# zalshaye_3

```
rm(list=ls())
setwd("C:/Users/Z/Desktop/Fall 2021/ML/Assignment 3")

bank = read.csv("UniversalBank.csv")
bank$Personal.Loan = as.factor(bank$Personal.Loan)
bank$Online = as.factor(bank$Online)
bank$CreditCard = as.factor(bank$CreditCard)
set.seed(1)
train.index <- sample(row.names(bank), 0.6*dim(bank)[1])
test.index <- setdiff(row.names(bank), train.index)
train.df <- bank[train.index, ]
test.df <- bank[test.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   1924
## 2          0             1    198
## 3          1             0    801
## 4          1             1     77

melted.bankc1 = melt(train,id=c("Personal.Loan"),variable = "Online")

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

melted.bankc2 = melt(train,id=c("CreditCard"),variable = "Online")

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

recast.bankc1=dcast(melted.bankc1,Personal.Loan~Online)
```

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

recast.bankc2=dcast(melted.bankc2,CreditCard~Online)

## Aggregation function missing: defaulting to length

Loanline=recast.bankc1[,c(1,13)]
LoanCC = recast.bankc2[,c(1,14)]

Loanline

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

LoanCC

##   CreditCard Online
## 1          0   2122
## 2          1    878

table(train[,c(14,10)])

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

table(train[,c(13,10)])

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

table(train[,c(10)])

##
##    0    1
## 2725  275

((77/(77+198))*(166/(166+109))*(275/(275+2725)))/(((77/(77+198))*(166/(166+10
9))*(275/(275+2725)))+((801/(801+1924))*(1588/(1588+1137))*2725/(2725+275)))

## [1] 0.09055758

naive.train = train.df[,c(10,13:14)]
naive.test = test.df[,c(10,13:14)]
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train)
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          0        1
##   0 0.706055 0.293945
##   1 0.720000 0.280000
```