Pgm 6: Assuming a set of documents that need to be classified, use the **naïve 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 data set.

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)

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("Accuracy:", accuracy\_score(twenty\_test.target, predicted))

print(classification\_report(twenty\_test.target,predicted,target\_names=twenty\_test.target\_names))

print("confusion matrix is \n",metrics.confusion\_matrix(twenty\_test.target, predicted))

**Output**

2257

1502

['alt.atheism', 'comp.graphics', 'sci.med', 'soc.religion.christian']

Accuracy: 0.8348868175765646

precision recall f1-score support

alt.atheism 0.97 0.60 0.74 319

comp.graphics 0.96 0.89 0.92 389

sci.med 0.97 0.81 0.88 396

soc.religion.christian 0.65 0.99 0.78 398

micro avg 0.83 0.83 0.83 1502

macro avg 0.89 0.82 0.83 1502

weighted avg 0.88 0.83 0.84 1502

confusion matrix is

[[192 2 6 119]

[ 2 347 4 36]

[ 2 11 322 61]

[ 2 2 1 393]]