In [22]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import metrics
from sklearn.naive_bayes import MultinomialNB
import pandas as pd

data = pd.read_csv('naivetext1.csv',names=['msg','label'])
print("The dimension of data: {0}".format(data.shape))
data['label_num'] = data.label.map({'pos':1,'neg':0})
```

```
The dimension of data: (18, 2)
```

In [23]:
```
X = data.msg
y = data.label_num
xtrain,xtest,ytrain,ytest = train_test_split(X,y)

cvect = CountVectorizer()
xtrain_dtm = cvect.fit_transform(xtrain)
xtest_dtm = cvect.transform(xtest)
print("the vocabulary: \n{0}".format(cvect.get_feature_names()))
```

```
the vocabulary:
['am', 'amazing', 'an', 'awesome', 'bad', 'best', 'boss', 'can', 'deal', 'do', 'enemy', 'fun', 'good',
'have', 'he', 'horrible', 'house', 'is', 'juice', 'like', 'locality', 'love', 'my', 'not', 'of', 'plac
e', 'sandwich', 'stay', 'stuff', 'sworn', 'taste', 'that', 'the', 'this', 'tired', 'to', 'today', 'tom
orrow', 'view', 'we', 'went', 'what', 'will', 'with', 'work']
```

In [24]:
```
clf = MultinomialNB().fit(xtrain_dtm,ytrain)
predicted = clf.predict(xtest_dtm)
```

In [26]:
```
print('Accuracy metrics')
accuracy = metrics.accuracy_score(ytest,predicted)
confusion = metrics.confusion_matrix(ytest,predicted)
recall = metrics.recall_score(ytest,predicted)
precision = metrics.precision_score(ytest,predicted)
print('Accuracy of the classifer is \n{0}'.format(accuracy))
print('\nConfusion matrix is: \n{0}'.format(confusion))
print('\nRecall:\n {0}'.format(recall))
print('\nPrecision:\n {0}'.format(precision))
```

```
Accuracy metrics
Accuracy of the classifer is
0.8

Confusion matrix is:
[[2 0]
 [1 2]]

Recall:
 0.6666666666666666

Precision:
 1.0
```

In [ ]:

In [ ]:

In [ ]: