```
import numpy as np
import pandas as pd
```

```
dataset = pd.read_csv("/content/SMSSpamCollection",sep='\t',names=['label','message'])
```

```
dataset
```

	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 ü 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 × 2 columns

```
dataset.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
```

```
dataset.describe()
```

	label	message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

```
dataset['label']=dataset['label'].map({"ham": 0,"spam": 1})
```

```
dataset
```

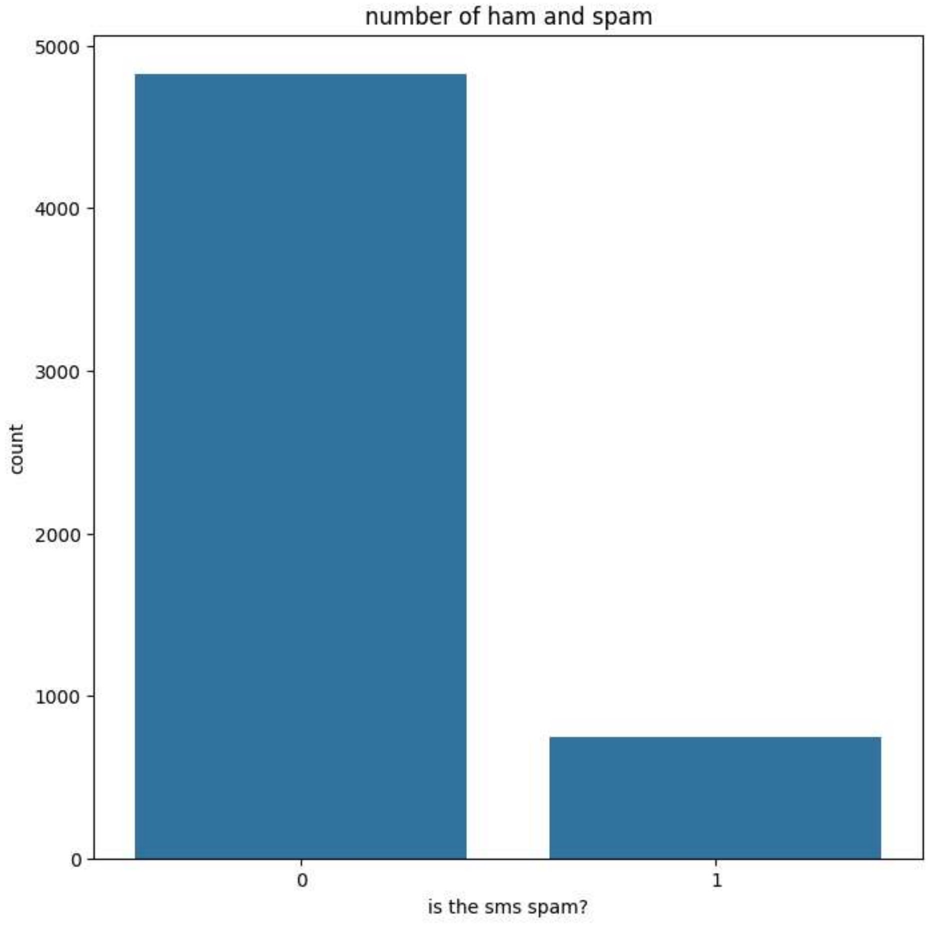
	label	message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	1	This is the 2nd time we have tried 2 contact u...
5568	0	Will ü b going to esplanade fr home?
5569	0	Pity, * was in mood for that. So...any other s...
5570	0	The guy did some bitching but I acted like i'd...
5571	0	Rofl. Its true to its name

5572 rows × 2 columns

```
#visualizing of data
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
#countplot for spam and ham as unbalanced dataset
plt.figure(figsize=(8,8))
g = sns.countplot(x="label",data=dataset)
plt.title("number of ham and spam")
plt.xlabel("is the sms spam?")
plt.ylabel("count")
```

```
Text(0, 0.5, 'count')
```



```
only_spam = dataset[dataset["label"]==1]
```

```
only_spam
```

	label	message
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
5	1	FreeMsg Hey there darling it's been 3 week's n...
8	1	WINNER!! As a valued network customer you have...
9	1	Had your mobile 11 months or more? U R entitle...
11	1	SIX chances to win CASH! From 100 to 20,000 po...
...	...	...
5537	1	Want explicit SEX in 30 secs? Ring 02073162414...
5540	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...
5547	1	Had your contract mobile 11 Mnths? Latest Moto...
5566	1	REMINDER FROM O2: To get 2.50 pounds free call...
5567	1	This is the 2nd time we have tried 2 contact u...

747 rows × 2 columns

```
count = int((dataset.shape[0]-only_spam.shape[0])/only_spam.shape[0])
```

```
count
```

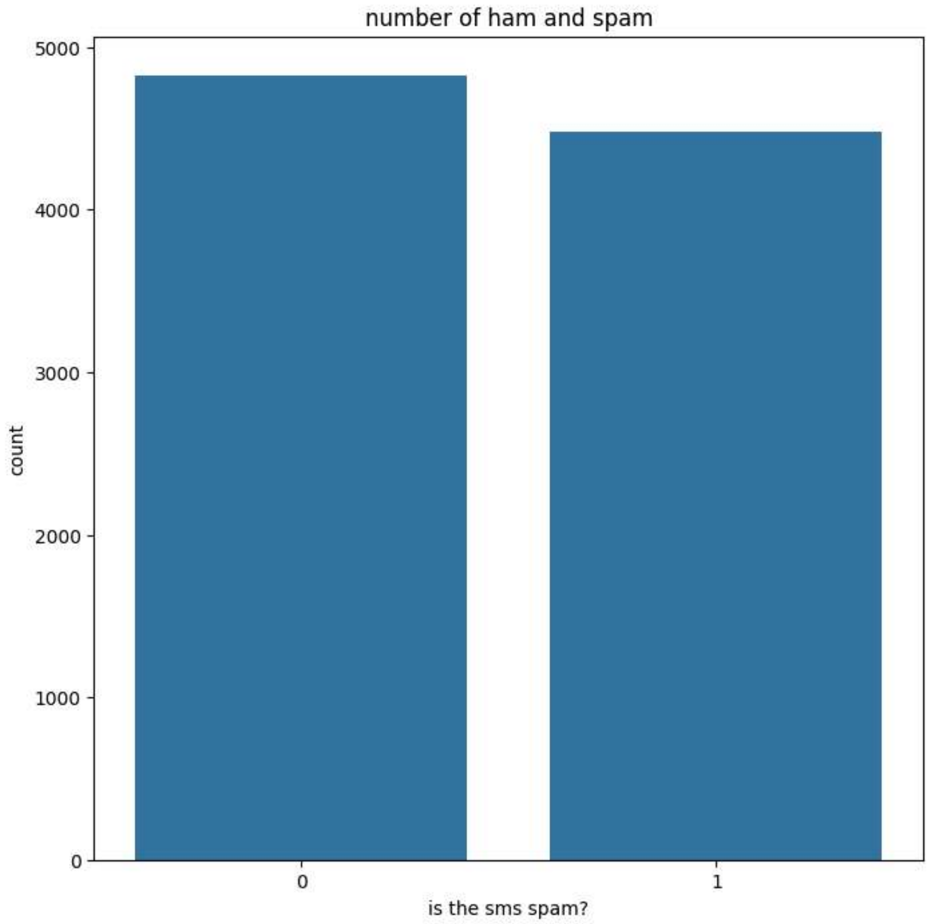
```
6
```

```
for i in range(0,count-1):
    dataset = pd.concat([dataset,only_spam])
dataset.shape
```

```
(9307, 2)
```

```
plt.figure(figsize=(8,8))
g = sns.countplot(x="label",data=dataset)
plt.title("number of ham and spam")
plt.xlabel("is the sms spam?")
plt.ylabel("count")
```

```
Text(0, 0.5, 'count')
```



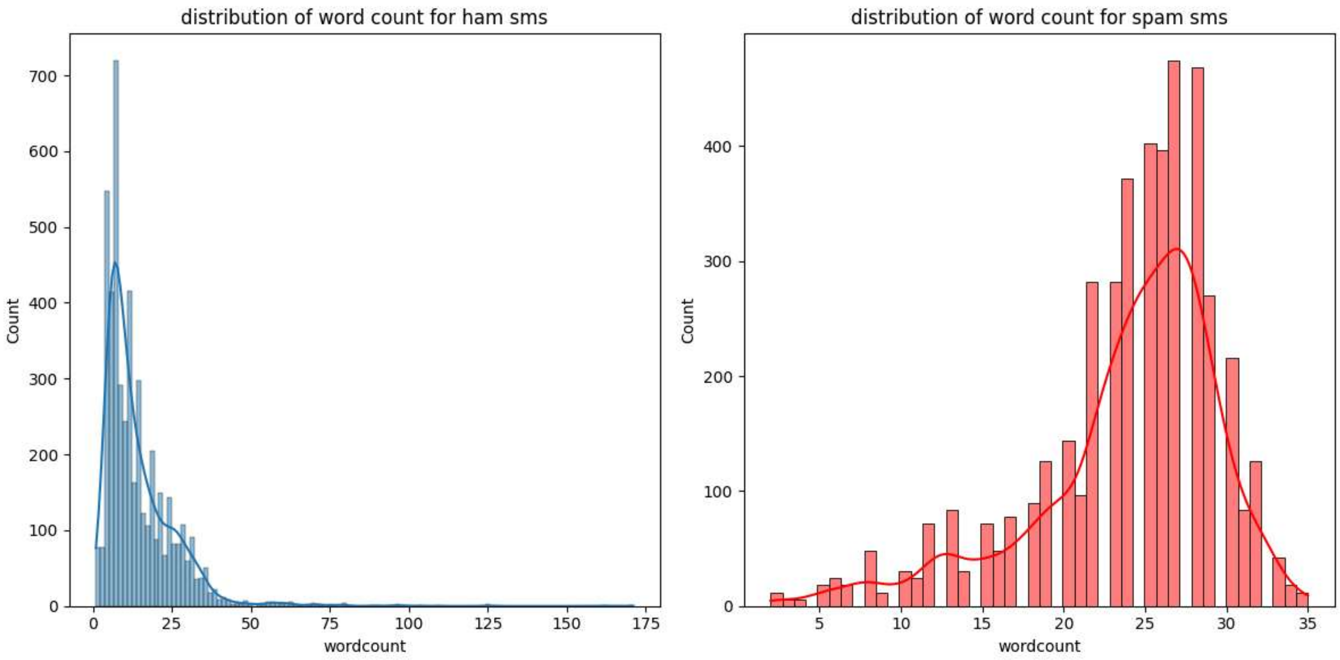
```
dataset["wordcount"] = dataset["message"].apply(lambda X:len(X.split()))
```

```
dataset
```

	label	message	wordcount
0	0	Go until jurong point, crazy.. Available only ...	20
1	0	Ok lar... Joking wif u oni...	6
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	28
3	0	U dun say so early hor... U c already then say...	11
4	0	Nah I don't think he goes to usf, he lives aro...	13
...	...	...	...
5537	1	Want explicit SEX in 30 secs? Ring 02073162414...	16
5540	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...	33
5547	1	Had your contract mobile 11 Mnths? Latest Moto...	28
5566	1	REMINDER FROM O2: To get 2.50 pounds free call...	28
5567	1	This is the 2nd time we have tried 2 contact u...	30

9307 rows × 3 columns

```
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
g = sns.histplot(dataset[dataset["label"] == 0].wordcount,kde=True)
plt.title("distribution of word count for ham sms")
plt.subplot(1,2,2)
g = sns.histplot(dataset[dataset["label"] == 1].wordcount,color="red",kde=True)
plt.title("distribution of word count for spam sms")
plt.tight_layout()
plt.show()
```



```
def currency(data):
    currency_symbols =['$','₹','€','¥','£']
    for i in currency_symbols:
         if i in data:
           return 1
    return 0
```

Double-click (or enter) to edit

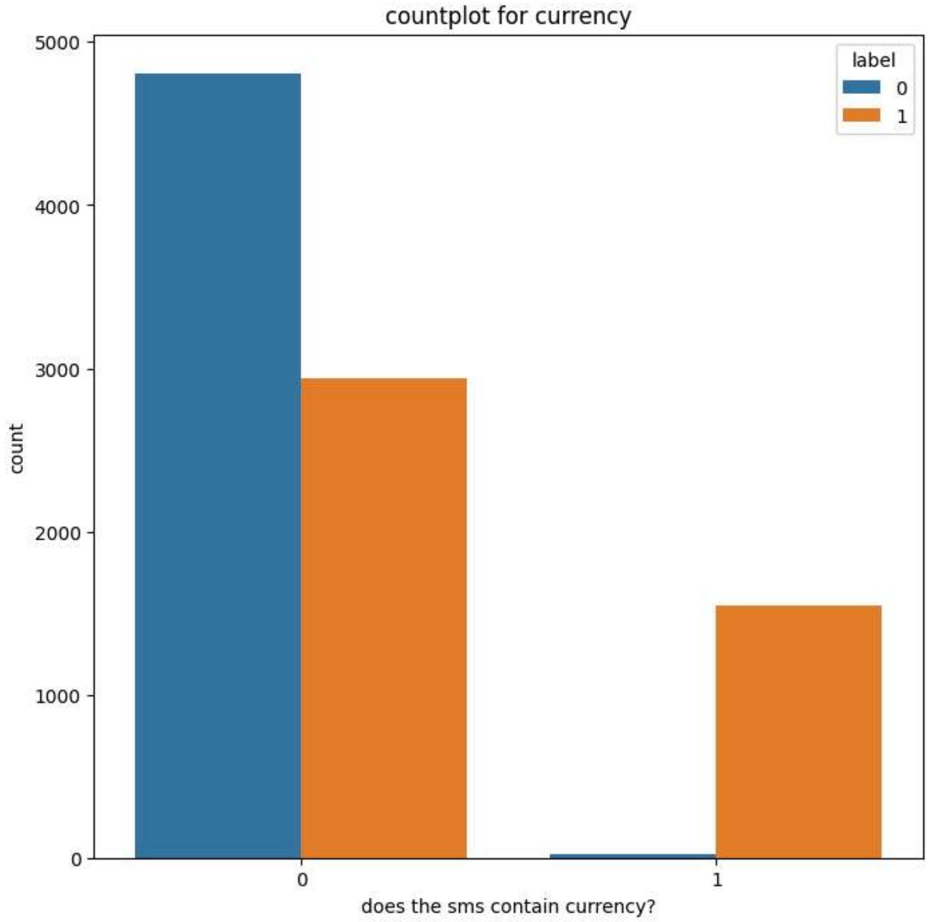
```
dataset["currency"] = dataset["message"].apply(currency)
```

```
dataset
```

	label	message	wordcount	currency
0	0	Go until jurong point, crazy.. Available only ...	20	0
1	0	Ok lar... Joking wif u oni...	6	0
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	28	0
3	0	U dun say so early hor... U c already then say...	11	0
4	0	Nah I don't think he goes to usf, he lives aro...	13	0
...	...	...	...	...
5537	1	Want explicit SEX in 30 secs? Ring 02073162414...	16	0
5540	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...	33	1
5547	1	Had your contract mobile 11 Mnths? Latest Moto...	28	0
5566	1	REMINDER FROM O2: To get 2.50 pounds free call...	28	0
5567	1	This is the 2nd time we have tried 2 contact u...	30	1

9307 rows × 4 columns

```
plt.figure(figsize=(8,8))
g = sns.countplot(x="currency",data=dataset,hue = "label")
plt.title("countplot for currency")
plt.xlabel("does the sms contain currency?")
plt.ylabel("count")
plt.show()
```



```
def has_number(data):
    for i in data:
        if ord(i) >= 48 and ord(i) <= 57:
          return 1
    return 0
```

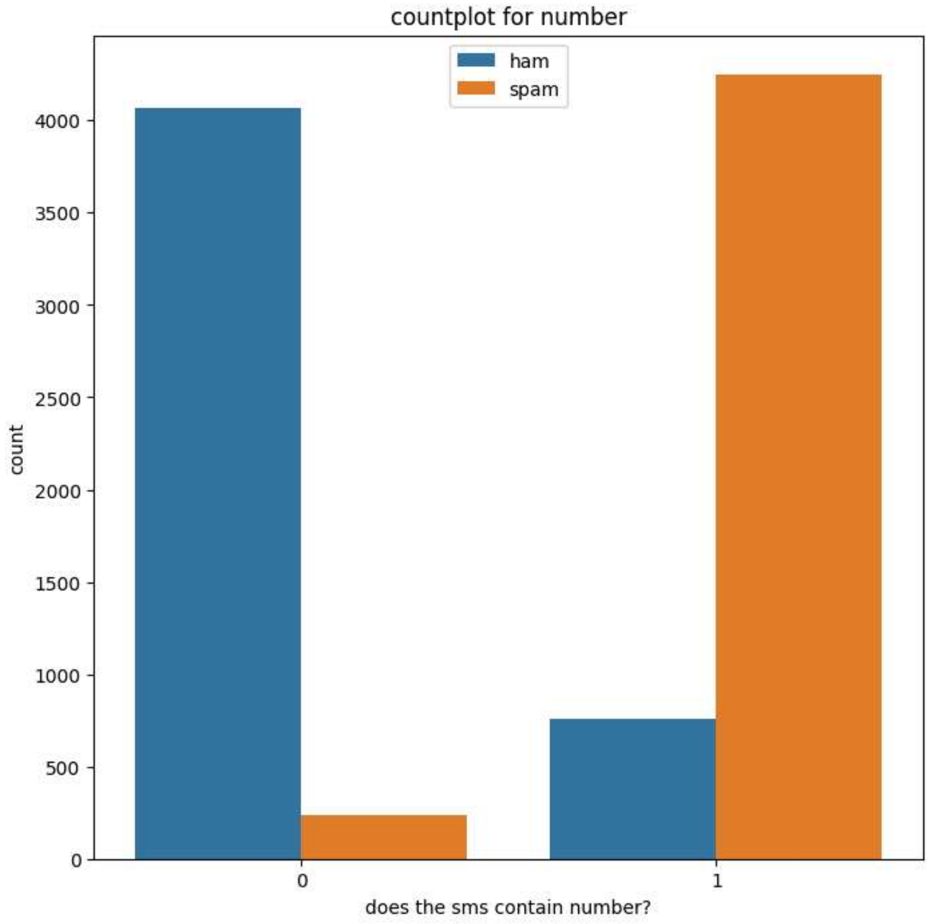
```
dataset["has_number"] = dataset["message"].apply(has_number)
```

```
dataset
```

	label	message	wordcount	currency	has_number
0	0	Go until jurong point, crazy.. Available only ...	20	0	0
1	0	Ok lar... Joking wif u oni...	6	0	0
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	28	0	1
3	0	U dun say so early hor... U c already then say...	11	0	0
4	0	Nah I don't think he goes to usf, he lives aro...	13	0	0
...	...	...	...	...	...
5537	1	Want explicit SEX in 30 secs? Ring 02073162414...	16	0	1
5540	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...	33	1	1
5547	1	Had your contract mobile 11 Mnths? Latest Moto...	28	0	1
5566	1	REMINDER FROM O2: To get 2.50 pounds free call...	28	0	1
5567	1	This is the 2nd time we have tried 2 contact u...	30	1	1

9307 rows × 5 columns

```
plt.figure(figsize=(8,8))
g = sns.countplot(x="has_number",data=dataset,hue = "label")
plt.title("countplot for number")
plt.xlabel("does the sms contain number?")
plt.ylabel("count")
plt.legend(labels=("ham","spam"),loc = 9)
plt.show()
```



```
import nltk
import re
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
corpus = []
wnl = WordNetLemmatizer()
for sms in list(dataset["message"]):
    message = re.sub("[^a-zA-Z]",repl=" ",string=sms)
    message = message.lower()
    words = message.split()
    words = [wnl.lemmatize(word) for word in words if word not in set (stopwords.words("english"))]
    message = " ".join(words)
    corpus.append(message)
```

```
corpus
```

```
['go jurong point crazy available bugis n great world la e buffet cine got amore wat',
 'ok lar joking wif u oni',
 'free entry wkly comp win fa cup final tkts st may text fa receive entry question std txt rate c apply',
 'u dun say early hor u c already say',
 'nah think go usf life around though',
 'freemsg hey darling week word back like fun still tb ok xxx std chgs send rcv',
 'even brother like speak treat like aid patent',
 'per request melle melle oru minnaminunginte nurungu vettam set callertune caller press copy friend callertune',
 'winner valued network customer selected receivea prize reward claim call claim code kl valid hour',
 'mobile month u r entitled update latest colour mobile camera free call mobile update co free',
 'gonna home soon want talk stuff anymore tonight k cried enough today',
 'six chance win cash pound txt csh send cost p day day tsandcs apply reply hl info',
 'urgent week free membership prize jackpot txt word claim c www dbuk net lccltd pobox ldnw rw',
 'searching right word thank breather promise wont take help granted fulfil promise wonderful blessing time',
 'date sunday',
 'xxxmobilemovieclub use credit click wap link next txt message click http wap xxxmobilemovieclub com n qjkgighjjgcbl',
 'oh k watching',
 'eh u remember spell name yes v naughty make v wet',
 'fine way u feel way gota b',
 'england v macedonia dont miss goal team news txt ur national team eg england try wale scotland txt poboxox w wq',
 'seriously spell name',
 'going try month ha ha joking',
 'pay first lar da stock comin',
 'aft finish lunch go str lor ard smth lor u finish ur lunch already',
 'ffffffffff alright way meet sooner',
 'forced eat slice really hungry tho suck mark getting worried know sick turn pizza lol',
 'lol always convincing',
 'catch bus frying egg make tea eating mom left dinner feel love',
 'back amp packing car let know room',
 'ahhh work vaguely remember feel like lol',
 'wait still clear sure sarcastic x want live u',
 'yeah got v apologetic n fallen actin like spoilt child got caught till go badly cheer',
 'k tell anything',
 'fear fainting housework quick cuppa',
 'thanks subscription ringtone uk mobile charged month please confirm replying yes reply charged',
 'yup ok go home look timing msg xuhui going learn nd may lesson',
 'oops let know roommate done',
 'see letter b car',
 'anything lor u decide',
 'hello saturday go texting see decided anything tomo trying invite anything',
 'pls go ahead watt wanted sure great weekend abiola',
 'forget tell want need crave love sweet arabian steed mmmmmm yummy',
 'rodger burn msg tried call reply sm free nokia mobile free camcorder please call delivery tomorrow',
 'seeing',
 'great hope like man well endowed lt gt inch',
 'call message missed call',
 'get hep b immunisation nigeria',
 'fair enough anything going',
 'yeah hopefully tyler could maybe ask around bit',
 'u know stubborn even want go hospital kept telling mark weak sucker hospital weak sucker',
 'thinked first time saw class',
 'gram usually run like lt gt half eighth smarter though get almost whole second gram lt gt',
 'k fyi x ride early tomorrow morning crashing place tonight',
 'wow never realized embarassed accomodations thought liked since best could always seemed happy cave sorry give sorry offered
sorry room embarassing',
 'sm ac sptv new jersey devil detroit red wing play ice hockey correct incorrect end reply end sptv',
 'know mallika sherawat yesterday find lt url gt',
 'congrats year special cinema pas call c suprman v matrix starwars etc free bx ip pm dont miss',
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=2500)
X = cv.fit_transform(corpus).toarray()
feature_names = cv.get_feature_names_out()
X = pd.DataFrame(X, columns= feature_names)
y = dataset["label"]
```

```
from sklearn.model_selection import cross_val_score,train_test_split
from sklearn.metrics import classification_report,confusion_matrix
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

```
X_train
```

	aathi	ab	aberdeen	abiola	able	abroad	abt	abta	abuse	ac	...	yoga	yourinclusive	yr	yt	yup	zebra	zed	zf	zoe	zou
3533	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
2592	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
4253	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
6976	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
7191	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
5734	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
5191	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
5390	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
860	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	
7270	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	

7445 rows × 2500 columns

```
#naive bayes model
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
cv = cross_val_score(model,X,y,scoring='f1',cv=10)
print(round(cv.mean(),3))
print(round(cv.std(),3))
```

```
0.972
0.004
```

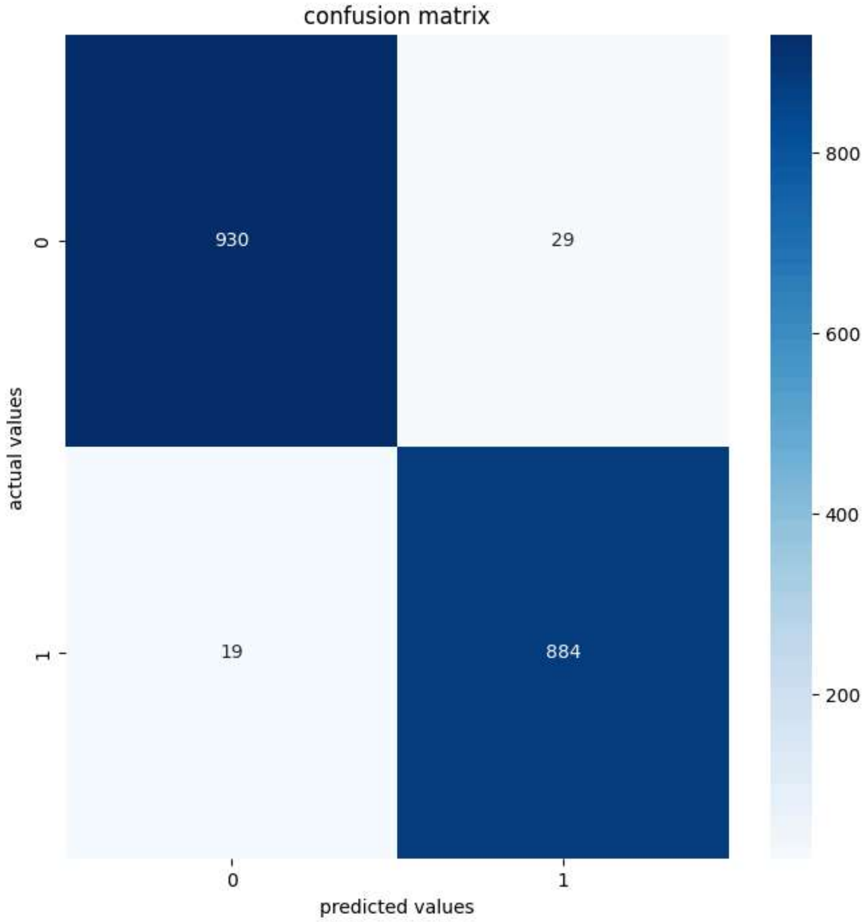
```
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.97      0.97       959
           1       0.97      0.98      0.97       903

    accuracy                           0.97      1862
   macro avg       0.97      0.97      0.97      1862
weighted avg       0.97      0.97      0.97      1862
```

```
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[930  29]
 [ 19 884]]
```

```
plt.figure(figsize=(8,8))
g=sns.heatmap(cm,annot=True,cmap="Blues",fmt="d")
plt.title("confusion matrix")
plt.xlabel("predicted values")
plt.ylabel("actual values")
plt.show()
```



```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
cv = cross_val_score(dt,X,y,scoring='f1',cv=10)
print(round(cv.mean(),3))
print(round(cv.std(),3))
```

```
0.991
0.006
```

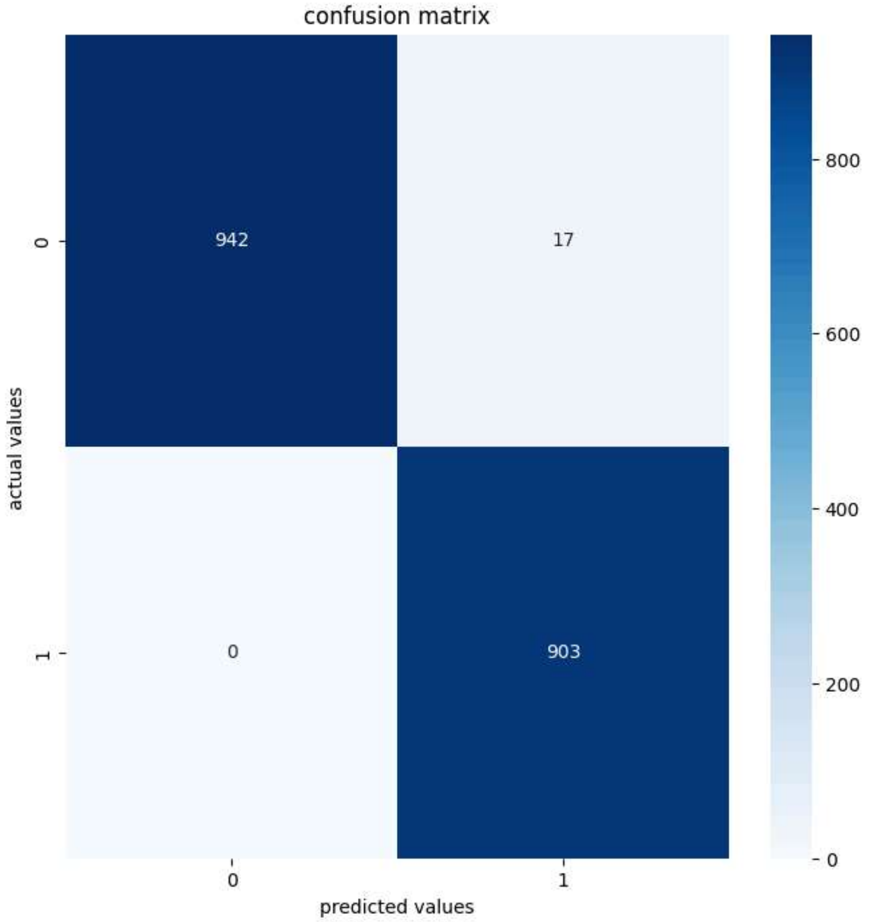
```
dt.fit(X_train,y_train)
y_pred = dt.predict(X_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       959
           1       0.98      1.00      0.99       903

    accuracy                           0.99      1862
   macro avg       0.99      0.99      0.99      1862
weighted avg       0.99      0.99      0.99      1862
```

```
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[942  17]
 [  0 903]]
```

```
plt.figure(figsize=(8,8))
g=sns.heatmap(cm,annot=True,cmap="Blues",fmt="d")
plt.title("confusion matrix")
plt.xlabel("predicted values")
plt.ylabel("actual values")
plt.show()
```



```
# Install NLTK and scikit-learn if not installed
!pip install nltk scikit-learn

import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

# Download NLTK resources if needed
nltk.download('stopwords')
nltk.download('wordnet')

# Sample data (replace with your actual data)
sms_data = [
    ("This is a spam message about winning a prize.", 1),  # 1 for spam
    ("Hey, how are you doing? Want to meet up?", 0),  # 0 for ham
    ("Congratulations! You've won a free iPhone!", 1),
    ("Just checking in to see how you're feeling.", 0)  # Fixed the missing label
]

# Separate messages and labels
messages = [sms[0] for sms in sms_data]
labels = [sms[1] for sms in sms_data]

# Preprocessing and feature extraction
wnl = WordNetLemmatizer()
cv = CountVectorizer(stop_words='english')

# Transform messages to a bag-of-words representation
X = cv.fit_transform(messages).toarray()

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)

# Train the model (DecisionTreeClassifier as an example)
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)

# Function to predict spam/ham
def predict_spam(sms):
    message = re.sub("[^a-zA-Z]", repl=" ", string=sms)  # Remove non-alphabetic characters
```

```
        message = re.sub('[^a-zA-Z]', repl=' ', string=sms)  # Remove non-alphabetic characters
        message = message.lower()
        words = message.split()
        words = [wnl.lemmatize(word) for word in words if word not in set(stopwords.words("english"))]
        message = " ".join(words)
        temp = cv.transform([message]).toarray()  # Transform the input message using the fitted vectorizer
        return dt.predict(temp)[0]  # Return the prediction (0 or 1)

    # Test the function
    test_messages = [
        "hi, how are you",
        "Congratulations! You've won a free gift!",
        "Meeting at 3 PM today?"
    ]
```