FML 3 ASSIGNMENT

JHANSI NAIDU

2024-03-02

SUMMARY

1. **Probability of Loan Acceptance** : Based on the pivot table, the likelihood of a client accepting a loan when they actively use online banking services (Online = 1) and hold a bank credit card (CC = 1) is estimated to be roughly 0.0297. But when this probability is calculated using the Naive Bayes technique, the outcome is far less—roughly 0.001—than before. This disparity raises the possibility that the Naive Bayes model may not adequately represent the correlation between the variables in this particular case.

2. **Model Evaluation** : Based on the available data, the pivot table's probability is judged to be more accurate when compared to the Naive Bayes algorithm's probability. This shows that in order to increase its accuracy, the Naive Bayes model might not be the best fit for this specific dataset or could need more adjusting and fine-tuning.

3. **Conditional Probabilities**: A number of conditional probabilities were computed, including the proportion of loan acceptors in the dataset overall and the likelihood of holding a credit card given loan acceptance as well as the likelihood of online banking activity given loan acceptance. These probabilities can be useful for comprehending consumer behavior and decision-making since they provide light on the correlations between various variables.

4. **Model Output Examination**: Based on credit card ownership and online banking behavior, the model output on the training data was analyzed to identify the entry that corresponded to the likelihood of loan acceptance. It is possible to ascertain whether the model appropriately depicts the underlying patterns in the data by comparing the estimates produced by the model with the estimated probability.

Overall, this analysis sheds light on the variables affecting bank customers' acceptance of loans and emphasizes the significance of comparing various modeling strategies in order to produce precise forecasts. To increase forecast accuracy and enhance the information available to decision-makers, more research and model improvement may be required.

Loading required libraries

We loaded required libraries like tidy verse, reshape, reshape2, and caret, which provide functions for data manipulation, visualization, and machine learning.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:lubridate':
##
##     stamp

## The following object is masked from 'package:dplyr':
##
##     rename

## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

We loaded the dataset UniversalBank.csv into R using the read.csv() function. The head() and tail() functions are used to display the first and last few rows of the dataset which makes it easier for us to analyse the data

Loading the data csv file

```
library(e1071)

Universal_Bank_data <- read.csv("/Users/chaithanayayennam/Downloads/UniversalBank.csv")

head(Universal_Bank_data)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```
tail(Universal_Bank_data)
```

```
##        ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 4995 4995  64         40     75    94588      3   2.0         3        0
## 4996 4996  29          3     40    92697      1   1.9         3        0
## 4997 4997  30          4     15    92037      4   0.4         1       85
## 4998 4998  63         39     24    93023      2   0.3         3        0
## 4999 4999  65         40     49    90034      3   0.5         2        0
## 5000 5000  28          4     83    92612      3   0.8         1        0
##      Personal.Loan Securities.Account CD.Account Online CreditCard
## 4995             0                  0          0      1          0
## 4996             0                  0          0      1          0
## 4997             0                  0          0      1          0
## 4998             0                  0          0      0          0
## 4999             0                  0          0      1          0
## 5000             0                  0          0      1          1
```

We defined the column names and transformed the data into factors and categories to perform the analysis.We then divided the dataset into training and validation sets using a 60/40 split.

Defining the Column Names

```
colnames(Universal_Bank_data)
```

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

Data transformation into factors and categories

```
Universal_Bank_data$`Personal Loan` = as.factor(Universal_Bank_data$Personal.Loan)
Universal_Bank_data$Online = as.factor(Universal_Bank_data$Online)
Universal_Bank_data$CreditCard = as.factor(Universal_Bank_data$CreditCard)
```

60 % training data and 40% Validation data respectively

```
set.seed(456)

Universal_bank_train <- sample(row.names(Universal_Bank_data), 0.6*dim(Universal_Bank_data)[1])
Universal_bank_valid <- setdiff(row.names(Universal_Bank_data), Universal_bank_train)
```

Putting training and validation data into separate dataframes from the dataset

```
Universal_bank_train_data <- Universal_Bank_data[Universal_bank_train, ]
Universal_bank_valid_data <- Universal_Bank_data[Universal_bank_valid, ]
```

Duplicating the data frame UB.train and UB.valid

```
train <- Universal_Bank_data[Universal_bank_train, ]

valid <- Universal_Bank_data[Universal_bank_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().

We created pivot tables to review the relationship between variables like Online, Credit Card, and Loan acceptance. These tables provided the counts of different combinations of variables which helped us in interpreting the distribution of data and identifying the patterns.

Melt data from data

```
m_data = melt(train,id=c("CreditCard","Personal.Loan"),variable= "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

Casting the melted data

```
cast_data <- dcast(m_data, CreditCard + Personal.Loan ~ value, fun.aggregate = length)

cast_data[,c(1,2,3,14)]
```

```
##   CreditCard Personal.Loan -1 0.7
## 1          0             0 14  72
## 2          0             1  0   0
## 3          1             0  5  31
## 4          1             1  0   2
```

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)].

We calculated various conditional probabilities, such as the probability of having a credit card given that a loan is accepted, and the probability of loan acceptance given online banking activity and credit card ownership. These probabilities' calculations helped us in assessing the likelihood of different outcomes and informing a better decision-making.

The pivot table indicates that the value for CC and the value for Loan is 89

```
Universal_bank_loan_crc <- 89/3000
Universal_bank_loan_crc
```

```
## [1] 0.02966667
```

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.

Transforming the train data frame into a lengthy format, using "Online" as a variable to be melted and "Personal.Loan" as an identification

```
m1 = melt(train,id=c("Personal.Loan"),variable = "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

Creating a long format out of the train data frame, using "CreditCard" as an identifier and "Online" as a variable that needs to be melted

```
m2 = melt(train,id=c("CreditCard"),variable = "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

Casting Personal loan and online values

```
c1=dcast(m1,`Personal.Loan`~Online)
```

```
## Aggregation function missing: defaulting to length
```

Casting Personal loan and online values

```
c2=dcast(m2,CreditCard~Online)
```

```
## Aggregation function missing: defaulting to length
```

Displaying the quantity of personal loans compared to online

```
Universal_bank_loan_online=c1[,c(1,13)]
Universal_Bank_Loan_CC = c2[,c(1,14)]
Universal_bank_loan_online
```

```
##   Personal.Loan Online
## 1             0   2711
## 2             1    289
```

Displaying the quantity of credit cards in relation to online

```
Universal_Bank_Loan_CC
```

```
##   CreditCard Online
## 1          0   2117
## 2          1    883
```

D. Compute the following quantities [P (A | B) means "the probability of A given B"]:

1.P (CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) #2.P(Online=1|Loan=1) #3.P (Loan = 1) (the proportion of loan acceptors) #4.P(CC=1|Loan=0) #5.P(Online=1|Loan=0) #6.P(Loan=0)

Making a pivot table where personal loans are represented by columns 14 and 10

```
table(train[,c(14,10)])
```

```
##            Personal.Loan
## CreditCard    0    1
##          0 1917  200
##          1  794   89
```

Making a pivot table for the online and personal loan columns 13 and 10

```
table(train[,c(13,10)])
```

```
##       Personal.Loan
## Online    0    1
##      0 1046  112
##      1 1665  177
```

Pivot table for personal loans There are, from training, 2725 and 275, respectively

```
table(train[,c(10)])
```

```
##
##    0    1
## 2711  289
```

By consulting the above p, we may determine the CC= 1 and Loan = 1 values

```
Universal_Bank_CCUni_Bank.Loan1 = 89/(89+200)
Universal_Bank_CCUni_Bank.Loan1
```

```
## [1] 0.3079585
```

P(Online=1|Loan=1)

The pivot table above UB.ONUB.Loan1 gives us the data for online = 1 and loan = 1

```
Universal_Bank_ONUni_Bank.Loan1 =177/(177+112)
```

P(Loan = 1)

By referring the above pivot table we can get the Loan = 1

```
Universal_Bank_Loan1 =289/(289+2711)
Universal_Bank_Loan1
```

```
## [1] 0.09633333
```

P(CC=1|Loan=0)

The CC = 1 and Loan = 0 values can be obtained by using the pivot table above

```
Universal_Bank_CCLoan.01= 794/(794+1917)
Universal_Bank_CCLoan.01
```

```
## [1] 0.2928809
```

P(Online=1|Loan=0)

The pivot table above gives us the numbers for online = 1 and loan = 0

```
Universal_Bank_ON1.L0= 1665/(1665+1046)
Universal_Bank_ON1.L0
```

```
## [1] 0.6141645
```

P(Loan=0)

The pivot table above allows us to extract the Loan = 0 values

```
Universal_Bank_Loan0= 2711/(2711+289)
Universal_Bank_Loan0
```

```
## [1] 0.9036667
```

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

We applied Naive Bayes algorithm to predict the probability of loan acceptance in the given specific conditions, such as having a credit card and using online banking services. This approach uses the calculated probabilities to make forecasts based on observed data.

Given probabilities

```
P_CC_Loan1 <- 0.096
P_Online_Loan1 <- 0.833
P_L1 <- 0.0125
```

Calculate Naive Bayes probability P(Loan = 1 | CC = 1, Online = 1)

```
Universal_Bank_Naive_bayes <- (P_CC_Loan1)*(P_Online_Loan1)*(P_L1)
Universal_Bank_Naive_bayes
```

```
## [1] 0.0009996
```

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

We compared the Naive Bayes Probability with the probability acquired from the pivot table to analyze the accuracy of the model. This comparison helped us in evaluating the performance of the Naive Bayes algorithm in estimating the loan acceptance.

Naive Bayes Probability (from calculation in E)

```
naive_bayes_probability <- 0.0009996
```

Pivot Table Probability

```
pivot_table_probability <- 0.02966667
```

Compare the odds and print a message stating which is more likely to be true

```
if (naive_bayes_probability > pivot_table_probability) {
  message("Naive Bayes Probability is more accurate: ", naive_bayes_probability)
} else if (naive_bayes_probability < pivot_table_probability) {
  message("Pivot Table Probability is more accurate: ", pivot_table_probability)
} else {
  message("Both Probabilities are the same: ", naive_bayes_probability)
}
```

```
## Pivot Table Probability is more accurate: 0.02966667
```

Based on the comparison, the pivot table probability (0.02966667) is considered more accurate compared to the Naive Bayes probability

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).

Lastly, we examined the model output on the training data to find the entry related to the probability of loan acceptance in the given criteria like credit card ownership and online banking activity. This evaluation helped us validate the model's estimates and evaluate its authenticity.

```
names(Universal_Bank_data)
```

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

```r
names(Universal_bank_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"         "Personal Loan"
```

Choosing the pertinent training columns

```r
Universal_Bank.train <- Universal_Bank_data[, c("CreditCard", "Online", "Personal Loan")]
```

Changing the columns' names to eliminate gaps

```r
colnames(Universal_Bank.train) <- c("CreditCard", "Online", "PersonalLoan")
```

Converting "Online" and "CreditCard" to factors with the proper levels

```r
Universal_Bank.train$CreditCard <- factor(Universal_Bank.train$CreditCard, levels = c(0, 1), labels = c
Universal_Bank.train$Online <- factor(Universal_Bank.train$Online, levels = c(0, 1), labels = c("No", "
```

Printing the probability

```r
print("Prob of Loan = 1 given CC = 1 and Online = 1:")
```

```
## [1] "Prob of Loan = 1 given CC = 1 and Online = 1:"
```