Assignment_3

Dengcheng

2022/3/6

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(caret)

## 载入需要的程辑包：ggplot2

## 载入需要的程辑包：lattice

library(class)
library(ISLR)
library(readr)
library(reshape)

##
## 载入程辑包：'reshape'

## The following object is masked from 'package:class':
##
##     condense

DF <- read_delim(file = 'UniversalBank.csv',delim=',')

## 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 m
essage.
```

```
DF <- rename(DF,c('CreditCard'='CC',`Personal Loan` = 'PL','Securities Accoun
t'='SA','CD Account'='CDA'))
names(DF)
```

```
##  [1] "ID"          "Age"         "Experience" "Income"      "ZIP Code"
##  [6] "Family"      "CCAvg"       "Education"  "Mortgage"    "PL"
## [11] "SA"          "CDA"         "Online"     "CC"
```

```
DF$PL=as.factor(DF$PL)
summary(DF)
```

```
##        ID            Age          Experience       Income          ZIP Cod
e
##  Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9
307
##  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91
911
##  Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93
437
##  Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93
153
##  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94
608
##  Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96
651
##      Family          CCAvg          Education        Mortgage     PL
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4520
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 480
##  Median :2.000   Median : 1.500   Median :2.000   Median :  0.0
##  Mean   :2.396   Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :635.0
##        SA              CDA             Online             CC
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.1044   Mean   :0.0604   Mean   :0.5968   Mean   :0.294
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

*Task 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(). In Python, use panda dataframe methods melt() and pivot().

```
Train_Index = createDataPartition(DF$PL,p=0.6, list=FALSE) # 60% reserved for
 Train
Train.df=DF[Train_Index,]
Validation.df=DF[-Train_Index,]
```

```
mytable <- xtabs(~ CC+PL+Online, data=Train.df)
ftable(mytable)
```

```
##       Online    0    1
## CC PL
## 0  0          778 1125
##    1           81  122
## 1  0          330  479
##    1           35   50
```

Task 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)].

p(PL=1&CC=1&Online=1 | CC=1&Online=1) = 50/(479+50)=0.095

Task 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.

```
table(PL=Train.df$PL, Online=Train.df$Online)
```

```
##    Online
## PL     0    1
##   0 1108 1604
##   1  116  172
```

```
table(PL=Train.df$PL, CC=Train.df$CC)
```

```
##    CC
## PL     0    1
##   0 1903  809
##   1  203   85
```

Task D Compute the following quantities [P(A | B) means “the probability ofA given B”]: i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) ii. P(Online = 1 | Loan = 1) iii. P(Loan = 1) (the proportion of loan acceptors) iv. P(CC = 1 | Loan = 0) v. P(Online = 1 | Loan = 0) vi. P(Loan = 0)

i. P(CC = 1 | Loan = 1) = 83/(205+83) = 0.29
ii. P(Online = 1 | Loan = 1) = 180/(108+180) = 0.625
iii. P(Loan = 1) = (108+180+205+83)/(1117+1595+108+180+1914+798+205+83) = 0.096
iv. P(CC = 1 | Loan = 0) = 798/(1914+798) = 0.29
v. P(Online = 1 | Loan = 0) = 1595/(1117 +1595)= 0.59
vi. P(Loan = 0) = 1-P(Loan = 1) = 0.904

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

P(CC = 1) = (798+83)/(1914+205+798+83) = 0.29 P(Online = 1) = (1595+180)/(1117+108+1595+180) = 0.59 P(Loan = 1 | CC = 1, Online = 1) = [P(CC = 1|Loan = 1)P(Online = 1|Loan = 1)P(Loan = 1)] /[P(CC = 1)P(Online = 1)] =0.29*0.625*0.096/(0.29*0.59) = 0.10

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

Task E is more accurate.

Task 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).

```
library(e1071)
nb.model<-naiveBayes (PL~CC+Online, data=Train.df)
To_Predict=data.frame(CC=1, Online = 1)
predict(nb.model,To_Predict,type='raw')

##              0          1
## [1,] 0.9042433 0.09575667
```

The result is very close to (E).

.