```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))



df = pd.read_csv("/content/spam.csv", encoding='ISO-8859-1')


df.head()
```

	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

Next steps: Generate code with df View recommended plots

```
df.drop(["Unnamed: 2" , "Unnamed: 3", "Unnamed: 4" ],axis = 1, inplace = True)


df.head()
```

	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...

Next steps: Generate code with df View recommended plots

```
df.rename(columns = {"v1" : "category",
          "v2" : "message"}, inplace = True)


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

```
category    0
message     0
dtype: int64
```

```
df.duplicated().sum()
```

```
403
```

```
df.drop_duplicates(inplace =True)


df.duplicated().sum()
```

```
0
```

```
df.groupby('category').describe()
```

	message			
	count	unique	top	freq
category				
ham	4516	4516	Go until jurong point, crazy.. Available only ...	1
spam	653	653	Free entry in 2 a wkly comp to win FA Cup fina...	1

```
df["spam_int"] = df["category"].apply(lambda x : 1 if x =="spam" else 0)
```

```
df.head()
```

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

Next steps: Generate code with df View recommended plots

```
x_train , x_test , y_train , y_test = train_test_split(df.message , df.spam_int)
```

```
count_words = CountVectorizer()
x_train_word_count = count_words.fit_transform(x_train)
```

```
model = MultinomialNB()
model.fit(x_train_word_count, y_train)
```

```
MultinomialNB()
```

```
#test ham
ham_email = ["Hi how are you my friend"]
ham_email_count = count_words.transform(ham_email)
confidence = model.predict_proba(ham_email_count)
```

```
if model.predict(ham_email_count) == 0:
        print(f"Prediction: not spam - Confidence: {confidence[0][1]:.2f} - Message: {ham_email[0]}")
```

```
Prediction: not spam - Confidence: 0.98 - Message: Hi how are you my friend
```

```
#test spam
spam_email = ["you won 1millon dollar click here to claim"]
spam_email_count = count_words.transform(spam_email)
confidence = model.predict_proba(spam_email_count)
```

```
if model.predict(spam_email_count) == 1:
        print(f"Prediction: Spam - Confidence: {confidence[0][1]:.2f} - Message: {spam_email[0]}")
```

```
Prediction: Spam - Confidence: 0.98 - Message: you won 1millon dollar click here to claim
```

```
x_test_word_count = count_words.transform(x_test)
```

```
model.score(x_test_word_count, y_test)
```

```
0.9837587006960556
```