In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv("spam",sep="\t",names=["label","message"])
```

In [3]:

```
df.head()
```

Out[3]:

	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 [4]:

```
df["label"] = df['label'].replace(['spam','ham'],[1,0])
```

In [5]:

```
df.head()
```

Out[5]:

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

In [6]:

```
df["label"].value_counts()
```

Out[6]:

```
0    4825
1     747
Name: label, dtype: int64
```

In [7]:

```
df.shape
```

Out[7]:

```
(5572, 2)
```

In [11]:

```
len(df)
```

Out[11]:

```
5572
```

In [14]:

```
!pip install tqdm
```

```
Requirement already satisfied: tqdm in d:\anacondainstalled\lib\site-packa
ges (4.35.0)

WARNING: You are using pip version 19.2.3, however version 19.3.1 is avail
able.
You should consider upgrading via the 'python -m pip install --upgrade pi
p' command.
```

data cleaning and preprocessing

In [15]:

```
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
import tqdm
```

In [17]:

```
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()
#sentences = nltk.sent_tokenize(paragraph)
corpus = []
for i in range(0,len(df)):
    review = re.sub("[^a-zA-Z]",' ',df['message'][i])
    review = review.lower()
    review = nltk.word_tokenize(review)
    review = [lemmatizer.lemmatize(word) for word in review if word not in set(stopword
s.words("english"))]
    review = " ".join(review)
    corpus.append(review)
```

In [18]:

```
corpus[:10]
```

Out[18]:

```
['go jurong point crazy available bugis n great world la e buffet cine got
amore wat',
 'ok lar joking wif u oni',
 'free entry wkly comp win fa cup final tkts st may text fa receive entry
question std txt rate c apply',
 'u dun say early hor u c already say',
 'nah think go usf life around though',
 'freemsg hey darling week word back like fun still tb ok xxx std chgs sen
d rcv',
 'even brother like speak treat like aid patent',
 'per request melle melle oru minnaminunginte nurungu vettam set callertun
e caller press copy friend callertune',
 'winner valued network customer selected receivea prize reward claim call
claim code kl valid hour',
 'mobile month u r entitled update latest colour mobile camera free call m
obile update co free']
```

In [27]:

```
from sklearn.feature_extraction.text import CountVectorizer
vc = CountVectorizer(max_features=5000)
X = vc.fit_transform(corpus).toarray()
```

In [30]:

```
y = df['label']
X.shape, y.shape
```

Out[30]:

```
((5572, 5000), (5572,))
```

In [33]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,y, test_size = 0.2,random_state = 7)
```

In [34]:

```
x_train.shape, x_test.shape
```

Out[34]:

```
((4457, 5000), (1115, 5000))
```

In [37]:

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x_train,y_train)
```

Out[37]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [38]:

```
y_pred = model.predict(x_test)
```

In [40]:

```
from sklearn.metrics import accuracy_score
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.9820627802690582
```

In [61]:

```
from sklearn.metrics import confusion_matrix
print("Spam:",len(y_pred[y_pred ==1]))
print("Ham:",len(y_pred[y_pred ==0]))
print("\n")
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
Spam: 149
Ham: 966


[[957  11]
 [  9 138]]
```

In [65]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vc = TfidfVectorizer(max_features=5000)
X = vc.fit_transform(corpus).toarray()
```

In [66]:

```
y = df['label']
X.shape, y.shape
```

Out[66]:

```
((5572, 5000), (5572,))
```

In [67]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,y, test_size = 0.2,random_state = 7)

x_train.shape, x_test.shape
```

Out[67]:

```
((4457, 5000), (1115, 5000))
```

In [68]:

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x_train,y_train)
```

Out[68]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [69]:

```
y_pred = model.predict(x_test)
```

In [70]:

```
from sklearn.metrics import accuracy_score
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.9659192825112107
```

In [71]:

```
from sklearn.metrics import confusion_matrix
print("Spam:",len(y_pred[y_pred ==1]))
print("Ham:",len(y_pred[y_pred ==0]))
print("\n")
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
Spam: 109
Ham: 1006


[[968   0]
 [ 38 109]]
```