6. Assuming a set of documents that need to classified use the naive Bayesian Classifier model to perform this task. Built- in Java Classes / API can be used to write the program. Calculate the accuracy, precision and recall for your dataset.

```
import pandas as pd
msg = pd.read_csv('C:/users/user/Desktop/ML Lab/
                         lab6.csv') names=['message','label'])
print ('Total instances in the dataset:", msg.shape [0])

msg['labelnum']=msg.label.map({'pos':1 'neg':0})
 x=msg.message
 y = msg.labelnum
print ("In The message and its label of first 5 instances
          are listed below")
 x5, y5 = x[0:5], msg.label [0:5]
 for x,y in zip(x5, y5):
     print (x, ',', y)

from sklearn_model_selection import train_test_split:
  xtrain, xtest, ytrain, ytest = train_test_split(x,y)
  print ("Dataset is split into Training and Testing
                Samples")
  print (" Total training instances:", xtrain.shape[0])
  print ("Total testing instances:", xtest.shape[0])
```

```
from sklearn.feature_extraction.text import
                                    CountVectorizer
count_vect = CountVectorizer()
xtrain_dtm = count_vect.fit_transform (xtrain)

xtest_dtm = count_vect.fit transform (xtest)
print ("In Total features extracted using CountVectorizer:"
                                   xtrain_dtm.shape[1])
print ("In features for first 5 training instances are
                             listed below")
df = pd.DataFrame (xtrain_dtm.toarray(),column=
                                   count_vect.get_feature_names())
print (df[0:5])

from sklearn.naive_bayes import MultiNominalNB
df = MultiNominalNB().fit (xtrain_dtm , ytrain)
predicted = df.predict (xtest_dtm)
print (" In Classification results of testing samples
           are given below")
for doc, p in zip (xtest, predicted):
        pred = 'pos' if p==1 else 'neg'
        print ("%s → %s" % (doc, pred))

from sklearn import metrics
print ("In Accuracy metrics")
print ("In Accuracy of the classifier is", matrices.
                         accuracy_score (ytest, predicted))
```

```
print ("Recall :", metrices.recall (ytest, predicted"))
print ("Precision:", metrices.precision_t~~ree~~score
                                  (ytest, predicted))
print (" Confusion matrix")
print(" metrices.confusion-matrix (ytest,predicted))
```

Teacher's Signature :

Output:-

I love this sandwich	pos
This is an amazing place	pos
I feel very good about these beens	pos
This is my best work	pos
What an awesome view	pos
I donot like restaurant	neg
Iam tired of this stuff	neg
I can't deal with this	neg
He is my sworn enemy	neg
My boss is horrible	neg
This is an awesome place	pos
Idont like taste of this juice	neg
I love to dance	pos
Iam sick and tired of this place	neg
what a great holiday	pos
This is a bad locality to stay	neg
We will have good fun tomarrow	pos
I went to my enemy's house today	neg

Dimension of Dataset : (18, 2)

Total number of training data : (13,)
Total number of testing data : (5,)

GAUTHAMI.U.G. 4MT17CS041

Total instances in the dataset : 18

The message and its label of first 5 instances are listed below :

I Love this sandwich , pos
This is an amazing place, pos
I feel very good about these beers, pos
This is my best work, pos
what an awesome view, pos

Dataset is split into Training and Testing samples
Total training instances : 13
Total testing instances : 5

Total features extracted using countVectorizer : 46

Features for first 5 training instances are listed below

	about	am	an	awesome	beers	best	boss	can	deal	do	...	today
0	0	0	0	0	0	0	0	0	0	1	...	0
1	0	0	0	0	0	0	0	0	0	0	...	1
2	0	0	0	0	0	0	0	1	1	0	...	0
3	0	0	1	1	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	0	...	0

```
   tomarrow  very  view  we  went  what  will  with  work
0     0       0     0    0    0     0     0     0     0
1     0       0     0    0    1     0     0     0     0
2     0       0     0    0    0     0     0     1     0
3     0       0     0    0    0     0     0     0     0
4     0       0     0    0    0     0     0     0     0

[5 rows x 46 columns]
```

classification results of testing samples are given below:

```
I Love to dance → pos
I am sick and tired of this place → neg
This is an amazing place → pos
what a great holiday → pos
This is a bad locality to stay → ~~pos~~ neg
```

Accuracy metrices

```
Accuracy of the decision is 1.0
Recall : 1.0
Precision : 1.0

Confusion Matrix:
[[2 0]
 [0 3]]
```