Assignment3

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#Installing Libraries
library(reshape2)
library(gmodels)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(ISLR)
library(e1071)
```

```
#Read universalbank CSV file
UnivBank <-read.csv("UniversalBank.CSV")
```

```
#conerting variables
UnivBank$Personal.Loan<-factor(UnivBank$Personal.Loan)
UnivBank$Online<-factor(UnivBank$Online)
UnivBank$CreditCard<-factor(UnivBank$CreditCard)
```

```
set.seed(10)
#Spliting data into training 60% and validation 40%
t.index <-sample(row.names(UnivBank), 0.6*dim(UnivBank)[1])
validt.index <- setdiff(row.names(UnivBank), t.index)
t.df <- UnivBank[t.index, ]
validt.df <- UnivBank[validt.index, ]
train <- UnivBank[t.index,]
validtest <- UnivBank[t.index, ]
```

#A Create a pivot table for the training data with Online as a column variable, CC as a row variable, and Loan as a secondary row variable. The values inside the table should convey the count. In R use functions melt() and cast(), or function table(). In Python, use panda dataframe methods melt() and pivot().

```
melt.bank <- melt(train, id=c("CreditCard", "Personal.Loan"),variable="Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
cast.bank <- dcast(melt.bank,CreditCard+Personal.Loan~Online)
```

```
## Aggregation function missing: defaulting to length
```

```
cast.bank[,c(1:2,14)]
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1923
## 2          0             1    202
## 3          1             0    782
## 4          1             1     93
```

#B Consider the task of classifying a customer who owns a bank credit card and is actively using online banking services. Looking at the pivot table, what is the probability that this customer will accept the loan offer? [This is the probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and being an active user of online banking services (Online = 1)].

```
x= table(train[,c(10,13,14)])
y<-as.data.frame(x)
y
```

```
##   Personal.Loan Online CreditCard Freq
## 1             0      0          0  772
## 2             1      0          0   79
## 3             0      1          0 1151
## 4             1      1          0  123
## 5             0      0          1  300
## 6             1      0          1   40
## 7             0      1          1  482
## 8             1      1          1   53
```

#C Create two separate pivot tables for the training data. One will have Loan (rows) as a function of Online (columns) and the other will have Loan (rows) as a function of CC. #Creating pivot table for Loan (rows) as a function of Online (columns)

```
table(train[,c(10,13)])
```

```
##              Online
## Personal.Loan    0    1
##             0 1072 1633
##             1  119  176
```

#Creating pivot table for Loan (rows) as a function of CC

```
table(train[,c(10,14)])
```

```
##              CreditCard
## Personal.Loan    0    1
##             0 1923  782
##             1  202   93
```

#D Compute the following quantities [P(A | B) means “the probability ofA given B”]: i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) ii. P(Online = 1 | Loan = 1) iii. P(Loan = 1) (the proportion of loan acceptors) iv. P(CC = 1 | Loan = 0) v. P(Online = 1 | Loan = 0) vi. P(Loan = 0)

```
#i P(CC = 1 | Loan = 1)
P1 <-table(train[,c(14,10)])
S1<- P1[2,2]/(P1[2,2]+P1[1,2])
S1
```

```
## [1] 0.3152542
```

#ii P(Online = 1 | Loan = 1)

```
P2 <- table(train[, c(13,10)])
S2 <- P2[2,2]/(P2[2,2]+P2[1,2])
S2
```

```
## [1] 0.5966102
```

#iii P(Loan = 1)

```
P3<-table(train[,10])
S3<-P3[2]/(P3[2]+P3[1])
S3
```

```
##          1
## 0.09833333
```

#iv P(CC = 1 | Loan = 0)

```
P4<-table(train[,c(14,10)])
S4<-P4[2,1]/(P4[2,1]+P4[1,1])
S4
```

```
## [1] 0.2890943
```

#v P(Online = 1 | Loan = 0)

```
P5<-table(train[,c(13,10)])
S5<-P5[2,1]/(P5[2,1]+P5[1,1])
S5
```

```
## [1] 0.6036969
```

#vi P(Loan = 0)

```
P6<-table(train[,10])
S6<-P6[1]/(P6[1]+P6[2])
S6
```

```
##         0
## 0.9016667
```

#E Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC= 1, Online = 1). #NaiveBayesProbability= (S1*S2*S3)/[(S1*S2*S3)+(S4*S5*S6)]
#0.01849491/(0.01849491+0.15736368)=0.1051692

#F Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate?

#The value we got from pivot table is 0.092831 and the naive bayes is 0.1051692 and are almost similar. Pivot table value is more accurate.

#G Which of the entries in this table are needed for computing P(Loan = 1 | CC = 1, Online = 1)? Run naive Bayes on the data. Examine the model output on training data, and find the entry that corresponds to P(Loan = 1 | CC = 1, Online = 1). Compare this to the number you obtained in (E).

#Naive Bayes on training data

```
table(train[,c(10,13:14)])
```

```
## , , CreditCard = 0
##
##              Online
## Personal.Loan    0    1
##             0  772 1151
##             1   79  123
##
## , , CreditCard = 1
##
##              Online
## Personal.Loan    0    1
##             0  300  482
##             1   40   53
```

```
train_Naive<-train[,c(10,13:14)]
UnivBank_NB<-naiveBayes(Personal.Loan~.,data = train_Naive)
UnivBank_NB
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90166667 0.09833333
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.3963031 0.6036969
##   1 0.4033898 0.5966102
##
##    CreditCard
## Y           0         1
##   0 0.7109057 0.2890943
##   1 0.6847458 0.3152542
```

After running Naive bayes on data Value obtained is 0.1051692 where as value from E is 0.1051692 which is almost similar.