Name:P.Asha Belcilda

Roll no:225229104

Lab6. Spam Filtering using Multinomial NB

In [1]:
```
import pandas as pd
```

STEP 1

In [2]:
```
data = pd.read_csv("SMSSpamCollection.csv",encoding="ISO-8859-1")
data.head()
```

Out[2]:

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

In [3]:
```
data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1,inplace=True)
```

In [4]:
```
data.head()
```

Out[4]:

	label	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...

STEP 2

In [5]:
```
data['text'].value_counts().sum()
```

Out[5]:
```
5572
```

STEP 3

In [6]:
```
data.groupby(['label']).count()
```

Out[6]:

	text
label	
ham	4825
spam	747

In [7]:
```
y = data['label']
```

In [8]:
```
X = data['text']
```

In [9]:
```
y
```

Out[9]:
```
0        ham
1        ham
2       spam
3        ham
4        ham
        ...
5567    spam
5568     ham
5569     ham
5570     ham
5571     ham
Name: label, Length: 5572, dtype: object
```

In [10]:
```
X
```

Out[10]:
```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                Will Ì_ b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                           Rofl. Its true to its name
Name: text, Length: 5572, dtype: object
```

STEP 4

In [11]:
```
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, rand
```

STEP 5

In [12]:
```
from nltk.corpus import stopwords
def process_text(msg):
    punctuations = '''!()-[]:;"\,<>./?@#${}%^_~*&'''
    no_punc = [char for char in msg if char not in punctuations]
    no_punc = ''.join(no_punc)
    return [word for word in no_punc.split()
            if word.lower() not in stopwords.words('english')]
```

In [13]:
```
import nltk
nltk.download('stopwords')
```

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

Out[13]:
```
True
```

STEP 6

In [14]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
data_1 = TfidfVectorizer(use_idf=True,analyzer = process_text, ngram_range=(1,
data_1
```

Out[14]:
```
TfidfVectorizer(analyzer=<function process_text at 0x000001D88FAFEF70>,
                ngram_range=(1, 3), stop_words='english')
```

In [15]:
```
a1 = data_1.fit_transform(X_train)
```

In [16]:
```
a2 = data_1.transform(X_test)
```

STEP 7

In [17]:
```
from sklearn.naive_bayes import MultinomialNB
mn = MultinomialNB()
mn.fit(a1,y_train)
```

Out[17]:
```
MultinomialNB()
```

STEP 8

```
In [18]: y_pred = mn.predict(a2)
         y_pred
```

```
Out[18]: array(['ham', 'ham', 'ham', ..., 'ham', 'ham', 'spam'], dtype='<U4')
```

STEP 9

```
In [19]: from sklearn.metrics import confusion_matrix
         confusion_matrix(y_test,y_pred)
```

```
Out[19]: array([[965,   0],
                [ 39, 111]], dtype=int64)
```

```
In [20]: from sklearn.metrics import classification_report
         print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         ham       0.96      1.00      0.98       965
        spam       1.00      0.74      0.85       150

    accuracy                           0.97      1115
   macro avg       0.98      0.87      0.92      1115
weighted avg       0.97      0.97      0.96      1115
```

STEP 10

```
In [21]: from sklearn.feature_extraction.text import TfidfVectorizer
         data2 = TfidfVectorizer(use_idf=True, analyzer = process_text, ngram_range=(1,
         data2
```

```
Out[21]: TfidfVectorizer(analyzer=<function process_text at 0x000001D88FAFEF70>,
                         ngram_range=(1, 2), stop_words='english')
```

```
In [22]: b = data2.fit_transform(X_train)
         b1= data2.transform(X_test)
```

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

```
Out[23]: MultinomialNB()
```

```
In [24]: y1_pred = mn.predict(b1)
         y1_pred
```

```
Out[24]: array(['ham', 'ham', 'ham', ..., 'ham', 'ham', 'spam'], dtype='<U4')
```

In [25]:
```
confusion_matrix(y_test,y1_pred)
```

Out[25]:
```
array([[965,   0],
       [ 39, 111]], dtype=int64)
```

In [26]:
```
print(classification_report(y_test,y1_pred))
```

```
              precision    recall  f1-score   support

         ham       0.96      1.00      0.98       965
        spam       1.00      0.74      0.85       150

    accuracy                           0.97      1115
   macro avg       0.98      0.87      0.92      1115
weighted avg       0.97      0.97      0.96      1115
```