```
#import basic packages
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
```

```
#Loading the dataset
spam_df = pd.read_csv('/content/drive/MyDrive/spam.csv')
```

```
spam_df
```

	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...
...	...	...
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

Next steps: Generate code with spam_df View recommended plots

```
spam_df.groupby('Category').describe()
```

	Message				
	count	unique	top	freq
Category				
ham	4825	4516	Sorry, I'll call later	30
spam	747	641	Please call our customer service representativ...	4

```
#turn spam into numeriacl
spam_df['spam'] = spam_df['Category'].apply(lambda x: 1 if x=='spam' else 0)
```

```
spam_df
```

	Category	Message	spam
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
...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	1
5568	ham	Will ü b going to esplanade fr home?	0
5569	ham	Pity, * was in mood for that. So...any other s...	0
5570	ham	The guy did some bitching but I acted like i'd...	0
5571	ham	Rofl. Its true to its name	0

5572 rows × 3 columns

Next steps: Generate code with spam_df View recommended plots

```
#creating the train test and split
x_train, x_test, y_train, y_test = train_test_split(spam_df.Message,spam_df.spam,test_size=0.30)
```

```
x_train.describe()
```

```
count                     3900
unique                    3668
top       Sorry, I'll call later
freq                        22
Name: Message, dtype: object
```

```
#find word count and store data as matrix
cv =CountVectorizer()
x_train_count = cv.fit_transform(x_train.values)
```

```
x_train_count.toarray()
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
#train the model
model = MultinomialNB()
model.fit(x_train_count,y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
#pre-test for ham
email_ham = ['hey wanna meet up for the game?']
email_ham_count = cv.transform(email_ham)
model.predict(email_ham_count)
```

```
array([0])
```

```
#pre-test for spam
email_spam = ['reward ticket money click']
email_spam_count = cv.transform(email_spam)
model.predict(email_spam_count)
```

```
array([1])
```

```
#test model
x_test_count = cv.transform(x_test)
model.score(x_test_count,y_test)
```

```
0.9886363636363636
```

```
Start coding or generate with AI.
```