SPAM EMAIL DETECTION

Objective of the project:

To determine whether an e-mail is genuine one or a spam.

Importing the dependencies

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

The Dataset

In [2]:
```
spam_df=pd.read_csv(r'C:\Users\HP\Downloads\mail_data.csv')
```

In [3]:
```
spam_df
```

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...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

Inspecting the dataset

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

Out[4]:

	Message			
	count	unique	top	freq
Category				
ham	4825	4516	Sorry, I'll call later	30
spam	747	641	Please call our customer service representativ...	4

Turning spam & ham into numerical data and creating a new column 'spam'

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

In [6]:
```
spam_df
```

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
...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	1
5568	ham	Will ü b going to esplanade fr home?	0
5569	ham	Pity, * was in mood for that. So...any other s...	0
5570	ham	The guy did some bitching but I acted like i'd...	0
5571	ham	Rofl. Its true to its name	0

5572 rows × 3 columns

spam ---> 1 ham ---> 0

Creating train/test split

In [7]:
```
X_train , X_test , Y_train , Y_test =train_test_split(spam_df.Message,spam_df.spam)
```

In [8]:
```
X_train
```

Out[8]:
```
3462    K.. I yan jiu liao... Sat we can go 4 bugis vi...
4041                              I'm at home n ready...
3561    Lol I know! Hey someone did a great inpersonat...
2108    Hmmm ... And imagine after you've come home fr...
2548    Text82228>> Get more ringtones, logos and game...
                              ...
3332            How much it will cost approx . Per month.
1521    URGENT! Your Mobile No was awarded a £2,000 Bo...
4967    URGENT! We are trying to contact U. Todays dra...
3424    Had your mobile 10 mths? Update to latest Oran...
5389                  Ok.ok ok..then..whats ur todays plan
Name: Message, Length: 4179, dtype: object
```

In [9]:
```
X_train.describe()
```

Out[9]:
```
count                     4179
unique                    3915
top       Sorry, I'll call later
freq                        23
Name: Message, dtype: object
```

Find word count and store data as a matrix

In [10]:
```
cv=CountVectorizer()
```

In [11]:
```
X_train_count=cv.fit_transform(X_train.values)
```

In [13]:
```
X_train_count.toarray()
```

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

Training the model

In [14]:
```
model=MultinomialNB()
```

In [15]:
```
model.fit(X_train_count,Y_train)
```

Out[15]:
```
▾ MultinomialNB
MultinomialNB()
```

Pre-Testing the model 1

Pre-Testing the model 1

In [16]:
```
email_ham=['hey wanna meet up for the game ?']
```

In [17]:
```
email_ham_count=cv.transform(email_ham)
```

In [18]:
```
model.predict(email_ham_count)
```

Out[18]:
```
array([0], dtype=int64)
```

Hence this is a genuine email.

Pre-Testing the model 2

In [20]:
```
email_spam=["10K lottery % win"]
```

In [21]:
```
email_spam_count=cv.transform(email_spam)
```

In [28]:
```
model.predict(email_spam_count)
```

Out[28]:
```
array([1], dtype=int64)
```

Hence the email is a SPAM.

Test Model

In [29]:
```
X_test_count = cv.transform(X_test)
```

In [30]:
```
model.score(X_test_count,Y_test)
```

Out[30]:
```
0.9849246231155779
```

Using Naive Bayes the model has obtained an accuracy of almost 98% which is really incredible.

In [ ]:
```

```