title: “Assignment_3” author: “Manaswini” date: ‘2022-03-06’ output: word_document —

R Markdown

#Set universalbank data as working directory #Converting characteristics attributes into factors

```
getwd()

## [1] "C:/Users/mpuru/OneDrive/Documents/Assignment3"

setwd("C:/Users/mpuru/OneDrive/Documents/Assignment3")
UniversalBank <- read.csv("~/Assignment3/UniversalBank.csv")
UniversalBank$Personal.Loan = as.factor(UniversalBank$Personal.Loan)
UniversalBank$Online = as.factor(UniversalBank$Online)
UniversalBank$CreditCard = as.factor(UniversalBank$CreditCard)
```

library(caret) library(ggplot2) library(lattice) library(e1071) library(dplyr) library(tidyr) library(ISLR) library(FNN)

#Partition data into train and test sets #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().

```
set.seed(1)
train.index <- sample(row.names(UniversalBank), 0.6*dim(UniversalBank)[1])
test.index <- setdiff(row.names(UniversalBank), train.index)
train.df <- UniversalBank[train.index, ]
test.df <- UniversalBank[test.index, ]
train <- UniversalBank[train.index, ]
test = UniversalBank[train.index,]
```

#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)].

#Calling libraries

```

library("dplyr")
library("tidyr")
library("ggplot2")
library("e1071")
install.packages("latexpdf")
install.packages("tinytex")
```

```
melted.UniversalBank =
melt(train,id=c("CreditCard","Personal.Loan"),variable= "Online")
recast.UniversalBank=dcast(melted.UniversalBank,CreditCard+Personal.Loan~Onli
ne)
recast.UniversalBank[,c(1:2,14)]
```

#Probability of Loan acceptance given having a bank credit card and user of online services is 77/3000 = 2.6%

#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.

```
library(reshape2)
library(ggplot2)
melted.UniversalBankc1 = melt(train,id=c("Personal.Loan"),variable =
"Online")

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

melted.UniversalBankc2 = melt(train,id=c("CreditCard"),variable = "Online")

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

recast.UniversalBankc1=dcast(melted.UniversalBankc1,Personal.Loan~Online)

## Aggregation function missing: defaulting to length

recast.UniversalBankc2=dcast(melted.UniversalBankc2,CreditCard~Online)

## Aggregation function missing: defaulting to length

Loanline=recast.UniversalBankc1[,c(1,13)]
LoanCC = recast.UniversalBankc2[,c(1,14)]

Loanline

##   Personal.Loan Online
## 1             0   2725
## 2             1    275

LoanCC

##   CreditCard Online
## 1          0   2122
## 2          1    878
```

#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)

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

##           Personal.Loan
## CreditCard    0    1
##          0 1924  198
##          1  801   77

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

##       Personal.Loan
## Online    0    1
##      0 1137  109
##      1 1588  166

table(train[,c(10)])

##
##    0    1
## 2725  275
```

#i. 77/(77+198)=28% #ii. 166/(166+109)= 60.3% #iii.275/(275+2725)=9.2% #iv. 801/(801+1924)=29.4% #v. 1588/(1588+1137) = 58.3% #vi. 2725/(2725+275) = 90.8%

#E. Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1).

```
((77/(77+198))*(166/(166+109))*(275/(275+2725)))/(((77/(77+198))*(166/(166+10
9))*(275/(275+2725)))+((801/(801+1924))*(1588/(1588+1137))*2725/(2725+275)))

## [1] 0.09055758
```

#F. Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate? 9.05% are very similar to the 9.7% the difference between the exact method and the naive-baise method is the exact method would need the the exact same independent variable classifications to predict, where the naive bayes method does not.

#G.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).

```
library(gmodels)
library(e1071)
naive.train = train.df[,c(10,13:14)]
naive.test = test.df[,c(10,13:14)]
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train)
naivebayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90833333 0.09166667
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4172477 0.5827523
##   1 0.3963636 0.6036364
##
##    CreditCard
## Y          0        1
##   0 0.706055 0.293945
##   1 0.720000 0.280000
```

#The naive bayes is the exact same output we recieved in the previous methods. (.280)(.603)(.09)/(.280.603.09+.29.58.908) = .09 which is the same response provided as above.