In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

In [3]:

```
msg=pd.read_csv('naivetext.csv',names=['message','label'])
print('The dimensions of the dataset',msg.shape)
```

```
The dimensions of the dataset (18, 2)
```

In [4]:

```
msg['labelnum']=msg.label.map({'pos':1,'neg':0})
X=msg.message
y=msg.labelnum
```

In [5]:

```
#splitting the dataset into train and test data
xtrain,xtest,ytrain,ytest=train_test_split(X,y)
print ('\n the total number of Training Data :',ytrain.shape)
print ('\n the total number of Test Data :',ytest.shape)
```

```
 the total number of Training Data : (13,)

 the total number of Test Data : (5,)
```

In [6]:

```
#output the words or Tokens in the text documents
cv = CountVectorizer()
xtrain_dtm = cv.fit_transform(xtrain)
xtest_dtm=cv.transform(xtest)
print('\n The words or Tokens in the text documents \n')
print(cv.get_feature_names())
df=pd.DataFrame(xtrain_dtm.toarray(),columns=cv.get_feature_names())
```

```
 The words or Tokens in the text documents

['about', 'am', 'amazing', 'an', 'awesome', 'beers', 'can', 'dance', 'dea
l', 'do', 'enemy', 'feel', 'fun', 'good', 'great', 'have', 'he', 'holida
y', 'house', 'is', 'juice', 'like', 'love', 'my', 'not', 'of', 'place', 'r
estaurant', 'sandwich', 'stuff', 'sworn', 'taste', 'the', 'these', 'this',
'tired', 'to', 'today', 'tomorrow', 'very', 'we', 'went', 'what', 'will',
'with']
```

In [7]:

```
# Training Naive Bayes (NB) classifier on training data.
clf = MultinomialNB().fit(xtrain_dtm,ytrain)
predicted = clf.predict(xtest_dtm)

#printing accuracy, Confusion matrix, Precision and Recall
print('\n Accuracy of the classifier is',metrics.accuracy_score(ytest,predicted))
print('\n Confusion matrix')
print(metrics.confusion_matrix(ytest,predicted))
print('\n The value of Precision', metrics.precision_score(ytest,predicted))
print('\n The value of Recall', metrics.recall_score(ytest,predicted))
```

```
 Accuracy of the classifier is 0.6

 Confusion matrix
[[2 1]
 [1 1]]

 The value of Precision 0.5

 The value of Recall 0.5
```

In [ ]: