Assignment_3

Tejasvini Mavuleti

3/06/2022

```
UniversalBank <-
read.csv("C:/Users/mavul/OneDrive/Desktop/UniversalBank.csv")
View(UniversalBank)

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(reshape)

##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
##
##     rename

library(reshape2)

##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast

library(ggplot2)
library(caret)

## Loading required package: lattice

library(ISLR)
library(naivebayes)

## naivebayes 0.9.7 loaded
```

```
library(lattice)

UniversalBank$Personal.Loan <- as.factor(UniversalBank$Personal.Loan)
UniversalBank$Online = as.factor(UniversalBank$Online)
UniversalBank$CreditCard = as.factor(UniversalBank$CreditCard)

# Parting the data as training 60% and testing 40%

set.seed(64060)
Index <- createDataPartition(UniversalBank$Income, p=0.6, list = FALSE)
Train_Data <- UniversalBank[Index,]
Test_Data <- UniversalBank[-Index,]

# A. Creating a pivot table

set.seed(64060)
Melt_Train <- melt(Train_Data,id=c("CreditCard","Personal.Loan"),variable=
"Online")

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

cast_Train <- dcast(Melt_Train,CreditCard+Personal.Loan~Online)

## Aggregation function missing: defaulting to length

cast_Train <-cast_Train[c(1,2,14)]
cast_Train

##   CreditCard Personal.Loan Online
## 1          0             0   1937
## 2          0             1    169
## 3          1             0    811
## 4          1             1     85

# B.  The probability that this customer will accept the loan offer
#P(Loan=1 | CC=1, Online=1)
(85)/(811)

## [1] 0.1048089

0.1048

## [1] 0.1048

# C.Create two separate pivot tables for the training data
set.seed(64060)
Melt_Train1 <- melt(Train_Data,id=c("Personal.Loan"),variable = "Online")

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

```r
cast_Train1  <- dcast(Melt_Train1,Personal.Loan~Online)

## Aggregation function missing: defaulting to length

cast_Train1 <-cast_Train1[c(1,13)]
cast_Train1

##   Personal.Loan Online
## 1             0   2748
## 2             1    254

set.seed(64060)
Melt_Train2 <- melt(Train_Data,id=c("CreditCard"),variable = "Online")

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

cast_Train2 <- dcast(Melt_Train2,CreditCard~Online)

## Aggregation function missing: defaulting to length

cast_Train2 <-cast_Train2[c(1,14)]
cast_Train2

##   CreditCard Online
## 1          0   2106
## 2          1    896

Train_Data1 <- Train_Data[c(13,10,14)]
table(Train_Data1[,c(3,2)])

##           Personal.Loan
## CreditCard    0    1
##          0 1937  169
##          1  811   85

table(Train_Data1[,c(1,2)])

##       Personal.Loan
## Online    0    1
##      0 1089  102
##      1 1659  152

table(Train_Data1[,c(2)])

##
##    0    1
## 2748  254

# D.Compute the following quantities
```

```r
# i. #P(CC = 1 | Loan = 1)
(85)/(85+169)

## [1] 0.3346457

0.334

## [1] 0.334

# ii. #P(Online = 1 | Loan = 1)
(152)/(152+102)

## [1] 0.5984252

0.598

## [1] 0.598

# iii. #P(Loan = 1)
(254)/(2748+254)

## [1] 0.08461026

0.084

## [1] 0.084

# iv. #P(CC = 1 | Loan = 0)
(811)/(811+1937)

## [1] 0.2951237

0.291

## [1] 0.291

# v. #P(Online = 1 | Loan = 0)
(1659)/(1659+1089)

## [1] 0.6037118

0.603

## [1] 0.603

# vi. #P(Loan = 0)
(2748)/(2748+254)

## [1] 0.9153897

0.915

## [1] 0.915
```

```r
# E.Use the quantities computed above to compute the naive Bayes probability
((0.334*0.598*0.084)/((0.334*0.598*0.084)+(0.291*0.603*0.915)))

## [1] 0.09460885

0.09460

## [1] 0.0946

# F.Compare this value with the one obtained from the pivot table in (B).
Which is a more accurate estimate.
##  0.09460 are very similar to the 0.1048 the difference between the exact
method and the naive-Bayes method is the exact method would need the the
exact same independent variable classifications to predict, where the naive
bayes method does not.

# G .Examine the model output on training data

library(e1071)

set.seed(64060)
naivebayes <- naiveBayes(Personal.Loan~.,data=Train_Data1)
naivebayes

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.91538974 0.08461026
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.3962882 0.6037118
##   1 0.4015748 0.5984252
##
##    CreditCard
## Y           0         1
##   0 0.7048763 0.2951237
##   1 0.6653543 0.3346457

(((0.334)*(0.598)*(0.084))/((0.334*0.598*0.084)+(0.295*0.603*0.915)))

## [1] 0.0934459

(0.3159)*(0.5972)*(0.097)/((0.3159)*(0.5972)*(0.097) +
(0.2971)*(0.6006)*(0.902))
```

```
## [1] 0.1020892

# Values from the naive Bayes model probability 0.0934 is very similar to
value of E that is 0.094.
```