reading the dataset and appling the ML algo

- dataset from uci website
- thre will a ML repository
- dataset:sms spam collection dataset just download it
- the dataset ia actually tab seperated so we should have to use \t for seperating them # jump into the code

In [9]:
```
import pandas as pd
messages = pd.read_csv('~/Desktop/smsspamcollection/SMSSpamCollection', sep='\t',
                           names=["label", "message"])
```

In [10]:
```
messages
```

Out[10]:

	label	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

In [11]:
```
messages.head()
```

Out[11]:

	label	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 [12]:
```
messages.describe()
```

Out[12]:

	label	message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

In [16]:
```
messages.tail()
```

Out[16]:

	label	message
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

data cleaning and pre processing

In [18]:
```
import re
import nltk
```

In [19]:
```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
corpus = []
for i in range(0, len(messages)):
    review = re.sub('[^a-zA-Z]', ' ', messages['message'][i])
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

In [20]:
```
# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=2500)
X = cv.fit_transform(corpus).toarray()

y=pd.get_dummies(messages['label'])
y=y.iloc[:,1].values
```

In [21]:
```
# Train Test Split

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0
)
```

training the model based on Naive bayes classifier

In [23]:
```
from sklearn.naive_bayes import MultinomialNB
spam_detect_model=MultinomialNB().fit(X_train,y_train)
```

In [25]:
```
y_pred=spam_detect_model.predict(X_test)
```

In [26]:
```
y_pred
```

Out[26]:
```
array([0, 1, 0, ..., 0, 1, 0], dtype=uint8)
```

to find the accuracy of the model we use confusion matrix

In [29]:
```
from sklearn.metrics import confusion_matrix
confusion_m=confusion_matrix(y_pred,y_test)
```

In [35]:
```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred,y_test)
y_test
```

Out[35]:
```
array([0, 1, 0, ..., 0, 1, 0], dtype=uint8)
```

In [34]:
```
y_pred
```

Out[34]:
```
array([0, 1, 0, ..., 0, 1, 0], dtype=uint8)
```

In [31]:
```
y_pred
```

Out[31]:
```
array([0, 1, 0, ..., 0, 1, 0], dtype=uint8)
```

In [32]:
```
y_test
```

Out[32]:
```
array([0, 1, 0, ..., 0, 1, 0], dtype=uint8)
```

In [ ]:
```

```