Assignment 3 Fundamentals of ML

R Markdown

library(reshape) library(caret) library(e1071)

readin the excel data into dataframe

```
rm(list=ls())
getwd()
```

```
## [1] "/Users/ankithdasu/Desktop/Spring 2022/Fundamentals of Machine Learning/Assignment 3"
```

```
setwd("/Users/ankithdasu/Desktop/Spring 2022/Fundamentals of Machine Learning/Assignment 3")
NvBay3 <- read.csv("UniversalBank.csv")
head(NvBay3)
```

```
##   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(NvBay3)
```

```
##        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
```

Converting data into factors(categorical) mainly the one which are important to this.

```
NvBay3$Personal.Loan = as.factor(NvBay3$Personal.Loan) # converting Personal Loan into categorical data
NvBay3$Online = as.factor(NvBay3$Online) # converting Online into categorical data
NvBay3$CreditCard = as.factor(NvBay3$CreditCard) # converting CreditCard into categorical data
```

#Data partition 60 % training and 40 % into validation

```
set.seed(64060)
train.index <- sample(row.names(NvBay3), 0.6*dim(NvBay3)[1])  # 60 % of data into training set
valid.index <- setdiff(row.names(NvBay3), train.index)  # 40 %  into validation set
train.df <- NvBay3[train.index, ] # assigning the train.index into data frame
valid.df <- NvBay3[valid.index, ] # assigning the validation index into data frame
train <- NvBay3[train.index, ] # Making a copy of the data frame train.df
valid = NvBay3[train.index,]#  Making a copy of the data frame valid.df
```

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().

Pivot table For CreditCard , Personal loan as row variables and Online in column.

```
library(reshape2)
melt = melt(train,id=c("CreditCard","Personal.Loan"),variable= "Online")
```

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

```
cast=dcast(melt,CreditCard+Personal.Loan~Online)  # dcast is to convert the data in CC , Personal loan
```

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

```
cast[,c(1,2,3,14)] # casting column no 14 which credit card  and 1 , 2 , 3 column is  , personal loan,
```

```
##   CreditCard Personal.Loan   ID Online
## 1          0             0 1931   1931
## 2          0             1  200    200
## 3          1             0  784    784
## 4          1             1   85     85
```

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)].

```
Loan_CC1 <- 77/3000  # 77 is the value for Loan and CC =1 as per pivot table.  and 3000 is the total co
Loan_CC1 # which is 26 %.
```

```
## [1] 0.02566667
```

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.

```
melt_1 = melt(train,id=c("Personal.Loan"),variable = "Online") # Melting Personal loan and Online data
```

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

```
melt_2 = melt(train,id=c("CreditCard"),variable = "Online") # Melting Credicard data with reference to
```

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

```
cast_1 =dcast(melt_1,Personal.Loan~Online) # Casting Personal loan and online values
```

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

```
cast_2=dcast(melt_2,CreditCard~Online) # Casting Personal loan and online values
```

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

```
Loanonline=cast_1[,c(1,13)]
LoanCC = cast_2[,c(1,14)]

Loanonline # indicates  personal loan count in reference with online
```

```
##   Personal.Loan Online
## 1             0   2715
## 2             1    285
```

```
LoanCC # Indicates Credit Card count in reference with online.
```

```
##   CreditCard Online
## 1          0   2131
## 2          1    869
```

D. Compute the following quantities [P (A | B) means "the probability of A given B"]: P (CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) P(Online=1|Loan=1) P (Loan = 1) (the proportion of loan acceptors) P(CC=1|Loan=0) P(Online=1|Loan=0) P(Loan=0)

```
table(train[,c(14,10)])  # Creating a pivot table for column 14 and 10 which is credit card and persona
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1931  200
##          1  784   85
```

```
table(train[,c(13,10)])  #  Creating a pivot table for column 13 and 10 which is online and personal lo
```

```
##       Personal.Loan
## Online    0    1
##      0 1094  111
##      1 1621  174
```

```
table(train[,c(10)]) # Pivot table for Personal loan.  There are 2725 and 275 from training
```

```
##
##    0    1
## 2715  285
```

P (CC = 1 | Loan = 1)

```
CCLoan_1 = 77/(77+198) # by referring the above pivot table we can get the CC= 1 and Loan = 1 values, w
CCLoan_1
```

```
## [1] 0.28
```

P(Online=1|Loan=1)

```
ONLoan_1 =166/(166+109) # by referring the above pivot table we can get the online = 1 and Loan = 1 val
ONLoan_1
```

```
## [1] 0.6036364
```

P (Loan = 1)

```
Loan_1 =275/(275+2725) # by referring the above pivot table we can get the Loan = 1
Loan_1
```

```
## [1] 0.09166667
```

P(CC=1|Loan=0)

```
CCLoan_01= 801/(801+1924) # by referring the above pivot table we can get the CC = 1 and Loan = 0 value
CCLoan_01
```

```
## [1] 0.293945
```

P(Online=1|Loan=0)

```
ON1L0= 1588/(1588+1137)  # by referring the above pivot table we can get the online = 1 and Loan = 0 va
ON1L0
```

```
## [1] 0.5827523
```

P(Loan=0)

```
Loan_0= 2725/(2725+275)  # by referring the above pivot table we can get the Loan = 0 values
Loan_0
```

```
## [1] 0.9083333
```

E. Use the quantities computed above to compute the naive Ba1 probability P(Loan = 1 | CC = 1, Online = 1).

```
Naivebayes = ((77/(77+198))*(166/(166+109))*(275/(275+2725)))/(((77/(77+198))*(166/(166+109))*(275/(275+

Naivebayes # 90 % is the probability
```

```
## [1] 0.09055758
```

F. Compare this value with the one obtained from the pivot table in (b). Which is a more accurate estimate? 9.05% are very similar to the 9.7% the difference between the exact method and the naive-baise method is the exact method would need the the exact same independent variable classifications to predict, where the naive bayes method does not.

```
library(caret)
```

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

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

```
library(e1071)
naive.train1 = train.df[,c(10,13,14)] # training data is from Personal loan, Creditcard and online. col
naive.test1 =valid.df[,c(10,13,14)] # testing set data from the same columns of data
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train1) # applying naivebayes algorithm to personal
naivebayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
## 0.905 0.095
##
```

```
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4029466 0.5970534
##   1 0.3894737 0.6105263
##
##    CreditCard
## Y           0         1
##   0 0.7112339 0.2887661
##   1 0.7017544 0.2982456
```

G. Which of the entries in this table are needed for computing P (Loan = 1 | CC = 1, Online = 1)? In R, 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).

Answer:

For the naivebayes, it is same output that we have got in the manual calculation prcoess. (.280)(.603)(.09)/(.280.603.09+.29.58. = .09 which is the same as the manual calculation process.