Importing libraries

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.feature_extraction.text import CountVectorizer
```

Reading the datasets

In [2]:
```
df = pd.read_excel("spam_train2.xlsx")
df_test= pd.read_excel("spam_test.xlsx")
```

data cleaning

In [3]:
```
df['spam'] = df['output'].apply(lambda x: 1 if x =='spam' else 0 )
df_test['spam'] = df_test['output'].apply(lambda x:1 if x=='spam' else 0 )
```

In [30]:
```
import nltk
# download the stopwords package
nltk.download("stopwords")
```

Out[30]:
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\mkali\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

In [31]:
```
from nltk.corpus import stopwords
import string
```

In [32]:
```
def clean(text):
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ''.join(nopunc)

    clean = [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
    return clean
```

Training with Naive Bayes

In [33]:
```
x_train = df['text']
y_train = df['spam']
```

In [34]:
```
x_train = x_train.values.tolist()
x_train = [str (item) for item in x_train]
```

In [35]:
```
cv = CountVectorizer()
x_train_new = cv.fit_transform(x_train)
```

In [36]:
```
model = MultinomialNB(alpha=1)
model.fit(x_train_new,y_train)
```

Out[36]:
```
MultinomialNB(alpha=1)
```

Testing the model

In [37]:
```
x_test = df_test['text']
y_test = df_test['spam']
```

In [38]:
```
x_test = x_test.values.tolist()
x_test_new = cv.transform(x_test)
predictions = model.predict(x_test_new)
score = model.score(x_test_new,y_test)
print('score:{:5f}'.format(score))
#print(predictions)
```

```
score:0.991039
```

Training & Testing with SVM

In [39]:
```
model = SVC(C=0.8,kernel = 'rbf') #radius 0.8
model.fit(x_train_new,y_train)
```

Out[39]:
```
SVC(C=0.8)
```

In [40]:

```
score = model.score(x_test_new,y_test)
predictions = model.predict(x_test_new)
print(score)
```

```
0.9802867383512545
```

In [ ]:

In [ ]: