Assignment\_3

Melissa Dennis

2025-10-10

## Load libraries  
library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Loading required package: lattice

library(reshape2)

## Warning: package 'reshape2' was built under R version 4.3.3

library(reshape)

##   
## Attaching package: 'reshape'

## The following objects are masked from 'package:reshape2':  
##   
## colsplit, melt, recast

library(e1071)

## Warning: package 'e1071' was built under R version 4.3.1

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:ggplot2':  
##   
## element

##Load dataset  
df <- read.csv("C:/Users/m\_den/Downloads/UniversalBank.csv")   
  
##Recall column names  
colnames(df)

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

##Rename change to requested columns to factors for easier evaluation  
  
df$CreditCard <- as.factor(df$CreditCard)  
df$Online <- as.factor(df$Online)  
df$Personal.Loan <- as.factor(df$Personal.Loan)

## Data partition dataset: training 60%, validation 40%  
set.seed(123)  
  
train.index <- createDataPartition(df$Personal.Loan,p = 0.6, list = FALSE) ##creates a partition of 60%  
train.df <- df[train.index, ] ##assigns training tag to the 60%  
valid.df <- df[-train.index, ] ##assigns validation tag to the 40%

##Print the stated question  
  
cat("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().")

## 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().

##Use melt() to stack columns CC and Loan   
  
mlt <- melt(train.df, id=c("CreditCard", "Personal.Loan"), measure = c("Online"))  
head(mlt,5)

## CreditCard Personal.Loan variable value  
## 1 0 0 Online 0  
## 2 0 0 Online 0  
## 3 0 0 Online 0  
## 4 1 0 Online 0  
## 5 0 0 Online 1

##Convert Value to a number  
  
mlt$value <- as.numeric(mlt$value)  
  
##Use cast to the results from melt() and reshape it into a pivot table  
  
cast(mlt, CreditCard ~ Personal.Loan, subset = variable == "Online", margins = c("grand\_row", "grand\_col"), fun.aggregate = sum)

## CreditCard 0 1 (all)  
## 1 0 3079 329 3408  
## 2 1 1244 135 1379  
## 3 (all) 4323 464 4787

##Print the stated question  
  
cat("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)].")

## 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)].

##Run Naive Bayes on Personal Loan only this will exclude all other columns as we are focusing on customer who will accept the offer  
  
df.nb <- naiveBayes(Personal.Loan ~ ., data = mlt)  
  
df.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:  
## CreditCard  
## Y 0 1  
## 0 0.7134956 0.2865044  
## 1 0.7083333 0.2916667  
##   
## variable  
## Y Online  
## 0 1  
## 1 1  
##   
## value  
## Y [,1] [,2]  
## 0 1.594027 0.4911700  
## 1 1.611111 0.4883466

##Create an overall probably of proportion of customers who accepted the loan  
  
prob.table <- prop.table(table(mlt$Personal.Loan))  
  
prob.table

##   
## 0 1   
## 0.904 0.096

##Print response to question B  
  
cat("\n","The probably based on the data evaluation that a customer who uses has a credit card and uses online banking has a 9.6% rate of accepting a personal loan offer.")

##   
## The probably based on the data evaluation that a customer who uses has a credit card and uses online banking has a 9.6% rate of accepting a personal loan offer.

##Print question  
  
cat("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.")

## 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.

# Pivot table for Personal.Loan and Online  
  
pivot.loan.online <- dcast(train.df, Personal.Loan ~ Online, value.var = "CreditCard", fun.aggregate = length)  
pivot.loan.online

## Personal.Loan 0 1  
## 1 0 1101 1611  
## 2 1 112 176

# Pivot table for Personal.Loan and CreditCard  
  
pivot.loan.cc <- dcast(train.df, Personal.Loan ~ CreditCard, value.var = "Online", fun.aggregate = length)  
pivot.loan.cc

## Personal.Loan 0 1  
## 1 0 1935 777  
## 2 1 204 84

##Print question D  
  
cat("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)")

## 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)

##Calculate the count of customers who accepted loans with a credit card  
sum.cc1.loan1 <- sum(train.df$CreditCard == 1 & train.df$Personal.Loan == 1)  
  
##Calculate the total count of customers who accepted loans  
sum.loan1 <- sum(train.df$Personal.Loan == 1)  
  
##Calculate the proportion  
prob.cc1.loan1 <- sum.cc1.loan1 / sum.loan1  
  
cat("i.",prob.cc1.loan1)

## i. 0.2916667

##Calculate the count of customers who accepted loans who use online banking  
sum.online1.loan1 <- sum(train.df$Online == 1 & train.df$Personal.Loan == 1)  
  
##Calculate the total count of customers who accepted loans  
sum.loan1 <- sum(train.df$Personal.Loan == 1)  
  
##Calculate the proportion  
prob.online1.loan1 <- sum.online1.loan1 / sum.loan1  
  
cat("ii.",prob.online1.loan1)

## ii. 0.6111111

##Calculate the total count of customers who accepted loans  
sum.loan1 <- sum(train.df$Personal.Loan == 1)  
  
##Calculate the total number of observations in the training data  
total.obs <- nrow(train.df)  
  
##Calculate the proportion  
prob.loan1 <- sum.loan1 / total.obs  
  
cat("iii.",prob.loan1)

## iii. 0.096

##Calculate the count of customers who did not accepted loans with a credit card  
sum.cc1.loan0 <- sum(train.df$CreditCard == 1 & train.df$Personal.Loan == 0)  
  
##Calculate the total count of customers who did not accepted loans  
sum.loan0 <- sum(train.df$Personal.Loan == 0)  
  
##Calculate the proportion  
prob.cc1.loan0 <- sum.cc1.loan0 / sum.loan0  
  
cat("iv.",prob.cc1.loan0)

## iv. 0.2865044

##Calculate the count of customers who did not accepted loans who use online banking  
sum.online1.loan0 <- sum(train.df$Online == 1 & train.df$Personal.Loan == 0)  
  
##Calculate the total count of customers who did not accepted loans  
sum.loan0 <- sum(train.df$Personal.Loan == 0)  
  
##Calculate the proportion  
prob.online1.loan0 <- sum.online1.loan0 / sum.loan0  
  
cat("v.",prob.online1.loan0)

## v. 0.5940265

##Calculate the count of customers who did not accepted loans  
sum.loan0 <- sum(train.df$Personal.Loan == 0)  
  
##Calculate the total number of observations in the training data  
total.obs <- nrow(train.df)  
  
##Calculate the proportion  
prob.loan0 <- sum.loan0 / total.obs  
  
cat("vi.",prob.loan0)

## vi. 0.904

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

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

##Calculate the numerator  
numerator <- prob.cc1.loan1 \* prob.online1.loan1 \* prob.loan1  
  
##Calculate the denominator  
denominator <- (prob.cc1.loan1 \* prob.online1.loan1 \* prob.loan1) + (prob.cc1.loan0 \* prob.online1.loan0 \* prob.loan0)  
  
##Calculate the Naive Bayes probability  
prob.nb <- numerator/denominator  
  
cat("Naive Bayes Probability P(Loan = 1 | CC = 1, Online = 1): ", prob.nb)

## Naive Bayes Probability P(Loan = 1 | CC = 1, Online = 1): 0.1000861

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

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

##Print the response to question F  
  
cat("The two probability values are almost the same with the empirical probability as 10% and the Naive Bayes value as 9.6%. The slight difference most likely due to rounding. The empirical probability is directly calculated from observed data counts. This is representing the relationship without relying on any modeling assumptions. The Naive Bayes probability is an estimate based on the assumption of conditional independence among the predictor variables. In this case, the variables appear to be nearly conditionally independent, so the Naive Bayes is very close to the empirical value. This shows that the model can produce similar results under these conditions. ")

## The two probability values are almost the same with the empirical probability as 10% and the Naive Bayes value as 9.6%. The slight difference most likely due to rounding. The empirical probability is directly calculated from observed data counts. This is representing the relationship without relying on any modeling assumptions. The Naive Bayes probability is an estimate based on the assumption of conditional independence among the predictor variables. In this case, the variables appear to be nearly conditionally independent, so the Naive Bayes is very close to the empirical value. This shows that the model can produce similar results under these conditions.

##Print question G  
  
cat("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).")

## 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).

##Calculate the naive bayes model  
df.nb.e <- naiveBayes(Personal.Loan ~ CreditCard + Online, data = train.df)  
  
df.nb.e

##   
## 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:  
## CreditCard  
## Y 0 1  
## 0 0.7134956 0.2865044  
## 1 0.7083333 0.2916667  
##   
## Online  
## Y 0 1  
## 0 0.4059735 0.5940265  
## 1 0.3888889 0.6111111

##Define a new data point for CC = 1 and Online = 1  
new.data.point <- data.frame(CreditCard = 1, Online = 1)  
  
new.data.point$CreditCard <- as.factor(new.data.point$CreditCard)  
new.data.point$Online <- as.factor(new.data.point$Online)  
  
##Predict the probability for the new data point  
prediction.prob <- predict(df.nb.e, newdata = new.data.point, type = "raw")  
  
##Extract the probability for Loan = 1  
prob.loan1.from.model <- prediction.prob[,"1"]  
  
##Print response to question G  
cat("Calculated naive bayes value:",prob.loan1.from.model, "\n\nNaive Bayes Probability P(Loan = 1 | CC = 1, Online = 1):", prob.nb, "\n\nThe Entries Needed for Naive Bayes Calculation to compute the naive Bayes probability P(Loan=1|CC=1,Online=1). You need the following six probabilities:   
• P(Loan=1)  
• P(Loan=0)  
• P(CC=1|Loan=1)  
• P(Online=1|Loan=1)  
• P(CC=1|Loan=0)  
• P(Online=1|Loan=0,  
\nThe value are exactly the same. This is because the manual calculation and the model's prediction are based on the same formula and probabilities.")

## Calculated naive bayes value: 0.1000861   
##   
## Naive Bayes Probability P(Loan = 1 | CC = 1, Online = 1): 0.1000861   
##   
## The Entries Needed for Naive Bayes Calculation to compute the naive Bayes probability P(Loan=1|CC=1,Online=1). You need the following six probabilities:   
## • P(Loan=1)  
## • P(Loan=0)  
## • P(CC=1|Loan=1)  
## • P(Online=1|Loan=1)  
## • P(CC=1|Loan=0)  
## • P(Online=1|Loan=0,  
##   
## The value are exactly the same. This is because the manual calculation and the model's prediction are based on the same formula and probabilities.