Assignment_3

2025-10-09

Loading packages

```
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 (ISLR)
library(class)
library(gmodels)
library(e1071)

##
## Attaching package: 'e1071'

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

Importing the dataset

```
UniversalBank <- read_csv("./UniversalBank.csv")

## Rows: 5000 Columns: 14
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education,
M...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Train and Valid data partition

```
set.seed(135)  #help to keep the random with the same seed number
train.index=createDataPartition(UniversalBank$`Personal
Loan`,p=0.6,list=FALSE)
train.data=UniversalBank[train.index,]     #60%(3000)
valid.data=UniversalBank[-train.index,]    #40%(2000)

#creating subset of train.data
train.subset <- train.data %>% select(CreditCard, Online,`Personal Loan`)
```

Task A: Use of table function to create pivot

```
pivot.table <- table(CreditCard=train.subset$CreditCard,
PersonalLoan=train.subset$`Personal Loan`, Online=train.subset$Online)
dimnames(pivot.table) <- list(
  CreditCard = c("CC_0", "CC_1"),
  PersonalLoan = c("Loan_0", "Loan_1"),
  Online = c("Online_0", "Online_1")
)
ftable(pivot.table)

##                         Online Online_0 Online_1
## CreditCard PersonalLoan
## CC_0       Loan_0                   782     1119
##            Loan_1                    85      119
## CC_1       Loan_0                   320      496
##            Loan_1                    34       45
```

Task B: Calculating 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)

```
#The equation for this task will be,
probability=(CC=1+Loan=1+Online=1)/(CC=1+Online=1)
numerator <- pivot.table["CC_1", "Loan_1", "Online_1"]
denominator <- sum(pivot.table["CC_1", , "Online_1"]) #both Loan=0 and Loan=1

conditional.probability <- numerator / denominator
print(conditional.probability)

## [1] 0.0831793
```

Task C: Creating two separate pivot tables for the training data.

```
#Pivot Table 1: PersonalLoan vs Online
pivot1 <- table(PersonalLoan = train.subset$`Personal Loan`, Online =
train.subset$Online)
dimnames(pivot1) <- list(
  PersonalLoan = c("Loan_0", "Loan_1"),
```

```
  Online = c("Online_0", "Online_1")
)
print(pivot1)

##             Online
## PersonalLoan Online_0 Online_1
##       Loan_0     1102     1615
##       Loan_1      119      164

#Pivot Table 2: PersonalLoan vs CreditCard
pivot2 <- table(PersonalLoan = train.subset$`Personal Loan`, CreditCard =
train.subset$CreditCard)
dimnames(pivot2) <- list(
  PersonalLoan = c("Loan_0", "Loan_1"),
  CreditCard = c("CC_0", "CC_1")
)
print(pivot2)

##             CreditCard
## PersonalLoan CC_0 CC_1
##       Loan_0 1901  816
##       Loan_1  204   79
```

Task D: Computing probabilities

```
#P(CC = 1 | Loan = 1)
p_cc_loan1 <- pivot2["Loan_1", "CC_1"] / sum(pivot2["Loan_1", ])
print(p_cc_loan1)

## [1] 0.2791519

#P(Online = 1 | Loan = 1)
p_online_loan1 <- pivot1["Loan_1", "Online_1"] / sum(pivot1["Loan_1", ])
print(p_online_loan1)

## [1] 0.5795053

#P(Loan = 1)
p_loan1 <- sum(train.subset$`Personal Loan` == 1) / nrow(train.subset)
print(p_loan1)

## [1] 0.09433333

#P(CC = 1 | Loan = 0)
p_cc_loan0 <- pivot2["Loan_0", "CC_1"] / sum(pivot2["Loan_0", ])
print(p_cc_loan0)

## [1] 0.3003312

#P(Online = 1 | Loan = 0)
p_online_loan0 <- pivot1["Loan_0", "Online_1"] / sum(pivot1["Loan_0", ])
print(p_online_loan0)
```

```
## [1] 0.5944056
```

```r
#P(Loan = 0)
p_loan0 <- sum(train.subset$`Personal Loan` == 0) / nrow(train.subset)
print(p_loan0)
```

```
## [1] 0.9056667
```

Task E: Naive Bayes probability P(Loan = 1 | CC = 1, Online = 1)

```r
#In NB method, all events are considered independent. That's why in the
equation all events are multiplied
p_cc_online_loan1 <- p_cc_loan1 * p_online_loan1 * p_loan1
p_cc_online_loan0 <- p_cc_loan0 * p_online_loan0 * p_loan0

conditional.probability.NB <- p_cc_online_loan1 /
(p_cc_online_loan1+p_cc_online_loan0) ##for the denominator, CC=1 and
Online=1 cases for both Loan=0 and Loan=1 will be considered
print(conditional.probability.NB)
```

```
## [1] 0.08624632
```

Task F: Comparing this value with the one obtained from the pivot table in (B).

```r
#The value of conditational.probability= 0.0831793
#The value of conditional.probability.NB= 0.08624632

#The difference between probability values (~0.3%) is very small, so using
Naive Bayes in this case still gives a good approximation.So, in terms of
computational time, NB method will be more appropriate to use. But as we do
some assumptions in NB method, considering that the result of B will be more
accurate estimation as it is directly created from the train data.
```

Task G

```r
#To commute P(Loan=1 | CC=1, Online=1), I need following entries.
#For numerator, P(CC=1/Loan=1);P(Online=1/Loan=1);P(Loan=1)
#For denominator, P(CC=1/Loan=1);P(Online=1/Loan=1);P(Loan=1;)
P(CC=1/Loan0);P(Online=1/Loan0);P(Loan0);

#Following marked entries of the PivotTable, created in task A, will be
needed for computing P(Loan=1 | CC=1, Online=1) without any prior assumption,
considering the regular conditional praobility equation
ftable(pivot.table)
```

```
##                        Online Online_0 Online_1
## CreditCard PersonalLoan
## CC_0       Loan_0                  782     1119
##            Loan_1                   85      119
```

```
## CC_1       Loan_0                         320        496
##            Loan_1                          34         45
```

```r
#Runing naive Bayes on the data.
nb_model <- naiveBayes(`Personal Loan`~CreditCard+Online,data=train.subset)
#Following marked entries of the nb_model will be needed for computing
P(Loan=1 | CC=1, Online=1), considering the equation that assumes the events
independent
print(nb_model)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90566667 0.09433333
##
## Conditional probabilities:
##    CreditCard
## Y         [,1]      [,2]
##   0 0.3003312 0.4584864
##   1 0.2791519 0.4493770
##
##    Online
## Y         [,1]      [,2]
##   0 0.5944056 0.4910971
##   1 0.5795053 0.4945129
```

```r
#Examining the model output on training data, corresponds to P(Loan=1 | CC=1,
Online=1)
#Creating a data to test/examine the model
new_customer <- data.frame(CreditCard = 1, Online = 1)
test <- predict(nb_model, newdata = new_customer, type = "raw")
print(test)
```

```
##              0          1
## [1,] 0.9161652 0.08383477
```

```r
#The value of the probability of loan acceptance in this task is 0.08383477,
where the value I got in task E is 0.08624632. Though the values are
different, the differences is not too significant.
```