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

msg=pd.read\_csv('DATA.csv',names=['message','label'])

print('The dimensions of the dataset',msg.shape)

msg.head()

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

X=msg.message

y=msg.labelnum

#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)

#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())

# 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))\*100)

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))