AI
Lab-8

MAJJIGA JASWANTH
20BCD7171
QUESTION: Write a program to implement bayes rule.
CODE:

```
package com.gg.ml;

import java.io.File;

import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayesMultinomial;
import weka.core.Instances;
import weka.core.converters.ArffLoader;
import weka.filters.Filter;
import
weka.filters.unsupervised.attribute.StringToWordVector;

public class NaiveBayesDemo {
    public static final String TRAINING_DATA_SET_FILENAME
= "naive-train.arff";
    public static final String TESTING_DATA_SET_FILENAME =
"naive-test.arff";
    public static final String
PREDICTION_DATA_SET_FILENAME = "naive-confused.arff";

    public static Instances getDataSet(String fileName)
throws Exception {

        StringToWordVector filter = new
StringToWordVector();
        int classIdx = 1;

        ArffLoader loader = new ArffLoader();
         loader.setSource(NaiveBayesDemo.class.getResource
AsStream("/"+fileName));

        Instances dataSet = loader.getDataSet();
        dataSet.setClassIndex(classIdx);
        filter.setInputFormat(dataSet);
        dataSet = Filter.useFilter(dataSet, filter);
        return dataSet;
    }

    public static void process() throws Exception {

        Instances trainingDataSet =
getDataSet(TRAINING_DATA_SET_FILENAME);
        Instances testingDataSet =
getDataSet(TESTING_DATA_SET_FILENAME);
        Instances predictingDataSet =
getDataSet(PREDICTION_DATA_SET_FILENAME);
        Classifier classifier = new
NaiveBayesMultinomial();

        classifier.buildClassifier(trainingDataSet);

        Evaluation eval = new Evaluation(trainingDataSet);
        eval.evaluateModel(classifier, testingDataSet);

        System.out.println("** Naive Bayes Evaluation with
Datasets **");
        System.out.println(eval.toSummaryString());
        System.out.print(" the expression for the input
data as per alogorithm is ");
        System.out.println(classifier);
        for (int i = 0; i <
predictingDataSet.numInstances(); i++) {
            System.out.println(predictingDataSet.instance(
i));
            double index =
classifier.classifyInstance(predictingDataSet.instance(i))
;
            String className =
trainingDataSet.attribute(0).value((int) index);
            System.out.println(className);
        }

    }
}
```

Output:

```
Correctly Classified Instances          7              100      %
Incorrectly Classified Instances        0                0      %
Kappa statistic                         1
Mean absolute error                     0.1378
Root mean squared error                 0.1444
Relative absolute error                28.0006 %
Root relative squared error            29.1716 %
Total Number of Instances               7

 the expression for the input data as per alogorithm is The independent
probability of a class
--------------------------------------
spam    0.5555555555555556
ham     0.4444444444444444

The probability of a word given the class
-----------------------------------------
        spam    ham
Congrats        0.07407407407407407    0.043478260869565216
cards   0.07407407407407407    0.043478260869565216
credit  0.07407407407407407    0.043478260869565216
for     0.11111111111111109    0.043478260869565216
free    0.07407407407407407    0.043478260869565216
lottery 0.11111111111111109    0.043478260869565216
selected        0.07407407407407407    0.08695652173913045
travel  0.07407407407407407    0.043478260869565216
won     0.07407407407407407    0.043478260869565216
you     0.07407407407407407    0.08695652173913045
Congratulation  0.037037037037037035   0.08695652173913045
Good    0.037037037037037035   0.13043478260869565
are     0.037037037037037035   0.08695652173913045
night   0.037037037037037035   0.08695652173913045
very    0.037037037037037035   0.08695652173913045

{0 ?,1 1,2 1,3 1}
spam
```