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

```python
In [25]:  df = pd.read_csv(r"C:\Users\91830\Desktop\DUK\AIML\NaiveB\spam.csv", encoding='latin-1')[['v1', 'v2
```

```python
In [3]:   df.columns = ['category', 'message']
```

```python
In [4]:   df.head()
```

Out[4]:

	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...

```python
In [5]:   df.groupby('category').describe()
```

Out[5]:

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

```python
In [6]:   df['spam']=df['category'].apply(lambda x: 1 if x=='spam' else 0)
          df.head()
```

Out[6]:

	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

```python
In [7]:   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)
```

```python
In [26]:  from sklearn.feature_extraction.text import CountVectorizer
          v = CountVectorizer()
          X_train_count = v.fit_transform(X_train.values)
```

```python
In [37]:  from sklearn.naive_bayes import MultinomialNB
          model = MultinomialNB()
          model.fit(X_train_count,y_train)
```

Out[37]: MultinomialNB()

```
In [28]:   X_test_count= v.transform(X_test)
           accuracy = model.score(X_test_count, y_test)
```

```
In [31]:   emails=[
               'Can i get your phone number',
               'Hi, Click for free mobile data!'
           ]
```

```
In [32]:   emails_count = v.transform(emails)
           model.predict(emails_count)
```

Out[32]: array([0, 1], dtype=int64)

```
In [33]:   print("Accuracy Score ={:.2f}".format(accuracy*100)+"%")
```

Accuracy Score =98.06%