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

```
df = pd.read_csv('/content/spam_ham_dataset[1].csv')
df.head()
```

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5171 entries, 0 to 5170
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  5171 non-null   int64 
 1   label       5171 non-null   object
 2   text        5171 non-null   object
 3   label_num   5171 non-null   int64 
dtypes: int64(2), object(2)
memory usage: 161.7+ KB
```

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

	Unnamed: 0								label_num							
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
label																
ham	3672.0	1835.5	1060.159422	0.0	917.75	1835.5	2753.25	3671.0	3672.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
spam	1499.0	4421.0	432.868340	3672.0	4046.50	4421.0	4795.50	5170.0	1499.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0

```
df['spam'] = df['label'].apply(lambda x:1 if x=='spam' else 0)
```

```
df.head()
```

	Unnamed: 0	label	text	label_num	spam
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0	0

```
new_df = df[['label','text','spam']]
```

```
new_df.head()
```

	label	text	spam
0	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	spam	Subject: photoshop , windows , office . cheap ...	1
4	ham	Subject: re : indian springs\r\nthis deal is t...	0

```
from sklearn.model_selection import train_test_split as tts
x_train,x_test,y_train,y_test=tts(df.text,df.spam)
```

```
from sklearn.feature_extraction.text import CountVectorizer


v=CountVectorizer()
x_train_count=v.fit_transform(x_train.values)
x_train_count.toarray()[:2]
```

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

```
from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train_count,y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
emails=["How are you brother?", "Free entry"]
email_count=v.transform(emails)
model.predict(email_count)
```

```
array([1, 0])
```

```
x_test_count=v.transform(x_test)
print("Accuracy is : ",model.score(x_test_count,y_test))
```

```
Accuracy is :  0.9767981438515081
```