Assignment3 Naive Bayes Classification

Pranay Kumar Kodeboyina

2023-03-05

# importing the libraries and including caret package  
library("caret")

## Loading required package: ggplot2

## Loading required package: lattice

# Loading required library: ISLR  
library('ISLR')  
  
# Loading required library: dplyr  
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

# Loading required library: class  
library('class')  
  
# Loading required library: ggplot2  
library(ggplot2)  
  
# Loading required library: lattice  
library(lattice)  
  
# Loading required library: knitr  
library(knitr)  
  
# Loading required library: rmarkdown  
library(rmarkdown)  
  
# Loading required library: e1071  
library(e1071)  
  
  
#Extracting the current working directory  
getwd()

## [1] "/Users/kodeboyina/Documents/Kent State/Sem1/Fundamentals of ML/Assignment3"

#setting the working directory to the Assignment Folder  
setwd("/Users/kodeboyina/Documents/Kent State/Sem1/Fundamentals of ML/Assignment3")  
  
#Loading Universal csv data Import the data set into R  
UniBank.df <- read.csv("data/UniversalBank.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)

#Converting the "Education", "Personal Loan", "Credit Card" and "Online" variable to a factor value  
UniBank.df$Education = as.factor(UniBank.df$Education)  
UniBank.df$Personal.Loan <- factor(UniBank.df$Personal.Loan)  
UniBank.df$CreditCard <- as.factor(UniBank.df$CreditCard)  
UniBank.df$Online <- as.factor(UniBank.df$Online)  
  
  
#Drop ID and Zip Code columns(classification with all predictors except ID and ZIP code)   
UniBank.df$ID <- NULL  
UniBank.df$ZIP.Code <- NULL  
  
#Observing the first 10 observations of the data set post removing ID and ZIP code  
head(UniBank.df, n=10L)

## Age Experience Income Family CCAvg Education Mortgage Personal.Loan  
## 1 25 1 49 4 1.6 1 0 0  
## 2 45 19 34 3 1.5 1 0 0  
## 3 39 15 11 1 1.0 1 0 0  
## 4 35 9 100 1 2.7 2 0 0  
## 5 35 8 45 4 1.0 2 0 0  
## 6 37 13 29 4 0.4 2 155 0  
## 7 53 27 72 2 1.5 2 0 0  
## 8 50 24 22 1 0.3 3 0 0  
## 9 35 10 81 3 0.6 2 104 0  
## 10 34 9 180 1 8.9 3 0 1  
## Securities.Account CD.Account Online CreditCard  
## 1 1 0 0 0  
## 2 1 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 1  
## 6 0 0 1 0  
## 7 0 0 1 0  
## 8 0 0 0 1  
## 9 0 0 1 0  
## 10 0 0 0 0

#Priniting the Structure of the data post removing the ID and Zip Code  
str(UniBank.df)

## 'data.frame': 5000 obs. of 12 variables:  
## $ Age : int 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : int 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : int 49 34 11 100 45 29 72 22 81 180 ...  
## $ Family : int 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : int 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ Securities.Account: int 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD.Account : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...  
## $ CreditCard : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...

#Summary of data for the observations  
summary(UniBank.df)

## Age Experience Income Family   
## Min. :23.00 Min. :-3.0 Min. : 8.00 Min. :1.000   
## 1st Qu.:35.00 1st Qu.:10.0 1st Qu.: 39.00 1st Qu.:1.000   
## Median :45.00 Median :20.0 Median : 64.00 Median :2.000   
## Mean :45.34 Mean :20.1 Mean : 73.77 Mean :2.396   
## 3rd Qu.:55.00 3rd Qu.:30.0 3rd Qu.: 98.00 3rd Qu.:3.000   
## Max. :67.00 Max. :43.0 Max. :224.00 Max. :4.000   
## CCAvg Education Mortgage Personal.Loan Securities.Account  
## Min. : 0.000 1:2096 Min. : 0.0 0:4520 Min. :0.0000   
## 1st Qu.: 0.700 2:1403 1st Qu.: 0.0 1: 480 1st Qu.:0.0000   
## Median : 1.500 3:1501 Median : 0.0 Median :0.0000   
## Mean : 1.938 Mean : 56.5 Mean :0.1044   
## 3rd Qu.: 2.500 3rd Qu.:101.0 3rd Qu.:0.0000   
## Max. :10.000 Max. :635.0 Max. :1.0000   
## CD.Account Online CreditCard  
## Min. :0.0000 0:2016 0:3530   
## 1st Qu.:0.0000 1:2984 1:1470   
## Median :0.0000   
## Mean :0.0604   
## 3rd Qu.:0.0000   
## Max. :1.0000

#Dropping the Original education data post creation of the dummy columns  
UniBank.df$Education <- NULL  
  
#Displaying the names of the data columns post columns  
names(UniBank.df)

## [1] "Age" "Experience" "Income"   
## [4] "Family" "CCAvg" "Mortgage"   
## [7] "Personal.Loan" "Securities.Account" "CD.Account"   
## [10] "Online" "CreditCard"

select.var <- c(8,11,12)  
  
#Randomization of the data and setting same random sequence  
set.seed(123)  
  
## Seperating 60% of data as Training set and remaining 40% as validation set  
train.index <-sample(row.names(UniBank.df), 0.6\*dim(UniBank.df)[1])  
valid.index <-setdiff(row.names(UniBank.df), train.index)   
  
#Assigning the data interms of indexes to tain and Validation sets  
train\_data <- UniBank.df[train.index,]  
val\_data <- UniBank.df[valid.index,]

#Question 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 the resulting pivot table CC and LOAN are both rows, and online is a column.  
attach(train\_data)  
##ftable is defined as "function table".   
ftable(CreditCard,Personal.Loan,Online)

## Online 0 1  
## CreditCard Personal.Loan   
## 0 0 785 1145  
## 1 65 122  
## 1 0 317 475  
## 1 34 57

detach(train\_data)

#Question 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)].  
  
prop.table(ftable(train\_data$CreditCard,train\_data$Online,train\_data$Personal.Loan),margin=1)

## 0 1  
##   
## 0 0 0.92352941 0.07647059  
## 1 0.90370955 0.09629045  
## 1 0 0.90313390 0.09686610  
## 1 0.89285714 0.10714286

#Manual caluculation  
prob <- 57/532  
  
  
cat("From the above observation the probability to accept loan with credit card and Online services is ", prob, "and the probability percentage in 10.7%")

## From the above observation the probability to accept loan with credit card and Online services is 0.1071429 and the probability percentage in 10.7%

#Question 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.  
  
attach(train\_data)  
#Personal Loan (rows) as a function of CC (columns)  
ftable(Personal.Loan,CreditCard)

## CreditCard 0 1  
## Personal.Loan   
## 0 1930 792  
## 1 187 91

#Personal Loan (rows) as a function of Online (columns)  
ftable(Personal.Loan,Online)

## Online 0 1  
## Personal.Loan   
## 0 1102 1620  
## 1 99 179

detach(train\_data)

#Question D. Compute the following quantities [P(A | B) means “the probability ofA given B”]:  
  
#Only passing the training data for the calculations  
attach(train\_data)  
  
# Probability table for Personal Loan vs Credit card  
prop.table(ftable(Personal.Loan,CreditCard),margin=1)

## CreditCard 0 1  
## Personal.Loan   
## 0 0.7090375 0.2909625  
## 1 0.6726619 0.3273381

# Probability table for Personal Loan vs Online  
prop.table(ftable(Personal.Loan,Online),margin=1)

## Online 0 1  
## Personal.Loan   
## 0 0.4048494 0.5951506  
## 1 0.3561151 0.6438849

#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)  
Prob\_i <- 91\*100/(91+187)  
Prob\_i

## [1] 32.73381

#ii. P(Online=1|Loan=1)  
Prob\_ii <- 179\*100/(179+99)  
Prob\_ii

## [1] 64.38849

#iii. P(Loan = 1) (the proportion of loan acceptors)  
Prob\_iii <- 278\*100/(1930+792+187+91)  
Prob\_iii

## [1] 9.266667

#iv P(CC=1|Loan=0)  
Prob\_iv <- 792\*100/(1930 + 792)  
Prob\_iv

## [1] 29.09625

#v P(Online=1|Loan=0)  
Prob\_v <- 1620\*100/(1102+1620)  
Prob\_v

## [1] 59.51506

#vi. P(Loan = 0)  
Prob\_vi <- 2772\*100/(1930+792+187+91)  
Prob\_vi

## [1] 92.4

detach(train\_data)

#Question E. Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1).  
  
naive\_Bayes\_prob <- (Prob\_i\*Prob\_ii\*Prob\_iii)/((Prob\_i\*Prob\_ii\*Prob\_iii)+(Prob\_iv\*Prob\_v\*Prob\_vi))  
naive\_Bayes\_prob

## [1] 0.1087863

#Question F. Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate?  
  
#Upon comparing the estimates obtained from above B and E, Naive Bayes probability being slightly higher than the matrix-based probability and it doen not compute to any significant difference. The pivot table calculatios are more accurate as it donot consider the probabilities being independent, E uses probability for every count, whereas B uses a direct calculation based on a count.

#Question 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).  
  
  
#Displaying Training dataset  
  
Train\_naive <- naiveBayes(Personal.Loan ~ Online + CreditCard, data = train\_data)  
Train\_naive

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.90733333 0.09266667   
##   
## Conditional probabilities:  
## Online  
## Y 0 1  
## 0 0.4048494 0.5951506  
## 1 0.3561151 0.6438849  
##   
## CreditCard  
## Y 0 1  
## 0 0.7090375 0.2909625  
## 1 0.6726619 0.3273381

#While using the two tables made in step C makes it simple to understand how you're computing P(LOAN=1|CC=1,Online=1) using the Naive Bayes model, you can also quickly compute P(LOAN=1|CC=1,Online=1) using the pivot table made in step B.  
  
#While it is less than that calculated manually in step E, the probability predicted by the Naive Bayes model is the same as that projected by the prior techniques. This probability is closer to the one discovered in step B. This could be the case since step E's calculations are done manually, which leaves space for mistake when rounding fractions and results in approximations.  
  
## confusion matrix for train\_data  
##Training  
prediction\_class <- predict(Train\_naive, newdata = train\_data)  
confusionMatrix(prediction\_class, train\_data$Personal.Loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2722 278  
## 1 0 0  
##   
## Accuracy : 0.9073   
## 95% CI : (0.8964, 0.9175)  
## No Information Rate : 0.9073   
## P-Value [Acc > NIR] : 0.516   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.0000   
## Specificity : 0.0000   
## Pos Pred Value : 0.9073   
## Neg Pred Value : NaN   
## Prevalence : 0.9073   
## Detection Rate : 0.9073   
## Detection Prevalence : 1.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : 0   
##

###This model's low specificity offset its high sensitivity. In the absence of all genuine values from the reference, the model predicted that all values would be 0. Because of the enormous number of 0, even if the model missed all values of 1, it still yields 90.73% accuracy.

```