Assignment\_3

2025-06-15

# Load necessary libraries  
if (!require("readr")) install.packages("readr", dependencies = TRUE)

## Loading required package: readr

if (!require("e1071")) install.packages("e1071", dependencies = TRUE)

## Loading required package: e1071

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

library(readr)   
library(e1071)  
  
# Load the data from CSV file  
bank\_data <- read.csv("C:/Users/arkha/Downloads/UniversalBank.csv")  
  
# Just keep the columns we need for this assignment  
# Personal Loan = if they accepted the loan  
# Online = if they use online banking  
# CreditCard = if they have a credit card  
bank\_data <- bank\_data[, c("Personal.Loan", "Online", "CreditCard")]  
  
# Make the column names easier to type  
colnames(bank\_data) <- c("loan", "online", "card")  
  
# We'll split the data into training and validation sets  
set.seed(100) # this makes sure the random split is the same each time  
sample\_rows <- sample(1:nrow(bank\_data), 0.6 \* nrow(bank\_data))  
train\_data <- bank\_data[sample\_rows, ]  
valid\_data <- bank\_data[-sample\_rows, ]  
  
# --------------------  
# A. Create a pivot table with Credit Card (row), Loan (2nd row), Online (column)  
table\_result <- table(train\_data$card, train\_data$loan, train\_data$online)  
print("Pivot Table (Card x Loan x Online):")

## [1] "Pivot Table (Card x Loan x Online):"

print(table\_result)

## , , = 0  
##   
##   
## 0 1  
## 0 784 76  
## 1 326 34  
##   
## , , = 1  
##   
##   
## 0 1  
## 0 1125 128  
## 1 486 41

# --------------------  
# B. Calculate the probability someone accepts the loan  
# if they have a credit card AND use online banking  
total\_people <- sum(table\_result["1", , "1"])  
accepted\_loan <- table\_result["1", "1", "1"]  
prob\_from\_pivot <- accepted\_loan / total\_people  
print(paste("B. P(Loan=1 | Card=1, Online=1) from pivot table:", round(prob\_from\_pivot, 4)))

## [1] "B. P(Loan=1 | Card=1, Online=1) from pivot table: 0.0778"

# --------------------  
# C. Make two simpler tables to compare  
# Loan vs Online  
table\_online <- table(train\_data$loan, train\_data$online)  
print("Loan vs Online Table:")

## [1] "Loan vs Online Table:"

print(table\_online)

##   
## 0 1  
## 0 1110 1611  
## 1 110 169

# Loan vs Credit Card  
table\_card <- table(train\_data$loan, train\_data$card)  
print("Loan vs Credit Card Table:")

## [1] "Loan vs Credit Card Table:"

print(table\_card)

##   
## 0 1  
## 0 1909 812  
## 1 204 75

# --------------------  
# D. Now calculate all the probabilities we need  
  
# Total number of people in training data  
total\_rows <- nrow(train\_data)  
# How many said yes or no to loan  
loan\_yes <- sum(train\_data$loan == 1)  
loan\_no <- sum(train\_data$loan == 0)  
  
# i. P(Card = 1 | Loan = 1)  
prob\_card\_loan\_yes <- table\_card["1", "1"] / loan\_yes  
  
# ii. P(Online = 1 | Loan = 1)  
prob\_online\_loan\_yes <- table\_online["1", "1"] / loan\_yes  
  
# iii. P(Loan = 1)  
prob\_loan\_yes <- loan\_yes / total\_rows  
  
# iv. P(Card = 1 | Loan = 0)  
prob\_card\_loan\_no <- table\_card["0", "1"] / loan\_no  
  
# v. P(Online = 1 | Loan = 0)  
prob\_online\_loan\_no <- table\_online["0", "1"] / loan\_no  
  
# vi. P(Loan = 0)  
prob\_loan\_no <- loan\_no / total\_rows  
  
# --------------------  
# E. Use Naive Bayes formula manually  
# We use: P(A|B) = [P(B|A)\*P(A)] / [P(B|A)\*P(A) + P(B|~A)\*P(~A)]  
top <- prob\_card\_loan\_yes \* prob\_online\_loan\_yes \* prob\_loan\_yes  
bottom <- top + (prob\_card\_loan\_no \* prob\_online\_loan\_no \* prob\_loan\_no)  
naive\_result <- top / bottom  
print(paste("E. Naive Bayes manually: P(Loan=1 | Card=1, Online=1) =", round(naive\_result, 4)))

## [1] "E. Naive Bayes manually: P(Loan=1 | Card=1, Online=1) = 0.0863"

# --------------------  
# F. Compare both answers  
print(paste("F. From Pivot Table: ", round(prob\_from\_pivot, 4)))

## [1] "F. From Pivot Table: 0.0778"

print(paste("F. From Naive Bayes: ", round(naive\_result, 4)))

## [1] "F. From Naive Bayes: 0.0863"

# --------------------  
# G. Run actual Naive Bayes model to double-check  
# We'll use e1071 package   
  
# Convert to factors for the model to work properly  
train\_data$loan <- as.factor(train\_data$loan)  
train\_data$card <- as.factor(train\_data$card)  
train\_data$online <- as.factor(train\_data$online)  
  
# Train the model using just card and online  
nb\_model <- naiveBayes(loan ~ card + online, data = train\_data)  
  
# Make a test input for someone with card = 1 and online = 1  
test\_person <- data.frame(card = factor(1, levels = c(0,1)),  
 online = factor(1, levels = c(0,1)))  
prediction <- predict(nb\_model, test\_person, type = "raw")  
  
# Print the model's predicted probability  
print(paste("G. Naive Bayes model output: P(Loan=1) =", round(prediction[,"1"], 4)))

## [1] "G. Naive Bayes model output: P(Loan=1) = 0.0863"

# Final note  
cat("\nWe used only two smaller tables for the calculations:\n")

##   
## We used only two smaller tables for the calculations:

cat("- Loan vs Online\n- Loan vs Credit Card\n")

## - Loan vs Online  
## - Loan vs Credit Card