BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

TEAM MEMBER
311121205003-Ajayroshan R

PHASE 4- DOCUMENT SUBMISSION

Project: Building a Smarter AI-Powered Spam Classifier

ML ALGORITHMS IN PYTHON FOR HAM AND SPAM MESSAGE CLASSIFICATION

Program Code for Loading and Preprocessing Dataset:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_csv("/content/drive/MyDrive/Mail Spam-NLP/spam.csv",
encoding="ISO-8859-1")

df.head(10)
```

Out[ ]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
5	spam	FreeMsg Hey there darling it's been 3 week's n...	NaN	NaN	NaN
6	ham	Even my brother is not like to speak with me. ...	NaN	NaN	NaN
7	ham	As per your request 'Melle Melle (Oru Minnamin...	NaN	NaN	NaN
8	spam	WINNER!! As a valued network customer you have...	NaN	NaN	NaN
9	spam	Had your mobile 11 months or more? U R entitle...	NaN	NaN	NaN

```
#Drop empty columns
cols = [2,3,4]
df.drop(df.columns[cols],axis=1,inplace=True)
```

df.head(10)

Out[ ]:

	v1	v2
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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

#Rename columns as category and message
df.rename(columns = {'v1':'Category', 'v2':'Message'}, inplace = True)

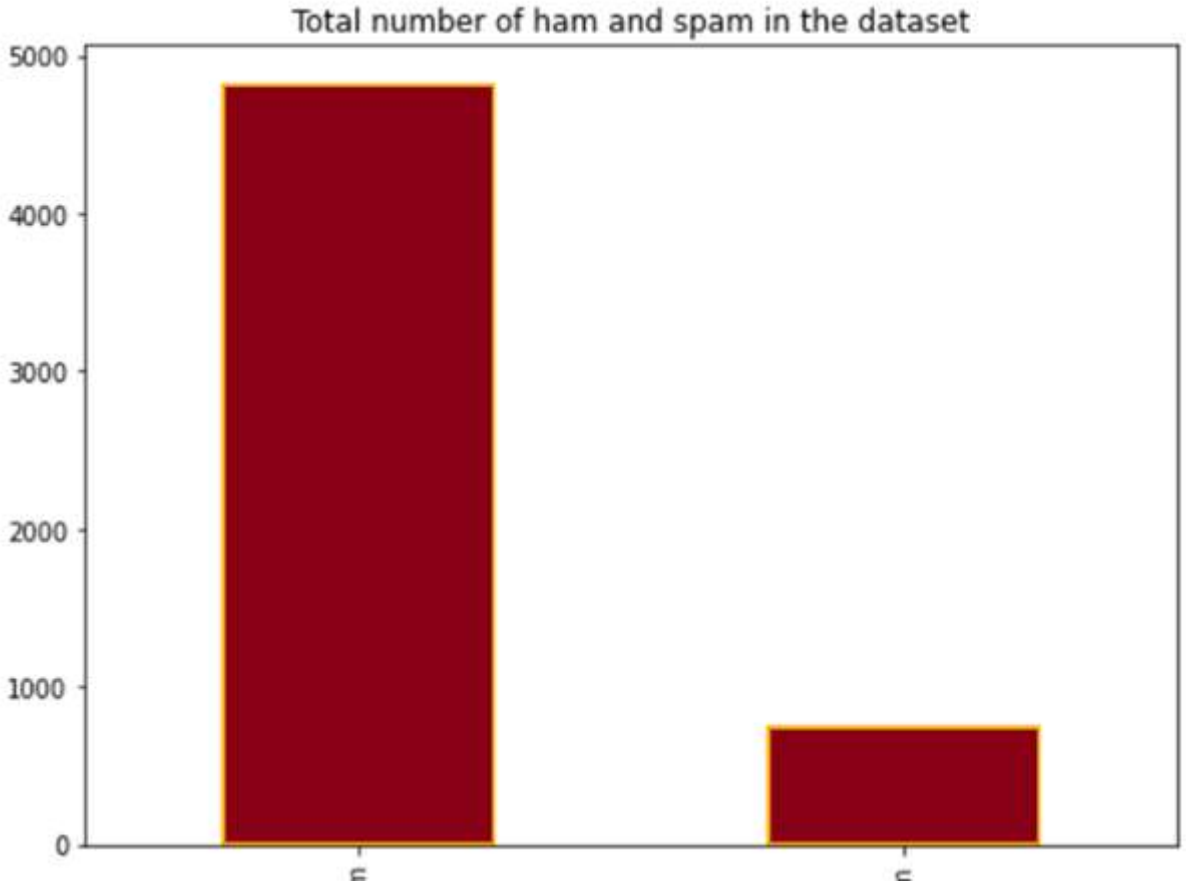
df.head(10)

Out[ ]:

	Category	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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

```
print(f'Dataset consist of {df.shape[0]} E-Mails.')
df['Category'].value_counts()
plt.figure(figsize=(8,6))
df['Category'].value_counts().plot.bar(color = ["orange","orange"])
plt.title('Total number of ham and spam in the dataset')
```

plt.show()



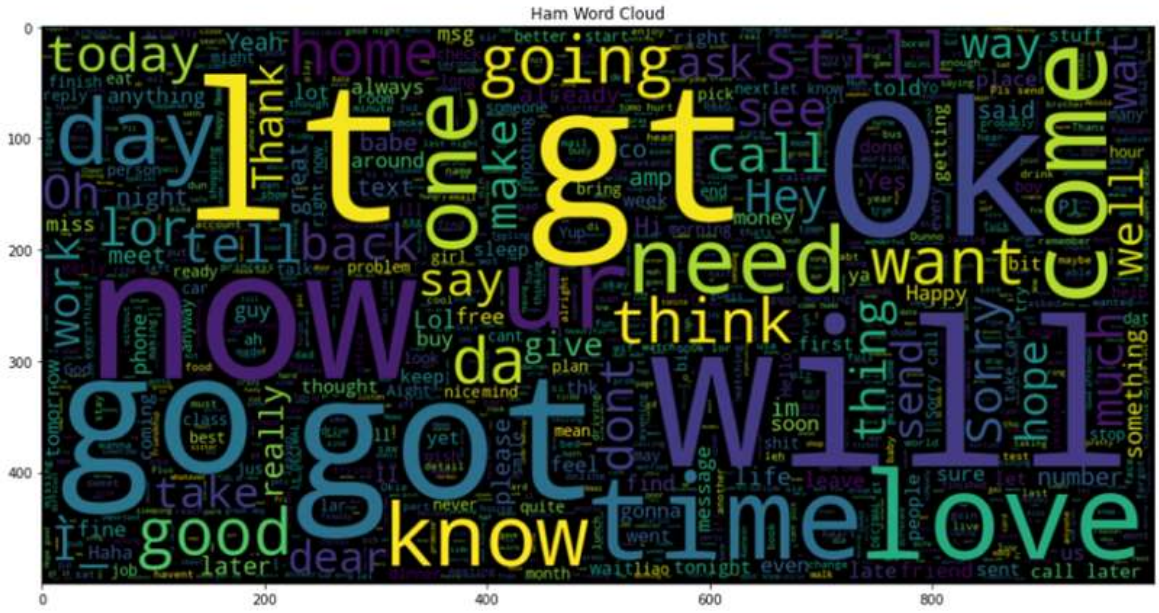
Program Code for Feature Extraction and Classification:

```
# WordCloud
from wordcloud import WordCloud
plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 2000 , width = 1000 , height = 500).generate("
".join(df[df.Category =="ham" ].Message))
plt.imshow(wc , interpolation = 'bilinear')
plt.title("Ham Word Cloud")

plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 2000 , width = 1000 , height = 500).generate("
".join(df[df.Category =="spam" ].Message))
plt.imshow(wc , interpolation = 'bilinear')
```

plt.title("Spam Word Cloud")

Out[ ]: Text(0.5, 1.0, 'Ham Word Cloud')



Test-Train Split

#0: Ham, 1: Spam
df['Category']=df['Category'].apply(lambda x: 1 if x=='spam' else 0)
df.head()

Out[ ]:

	Category	Message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

X=df['Message']
Y=df['Category']
X_train, X_test, y_train, y_test = train_test_split(X,Y)

```
# Naive Baised Model
#Defineing Naive Baised
clf_NaiveBaised= Pipeline([
    ('vectorizer', CountVectorizer()),
    ('nd', MultinomialNB())
])
 #Fiting the algorithm
clf_NaiveBaised.fit(X_train,y_train)
```

```
Out[ ]: Pipeline(steps=[('vectorizer', CountVectorizer()), ('nd', MultinomialNB())])
```

```
#Make prediction on X_test
y_pred_NB=clf_NaiveBaised.predict(X_test)
```

```
Out[ ]: 0.9820531227566404
```

```
# Random Forest Model
clf_rf= Pipeline([
    ('vectorizer', CountVectorizer()),
    ('rf', RandomForestClassifier(n_estimators=100))
])
clf_rf.fit(X_train,y_train)
```

```
Out[ ]: Pipeline(steps=[('vectorizer', CountVectorizer()),
                       ('rf', RandomForestClassifier())])
```

```
y_pred_RF=clf_rf.predict(X_test)
rf_acc=accuracy_score(y_test,y_pred_RF)
```

```
Out[ ]: 0.9712849964106246
```

```
rf_acc
```