Assignment 3

Harsh Patel

2025-10-06

# -------------------------------------------------------  
# Assignment 3: Naive Bayes Classification - Universal Bank  
# Name: Harsh Patel  
# -------------------------------------------------------  
  
# Load libraries  
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(e1071)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

# Reading the data  
df <- read.csv("C:/Users/jbhsp/Downloads/UniversalBank - Copy.csv")  
  
# Checking column names  
names(df)

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

# Keeping only required columns  
df <- df %>%  
 rename(  
 Loan = Personal.Loan,  
 CC = CreditCard,  
 Online = Online  
 ) %>%  
 select(Online, CC, Loan)  
  
# Making sure variables are numeric (0/1)  
df$Online <- as.integer(df$Online)  
df$CC <- as.integer(df$CC)  
df$Loan <- as.integer(df$Loan)  
  
# Splitting data into training (60%) and validation (40%)  
set.seed(123)  
train\_index <- createDataPartition(df$Loan, p = 0.6, list = FALSE)  
train <- df[train\_index, ]  
val <- df[-train\_index, ]  
  
# =======================================================  
# A. Pivot table with Online (columns), CC and Loan (rows)  
# =======================================================  
pivot <- with(train, table(CC, Loan, Online))  
pivot

## , , Online = 0  
##   
## Loan  
## CC 0 1  
## 0 785 65  
## 1 317 34  
##   
## , , Online = 1  
##   
## Loan  
## CC 0 1  
## 0 1145 122  
## 1 475 57

# =======================================================  
# B. Probability P(Loan=1 | CC=1, Online=1)  
# =======================================================  
count\_cc1\_online1\_loan1 <- nrow(filter(train, CC == 1, Online == 1, Loan == 1))  
count\_cc1\_online1\_total <- nrow(filter(train, CC == 1, Online == 1))  
p\_table\_B <- count\_cc1\_online1\_loan1 / count\_cc1\_online1\_total  
cat("\nB) P(Loan=1 | CC=1, Online=1) =", p\_table\_B, "\n")

##   
## B) P(Loan=1 | CC=1, Online=1) = 0.1071429

# =======================================================  
# C. Pivot tables for Loan vs Online and Loan vs CC  
# =======================================================  
pivot\_online <- table(train$Loan, train$Online)  
pivot\_cc <- table(train$Loan, train$CC)  
  
cat("\nC) Pivot: Loan vs Online\n")

##   
## C) Pivot: Loan vs Online

print(pivot\_online)

##   
## 0 1  
## 0 1102 1620  
## 1 99 179

cat("\nC) Pivot: Loan vs CreditCard\n")

##   
## C) Pivot: Loan vs CreditCard

print(pivot\_cc)

##   
## 0 1  
## 0 1930 792  
## 1 187 91

# =======================================================  
# D. Computing probabilities  
# =======================================================  
p\_cc\_given\_loan1 <- sum(train$CC == 1 & train$Loan == 1) / sum(train$Loan == 1)  
p\_online\_given\_loan1 <- sum(train$Online == 1 & train$Loan == 1) / sum(train$Loan == 1)  
p\_loan1 <- mean(train$Loan == 1)  
  
p\_cc\_given\_loan0 <- sum(train$CC == 1 & train$Loan == 0) / sum(train$Loan == 0)  
p\_online\_given\_loan0 <- sum(train$Online == 1 & train$Loan == 0) / sum(train$Loan == 0)  
p\_loan0 <- mean(train$Loan == 0)  
  
cat("\nD) Probabilities:\n")

##   
## D) Probabilities:

cat("P(CC=1 | Loan=1) =", p\_cc\_given\_loan1, "\n")

## P(CC=1 | Loan=1) = 0.3273381

cat("P(Online=1 | Loan=1) =", p\_online\_given\_loan1, "\n")

## P(Online=1 | Loan=1) = 0.6438849

cat("P(Loan=1) =", p\_loan1, "\n")

## P(Loan=1) = 0.09266667

cat("P(CC=1 | Loan=0) =", p\_cc\_given\_loan0, "\n")

## P(CC=1 | Loan=0) = 0.2909625

cat("P(Online=1 | Loan=0) =", p\_online\_given\_loan0, "\n")

## P(Online=1 | Loan=0) = 0.5951506

cat("P(Loan=0) =", p\_loan0, "\n")

## P(Loan=0) = 0.9073333

# =======================================================  
# E. Naive Bayes formula for P(Loan=1 | CC=1, Online=1)  
# =======================================================  
num1 <- p\_cc\_given\_loan1 \* p\_online\_given\_loan1 \* p\_loan1  
num0 <- p\_cc\_given\_loan0 \* p\_online\_given\_loan0 \* p\_loan0  
p\_nb <- num1 / (num1 + num0)  
cat("\nE) Naive Bayes P(Loan=1 | CC=1, Online=1) =", p\_nb, "\n")

##   
## E) Naive Bayes P(Loan=1 | CC=1, Online=1) = 0.1105637

# =======================================================  
# F. Comparison between pivot and Naive Bayes  
# =======================================================  
cat("\nF) Comparison:\n")

##   
## F) Comparison:

cat("Pivot-table estimate:", p\_table\_B, "\n")

## Pivot-table estimate: 0.1071429

cat("Naive Bayes estimate:", p\_nb, "\n")

## Naive Bayes estimate: 0.1105637

# =======================================================  
# G. Naive Bayes model using e1071  
# =======================================================  
model <- naiveBayes(as.factor(Loan) ~ CC + Online, data = train)  
predict\_case <- predict(model, newdata = data.frame(CC = 1, Online = 1), type = "raw")  
cat("\nG) Model posterior P(Loan=1 | CC=1, Online=1):", predict\_case[2], "\n")

##   
## G) Model posterior P(Loan=1 | CC=1, Online=1): 0.1156935