BA64060\_Assignment3

Brian Yim

2025-10-09

####The file UniversalBank.csv contains data on 5000 customers of Universal Bank. The data include customer demographic information (age, income, etc.), the customer’s relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign. In this exercise, we focus on two predictors: Online (whether or not the customer is an active user of online banking services) and Credit Card (abbreviated CC below) (does the customer hold a credit card issued by the bank), and the outcome Personal Loan (abbreviated Loan below).

library(e1071)

## Warning: package 'e1071' was built under R version 4.4.3

library(caret)

## Warning: package 'caret' was built under R version 4.4.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.4.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.4.2

#reading the data

library(readr)  
UniversalBank <- read.csv("C:/Users/hyim/OneDrive - Kent State University/Desktop/64060 Assignment2/UniversalBank (2).csv")

mydata<-UniversalBank  
data.frame(colnames(mydata))

## colnames.mydata.  
## 1 ID  
## 2 Age  
## 3 Experience  
## 4 Income  
## 5 ZIP.Code  
## 6 Family  
## 7 CCAvg  
## 8 Education  
## 9 Mortgage  
## 10 Personal.Loan  
## 11 Securities.Account  
## 12 CD.Account  
## 13 Online  
## 14 CreditCard

mydata<-UniversalBank  
data.frame(colnames(mydata))

## colnames.mydata.  
## 1 ID  
## 2 Age  
## 3 Experience  
## 4 Income  
## 5 ZIP.Code  
## 6 Family  
## 7 CCAvg  
## 8 Education  
## 9 Mortgage  
## 10 Personal.Loan  
## 11 Securities.Account  
## 12 CD.Account  
## 13 Online  
## 14 CreditCard

#I will select the variables, personal loan, online, credit card which are 10, 13, and 14

mydata<-mydata[c(10,13,14)]  
table(is.na(mydata))

##   
## FALSE   
## 15000

View(mydata)  
mydata$PersonalLoan<-mydata$`Personal.Loan`   
class(mydata$PersonalLoan)

## [1] "integer"

mydata$PersonalLoan <- as.factor(mydata$PersonalLoan)

class(mydata$PersonalLoan)

## [1] "factor"

mydata$Online <- as.factor(mydata$Online)

class(mydata$Online)

## [1] "factor"

mydata$CreditCard <- as.factor(mydata$CreditCard)

class(mydata$CreditCard)

## [1] "factor"

str(mydata) # check if the converting the variables was successful.

## 'data.frame': 5000 obs. of 4 variables:  
## $ Personal.Loan: int 0 0 0 0 0 0 0 0 0 1 ...  
## $ 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 ...  
## $ PersonalLoan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...

table(mydata$PersonalLoan) # Check the frequency of PersonalLoan and see if the sum is 5000

##   
## 0 1   
## 4520 480

table(mydata$Online) # Check the frequency of Online and see if the sum is 5000

##   
## 0 1   
## 2016 2984

table(mydata$CreditCard) # Check the frequency of CreditCard and see if the sum is 5000

##   
## 0 1   
## 3530 1470

#dividing the data into 60% and 40% for training and validation

set.seed(1)  
Train\_Index=createDataPartition(mydata$PersonalLoan,p=0.6,list=FALSE)  
Train=mydata[Train\_Index,]  
Validation=mydata[-Train\_Index,]  
summary(Train)

## Personal.Loan Online CreditCard PersonalLoan  
## Min. :0.000 0:1199 0:2103 0:2712   
## 1st Qu.:0.000 1:1801 1: 897 1: 288   
## Median :0.000   
## Mean :0.096   
## 3rd Qu.:0.000   
## Max. :1.000

summary(Validation)

## Personal.Loan Online CreditCard PersonalLoan  
## Min. :0.000 0: 817 0:1427 0:1808   
## 1st Qu.:0.000 1:1183 1: 573 1: 192   
## Median :0.000   
## Mean :0.096   
## 3rd Qu.:0.000   
## Max. :1.000

#Now the data is ready to be analyzed

####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

attach(Train)  
ftable(CreditCard,PersonalLoan,Online)

## Online 0 1  
## CreditCard PersonalLoan   
## 0 0 780 1126  
## 1 77 120  
## 1 0 303 503  
## 1 39 52

#Answer for A: As directed I created the pivot table. Online is the column variable and first row is the creditcard and second row is the personalloan and the counts are for the training set

####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)]

ftable(CreditCard, Online, PersonalLoan)

## PersonalLoan 0 1  
## CreditCard Online   
## 0 0 780 77  
## 1 1126 120  
## 1 0 303 39  
## 1 503 52

prop.table(ftable(CreditCard, Online, PersonalLoan), margin=1) # A pivot table was created using the training data, with Personal Loan as the column variable, CreditCard as the primary row variable, and Online as the secondary row variable. This table displays the conditional probability of loan acceptance based on each combination of CreditCard and Online statuses, ensuring that the sum of probabilities across each row equals 1.

## PersonalLoan 0 1  
## CreditCard Online   
## 0 0 0.91015169 0.08984831  
## 1 0.90369181 0.09630819  
## 1 0 0.88596491 0.11403509  
## 1 0.90630631 0.09369369

####Answer for B: The probability is 0.09369. The number of customers (cc ∩ online = 503 + 52) = 555. The number of customers (Loan = 1) within the (cc∩online)=52. 52/555

#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

ftable(PersonalLoan,Online)

## Online 0 1  
## PersonalLoan   
## 0 1083 1629  
## 1 116 172

prop.table(ftable(PersonalLoan,Online), margin=1)

## Online 0 1  
## PersonalLoan   
## 0 0.3993363 0.6006637  
## 1 0.4027778 0.5972222

ftable(PersonalLoan, CreditCard) #The values inside the table should convey the count

## CreditCard 0 1  
## PersonalLoan   
## 0 1906 806  
## 1 197 91

prop.table(ftable(PersonalLoan, CreditCard), margin=1) #The pivot table calculates the probability of having CC conditional on Loan acceptance (the sum probability of row (the condition, loan acceptance) is 1).

## CreditCard 0 1  
## PersonalLoan   
## 0 0.7028024 0.2971976  
## 1 0.6840278 0.3159722

####Answer to C: 2 tables were created

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

ftable(PersonalLoan, CreditCard)

## CreditCard 0 1  
## PersonalLoan   
## 0 1906 806  
## 1 197 91

prop.table(ftable(PersonalLoan, CreditCard), margin=1)

## CreditCard 0 1  
## PersonalLoan   
## 0 0.7028024 0.2971976  
## 1 0.6840278 0.3159722

####Answer to D(i) is 0.31597

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

ftable(PersonalLoan, Online)

## Online 0 1  
## PersonalLoan   
## 0 1083 1629  
## 1 116 172

prop.table(ftable(PersonalLoan, Online), margin=1)

## Online 0 1  
## PersonalLoan   
## 0 0.3993363 0.6006637  
## 1 0.4027778 0.5972222

####Answer to D(ii) is 0.59722

#D(iii). P(Loan = 1) (the proportion of loan acceptors))

ftable(PersonalLoan)

## PersonalLoan 0 1  
##   
## 2712 288

prop.table(ftable(PersonalLoan))

## PersonalLoan 0 1  
##   
## 0.904 0.096

####Answer to D(iii) is 0.096

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

ftable(PersonalLoan, CreditCard)

## CreditCard 0 1  
## PersonalLoan   
## 0 1906 806  
## 1 197 91

prop.table(ftable(PersonalLoan, CreditCard), margin=1)

## CreditCard 0 1  
## PersonalLoan   
## 0 0.7028024 0.2971976  
## 1 0.6840278 0.3159722

####Answer to D(iv) is 0.29719

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

ftable(PersonalLoan, Online)

## Online 0 1  
## PersonalLoan   
## 0 1083 1629  
## 1 116 172

prop.table(ftable(PersonalLoan, Online), margin=1)

## Online 0 1  
## PersonalLoan   
## 0 0.3993363 0.6006637  
## 1 0.4027778 0.5972222

####Answer to D(v) is 0.6006

#D(vi). P(Loan = 0)

prop.table(ftable(PersonalLoan))

## PersonalLoan 0 1  
##   
## 0.904 0.096

####Answer to D(vi) is 0.904

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

#{(91/288) \* (172/288) \* (288/3000)} / [{(91/288) \* (172/288) \* (288/3000)} + {(806/2712) \* (1629/2712) \* (2712/3000)}] = (15652/864000) / {(15652/864000) + (1312974/8136000) = 0.01811 / (0.01811 + 0.161378) = 0.01811 / 0.179488 = 0.100926687 ≈ 0.1009267 #Answer for E: Nave Bayes P(Loan = 1 | CC = 1, Online = 1) = 0.1009267

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

#Answer for F: The value obtained from the Exact Bayes calculation (0.094) is very close to that from the Naïve Bayes calculation (0.1009267), with a small difference of 0.0069. The Exact Bayes method is expected to yield a more accurate result because it identifies all records that share the same combination of predictor values. In contrast, the Naïve Bayes approach estimates the conditional probabilities of each predictor individually (P(Xi|C1)) and multiplies them together under the assumption that the predictors are independent. Although this independence assumption is rarely true in real-world data—since predictors often exhibit some correlation—the Naïve Bayes method generally performs well because it tends to preserve the correct rank ordering of predicted probabilities.

#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)

Loan\_nb<-naiveBayes(PersonalLoan ~ Online+CreditCard,data=Train)  
Loan\_nb

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.904 0.096   
##   
## Conditional probabilities:  
## Online  
## Y 0 1  
## 0 0.3993363 0.6006637  
## 1 0.4027778 0.5972222  
##   
## CreditCard  
## Y 0 1  
## 0 0.7028024 0.2971976  
## 1 0.6840278 0.3159722

pred.class <- predict(Loan\_nb, newdata = Train)  
confusionMatrix(pred.class, Train$PersonalLoan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2712 288  
## 1 0 0  
##   
## Accuracy : 0.904   
## 95% CI : (0.8929, 0.9143)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : 0.5157   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.000   
## Specificity : 0.000   
## Pos Pred Value : 0.904   
## Neg Pred Value : NaN   
## Prevalence : 0.904   
## Detection Rate : 0.904   
## Detection Prevalence : 1.000   
## Balanced Accuracy : 0.500   
##   
## 'Positive' Class : 0   
##

pred.prob <- predict(Loan\_nb, newdata=Train, type="raw")  
df <- data.frame(actual = Train$PersonalLoan, predicted = pred.class, pred.prob)  
df[Train$PersonalLoan==1&Train$Online==1&Train$CreditCard==1,]

## actual predicted X0 X1  
## 18 1 0 0.8990733 0.1009267  
## 30 1 0 0.8990733 0.1009267  
## 47 1 0 0.8990733 0.1009267  
## 82 1 0 0.8990733 0.1009267  
## 271 1 0 0.8990733 0.1009267  
## 351 1 0 0.8990733 0.1009267  
## 449 1 0 0.8990733 0.1009267  
## 486 1 0 0.8990733 0.1009267  
## 492 1 0 0.8990733 0.1009267  
## 561 1 0 0.8990733 0.1009267  
## 633 1 0 0.8990733 0.1009267  
## 645 1 0 0.8990733 0.1009267  
## 782 1 0 0.8990733 0.1009267  
## 796 1 0 0.8990733 0.1009267  
## 866 1 0 0.8990733 0.1009267  
## 870 1 0 0.8990733 0.1009267  
## 875 1 0 0.8990733 0.1009267  
## 920 1 0 0.8990733 0.1009267  
## 933 1 0 0.8990733 0.1009267  
## 966 1 0 0.8990733 0.1009267  
## 1013 1 0 0.8990733 0.1009267  
## 1173 1 0 0.8990733 0.1009267  
## 1223 1 0 0.8990733 0.1009267  
## 1248 1 0 0.8990733 0.1009267  
## 1325 1 0 0.8990733 0.1009267  
## 1433 1 0 0.8990733 0.1009267  
## 1436 1 0 0.8990733 0.1009267  
## 1489 1 0 0.8990733 0.1009267  
## 1525 1 0 0.8990733 0.1009267  
## 1574 1 0 0.8990733 0.1009267  
## 1583 1 0 0.8990733 0.1009267  
## 1649 1 0 0.8990733 0.1009267  
## 1726 1 0 0.8990733 0.1009267  
## 1803 1 0 0.8990733 0.1009267  
## 1823 1 0 0.8990733 0.1009267  
## 1830 1 0 0.8990733 0.1009267  
## 1942 1 0 0.8990733 0.1009267  
## 1949 1 0 0.8990733 0.1009267  
## 2007 1 0 0.8990733 0.1009267  
## 2369 1 0 0.8990733 0.1009267  
## 2400 1 0 0.8990733 0.1009267  
## 2405 1 0 0.8990733 0.1009267  
## 2501 1 0 0.8990733 0.1009267  
## 2607 1 0 0.8990733 0.1009267  
## 2649 1 0 0.8990733 0.1009267  
## 2779 1 0 0.8990733 0.1009267  
## 2823 1 0 0.8990733 0.1009267  
## 2891 1 0 0.8990733 0.1009267  
## 2930 1 0 0.8990733 0.1009267  
## 2959 1 0 0.8990733 0.1009267  
## 2978 1 0 0.8990733 0.1009267  
## 2989 1 0 0.8990733 0.1009267

#Answer (Model Evaluation): The Naïve Bayes Classifier achieved an accuracy of 0.904, calculated as the proportion of correctly classified samples to the total number of samples. Since the model predicted all cases as “Loan reject (0),” the accuracy is computed as 2712/3000. The sensitivity (true positive rate) is 1, calculated as (true positives)/(true positives + false negatives) = 2712/(2712 + 0) = 1. The specificity (true negative rate) is 0, calculated as (true negatives)/(true negatives + false positives) = 0/(0 + 288) = 0 #Answer (Predicted P(Loan = 1 | CC = 1, Online = 1)): The Naïve Bayes model predicted a probability of 0.1009267, which matches the value obtained previously in (E)