sbaig1_assgnment3

```
library(readr)
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(dummies)

## dummies-1.5.6 provided by Decision Patterns

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(class)
library(reshape)

##
## Attaching package: 'reshape'

## The following object is masked from 'package:class':
##
##     condense

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

library(reshape2)

##
## Attaching package: 'reshape2'

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

library(ggplot2)
library(ISLR)
library(e1071)

setwd("C:/Users/shari/OneDrive/Desktop/Business Analytics/Sem 1/Machine Le
arning/ML_Assignment3")
```

```
Bankdata<- read.csv("C:/Users/shari/OneDrive/Desktop/Business Analytics/Se
m 1/Machine Learning/ML_Assignment3/UniversalBank.csv")
```

##Transform Data

```
Bankdata$PersonalLoan <- as.factor(Bankdata$Personal.Loan)
Bankdata$Online <- as.factor(Bankdata$Online)
Bankdata$CreditCard <- as.factor(Bankdata$CreditCard)
View(Bankdata)
set.seed(15)
```

##Partition the data into training (60%) and validation (40%) sets.

```
Train_Index <- createDataPartition(Bankdata$Personal.Loan, p=0.6, list=FAL
SE)
Train <-Bankdata[Train_Index,]
Valid <- Bankdata[-Train_Index,]
```

#{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.

```
Melt_Train <- melt(Train,id=c("CreditCard","Personal.Loan"),variable= "Onl
ine")

## Warning: attributes are not identical across measure variables; they wi
ll be dropped

cast_Train <- dcast(Melt_Train,CreditCard+Personal.Loan~Online)

## Aggregation function missing: defaulting to length

cast_Train <-cast_Train[c(1,2,14)]
cast_Train

##   CreditCard Personal.Loan Online
## 1          0             0   1903
## 2          0             1    209
## 3          1             0    798
## 4          1             1     90

View(cast_Train)
```

#{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?

##Answer - The probability of the customer accepting the loan is 90/888 =0.10135

##{C} Create two separate pivot tables for the training data. One will have Loan (rows) as afunction of Online (columns) and the other will have Loan (rows) as a function of CC.

```
#pivot table1
Melt_Train1 <- melt(Train,id=c("Personal.Loan"),variable= "Online")
```

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

cast_Train1 <- dcast(Melt_Train1,Personal.Loan~Online)

## Aggregation function missing: defaulting to length

cast_Train1 <-cast_Train1[c(1,13)]
cast_Train1

##   Personal.Loan Online
## 1             0   2701
## 2             1    299

#pivot table2
Melt_Train2 <- melt(Train,id=c("CreditCard"),variable= "Online")

## Warning: attributes are not identical across measure variables; they wi
ll be dropped

cast_Train2 <- dcast(Melt_Train2,CreditCard ~ Online)

## Aggregation function missing: defaulting to length

cast_Train2 <-cast_Train2[c(1,14)]
cast_Train2

##   CreditCard Online
## 1          0   2112
## 2          1    888
```

##{D} Compute the following quantities [P(A | B) means "the probability ofA given B"]: i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) ii. P(Online = 1 | Loan = 1) iii. P(Loan = 1) (the proportion of loan acceptors) iv. P(CC = 1 | Loan = 0) v. P(Online = 1 | Loan = 0)vi. P(Loan = 0)

```
Train_Data <- Train[c(13,10,14)]
table(Train_Data[,c(3,2)])

##           Personal.Loan
## CreditCard    0    1
##          0 1903  209
##          1  798   90

table(Train_Data[,c(1,2)])

##       Personal.Loan
## Online    0    1
##      0 1094  119
##      1 1607  180

table(Train_Data[,c(2)])

##
##    0    1
## 2701  299
```

i) P(CC=1 | PL=1) = 90/(798+90)= 0.10135

ii) P(OL=1 | PL=1) = 180/(180+119)=0.60200

iii) P(PL=1) = (209+90)/3000=0.09966

iv) P(CC=1 | PL=0) = 798/(798+1903)=0.29544

v) P(OL=1 | PL=0) = 1607/(1607+1094)=0.59496

vi) P(PL=0) = (1903+798)/3000=0.90033

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

=(0.10135*0.60266*0.09966)/((0.10135*0.60266*0.09966)+(0.29544*0.59496*0.90033))

= (0.00608052)/(0.00608052+0.15825548)

= 0.03700053

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

ANS:- Part B = 0.10135 and Part E = 0.03700053 by comparing both the values part B is more accurate.

#{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).

```
train.df <- Bankdata[Train_Index, ]
test.df <- Bankdata[ -Train_Index,]
train <- Bankdata[Train_Index, ]
test <- Bankdata[-Train_Index,]
nb_train = train.df[,c(10,13:14)]
nb_test = test.df[,c(10,13:14)]
naivebayes = naiveBayes(Personal.Loan~.,data=nb_train)
naivebayes

##
## 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:
##    Online
## Y           0         1
##   0 0.4050352 0.5949648
```

```
##    1 0.3979933 0.6020067
##
##     CreditCard
## Y           0         1
##    0 0.7045539 0.2954461
##    1 0.6989967 0.3010033
```