GAWIN BRITO 4MT17CS039

PAGE NO : 21
DATE : 22/12/20
EXP.NO. : 7

7 Assuming a set of documents that need to be classified, use the naive Bayesian classifier model to perform this task. Built in JAVA class/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('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("\n 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)
```

ARUN'S

Scanned with CamScanner

GAGWIN BRITO 4MT17CS039

PAGE NO : 22
DATE : 22/12/20
EXP.NO. : 7

```
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.transform
                        (xtest)
print("\nTotal features extracted
        using CountVectorizer :")
                xtrain_dtm.shape[1])
print("\n Features for first 5 training
instances are listed below")
df = pd.DataFrame(xtrain_dtm.
toarray(), columns = count_vect.
        get_feature_names())
```

GAHWIN BRITO 4MT17CS039

```
print(df[0:5])

from sklearn.naive_bayes import
                          MultiNomialNB
df = MultiNomialNB().fit(xtrain_
                          dtm, ytrain)
predicted = df.predict(xtest_dtm)
print("\n 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("\n Accuracy metrices")
print("\n Accuracy of the classifier
is" matrices.accuracy_score(ytest,
                          predicted))
print("recall:", metrices.recall_score
              (ytest, predicted))
print("Precision:", metrices.precision
          _score(ytest, predicted))
print("confusion matrix")
print(metrices.confusion_matrix
    (ytest, predicted))
```

GALWIN BRITO . 4MT17CS039

Output:

```
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 video, pos

Dataset is split into Training &
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
```

```
   tomorrow  very  view  we  went  what  will  with  wor
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 -> neg

Accuracy metrics
Accuracy of the classifier is 1.0
Recall : 1.0
Precision :1.0
confusion matrix:

```
[[2 0]
 [0 3]]
```