**PROBLEM**

This problem is to be implemented in R.

**Dataset**

The dataset is taken and modified from causality workbench:

http://www.causality.inf.ethz.ch/challenge.php?page=datasetshttp://www.causality.inf.ethz.ch/challenge.php?page=datasets

**To do**

1. Read the dataset training.txt First 9 columns are features and the last column is the label.

2. Generate your naïve Bayes' model using the training set

3. Determine the predictions on the test data given in testing.txt file.

**Coding in R**

To understand implementation, let’s take this example:

TRAINING DATA (labels known)

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | Label |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

TEST DATA (labels not known)

|  |  |  |
| --- | --- | --- |
| X1 | X2 | X3 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |

**NOTE:** Be systematic when implementing your program. You can implement an individual script or two scripts, one for training and the other for testing in R.

**R Commands: Accessing data, rows and columns**

**All indices in R start from 1**

datAll=read.table(“…”) #replace … with filename

# above is a built in function for reading text files. The entire dataset will be stored in datAll

f1 = datAll[:,1] #this stores first column in f1

r1 = datAll[1,] #this stores first row in r1

ncol(datAll) # this returns total cols in datAll

x= dat[,-1] #stores all columns of dat except column 1 in x

labels = datAll[,ncol(datAll)] #this will store last column of datAll in labels

trainX = datAll[,-ncol(datAll)] #this will store all features except label in dat

oneClass = labels==1 #this will indicate true for all values for class = 1

zeroClass = labels==0 #this will indicate true for all values for class = 0

oneDat = trainX[oneClass,] #this will give you the data matrix for class = 1

zeroDat = trainX[zeroClass,] #this will give you the data matrix for class = 0

**Example commands needed for training phase**

probOf1Class1 = colMeans(oneDat) #this gives the mean vector of columns

**Example commands needed for test phase**

Row1 = testX[1,] #gives first row of testX

ind = Row1 == 1 #ind has indices of all values of Row1 equal to one

Row1[ind] = probOf1Class1[ind] #this replaces all values at ind with corresponding

#probability

#do the same for probOf0Class1 (replace zero by their

#probabilities

prod(Row1) #will return product of all elements of Row1

**Generating model**

Write a script in R to learn all probabilities using trainX and labels. You need to compute the following:

1. probOf1Class0 and probOf0Class0 (both these are vectors)
2. probOf1Class1 and probOf1Class1 (both these are vectors. Computed above for class 1)
3. prior0
4. prior1

For example for three binary features in the example ProbOf1Class1 would be a vector of probabilities

[3/7 4/7 4/7] T. **NOTE:** For all the above 4 parameters you do not need a for loop

**Testing the model / Predicting the labels**

1. Compute the likelihood P(**x**|C=0) and P(x|C=1) for each row and predict the label using ML
2. Compute the posterior P(C=0|**x**) and P(C=1|**x**) for each row and predict the label using MAP

3. One answer for posterior is given in the report and you can use it.

NOTE: Both the above steps can be done without a for loop. However, for this assignment you can use a for loop if you find it easier to use it.

**Example of test**

If your test point is **z**=[0 1 0] T then P(**z**|C=1) = (1-3/7)\*(4/7)\*(1-4/7)

Also, P(C=1|**z**) = (1-3/7)\*(4/7)\*(1-4/7)\*(7/10) / P(**z**)

P(**z**) = (1-3/7)\*(4/7)\*(1-4/7)\*(7/10) + (2/3)\*(2/3)\*(2/3)\*(3/10)

**Main script**

Once you have implemented the above functions write a main script that:

1. Reads training data
2. Finds the model parameters (relevant to the distribution to use)
3. Reads the test data and classifies the test data

**OPTIONAL**

You can smooth the probability estimates to obtain better results as discussed in class. In this case you need to mention the value of smoothing parameter used.