Assignment 3

SINGAMALLA SAIRITHIK

10-15-2023

Install and import libraries

```
# Load necessary libraries
options(warn = -1)  # Suppress all warnings
if (!require("caret")) install.packages("caret")
```

```
## Loading required package: caret
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
if (!require("e1071")) install.packages("e1071")
```

```
## Loading required package: e1071
```

```
if (!require("reshape2")) install.packages("reshape2")
```

```
## Loading required package: reshape2
```

```
library(caret)
library(e1071)
library(reshape2)
```

Load the dataset

```
# Load the data
data <-  read.csv("UniversalBank.csv")
```

partition of data into training and validation sets

```
set.seed(123)
index <- createDataPartition(data$Personal.Loan, p = 0.6, list = FALSE)
train_data <- data[index, ]
val_data <- data[-index, ]
colnames(train_data)
```

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

Task A

```
# Melt the data, specifying 'variable.name' and 'value.name'
data_melted <- melt(train_data, id.vars = c("Online", "CreditCard"), measure.vars = "Personal.Loan", va

# Cast the melted data into a wider format for the pivot table
pivot <- dcast(data_melted, CreditCard + Personal_Loan ~ Online, fun.aggregate = length)
```

```
## Using Personal_Loan as value column: use value.var to override.
```

```
print(pivot)
```

```
##   CreditCard Personal_Loan   0    1
## 1          0             0 785 1145
## 2          0             1  65  122
## 3          1             0 317  475
## 4          1             1  34   57
```

Task B

```
# Probability calculation for part b
subset_data <- subset(train_data, CreditCard == 1 & Online == 1)
prob_b <- nrow(subset(subset_data, `Personal.Loan` == 1)) / nrow(subset_data)

print(prob_b)
```

```
## [1] 0.1071429
```

Task C

```
# Pivot tables for part c
pivot_loan_online <- table(train_data$`Personal.Loan`, train_data$Online)
pivot_loan_cc <- table(train_data$`Personal.Loan`, train_data$CreditCard)

print(pivot_loan_online)
```

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

```r
print(pivot_loan_cc)
```

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

Task D

```r
# Probabilities for part d
P_CC_given_loan_1 <- pivot_loan_cc[2,2] / sum(pivot_loan_cc[2,])
P_online_given_loan_1 <- pivot_loan_online[2,2] / sum(pivot_loan_online[2,])
P_loan_1 <- sum(pivot_loan_cc[2,]) / sum(pivot_loan_cc)
P_CC_given_loan_0 <- pivot_loan_cc[1,2] / sum(pivot_loan_cc[1,])
P_online_given_loan_0 <- pivot_loan_online[1,2] / sum(pivot_loan_online[1,])
P_loan_0 <- sum(pivot_loan_cc[1,]) / sum(pivot_loan_cc)
```

Task E

```r
# Naive Bayes probability for part e
total <- nrow(train_data)
P_CC_1 <- sum(train_data$CreditCard == 1) / total
P_online_1 <- sum(train_data$Online == 1) / total
P_loan_1_given_CC_1_and_online_1 <- (P_CC_given_loan_1 * P_online_given_loan_1 * P_loan_1) / (P_CC_1 * P
```

Task F

```r
comparison <- data.frame(
  Method = c("Pivot Table", "Naive Bayes"),
  Probability = c(prob_b, P_loan_1_given_CC_1_and_online_1)
)

print(comparison)
```

```
##        Method Probability
## 1 Pivot Table   0.1071429
## 2 Naive Bayes   0.1106570
```

Task G

```r
# --- Task: Train a naive Bayes model and make a prediction for P(Loan = 1 | CC = 1, Online = 1) ---
model <- naiveBayes(as.factor(`Personal.Loan`) ~ Online + CreditCard, data = train_data)
newdata <- data.frame(Online = 1, CreditCard = 1)
pred <- predict(model, newdata, type = "raw")
prob_from_model <- pred[1, "1"]  # "1" is the factor level for loan acceptance
print(prob_from_model)  # This prints the probability from the naive Bayes model
```

```
##         1
## 0.1156935
```