In [2]:

```
#import required libraries
import pandas as pd
import string
from nltk.corpus import stopwords
```

In [5]:

```
#Get the spam data collection
df=pd.read_csv("SpamCollection",sep='\t',names=["response","message"])

df
```

Out[5]:

	response	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 [6]:

```
df.head()
```

Out[6]:

	response	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 [7]:

```
#view the information about spam data using describe method
df.describe()
```

Out[7]:

	response	message
count	5572	5572
unique	2	5169

response	message
top	ham	Sorry, I'll call later
freq	4825	30

In [11]:

```
#view response
df.groupby("response").describe()
```

Out[11]:

response	message			
	count	unique	top	freq
ham	4825	4516	Sorry, I'll call later	30
spam	747	653	Please call our customer service representativ...	4

In [12]:

```
#Verify length of the messages and also add it as a new column
df["length"]=df["message"].apply(len)
```

In [13]:

```
#view the first 5 rows of messages
df.head()
```

Out[13]:

	response	message	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

In [16]:

```
#define a function to get rid of stopwords present in the messages
def message_text_process(mess):
    #check for punctuations
    no_punctuation=[char for char in mess if char not in string.punctuation]
    #from sentense
    no_punctuation="".join(no_punctuation)
    #eliminate stopwords
    return [word for word in no_punctuation.split() if word.lower not in stopwords.words
("english")]
```

In [20]:

```
#import NLTK and download stopwords
import nltk
nltk.download("stopwords")
```

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

Out[20]:

```
True
```

In [21]:

```
df["message"].head(5).apply(message_text_process)
```

Out[21]:

```
0    [Go, until, jurong, point, crazy, Available, o...
1                        [Ok, lar, Joking, wif, u, oni]
2    [Free, entry, in, 2, a, wkly, comp, to, win, F...
3    [U, dun, say, so, early, hor, U, c, already, t...
4    [Nah, I, dont, think, he, goes, to, usf, he, l...
Name: message, dtype: object
```

In [22]:

```
#start text processing with vectorizer
from sklearn.feature_extraction.text import CountVectorizer
```

In [24]:

```
#use bag of words by applying the function and fit the data into it
bag_of_words_transeformer=CountVectorizer(analyzer=message_text_process).fit(df["message"
])
```

In [25]:

```
#print length of bag of words stored in the vocabulary_ attribute
print(len(bag_of_words_transeformer.vocabulary_))
```

```
11747
```

In [27]:

```
message_bagofwords=bag_of_words_transeformer.transform(df["message"])
```

In [29]:

```
#apply tfidf transformer and fit the bag of words into it (transformed version)
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer=TfidfTransformer().fit(message_bagofwords)
```

In [33]:

```
#print shape of the tfidf
message_tfidf=tfidf_transformer.transform(message_bagofwords)
print(message_tfidf.shape)
```

```
(5572, 11747)
```

In [41]:

```
#choose naive Bayes model to detect the spam and fit the tfidf data into it
from sklearn.naive_bayes import MultinomialNB
spam_detect_model=MultinomialNB().fit(message_tfidf,df["response"])
```

In [47]:

```
#check model for the predicted and expected value say for message#2 and message#5
message=df["message"][2]
bagofwords_message=bag_of_words_transeformer.transform([message])
tfidf=tfidf_transformer.transform(bagofwords_message)

print("predicted:",spam_detect_model.predict(tfidf)[0])
print("expected:",df.response[2])
```

```
predicted: spam
expected: spam
```

In [48]:

```
print("sucessfully completed project on NLP spam detetion")
```

```
sucessfully completed project on NLP spam detetion
```

In [49]:

```
print("Thank You Simplilearn")
```

```
Thank You Simplilearn
```

In [ ]:

```

```