Assignment_3

Vamshee Deepak Goud Katta

10/17/2021

1. Inserting Data and Libraries

Reading the UniversalBank csv file and inserting approproate libraries

```
library(class)
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(ISLR)
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
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(tidyr)
library(ggplot2)

UBank_data <- read.csv("UniversalBank.csv")
```

2. Data Selection

```r
UBank_data_d <- dummy.data.frame(select(UBank_data, c(Personal.Loan, CreditCard, Online)))
head(UBank_data_d)
```

```
##   Personal.Loan CreditCard Online
## 1             0          0      0
## 2             0          0      0
## 3             0          0      0
## 4             0          0      0
## 5             0          1      0
## 6             0          0      1
```

3. Data Partition

Partitioning the UniversalBank dataset into Training and validation sets

```r
set.seed(123)
Index_Train <- createDataPartition(UBank_data_d$Personal.Loan, p=0.6, list = FALSE)
# 60% of data is taken as Training Data

Train <- UBank_data_d[Index_Train,]
Validation <- UBank_data_d[-Index_Train,]
# Rest of the data is taken as Validation Data

summary(Train)
```

```
##  Personal.Loan       CreditCard         Online
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000   Median :1.0000
##  Mean   :0.09267   Mean   :0.2943   Mean   :0.5997
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
```

```r
summary(Validation)
```

```
##  Personal.Loan     CreditCard         Online
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000   Median :1.0000
##  Mean   :0.101   Mean   :0.2935   Mean   :0.5925
##  3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000
```

4. A. Creating Pivot table using Online, CC and Loan variables

```r
library(reshape2)
```

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

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

```r
names(Train)
```

```
## [1] "Personal.Loan" "CreditCard"    "Online"
```

```r
dcast(Train, Personal.Loan + CreditCard ~ Online)
```

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

## Aggregation function missing: defaulting to length

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

5. B. Naive Bayes Classification for customer with CC=1, Online=1, Personal.Loan=1

Total number of customers with CC=1 and Online=1 is 475+57 = 532

Number of customers with all variables as 1 is 57

Hence probability of customer taking personal loan is 57/532 = "0.1071"

6. C. Preparing separate pivot tables for Loan against Online and Loan against CC

a) Loan v/s Online

```r
dcast(Train, Personal.Loan ~ Online)
```

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

## Aggregation function missing: defaulting to length

##   Personal.Loan    0    1
## 1             0 1102 1620
## 2             1   99  179
```

b) Loan v/s CC

```r
dcast(Train, Personal.Loan ~ CreditCard)
```

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

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

```
##   Personal.Loan    0   1
## 1             0 1930 792
## 2             1  187  91
```

7. D. Computations of individual probabilities

i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)

Number of loan acceptors is 187+91 = 278

Number of Acceptors using credit cards is 91

Hence P(CC=1| Loan=1) = 91/278 = "0.3273"

ii. P(Online = 1 | Loan = 1)

Number of loan acceptors is 99+179 = 278

Number of acceptors with online presence is 179

Hence P(Online = 1 | Loan = 1) = 179/278 = "0.6438"

iii. P(Loan = 1) (the proportion of loan acceptors)

Total number of loan acceptors is 65+122+34+57 = 278

Hence proportion of loan acceptors = total of loan acceptors/total of customers = 278/3000 = "0.0926"

iv. P(CC = 1 | Loan = 0)

Number of loan rejectors is 1930+792 = 2722

Number of rejectors with credit card is 792

Hence P(CC = 1 | Loan = 0) = 792/2722 = "0.2909"

v. P(Online = 1 | Loan = 0)

Number of loan rejectors is 1102+1620 = 2722

Number of rejectors with online presence is 1620

Hence P(Online = 1 | Loan = 0) = 1620/2722 = "0.5951"

vi. P(Loan = 0) (the proportion of loan rejectors)

Total number of loan rejectors is 785+1145+317+475 = 2722

Hence P(Loan = 0) = total of rejectors/total of customers = 2722/3000 = "0.9073"

```r
# 10. G. Modelling Naive Bayes on the Data set
library(e1071)

nb_model <-naiveBayes(Personal.Loan ~ CreditCard + Online , data = Train)
nb_model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90733333 0.09266667
##
## Conditional probabilities:
##    CreditCard
## Y        [,1]      [,2]
##   0 0.2909625 0.4542897
##   1 0.3273381 0.4700881
##
##    Online
## Y        [,1]      [,2]
##   0 0.5951506 0.4909531
##   1 0.6438849 0.4797134
```

8. E. Naive Bayes probability P(Loan = 1 | CC = 1, Online = 1)

= P(CC = 1 | Loan = 1) * P(Online = 1 | Loan = 1) * P(Loan = 1) /

[P(CC = 1 | Loan = 1) * P(Online = 1 | Loan = 1) * P(Loan = 1)] + [P(CC = 1 | Loan = 0) * P(Online = 1 | Loan = 0) * P(Loan = 0)]

= 0.3273*0.6438*0.0926/(03273*0.6438*0.0926)+(0.2909*0.5951*0.9073)

= 0.0195/(0.0195+0.1570)

Hence P(Loan = 1 | CC = 1, Online = 1) = "0.1104"

9. F. The value obtained through Naive Bayes is slightly higher compared to the value of 0.1071, obtained from the pivot table which is more accurate compared to the Naive Bayes value

10. G. Using the model to find the entry corresponding to P(Loan = 1 | CC = 1, Online = 1)

```r
# Train
pred.class <- predict(nb_model, newdata = Train)

# validation
pred.prob <- predict(nb_model, newdata=Validation, type="raw") # probabilities
pred.class <- predict(nb_model, newdata = Validation) # class membership

# For the test set
df <- data.frame(actual = Validation$Personal.Loan, predicted = pred.class, pred.prob)

df[Validation$Personal.Loan == 1 & Validation$CreditCard == 1 & Validation$Online == 1,]
```

```
##     actual predicted        X0        X1
## 20       1         0 0.8843065 0.1156935
## 87       1         0 0.8843065 0.1156935
## 140      1         0 0.8843065 0.1156935
## 176      1         0 0.8843065 0.1156935
## 366      1         0 0.8843065 0.1156935
## 428      1         0 0.8843065 0.1156935
## 434      1         0 0.8843065 0.1156935
## 455      1         0 0.8843065 0.1156935
## 461      1         0 0.8843065 0.1156935
## 499      1         0 0.8843065 0.1156935
## 524      1         0 0.8843065 0.1156935
## 612      1         0 0.8843065 0.1156935
## 628      1         0 0.8843065 0.1156935
## 652      1         0 0.8843065 0.1156935
## 682      1         0 0.8843065 0.1156935
```

```
## 774        1          0 0.8843065 0.1156935
## 884        1          0 0.8843065 0.1156935
## 1132       1          0 0.8843065 0.1156935
## 1258       1          0 0.8843065 0.1156935
## 1390       1          0 0.8843065 0.1156935
## 1601       1          0 0.8843065 0.1156935
## 1722       1          0 0.8843065 0.1156935
## 1742       1          0 0.8843065 0.1156935
## 1938       1          0 0.8843065 0.1156935
## 1979       1          0 0.8843065 0.1156935
```