Lab Program-6

Assuming a set of documents that need to be 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.

Algorithm to train and derive inference from the Naive Bayes model

```
TRAINMULTINOMIALNB(C, D)
  V ← ExtractVocabulary(D)
  N ← CountDocs(D)
  for each c ∈ C
  do N_c ← Countdocsinclass(D, c)
       prior[c] ← N_c/N
       text_c ← ConcatenateTextofalldocsinclass(D, c)
       for each t ∈ V
           do T_ct ← Counttokensofterms(text_c, t)
       for each t ∈ V
       do condprob[t][c] ← $\frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$

  return V, prior, condprob
```

```
ApplyMultinomialNB(C, V, prior, condprob, d)
  W ← ExtractTokensFromDoc(V, d)
  for each c ∈ C
  do score[c] ← log prior[c]
        for each t ∈ W
        do score[c] += log condprob[t][c]
  return arg max_{c∈C} score[c]
```

$$P(t/c) = \frac{T_{ct}+1}{\Sigma_{t'\in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\Sigma_{t'\in V} T_{ct'}) + B'}$$

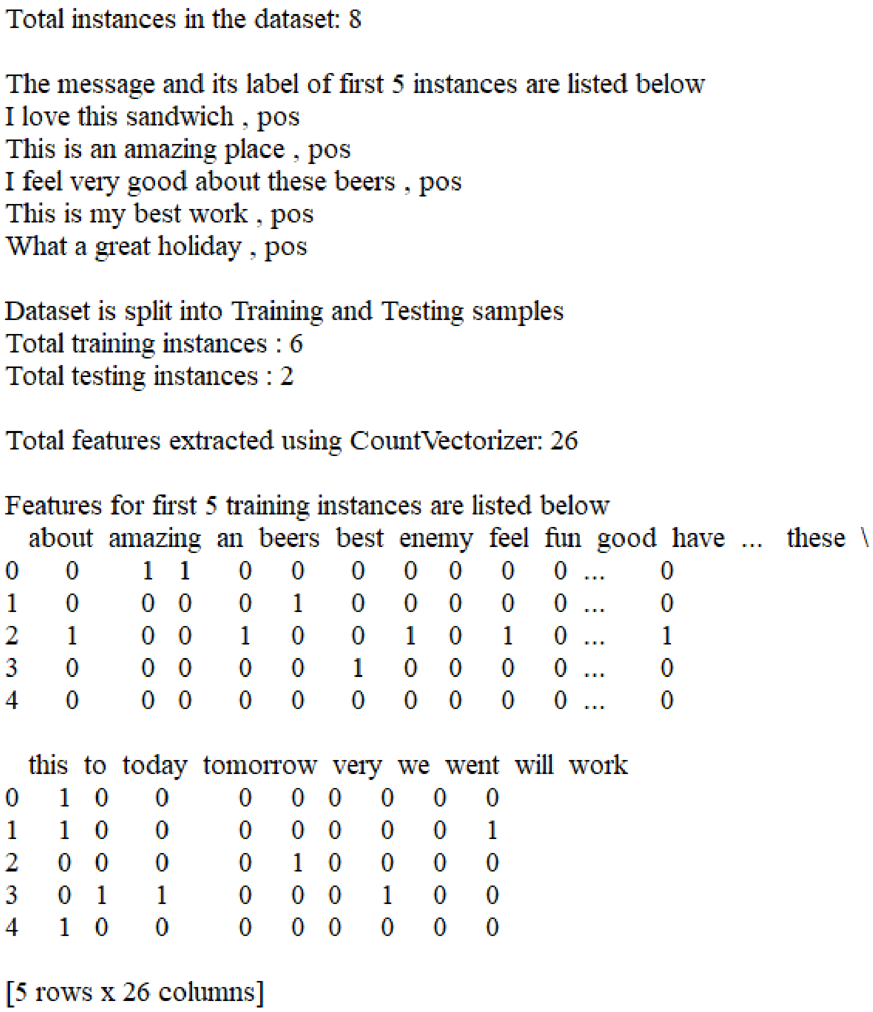
Program-

```
from sklearn.datasets import fetch_20newsgroups.
from sklearn.metrics import confusion_matrix.
from sklearn.metrics import classification_report
import numpy as np
categories = ['alt.atheism', 'soc.religion.christian',
              'comp.graphics', 'sci.med'].

twenty_train = fetch_20newsgroups(subset='train',
        categories=categories, shuffle=True)

twenty_test = fetch_20newsgroups(subset='test',
        categories=categories, shuffle=True)
```

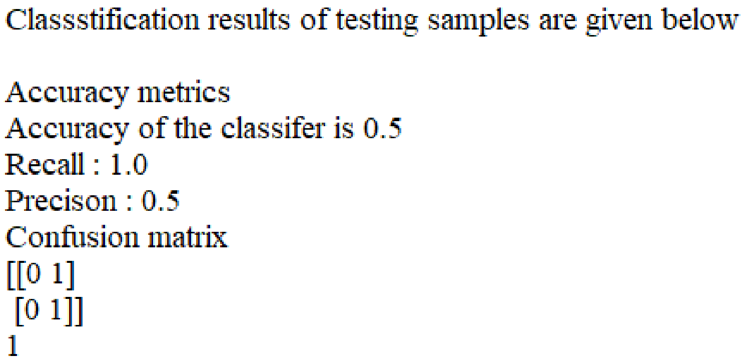
```
print(len(twenty_train.data))
print(len(twenty_test.data))
print(twenty_train.target_names)
print("\n".join(twenty_train.data[0].split("\n")))
print("Target", twenty_train.target)
from sklearn.feature_extraction.text import CountVectorizer
count_vect= CountVectorizer()
X_train_tf = count_vect.fit_transform(twenty_train.data)
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_tf)

from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn import metrics
mod = MultinomialNB()
mod.fit(X_train_tfidf, twenty_train.target)
X_test_tf = count_vect.transform(twenty_test.data)
X_test_tfidf = tfidf_transformer.transform(X_test_tf)
predicted = mod.predict(X_test_tfidf)
print("Predicted", predicted)
print("Accuracy", accuracy_score(twenty_test.target,
                      predicted))
print("Classification_report(twenty_test.target, predicted,
          target_names= twenty_test.target_names))
print("confusion matrix \n", metrics.confusion_matrix(twenty_test.
                      target, predicted)).
```

Teacher's Remarks

Teacher's Signature



```
Total instances in the dataset: 8

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 a great holiday , pos

Dataset is split into Training and Testing samples
Total training instances : 6
Total testing instances : 2

Total features extracted using CountVectorizer: 26

Features for first 5 training instances are listed below
   about  amazing  an  beers  best  enemy  feel  fun  good  have  ...   these  \
0      0        1   1      0     0      0     0    0     0     0  ...       0
1      0        0   0      0     1      0     0    0     0     0  ...       0
2      1        0   0      1     0      0     1    0     1     0  ...       1
3      0        0   0      0     0      1     0    0     0     0  ...       0
4      0        0   0      0     0      0     0    0     0     0  ...       0

   this  to  today  tomorrow  very  we  went  will  work
0     1   0      0         0     0   0     0     0     0
1     1   0      0         0     0   0     0     0     1
2     0   0      0         0     1   0     0     0     0
3     0   1      1         0     0   0     1     0     0
4     1   0      0         0     0   0     0     0     0

[5 rows x 26 columns]
```



```
Classstification results of testing samples are given below

Accuracy metrics
Accuracy of the classifer is 0.5
Recall : 1.0
Precison : 0.5
Confusion matrix
[[0 1]
 [0 1]]
1
```