

PART B

Roll No. C035	Name: Krisha Goti
Class : B	Batch : B1
Date of Experiment: 17/09/2022	Date of Submission: 17/09/2022
Grade :	

B.1 Classification Code and screenshot by student:

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for loading libraries, reading data from 'sample1.csv', and performing NB classification. The code includes steps for data preparation, model building, and evaluation.
- Environment:** Lists the objects created in the R session: 'model', 'model1', 'sample', 'testdata', and 'traindata'. It also shows the 'Values' for 'agecounts', 'desirecounts', 'incomecounts', 'jscounts', 'pno', 'pyes', 'results', and 'results1'.
- Console:** Shows the execution output of the R code, including the data frames 'traindata' and 'testdata', and the results of the classification process.
- Files:** A file explorer showing the project structure, including folders like 'SAS_Enterprise_Guide_Independent_Installer', 'SAP', 'R', 'oracle 11g exp ed', 'My Videos', 'My Pictures', 'My Music', 'MATLAB', 'LabVIEW Data', 'ER diagram.drawio', 'ds2c', 'ds2c.exe', 'ds2capp', 'desktop.ini', 'Custom Office Templates', 'Rhistory', and 'sample1.csv'.

RStudio interface showing a script for Naïve Bayes classification. The script defines data, computes probabilities, and uses the `naiveBayes` function. The console output shows the results of the classification, including the maximum probability and the predicted class for a specific instance.

```
21 incomecounts <- incomecounts/rowSums(incomecounts)
22 jscounts <- table(traindata[,c("enrolls", "jobsatisfaction")])
23 jscounts <- jscounts/rowSums(jscounts)
24 desircounts <- table(traindata[,c("enrolls", "desire")])
25 desircounts <- desircounts/rowSums(desircounts)
26 #predict - Compute the probabilities for Age<=30, Income = Medium,
27 #jobsatisfaction = yes and desire = Fair
28 pyes <-
29   agecounts["yes", "<=30"]+
30   incomecounts["yes", "medium"]+
31   jscounts["yes", "yes"]+
32   desircounts["yes", "Fair"]+
33   tprior["yes"]
34 pno <-
35   agecounts["no", "<=30"]+
36   incomecounts["no", "medium"]+
37   jscounts["no", "yes"]+
38   desircounts["no", "Fair"]+
39   tprior["no"]
40 print(pyes)
41 print(pno)
42 print(max(pyes,pno))
43 # use the nb classifier
44 model <- naiveBayes(enrolls ~., traindata)
45 # display model
46 model
```

Console output:

```
R 4.1.1 ~ /
> jscounts <- table(traindata[,c("enrolls", "jobsatisfaction")])
> jscounts <- jscounts/rowSums(jscounts)
> desircounts <- table(traindata[,c("enrolls", "desire")])
> desircounts <- desircounts/rowSums(desircounts)
> #predict - Compute the probabilities for Age<=30, Income = Medium,
> #jobsatisfaction = yes and desire = Fair
> pyes <-
+   agecounts["yes", "<=30"]+
+   incomecounts["yes", "medium"]+
+   jscounts["yes", "yes"]+
+   desircounts["yes", "Fair"]+
+   tprior["yes"]
> pno <-
+   agecounts["no", "<=30"]+
+   incomecounts["no", "medium"]+
+   jscounts["no", "yes"]+
+   desircounts["no", "Fair"]+
+   tprior["no"]
> print(pyes)
+
+   0.02821869
> print(pno)
+
+   0.006857143
> print(max(pyes,pno))
+ [1] 0.02821869
> # use the nb classifier
> model <- naiveBayes(enrolls ~., traindata)
> # display model
> model
```

Environment pane:

Object	Class	Attributes
data	data.frame	
model1	list	of 5
model11	list	of 5
sample	list	of 5 variables
testdata	list	of 5 variables
traindata	list	of 5 variables
values	list	of 5 variables
agecounts	table	num [1:2, 1:3] 0.6 0.222 0.4 0.333 0 ...
desircounts	table	num [1:2, 1:2] 0.6 0.333 0.4 0.667
incomecounts	table	num [1:2, 1:3] 0.4 0.222 0.2 0.333 0.4 ...
jscounts	table	num [1:2, 1:2] 0.8 0.333 0.2 0.667
pno	named num	0.00686
pyes	named num	0.0282
results	Factor w/ 2 levels "no","yes": 2	
results1	Factor w/ 2 levels "no","yes": 2	

RStudio interface showing the same script as above, but with additional output from the `naiveBayes` function. The console output displays the prior probabilities, conditional probabilities, and the predicted class for a specific instance.

```
25 desircounts <- desircounts/rowSums(desircounts)
26 #predict - Compute the probabilities for Age<=30, Income = Medium,
27 #jobsatisfaction = yes and desire = Fair
28 pyes <-
29   agecounts["yes", "<=30"]+
30   incomecounts["yes", "medium"]+
31   jscounts["yes", "yes"]+
32   desircounts["yes", "Fair"]+
33   tprior["yes"]
34 pno <-
35   agecounts["no", "<=30"]+
36   incomecounts["no", "medium"]+
37   jscounts["no", "yes"]+
38   desircounts["no", "Fair"]+
39   tprior["no"]
40 print(pyes)
41 print(pno)
42 print(max(pyes,pno))
43 # use the nb classifier
44 model <- naiveBayes(enrolls ~., traindata)
45 # display model
46 model
47 # predict with testdata
48 results <- predict(model, testdata)
49 # display results
50 results
```

Console output:

```
R 4.1.1 ~ /
Call:
naiveBayes.default(x = X, y = y, laplace = laplace)

A-priori probabilities:
y
  no  yes
0.3571429 0.6428571

conditional probabilities:
Age
  no <=30  >30  no >30  >30
no 0.6000000 0.4000000 0.0000000
yes 0.2222222 0.3333333 0.4444444

Income
  no  high  low  medium
no 0.4000000 0.2000000 0.4000000
yes 0.2222222 0.3333333 0.4444444

Jobsatisfaction
  no  yes
no 0.8000000 0.2000000
yes 0.3333333 0.6666667

Desire
  no  excellent  fair
no 0.6000000 0.4000000
yes 0.3333333 0.6666667

> # predict with testdata
> results <- predict(model, testdata)
> results
```

Environment pane:

Object	Class	Attributes
data	data.frame	
model1	list	of 5
model11	list	of 5
sample	list	of 5 variables
testdata	list	of 5 variables
traindata	list	of 5 variables
values	list	of 5 variables
agecounts	table	num [1:2, 1:3] 0.6 0.222 0.4 0.333 0 ...
desircounts	table	num [1:2, 1:2] 0.6 0.333 0.4 0.667
incomecounts	table	num [1:2, 1:3] 0.4 0.222 0.2 0.333 0.4 ...
jscounts	table	num [1:2, 1:2] 0.8 0.333 0.2 0.667
pno	named num	0.00686
pyes	named num	0.0282
results	Factor w/ 2 levels "no","yes": 2	
results1	Factor w/ 2 levels "no","yes": 2	

```
53 model1 <- naiveBayes(enrolls ~., traindata, laplace=.01)
54 # display model
55 model1
56 # predict with testdata
57 results1 <- predict(model1, testdata)
58 # display results
59 results1
60
61 #part2 - Read in the data for the model
62 #read input data from the database
63 #first up let us load the library RODBC
64 library("RODBC")
65 ch <- odbcconnect("Greenplum", uid="gpadmin",
66 case="postgres", pwd="changeme")
67 sqlQuery(ch, "select * from train")
68 sqlQuery(ch, "select * from test")
69 CREATE TABLE nbtrain (
70 age VARCHAR(8),
71 sex VARCHAR(8),
72 educ VARCHAR(8),
73 income VARCHAR(8) )
74 DISTRIBUTED BY (age);
75 INSERT INTO nbtrain
76 SELECT
77 t1.age
78 FROM
79 t1
80 WHERE
81 t1.age < 40
```

Console output:

```
R 4.1.1 > # predict with testdata
> results1 <- predict(model1, testdata)
> # display results
> results1
[1] Yes
Levels: No Yes
> # use the nb classifier with Laplace smoothing
> model1 <- naiveBayes(enrolls ~., traindata, laplace=.01)
> # display model
> model1
Naive Bayes Classifier for Discrete Predictors

call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y          No      Yes
0.3571429 0.6428571

Conditional probabilities:
Y      Age      <=30      >40      31 to 40
No 0.6020000 0.4020000 0.0020000
Yes 0.2233333 0.3344444 0.4455556

Y      Income      High      Low      Medium
No 0.4020000 0.2020000 0.4020000
Yes 0.2233333 0.3344444 0.4455556
```

B.2 Observations and learning:

In this experiment, I have understood the concept and written the program for Naive Bayes Classification.

B.3 Conclusion:

After successfully completing this experiment we are able to:

1. To understand the concept of Data Mining by implementing some data mining algorithm.
2. To understand the various Classification techniques in Mining.
3. To understand Naive Bayes Classification Algorithm.
