Assignment 3

Naive Bayes for classification

Hollister Victor

Fundamentals of Machine Learning

June 10,2025

Load Libraries

library(readr)  
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(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

Load Data Set/clean data

MyData <- read.csv("/Users/hollyvictor/Downloads/UniversalBank.csv", header = TRUE, stringsAsFactors = FALSE)  
  
# Remove unneeded columns (ID and ZIP Code)  
MyData <- MyData %>% select(-ID, -ZIP.Code)  
  
# Convert CreditCard, Online, and Personal.Loan to factors (for Naive Bayes)  
MyData$CreditCard <- as.factor(MyData$CreditCard)  
MyData$Online <- as.factor(MyData$Online)  
MyData$Personal.Loan <- as.factor(MyData$Personal.Loan)  
  
Partition the data into training (60%) and validation (40%) sets.

#Partition the data into training (60%) and validation (40%) sets.  
set.seed(123)  
Index\_Train <- createDataPartition(MyData$Personal.Loan, p = 0.6, list = FALSE)  
Train <- MyData[Index\_Train, ]  
Test <- MyData[-Index\_Train, ]

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().

# A. Create a pivot table for the training data  
Pivot\_Table <- table(CC = Train$CreditCard,  
 Online = Train$Online,  
 Loan = Train$Personal.Loan)  
print(Pivot\_Table)

## , , Loan = 0  
##   
## Online  
## CC 0 1  
## 0 791 1144  
## 1 310 467  
##   
## , , Loan = 1  
##   
## Online  
## CC 0 1  
## 0 79 125  
## 1 33 51

51 customers had Loan =1 CC=1 Online =1

125Customers had Loan =1 CC=0 Online =1

467 customers had Loan = 0 CC=1 Online =1

1144 customers had loan =0 CC=0 Online =0

Etc…

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)]. Observed probability of a loan being accepted among customers with CC = 1 and Online = 1 is **9.85%**

# B. Compute P(Loan = 1 | CC = 1, Online = 1) — Empirical estimate  
numerator <- Pivot\_Table["1", "1", "1"]  
denominator <- Pivot\_Table["1", "1", "1"] + Pivot\_Table["1", "1", "0"]  
Empirical\_Prob <- numerator / denominator  
cat("Empirical P(Loan = 1 | CC = 1, Online = 1):", round(Empirical\_Prob, 4), "\n")

## Empirical P(Loan = 1 | CC = 1, Online = 1): 0.0985

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  
  
# Loan vs Online  
table\_online <- table(Loan = Train$Personal.Loan, Online = Train$Online)  
print(table\_online)

## Online  
## Loan 0 1  
## 0 1101 1611  
## 1 112 176

# Loan vs Credit Card  
table\_cc <- table(Loan = Train$Personal.Loan, CC = Train$CreditCard)  
print(table\_cc)

## CC  
## Loan 0 1  
## 0 1935 777  
## 1 204 84

A white background with black text

AI-generated content may be incorrect.

#D. Compute conditional probabilities  
  
# Contingency tables for conditional calculations  
Table\_CC <- table(Loan = Train$Personal.Loan, CC = Train$CreditCard)  
Table\_Online <- table(Loan = Train$Personal.Loan, Online = Train$Online)  
  
# i. P(CC = 1 | Loan = 1)  
P\_CC1\_given\_Loan1 <- Table\_CC["1", "1"] / sum(Table\_CC["1", ])  
  
# ii. P(Online = 1 | Loan = 1)  
P\_Online1\_given\_Loan1 <- Table\_Online["1", "1"] / sum(Table\_Online["1", ])  
  
# iii. P(Loan = 1)  
P\_Loan1 <- mean(Train$Personal.Loan == "1")  
  
# iv. P(CC = 1 | Loan = 0)  
P\_CC1\_given\_Loan0 <- Table\_CC["0", "1"] / sum(Table\_CC["0", ])  
  
# v. P(Online = 1 | Loan = 0)  
P\_Online1\_given\_Loan0 <- Table\_Online["0", "1"] / sum(Table\_Online["0", ])  
  
# vi. P(Loan = 0)  
P\_Loan0 <- 1 - P\_Loan1  
  
# Print the results clearly  
cat("i. P(CC = 1 | Loan = 1):", round(P\_CC1\_given\_Loan1, 4), "\n")

## i. P(CC = 1 | Loan = 1): 0.2917

cat("ii. P(Online = 1 | Loan = 1):", round(P\_Online1\_given\_Loan1, 4), "\n")

## ii. P(Online = 1 | Loan = 1): 0.6111

cat("iii.P(Loan = 1):", round(P\_Loan1, 4), "\n")

## iii.P(Loan = 1): 0.096

cat("iv. P(CC = 1 | Loan = 0):", round(P\_CC1\_given\_Loan0, 4), "\n")

## iv. P(CC = 1 | Loan = 0): 0.2865

cat("v. P(Online = 1 | Loan = 0):", round(P\_Online1\_given\_Loan0, 4), "\n")

## v. P(Online = 1 | Loan = 0): 0.594

cat("vi. P(Loan = 0):", round(P\_Loan0, 4), "\n")

## vi. P(Loan = 0): 0.904

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

#E. Compute Naive Bayes Probability: P(Loan = 1 | CC = 1, Online = 1)  
  
# Naive Bayes numerator and denominator  
# Numerator for Loan = 1  
Numerator <- P\_CC1\_given\_Loan1 \* P\_Online1\_given\_Loan1 \* P\_Loan1  
  
# Denominator includes both Loan = 1 and Loan = 0 branches  
Denominator <- Numerator + (P\_CC1\_given\_Loan0 \* P\_Online1\_given\_Loan0 \* P\_Loan0)  
  
# Final Naive Bayes probability  
NB\_Prob <- Numerator / Denominator

Naive Bayes Estimate: 0.1001

\* this number is the model’s estimate that a customer with credit card and online banking will accept a loan. Uses all training data.

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

# F. Compare result to part B  
  
cat("Empirical P(Loan = 1 | CC = 1, Online = 1):", round(Empirical\_Prob, 4), "\n")

## Empirical P(Loan = 1 | CC = 1, Online = 1): 0.0985

cat("Naive Bayes Estimate:", round(NB\_Prob, 4), "\n")

## Naive Bayes Estimate: 0.1001

The Empirical probability is 0.098 for customers who accept loan and are Credit Card holders and participate in online banking.

The Naïve Bayes estimate is 0.1001. While both values are close, the Naïve Bayes Estimate is considered more accurate for prediction purposes, since it draws on the conditional probabilities across the entire training set. It is more generalized and less susceptible to noise or the small sample bias.

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).

\*need two entries each from both table\_cc and table\_online, plus counts of Loan = 1 and Loan = 0

To Compute P(Loan=1∣CC=1,Online=1), we need six probabilities:

P(CC=1|Loan=1)

P(Online=1|Loan=1)

P(Loan=1)

P(CC=1|Loan=0)

P(Online=1|Loan=0)

P(Loan=0)

In Part G: the Naïve bayes estimate returned the same result of 0.1001

This is the same result that was obtained in Part E – manual type calculation.

# G. Run Naive Bayes model and examine output  
NB\_Model <- naiveBayes(Personal.Loan ~ CreditCard + Online, data = Train)  
# Predict probabilities on the training data  
NB\_Preds <- predict(NB\_Model, newdata = Train, type = "raw")  
  
# Find estimated probability for a hypothetical case: CC = 1, Online = 1  
newdata <- data.frame(CreditCard = factor(1, levels = c(0, 1)),  
 Online = factor(1, levels = c(0, 1)))  
Predicted\_Prob <- predict(NB\_Model, newdata, type = "raw")  
cat("Model-based P(Loan = 1 | CC = 1, Online = 1):", round(Predicted\_Prob[2], 4), "\n")

## Model-based P(Loan = 1 | CC = 1, Online = 1): 0.1001

Summary:

|  |  |
| --- | --- |
| **Estimate Type** | **Probability** |
| Empirical | 0.0985 |
| Naïve Bayes (Manual) | 0.1001 |
| Naïve Bayes( Model) | 0.1001 |