Ex.no: 7

Naive Bayes Algorithm for Learning and classifying Text Documents.

Aim:

To Assume a set of Documents that need to be classified, use the naive Bayesian classifier Model to perform this task.

Algorithm:

1. Given training data set D which consits of Documents belonging to different class say class A and B.

2. Calculate the prior probability of class A = number of objects of class A | Total number of objects.

Calculate the prior probability of class B = number of objects of class B | Total number of objects.

3. Find ni, the Total number of word frequency of each class.

na = Total number of word frequency of class A.

nb = Total number of word frequency of class B.

4. Find Conditional probability of Keyword Occurence given a class

P (word 1 | class A) = wordcount / (ni(A)

P (word 1 | class B) = wordcount / (ni(B)

P (word 2 | class A) = word count / (ni (A)

P (word 2 | class B) = wordcount | ni (B)

. . . . – – – – – – –

. . . . – – – – – – – –

P (word n | class B) = word count | ni (B)

5. Avoid zero frequency problems by applying uniform Distribution.

6. classify a new Document C based on the probability P(C|W)

a) Find P(A|W) = P(A)*P(word1|classA)*P(word2|class A) . . . . *P(wordn| class A)

b) Find P(B|W) = P(B)*P(word1|classB)*P(word2|classB) . . . . *P(wordn|class B)

7. Assign Document to class the higher probability.

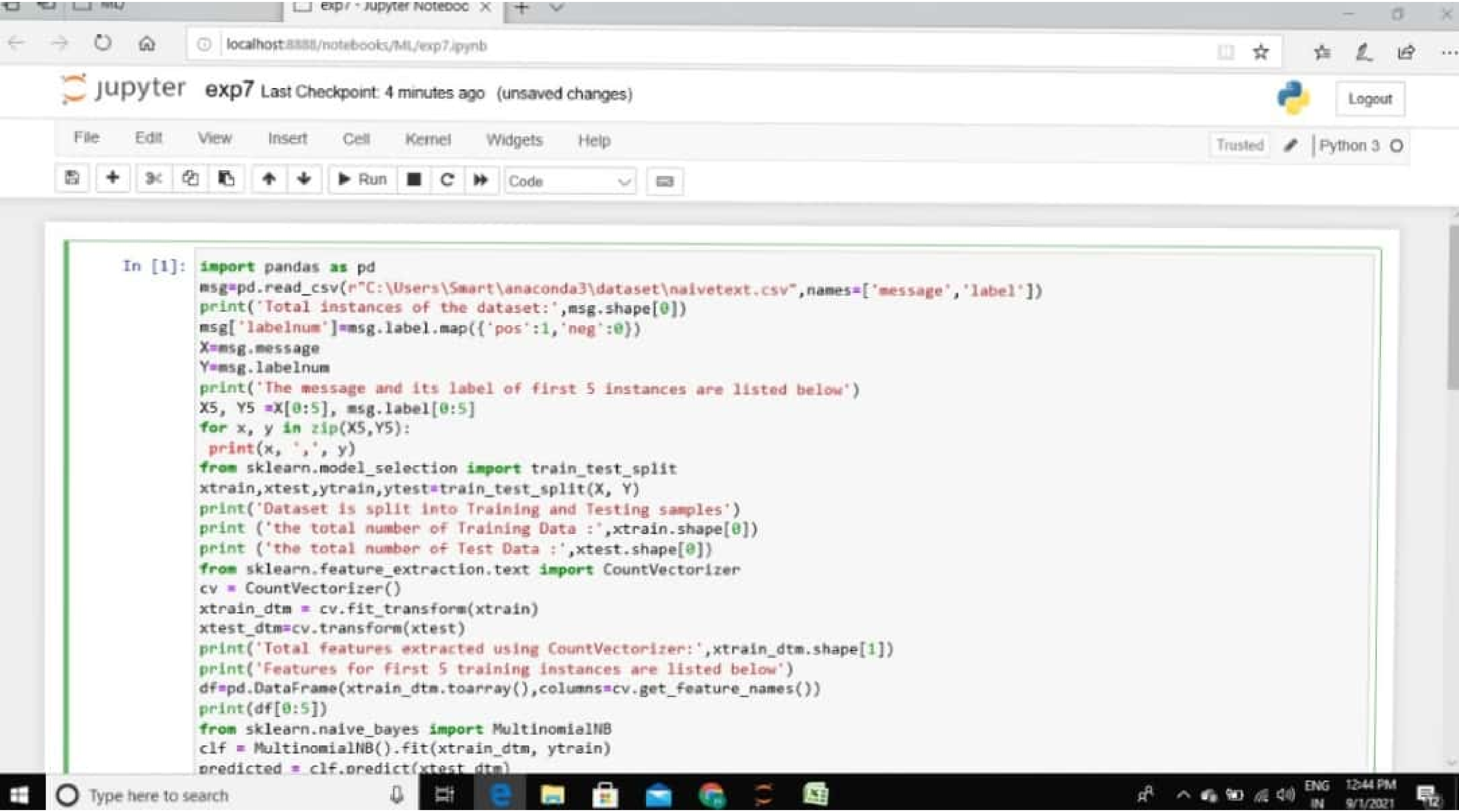
Code:

```
import pandas as pd
msg = pd.read-csv(r"c:\users\Smart\anaconda3\dataset\naivetext.csv",
                                    names=['Message','label'])
Print("Total Instances of the dataset:", msg.shape[0])
~~Print~~ msg['labelnum'] = msg.label.map({'pos':1','neg':0})
 x = msg.Message
 y = msg.labelnum
 print("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('The total number of Training Data :', xtrain.shape[0])
print('The total number of Test Data :', xtest.shape[0])
from Sklearn.feature-extraction.test import Count Vectorizer
 cv= Count Vectorizer()
```

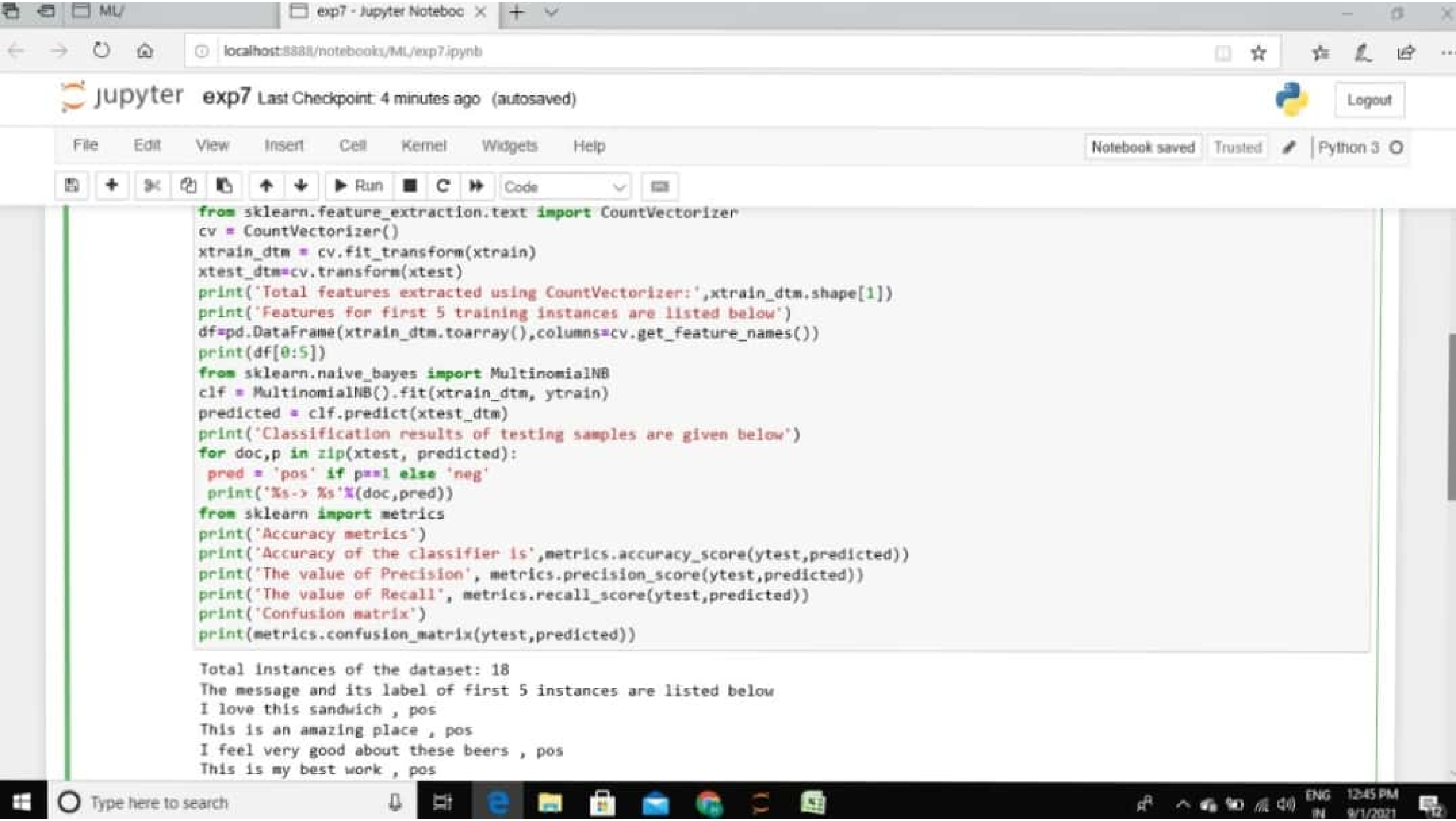
```
xtrain_dtm=cv.fit_transform(xtrain)
xtest_dtm=cv.transform(xtest)
print("Total features extracted using count vectorizer:', xtrain_dtm.
                                                          shape[1])
Print("Features for first 5 training instances are listed below')
df=pd.dataframe(xtrain_dtm.toarray(), columns=cv.get_feature_
                                                          names())
Print(df[0:5])
from sklearn.naive_bayes import MultinomialNB
clf=MultinomialNB().fit(xtrain_dtm,ytrain)
predicted =clf.predict(xtest_dtm)
print('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('Accuracy Matrix')
print('Accuracy of the classifier is', Metrics.accuracy_score
                                              (ytest, predicted))
print('The value of Precision', Metrics.precision_score(ytest, predicted))
print('The value of Recall', Metrics.recall_score(ytest, predicted))
print('Confusion Matrix')
print(Metrics.Confusion_Matrix(ytest, predicted))
```

Result:

Thus the naive bayes Algorithm has been Implemented Successfully.

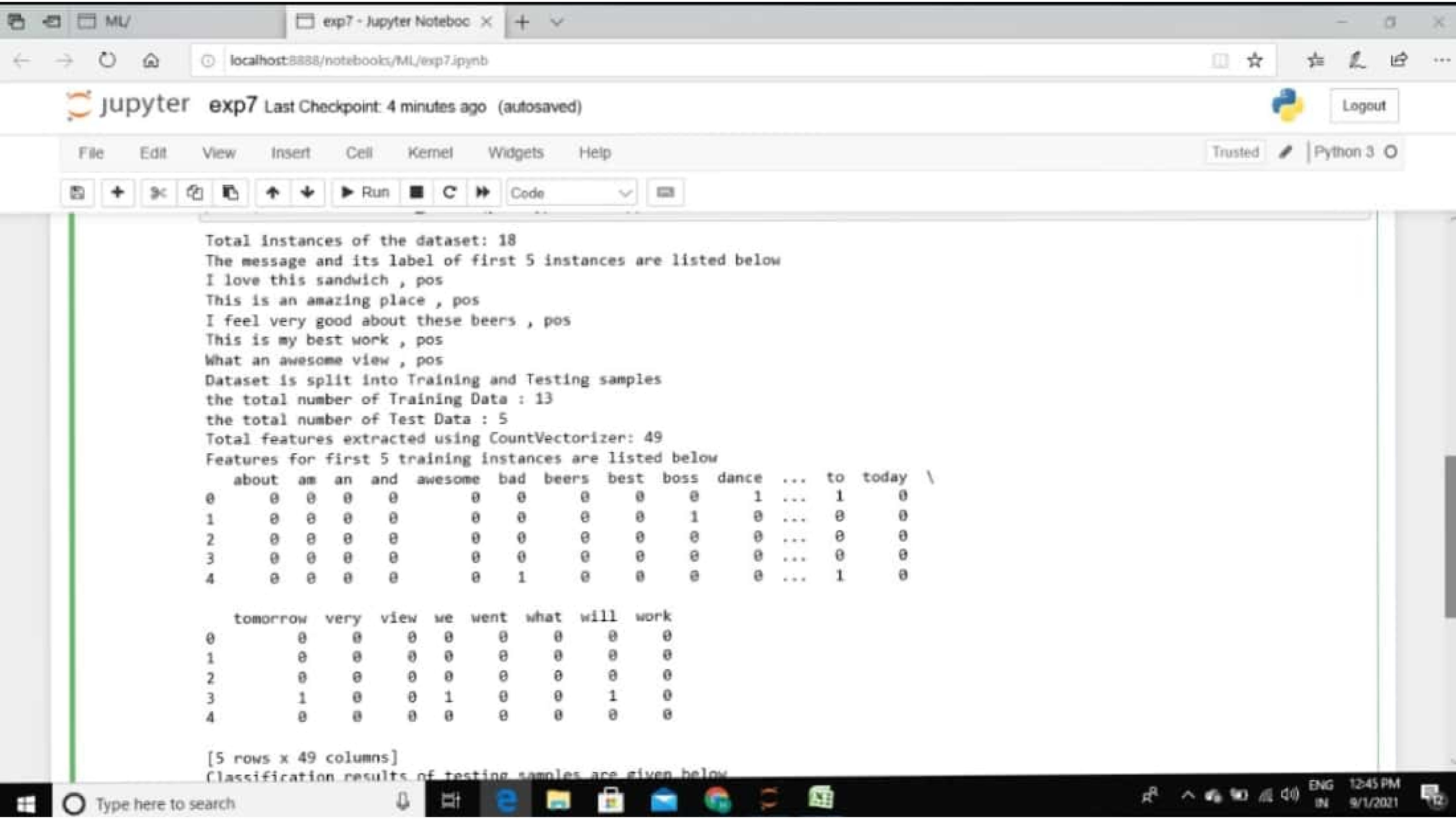


```
import pandas as pd
msg=pd.read_csv(r"C:\Users\Smart\anaconda3\dataset\naivetext.csv",names=['message','label'])
print('Total instances of the dataset:',msg.shape[0])
msg['labelnum']=msg.label.map({'pos':1,'neg':0})
X=msg.message
Y=msg.labelnum
print('The message and its label of first 5 instances 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 ('the total number of Training Data :',xtrain.shape[0])
print ('the total number of Test Data :',xtest.shape[0])
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
xtrain_dtm = cv.fit_transform(xtrain)
xtest_dtm=cv.transform(xtest)
print('Total features extracted using CountVectorizer:',xtrain_dtm.shape[1])
print('Features for first 5 training instances are listed below')
df=pd.DataFrame(xtrain_dtm.toarray(),columns=cv.get_feature_names())
print(df[0:5])
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(xtrain_dtm, ytrain)
predicted = clf.predict(xtest_dtm)
```



```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
xtrain_dtm = cv.fit_transform(xtrain)
xtest_dtm=cv.transform(xtest)
print('Total features extracted using CountVectorizer:',xtrain_dtm.shape[1])
print('Features for first 5 training instances are listed below')
df=pd.DataFrame(xtrain_dtm.toarray(),columns=cv.get_feature_names())
print(df[0:5])
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(xtrain_dtm, ytrain)
predicted = clf.predict(xtest_dtm)
print('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('Accuracy metrics')
print('Accuracy of the classifier is',metrics.accuracy_score(ytest,predicted))
print('The value of Precision', metrics.precision_score(ytest,predicted))
print('The value of Recall', metrics.recall_score(ytest,predicted))
print('Confusion matrix')
print(metrics.confusion_matrix(ytest,predicted))
```

```
Total instances of 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
```

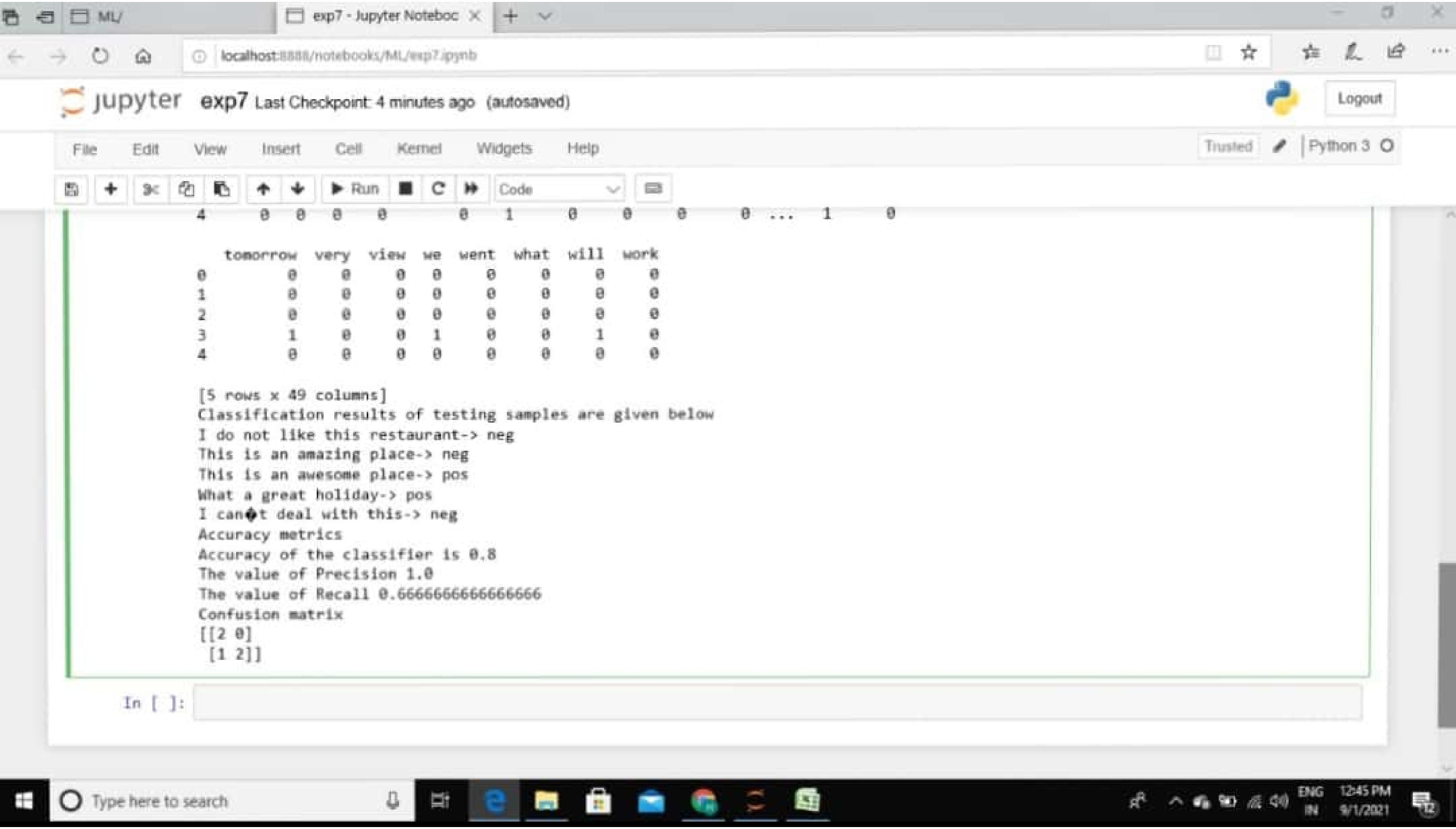

ML/
exp7 - Jupyter Noteboo
localhost:8888/notebooks/ML/exp7.ipynb
jupyter exp7 Last Checkpoint: 4 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG IN
12:45 PM 9/1/2021

```
Total instances of 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
the total number of Training Data : 13
the total number of Test Data : 5
Total features extracted using CountVectorizer: 49
Features for first 5 training instances are listed below
   about  am  an  and  awesome  bad  beers  best  boss  dance  ...  to  today  \
0      0   0   0    0        0    0      0     0     0      1  ...   1      0
1      0   0   0    0        0    0      0     0     1      0  ...   0      0
2      0   0   0    0        0    0      0     0     0      0  ...   0      0
3      0   0   0    0        0    0      0     0     0      0  ...   0      0
4      0   0   0    0        0    1      0     0     0      0  ...   1      0

   tomorrow  very  view  we  went  what  will  work
0         0     0     0   0     0     0     0     0
1         0     0     0   0     0     0     0     0
2         0     0     0   0     0     0     0     0
3         1     0     0   1     0     0     1     0
4         0     0     0   0     0     0     0     0

[5 rows x 49 columns]
Classification results of testing samples are given below
```

Scanned with CamScanner



```
4      0   0   0    0        0    1      0     0     0      0 ...   1     0

   tomorrow  very  view  we  went  what  will  work
0         0     0     0   0     0     0     0     0
1         0     0     0   0     0     0     0     0
2         0     0     0   0     0     0     0     0
3         1     0     0   1     0     0     1     0
4         0     0     0   0     0     0     0     0

[5 rows x 49 columns]
Classification results of testing samples are given below
I do not like this restaurant-> neg
This is an amazing place-> neg
This is an awesome place-> pos
What a great holiday-> pos
I can�t deal with this-> neg
Accuracy metrics
Accuracy of the classifier is 0.8
The value of Precision 1.0
The value of Recall 0.6666666666666666
Confusion matrix
[[2 0]
 [1 2]]
```