

```
In [1]: import pandas as pd
```

```
In [3]: df = pd.read_csv(r"C:\Users\NITISH SINGH\Downloads\archive (31)\Spam_Data.csv")
df.head()
```

```
Out[3]:
```

	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...

```
In [4]: df.groupby('Category').describe()
```

```
Out[4]:
```

		count	unique		Message	top	freq
	Category						
	ham	4825	4516		Sorry, I'll call later		30
	spam	747	653		Please call our customer service representativ...		4

```
In [5]: df['spam']=df['Category'].apply(lambda x: 1 if x=='spam' else 0)
df.head()
```

```
Out[5]:
```

	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

```
In [6]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.Message,df.spam, test_size=0.25)
```

```
In [11]: from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer()
X_train_count = v.fit_transform(X_train.values)
X_train_count.toarray()[:3]
```

```
Out[11]: array([[0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
In [12]: from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

```
Out[12]: MultinomialNB()
```

```
In [17]: emails = [
    'K tell me anything about you.',
    'SMS. ac Sptv: The New Jersey Devils and the Detroit Red Wings play Ice Hockey. Correct or Incorrect? End?'
]

emails_count = v.transform(emails)
model.predict(emails_count)
```

```
Out[17]: array([0, 1], dtype=int64)
```

```
In [18]: X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

```
Out[18]: 0.9827709978463748
```

```
In [ ]:
```