ASSIGNMENT 06

NAME: Bhosale Srushti Amar

CLASS: AIDS-A

ROLL NO.: 23107008

```
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import classification_report

df = pd.read_csv("Downloads/spam.csv", encoding="latin-1")

df = df[['v1', 'v2']]
df.columns = ['label', 'message']

df
```

```
     label                                            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...
...    ...                                              ...
5567  spam  This is the 2nd time we have tried 2 contact u...
5568   ham              Will Ì_ b going to esplanade fr home?
5569   ham  Pity, * was in mood for that. So...any other s...
5570   ham  The guy did some bitching but I acted like i'd...
5571   ham                         Rofl. Its true to its name

[5572 rows x 2 columns]
```

```
df.shape
```

```
(5572, 2)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   label    5572 non-null   object
 1   message  5572 non-null   object
```

```
dtypes: object(2)
memory usage: 87.2+ KB
```

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

```
label      0
message    0
dtype: int64
```

```
import re
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    return text
df['message'] = df['message'].apply(clean_text)

df['length'] = df['message'].apply(len)

Q1 = df['length'].quantile(0.25)
Q3 = df['length'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR
df1= df[(df['length'] >= lower) & (df['length'] <= upper)]
print("Shape after removing outliers:", df.shape)
```

```
Shape after removing outliers: (5572, 3)
```

```
df1
```

	label	message	length
0	ham	go until jurong point crazy available only in ...	102
1	ham	ok lar joking wif u oni	23
2	spam	free entry in a wkly comp to win fa cup final...	124
3	ham	u dun say so early hor u c already then say	43
4	ham	nah i dont think he goes to usf he lives aroun...	59
...	...	...	...
5567	spam	this is the nd time we have tried contact u u...	130
5568	ham	will b going to esplanade fr home	34
5569	ham	pity was in mood for that soany other suggest...	50
5570	ham	the guy did some bitching but i acted like id ...	124
5571	ham	rofl its true to its name	25

```
[5492 rows x 3 columns]
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english', max_features=3000)
X = vectorizer.fit_transform(df['message'])
y = df['label']
```

NAIVE BAYES

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
```

```
    X, y, test_size=0.2, random_state=42
)

from sklearn.naive_bayes import MultinomialNB
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
nb_pred = nb_model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, nb_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, nb_pred))
print("\nClassification Report:\n", classification_report(y_test, nb_pred))
```

```
Accuracy: 0.9739910313901345
Confusion Matrix:
 [[965   0]
 [ 29 121]]

Classification Report:
              precision    recall  f1-score   support

         ham       0.97      1.00      0.99       965
        spam       1.00      0.81      0.89       150

    accuracy                           0.97      1115
   macro avg       0.99      0.90      0.94      1115
weighted avg       0.97      0.97      0.97      1115
```

LOGITIC REGRESSION

```
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)
lr_pred = lr_model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, lr_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, lr_pred))
print("\nClassification Report:")
print(classification_report(y_test, lr_pred))
```

```
Accuracy: 0.9524663677130045

Confusion Matrix:
[[963   2]
 [ 51  99]]

Classification Report:
              precision    recall  f1-score   support

         ham       0.95      1.00      0.97       965
        spam       0.98      0.66      0.79       150

    accuracy                           0.95      1115
```

```
   macro avg       0.96      0.83      0.88      1115
weighted avg       0.95      0.95      0.95      1115
```