```
Objective :Design Spam Ham Classifier using naive bayes classificaton
```

In [1]:
```
#Step1 : Import necessary libraries
import pandas as pd
import numpy as np
```

In [3]:
```
#step2 : load the dataset
docs=pd.read_csv("C:/1562_AIML/spamfinal.csv",encoding="latin-1")
docs.head()
```

Out[3]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

In [4]:
```
#step3 : data preprocessing
#a. remove unnecessary columns
docs1=docs.drop("Unnamed: 2",axis=1)
docs1.head()
```

Out[4]:

	v1	v2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN

In [5]:
```
docs.head()
```

Out[5]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

In [6]:
```
#removing the column permanantly
docs1.drop('Unnamed: 3', axis=1,inplace=True)
```

In [7]:
```
docs1.head()
```

Out[7]:

	v1	v2	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN
1	ham	Ok lar... Joking wif u oni...	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN
3	ham	U dun say so early hor... U c already then say...	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN

In [8]:
```
#dropping using del method
del docs1['Unnamed: 4']
```

In [9]:
```
docs1.head()
```

Out[9]:

	v1	v2
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 [10]:
```
#b. encode the target attribute as 1 and 0 - spam as 1 and ham as 0
docs1['label']=docs1.v1.map({'ham':0,'spam':1})
docs1.head()
```

Out[10]:

	v1	v2	label
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]:
```
#information of our dataset
docs1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
 2   label   5572 non-null   int64 
dtypes: int64(1), object(2)
memory usage: 130.7+ KB
```

In [12]:
```
h1=docs1.v1.value_counts()
h1
```

Out[12]:
```
ham     4825
spam     747
Name: v1, dtype: int64
```

In [15]:
```
#percentage of spam and ham
spam_perc=(100*h1[1])/(h1[0]+h1[1])
```

In [16]:
```
print('Spam Percentage:',spam_perc)
```

```
Spam Percentage: 13.406317300789663
```

In [17]:
```
ham_perc=100-spam_perc
```

In [18]:
```
print('ham percentage:',ham_perc)
```

```
ham percentage: 86.59368269921033
```

In [19]:
```
#c.prepare x and y
x=docs.v2
y=docs1.label
```

In [20]:
```
# to check the shape
x.shape
```

Out[20]:
```
(5572,)
```

In [21]:
```
y.shape
```

Out[21]:
```
(5572,)
```

In [23]:
```
#step4 : split the dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=.8,random_state=
x_train.shape
```

Out[23]:
```
(4457,)
```

In [24]:
```
x_train
```

Out[24]:
```
3366              Hey what are you doing. Y no reply pa..
3022     You are a very very very very bad girl. Or lady.
1160             You sure your neighbors didnt pick it up
3778    Claim a 200 shopping spree, just call 08717895...
585     Tell them u have a headache and just want to u...
                              ...
4473    Want explicit SEX in 30 secs? Ring 02073162414...
580     Huh so early.. Then Ì_ having dinner outside i...
163     -PLS STOP bootydelious (32/F) is inviting you ...
4703                    Yar but they say got some error.
3616    Sorry sent blank msg again. Yup but trying 2 d...
Name: v2, Length: 4457, dtype: object
```

In [27]:
```
#step 5: Prepare a bag of words
# import CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer(stop_words = 'english')
vec.fit(x_train)
vec.vocabulary_
```

Out[27]:
```
{'hey': 3264,
 'doing': 2309,
 'reply': 5450,
 'pa': 4796,
 'bad': 1142,
 'girl': 3001,
 'lady': 3812,
 'sure': 6327,
 'neighbors': 4520,
 'didnt': 2221,
 'pick': 4954,
 'claim': 1744,
 '200': 329,
 'shopping': 5828,
 'spree': 6115,
 'just': 3693,
 '08717895698': 128,
 'won': 7193,
 'mobstorequiz10ppm': 4340,
```

In [28]:
```
print('Length:',len(vec.get_feature_names()))
```

```
Length: 7371
```

In [29]:
```
#prepare a sparce matrix
x_train_trans=vec.transform(x_train)
x_test_trans=vec.transform(x_test)
```

In [30]:
```
print(x_train_trans)
```

```
  (0, 2309)     1
  (0, 3264)     1
  (0, 4796)     1
  (0, 5450)     1
  (1, 1142)     1
  (1, 3001)     1
  (1, 3812)     1
  (2, 2221)     1
  (2, 4520)     1
  (2, 4954)     1
  (2, 6327)     1
  (3, 128)      1
  (3, 329)      1
  (3, 1744)     1
  (3, 3693)     1
  (3, 4340)     1
  (3, 5828)     1
  (3, 6115)     1
  (3, 7193)     1
  (4, 3219)     1
  (4, 3357)     1
  (4, 3693)     1
  (4, 5866)     1
  (4, 6457)     1
  (4, 6574)     1
  :     :
  (4454, 468)   2
  (4454, 552)   1
  (4454, 742)   1
  (4454, 1374)  2
  (4454, 2878)  1
  (4454, 2884)  1
  (4454, 3549)  1
  (4454, 5006)  1
  (4454, 5450)  1
  (4454, 5735)  1
  (4454, 5976)  1
  (4454, 6195)  3
  (4454, 7257)  1
  (4454, 7306)  1
  (4455, 2530)  1
  (4455, 3058)  1
  (4455, 5654)  1
  (4455, 7288)  1
  (4456, 1314)  1
  (4456, 4400)  1
  (4456, 5744)  1
  (4456, 6031)  1
  (4456, 6241)  1
  (4456, 6720)  1
  (4456, 7340)  1
```

In [33]:
```
#step 6: build the model
from sklearn.naive_bayes import BernoulliNB
#create instance of the class

bnb = BernoulliNB()
#train the model
bnb.fit(x_train_trans,y_train)
```

Out[33]:
```
BernoulliNB()
```

In [34]:
```
y_pred=bnb.predict(x_test_trans)
```

In [35]:
```
#y_test - actual values and y_pred - predicted values
y_pred , y_test
```

Out[35]:
```
(array([0, 0, 0, ..., 0, 1, 0], dtype=int64),
 5005    0
 3286    0
 4580    0
 3328    0
 1508    0
        ..
 3961    1
 2172    0
 741     0
 3895    1
 3945    0
 Name: label, Length: 1115, dtype: int64)
```

In [36]:
```
#step 7 : measure the performance of the model
from sklearn import metrics
metrics.confusion_matrix(y_test,y_pred)
```

Out[36]:
```
array([[940,   0],
       [ 39, 136]], dtype=int64)
```

In [37]:
```
metrics.accuracy_score(y_test,y_pred)
```

Out[37]:
```
0.9650224215246637
```

In [ ]:
```

```