assign3

Hannah Cronin

2022-10-17

```
library(caret)
```

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

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

```
library(ISLR)
library(e1071)
library(readr)
library(ggplot2)
library(lattice)
library(reshape)
library(reshape2)
```

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

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

```
UniversalBank <- read_csv("Desktop/GITHUB/64060_-HCRONIN-FML/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.
```

```
loan = UniversalBank[,-c(1,2,3,4,5,6,7,8,9,11,12)] #more narrowed dataset
colnames(loan)[1] = 'Personal_Loan'
```

```
Index_Train = createDataPartition(loan$`Personal_Loan`, p = .6, list=FALSE)
Train = loan[Index_Train,]
Valid = loan[-Index_Train,]
```

A.

```
pivot = melt(Train, id = c('CreditCard','Personal_Loan'), variable = 'Online')
cast.loan = cast(pivot, CreditCard+Personal_Loan~Online)
```

```
## Aggregation requires fun.aggregate: length used as default
```

```
cast.loan[, c(1:3)]
```

```
##   CreditCard Personal_Loan Online
## 1          0             0   1909
## 2          0             1    216
## 3          1             0    792
## 4          1             1     83
```

B.

```
prob = 87 / (1919+206+788+87)
prob * 100
```

```
## [1] 2.9
```

```
# The probability that this customer will accept a loan is 2.90%.
```

C.

```
pivot2 = melt(Train, id = c('Personal_Loan'), variable = 'Online')
```

```
cast.loan2 = cast(pivot2, Personal_Loan~Online)
```

```
## Aggregation requires fun.aggregate: length used as default
```

```
cast.loan2[,c(1:2)]
```

```
##   Personal_Loan Online
## 1             0   2701
## 2             1    299
```

```
pivot3 = melt(Train, id = c('Personal_Loan'), variable = 'CreditCard')
```

```
cast.loan3 = cast(pivot3, Personal_Loan~CreditCard)
```

```
## Aggregation requires fun.aggregate: length used as default
```

```
cast.loan3[,c(1,3)]
```

```
##   Personal_Loan CreditCard
## 1             0       2701
## 2             1        299
```

I can’t get the pivot tables above to display the correct answer - they seem to be defaulted to the number of people who accepted the loan regardless of the other variable we’re looking at. (293 & 2707)

I’m going to use regular tables instead so at least I can compute the right answers for D.

```
table(Train[,c(1,2)])
```

```
##              Online
## Personal_Loan    0    1
##             0 1085 1616
##             1  121  178
```

```
table(Train[,c(1,3)])
```

```
##              CreditCard
## Personal_Loan    0    1
##             0 1909  792
##             1  216   83
```

D.

```
# 1. 29.693%
((87)/(206+87))
```

```
## [1] 0.2969283
```

```
# 2. 60.751%
(178/(115+178))
```

```
## [1] 0.6075085
```

```
# 3. 9.767%
(293/3000)
```

```
## [1] 0.09766667
```

```
# 4. 29.110 %
(788/(788+1919))
```

```
## [1] 0.2910972
```

```
# 5. 59.660
(1615/(1615+1092))
```

```
## [1] 0.5966014
```

```
# 6. 90.233%
((3000-293)/3000)
```

```
## [1] 0.9023333
```

E.

```
((87/(206+87))*(178/(178+115))*(293/(293+2707)))/(((77/(77+198))*(178/(178+115))*(293/(2
93+2707)))+((788/(788+1919))*(1615/(1615+1092))*2707/(2707+293)))
```

```
## [1] 0.1016483
```

F. 10.16% and 10.73% are very similar. Since Naive assumes independence of the variables- it's a more accurate estimate.

G.

```
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
## 0.90033333 0.09966667
##
## Conditional probabilities:
##    CreditCard
## Y        [,1]      [,2]
##   0 0.2932247 0.4553249
##   1 0.2775920 0.4485617
##
##    Online
## Y        [,1]      [,2]
##   0 0.5982969 0.4903333
##   1 0.5953177 0.4916533
```

```
(.458)*(.489)*(.10)/((.458*.489*.098)+(.454*.491*.90))
```

```
## [1] 0.100625
```

This number is very close to the one found in E.