Assignment 3

Sean Bradford

```
rm(list=ls())
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.1
```

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

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

```
library(ISLR)
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.1
```

```
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 4.2.1
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.2.1
```

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

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

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.

```
bank = read.csv("C:\\Users\\Sean\\OneDrive\\Desktop\\Grad School\\Machine Learning\\Module 5 - Naive Ba
bank$Personal.Loan = as.factor(bank$Personal.Loan)
bank$Online = as.factor(bank$Online)
bank$CreditCard = as.factor(bank$CreditCard)
set.seed(111)

train.index=createDataPartition(bank$Personal.Loan,p=0.6,list=FALSE)
train.df = bank[train.index, ]
test.df = bank[-train.index, ]
train = bank[train.index, ]
test = bank[train.index,]
```

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

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

```
recast.bank =dcast(melted.bank,CreditCard+Personal.Loan~Online)
```

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

```
recast.bank[,c(1:2,14)]
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1918
## 2          0             1    200
## 3          1             0    794
## 4          1             1     88
```

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)].

```
Online_sum=1918+200+794+88
Online_sum
```

```
## [1] 3000
```

```
CustomerB=88/3000
CustomerB
```

```
## [1] 0.02933333
```

The probability of a customer (who owns a bank credit card and is actively using online services) to accept the loan offer is 2.9%.

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.bank1 = melt(train,id=c("Personal.Loan"),variable = "Online")
```

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

```
melt.bank2 = melt(train,id=c("CreditCard"),variable = "Online")
```

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

```
recast.bank1=dcast(melt.bank1,Personal.Loan~Online)
```

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

```
recast.bank2=dcast(melt.bank2,CreditCard~Online)
```

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

```
Loanline=recast.bank1[,c(1,13)]
LoanCC = recast.bank2[,c(1,14)]
```

```
Loanline
```

```
##   Personal.Loan Online
## 1             0   2712
## 2             1    288
```

```
LoanCC
```

```
##   CreditCard Online
## 1          0   2118
## 2          1    882
```

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)

```
#i & iv
table(train[,c(14,10)])
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1918  200
##          1  794   88
```

```
Di=87/(87+205)
Di
```

```
## [1] 0.2979452
```

```
Div=812/(812+1896)
Div
```

```
## [1] 0.2998523
```

```
#ii & v
table(train[,c(13,10)])
```

```
##       Personal.Loan
## Online    0    1
##      0 1111  111
##      1 1601  177
```

```
Dii=180/(180+112)
Dii
```

```
## [1] 0.6164384
```

```
Dv=1596/(1596+1112)
Dv
```

```
## [1] 0.5893648
```

```
#iii & vi
table(train[,c(10)])
```

```
##
##    0    1
## 2712  288
```

```
Diii=292/(292+2708)
Diii
```

```
## [1] 0.09733333
```

```
Dvi=2708/(2708+292)
Dvi
```

```
## [1] 0.9026667
```

i. 29.8%
ii. 61.6%
iii. 9.7%
iv. 30.0%
v. 58.9%
vi. 90.3%

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

```
De=(Di*Dii*Diii)/((Di*Dii*Diii)+(Div*Dv*Dvi))
De
```

```
## [1] 0.1007717
```

Probability is 10.1%

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

- The 10.1% from the naive Ba1 probability is more accurate because the naive bayes calculation (unlike the method from question B) does not rely on all predictor values being identical.

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).

```
nb.train = train.df[,c(10,13:14)]
nb.test = test.df[,c(10,13:14)]
nb = naiveBayes(Personal.Loan~.,data=nb.train)
nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
## 0.904 0.096
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4096608 0.5903392
##   1 0.3854167 0.6145833
##
##    CreditCard
## Y           0         1
##   0 0.7072271 0.2927729
##   1 0.6944444 0.3055556
```

```
Dg=(0.096*0.6145833*0.3055556)/((0.096*0.6145833*0.3055556)+(0.904*0.5903392*0.2927729))
Dg
```

```
## [1] 0.1034469
```

Probability is 10.3%.

The probability obtained in G. is almost the same as the probability obtained in E. There is a slight 0.2% difference between the values.