In [4]:
```
# Import the required library
import pandas as pd
```

In [8]:
```
#Load the dataset
df=pd.read_csv(r"C:\Users\user\Desktop\AI LAB WORKS\Naive Bayes\spam.csv", enc
df.columns=["category","message"]
df.head()
```

Out[8]:

	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 [9]:
```
df.groupby('category').describe()
```

Out[9]:

	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

In [10]:
```
df['spam']=df['category'].apply(lambda x:1 if x=='spam' else 0)
df.head()
```

Out[10]:

	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 [11]:
```
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.
```

In [13]:
```
# Preprocess the dataset if required
from sklearn.feature_extraction.text import CountVectorizer
v=CountVectorizer()
x_train_count=v.fit_transform(x_train.values)
x_train_count.toarray()[ :3]
```

Out[13]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [14]:
```
# Train the model
from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train_count,y_train)
```

Out[14]:
```
▾ MultinomialNB
MultinomialNB()
```

In [15]:
```
# Testing/Predicting the model
emails=[
     'Sounds great! Are you home now?',
     'Upto 20% discount on parking, exclusive offer just for you. Dont miss th
]
emails_count = v.transform(emails)
model.predict(emails_count)
```

Out[15]:
```
array([0, 1], dtype=int64)
```

In [16]:
```
# Performance of ML model -> Accuracy
x_test_count=v.transform(x_test)
accuracy=model.score(x_test_count,y_test)
print("Accuracy score :",accuracy)
```

```
Accuracy score : 0.9865470852017937
```

In [ ]:

In [ ]: