In [1]:
```
import numpy as np
import pandas as pd
import nltk
nltk.download('wordnet')
nltk.download('stopwords')
```

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

Out[1]:
```
True
```

In [2]:
```
dt = pd.read_csv("SPAM.csv")
dt.head(10)
```

Out[2]:

	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 [3]:
```
dt["spam"] = dt['type'].map({'spam':1,'ham':0}).astype(int)
dt.head(10)
```

Out[3]:

	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
5	spam	FreeMsg Hey there darling it's been 3 week's n...	1
6	ham	Even my brother is not like to speak with me. ...	0
7	ham	As per your request 'Melle Melle (Oru Minnamin...	0
8	spam	WINNER!! As a valued network customer you have...	1
9	spam	Had your mobile 11 months or more? U R entitle...	1

In [4]:
```
print('Columns in the data set: ')
for col in dt.columns:
    print(col)
```

```
Columns in the data set:
type
text
spam
```

In [5]:
```
t = len(dt['type'])
print('No. of rows in review column: ',t)
t = len(dt['text'])
print('No. of rows in linked column: ',t)
```

```
No. of rows in review column:  116
No. of rows in linked column:  116
```

In [6]:
```
# Tokenization
```

In [7]:
```
dt['text'][1] #Before
```

Out[7]:
```
'Ok lar... Joking wif u oni...'
```

In [8]:
```
def tokenizer(text):
    return text.split()
dt['text'] = dt['text'].apply(tokenizer)
```

In [9]:
```
dt['text'][1] # after
```

Out[9]:
```
['Ok', 'lar...', 'Joking', 'wif', 'u', 'oni...']
```

In [10]:
```
# Stemming
```

In [11]:
```
dt['text'][1] #Before
```

Out[11]:
```
['Ok', 'lar...', 'Joking', 'wif', 'u', 'oni...']
```

In [12]:
```
from nltk.stem.snowball import SnowballStemmer
poter = SnowballStemmer('english', ignore_stopwords = False)
def stem_it(text):
    return[poter.stem(word) for word in text]
dt['text'] = dt['text'].apply(stem_it)
```

In [13]:
```
dt['text'][1] #after
```

Out[13]:
```
['ok', 'lar...', 'joke', 'wif', 'u', 'oni...']
```

In [14]:
```
#lemmitization
```

In [15]:
```
dt['text'][10] #Before
```

Out[15]:
```
["i'm",
 'gonna',
 'be',
 'home',
 'soon',
 'and',
 'i',
 "don't",
 'want',
 'to',
 'talk',
 'about',
 'this',
 'stuff',
 'anymor',
 'tonight,',
 'k?',
 "i've",
 'cri',
 'enough',
 'today.']
```

In [16]:
```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
def lemmit_it(text):
    return[lemmatizer.lemmatize(word, pos='a') for word in text]
dt['text'] = dt['text'].apply(lemmit_it)
```

In [17]:
```
dt['text'][10] #after
```

Out[17]:
```
["i'm",
 'gonna',
 'be',
 'home',
 'soon',
 'and',
 'i',
 "don't",
 'want',
 'to',
 'talk',
 'about',
 'this',
 'stuff',
 'anymor',
 'tonight,',
 'k?',
 "i've",
 'cri',
 'enough',
 'today.']
```

In [18]:
```
#stopword removal
```

In [19]:
```
dt['text'][17] #Before
```

Out[19]:
```
['eh',
 'u',
 'rememb',
 'how',
 '2',
 'spell',
 'his',
 'name...',
 'yes',
 'i',
 'did.',
 'he',
 'v',
 'naughti',
 'make',
 'until',
 'v.',
 'i',
 'wet.']
```

In [20]:
```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
def stop_it(text):
    review=[word for word in text if not word in stop_words]
    return review
dt['text'] = dt['text'].apply(stop_it)
```

In [21]:
```
dt['text'][17] #after
```

Out[21]:
```
['eh',
 'u',
 'rememb',
 '2',
 'spell',
 'name...',
 'yes',
 'did.',
 'v',
 'naughti',
 'make',
 'v',
 'wet.']
```

In [22]:
```
dt.head(10)
```

Out[22]:

	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 [23]:
```
dt['text'] = dt['text'].apply(' '.join)
```

In [24]:
```
dt.head(10)
```

Out[24]:

	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
5	spam	freemsg hey darl 3 week word back! i'd like fu...	1
6	ham	even brother like speak me. treat like aid pat...	0
7	ham	per request mell mell (oru minnaminungint nuru...	0
8	spam	winner!! valu network custom select receivea £...	1
9	spam	mobil 11 month more? u r entitl updat late col...	1

In [25]:
```
# text data to vector form
```

In [26]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer()
y = dt.spam.values
x = tfidf.fit_transform(dt['text'])
```

In [29]:
```
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 [30]:
```
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 [31]:
```
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 [ ]:
```

```