In [3]:

```python
import numpy as np
import pandas as pd
dt = pd.read_csv("spam.csv" ,encoding = 'Windows-1252')
dt.head(10)
```

Out[3]:

	type	text
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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

In [5]:

```python
dt['spam'] = dt['type'].map( {'spam': 1, 'ham': 0} ).astype(int)
dt.head(5)
```

Out[5]:

	type	text	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 [8]:

```python
print("COLUMS IN THE GIVEN DATA:")
for col in dt.columns:
    print(col)
```

```
COLUMS IN THE GIVEN DATA:
type
text
spam
```

In [10]:

```python
t=len(dt['type'])
print("no of rows in review column:",t)
t=len(dt['text'])
print("no of rows in liked column:",t)
```

```
no of rows in review column: 116
no of rows in liked column: 116
```

In [12]:

```
dt['text'][1] #before
```

Out[12]:

```
'Ok lar... Joking wif u oni...'
```

In [13]:

```
def tokenizer(text):
    return text.split()
```

In [14]:

```
dt['text']=dt['text'].apply(tokenizer)
```

In [16]:

```
dt['text'][1]#after
```

Out[16]:

```
['Ok', 'lar...', 'Joking', 'wif', 'u', 'oni...']
```

In [18]:

```
dt['text'][1] #before
```

Out[18]:

```
['Ok', 'lar...', 'Joking', 'wif', 'u', 'oni...']
```

In [20]:

```
from nltk.stem.snowball import SnowballStemmer
porter = SnowballStemmer("english", ignore_stopwords=False)
```

In [21]:

```
def stem_it(text):
    return[porter.stem(word) for word in text]
```

In [22]:

```
dt['text']=dt['text'].apply(stem_it)
```

In [23]:

```
dt['text'][1]#after stemming
```

Out[23]:

```
['ok', 'lar...', 'joke', 'wif', 'u', 'oni...']
```

In [26]:

```
dt['text'][27]#before
```

Out[26]:

```
['did',
 'you',
 'catch',
 'the',
 'bus',
 '?',
 'are',
 'you',
 'fri',
 'an',
 'egg',
 '?',
 'did',
 'you',
 'make',
 'a',
 'tea?',
 'are',
 'you',
 'eat',
 'your',
 'mom',
 'left',
 'over',
 'dinner',
 '?',
 'do',
 'you',
 'feel',
 'my',
 'love',
 '?']
```

In [28]:

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

In [29]:

```
def lemmit_it(text):
    return [lemmatizer.lemmatize(word, pos ='a') for word in text]
```

In [32]:

```
dt['text']=dt['text'].apply(lemmit_it)
```

In [33]:

```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to C:\Users\RAJESH
[nltk_data]     SAXENA\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[33]:

```
True
```

In [35]:

```
dt['text'][36]
```

Out[35]:

```
['oops,', "i'll", 'let', 'you', 'know', 'when', 'my', 'roommat', 'done']
```

In [40]:

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
```

In [39]:

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to C:\Users\RAJESH
[nltk_data]     SAXENA\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[39]:

```
True
```

In [41]:

```
def stop_it(text):
    review = [word for word in text if not word in stop_words]
    return review
```

In [42]:

```
dt['text']=dt['text'].apply(stop_it)
```

In [43]:

```
dt['text'][36]
```

Out[43]:

```
['oops,', "i'll", 'let', 'know', 'roommat', 'done']
```

In [44]:

```
dt.head(10)
```

Out[44]:

	type	text	spam
0	ham	[go, jurong, point,, crazy.., avail, onli, bug...	0
1	ham	[ok, lar..., joke, wif, u, oni...]	0
2	spam	[free, entri, 2, wkli, comp, win, fa, cup, fin...	1
3	ham	[u, dun, say, earli, hor..., u, c, alreadi, sa...	0
4	ham	[nah, think, goe, usf,, live, around, though]	0
5	spam	[freemsg, hey, darl, 3, week, word, back!, i'd...	1
6	ham	[even, brother, like, speak, me., treat, like,...	0
7	ham	[per, request, mell, mell, (oru, minnaminungin...	0
8	spam	[winner!!, valu, network, custom, select, rece...	1
9	spam	[mobil, 11, month, more?, u, r, entitl, updat,...	1

In [45]:

```
dt['text']=dt['text'].apply(' '.join)
```

In [46]:

```
dt.head()
```

Out[46]:

	type	text	spam
0	ham	go jurong point, crazy.. avail onli bugi n gre...	0
1	ham	ok lar... joke wif u oni...	0
2	spam	free entri 2 wkli comp win fa cup final tkts 2...	1
3	ham	u dun say earli hor... u c alreadi say...	0
4	ham	nah think goe usf, live around though	0

In [47]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer()
y=dt.spam.values
x=tfidf.fit_transform(dt['text'])
```

In [49]:

```
from sklearn.model_selection import train_test_split
x_train,x_text,y_train,y_text=train_test_split(x,y,random_state=1,test_size=0.2,shuffle=
False)
```

In [51]:

```
from sklearn.linear_model import LogisticRegression
clf=LogisticRegression()
clf.fit(x_train,y_train)
y_pred=clf.predict(x_text)
from sklearn.metrics import accuracy_score
acc_log = accuracy_score(y_pred, y_text)*100
print("accuracy:",acc_log)
```

```
accuracy: 87.5
```

In [52]:

```
from sklearn.svm import LinearSVC

linear_svc = LinearSVC(random_state=0)
linear_svc.fit(x_train, y_train)
y_pred = linear_svc.predict(x_text)
acc_linear_svc =accuracy_score(y_pred, y_text)*100
print("accuracy:", acc_linear_svc)
```

```
accuracy: 87.5
```

In [ ]: