Assignment 3

#Packages loaded library(class) library(dplyr) library(tidyverse) library(ISLR) library(dummies) library(caret) library(dcast) library(pivottabler) library(reshape) library(e1071) library(naivebayes) library(klaR) library(bnclassify) library(rmarkdown) library(tinytex)

#Reading the Universal Bank file

```
Myfile <- read.csv("UniversalBank (1).csv")

head(Myfile)
```

```
##   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
```

```
Myfile$Personal.Loan =as.factor(Myfile$Personal.Loan)
Myfile$CCAvg =as.factor(Myfile$CCAvg)
Myfile$Online =as.factor(Myfile$Online)
```

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

```
set.seed(1)
train_index = sample(row.names(Myfile),0.6*dim(Myfile))
test.index <- setdiff(row.names(Myfile),train_index)

train_Data = Myfile[train_index,]
Validation_Data= Myfile[test.index,]
head(train_Data)
```

```
##        ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1017 1017  30          5     69    94720      1   0.8         2        0
## 4775 4775  56         32     22    91768      1   1.2         3        0
## 2177 2177  41         14     51    91320      3  2.33         2        0
## 1533 1533  45         20     55    94588      1   0.3         1        0
## 4567 4567  24          0    131    92831      1   5.4         1        0
## 2347 2347  52         26     59    92660      2   1.5         2      239
```

```
##      Personal.Loan Securities.Account CD.Account Online CreditCard
## 1017             0                  1          0      1          0
## 4775             0                  0          0      1          1
## 2177             0                  0          0      1          0
## 1533             0                  0          0      1          1
## 4567             0                  0          0      1          0
## 2347             0                  0          0      0          1
```

#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

```
library(maditr)

##
## To get total summary skip 'by' argument: take_all(mtcars, mean)

library(reshape)

##
## Attaching package: 'reshape'

## The following object is masked from 'package:maditr':
##
##     melt

library(reshape2)

##
## Attaching package: 'reshape2'

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

## The following objects are masked from 'package:maditr':
##
##     dcast, melt

library(ggplot2)
melted.bank = melt(train_Data,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   1924
## 2          0             1    198
## 3          1             0    801
## 4          1             1     77
```

#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)].

```
#Probability of customer accepting loan offer= .025 = 2.56%
77/(1924+198+801+77)
```

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

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

```
#pivot table for training data that has loan as a function of Online
melted.bank1 = melt(train_Data,id=c("Personal.Loan"),variable = "Online")
```

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

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

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

```
Loan=recast.bank1[,c(1,13)]
Loan
```

```
##   Personal.Loan Online
## 1             0   2725
## 2             1    275
```

```
#pivot table for training data that has loan as a function of CC
melted.bank2 = melt(train_Data,id=c("Personal.Loan"),variable = "CreditCard")
```

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

```
recast.bank2=dcast(melted.bank2,Personal.Loan~CreditCard)
```

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

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

```
##   Personal.Loan CreditCard
## 1             0       2725
## 2             1        275
```

#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)

#i

```
#P(CC = 1 | Loan = 1)
table(train_Data[,c(14,10)])
```

```
##            Personal.Loan
## CreditCard    0    1
##          0 1924  198
##          1  801   77
```

```
#Probability of P(CC = 1 | Loan = 1) = 77/(77+198) = .28 = 28 % is the
probability of the Credit Card being 1 and the Personal loan being 1
77/(77+198)
```

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

#ii

```
#P(Online = 1 | Loan = 1)
table(train_Data[,c(13,10)])
```

```
##        Personal.Loan
## Online    0    1
##      0 1137  109
##      1 1588  166
```

```
#P(Online = 1 | Loan = 1) = 166/(166+109) = .6036 = 60.36%
166/(166+109)
```

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

#iii

```
#P(Loan = 1) (the proportion of loan acceptors)
table(train_Data[,c(10)])
```

```
##
##    0    1
## 2725  275
```

```
#P(Loan = 1)= 279/ (279+2721) = .091 = 9.1%
275/ (275+2725)
```

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

#iv

```
#P(CC = 1 | Loan = 0)
table(train_Data[,c(14,10)])
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1924  198
##          1  801   77
```

```
#P(CC = 1 | Loan = 0) =  801/ (801+1924) = .2939 = 29.39%
801/ (801+1924)
```

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

#v

```
#P(Online = 1 | Loan = 0)
table(train_Data[,c(13,10)])
```

```
##       Personal.Loan
## Online    0    1
##      0 1137  109
##      1 1588  166
```

```
#P(Online = 1 | Loan = 0) = 1588/ (1588+1137) = .5827 = 58.27%
1588/ (1588+1137)
```

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

#vi

```
#P(Loan = 0)
table(train_Data[,c(10)])
```

```
##
##    0    1
## 2725  275
```

```
#P(Loan = 0) = 2725 / (2725+275) = .908 = 90.8%
2725 / (2725+275)
```

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

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

```
((77/(77+198)) * (166/(166+109)) * (275/(275+2725))) / (((77/(77+198)) *
(166/(166+109)) * (275/(275+2725))) + ((801/(801+1924)) * (1588/(1588+1137))
* 2725/(2725+275)))
```

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

```
#probability is .0905 = 9.05%
```

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

```
#Comparing the value above, 9.05 % to the value I found in the pivot table which was 2.56%, I think the naive bayes calculation done above is more accurate estimate.
```

#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).

```
library(e1071)
library(naivebayes)

## naivebayes 0.9.7 loaded

library(dplyr)

##
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:maditr':
##
##     between, coalesce, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

naive.traindata = train_Data[,c(10,13:14)]
naivebayes= naiveBayes(Personal.Loan~.,data=naive.traindata)
naivebayes

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90833333 0.09166667
```

```
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4172477 0.5827523
##   1 0.3963636 0.6036364
##
##    CreditCard
## Y       [,1]      [,2]
##   0 0.293945 0.4556506
##   1 0.280000 0.4498175
```

#Compare this to the number you obtained in (E).

```
#The value I found above for the naive bayes probability, is .0916 or 9.16% .
The number I found in E, was .0905 or 9.05%. The two values are close, only
off by .11 %.
```