.join(spam\_words))

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

plt.imshow(spam\_wc, interpolation='bilinear')

plt.axis('off')

plt.title('Spam Word Cloud')

plt.show()

ham\_wc = WordCloud(width=600, height=400, background\_color='white', colormap='viridis').generate(' '.join(ham\_words))

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

plt.imshow(ham\_wc, interpolation='bilinear')

plt.axis('off')

plt.title('Ham Word Cloud')

plt.show()

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

sns.boxplot(x='label', y='message\_length', data=df)

plt.title('Message Length by Class')

plt.show()

print("Average Message Length (Overall):", df['message\_length'].mean())

print("Average Message Length (Spam):", df[df['label'] == 'spam']['message\_length'].mean())

print("Average Message Length (Ham):", df[df['label'] == 'ham']['message\_length'].mean())

q1 = df['message\_length'].quantile(0.25)

q3 = df['message\_length'].quantile(0.75)

iqr = q3 - q1

outlier\_limit = q3 + 1.5 \* iqr

outliers = df[df['message\_length'] > outlier\_limit]

print(f"Number of outlier messages (length > {outlier\_limit:.2f}):", outliers.shape[0])

OUTPUT:









