R Notebook

```
#setwd("~/R_KSU/ML/Assignment 3")
bank <- read.csv("UniversalBank.csv")
```

```
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 4.0.3
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.0.3
```

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

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

```
str(bank)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```
summary(bank)
```

```
##        ID            Age          Experience       Income          ZIP.Code
##  Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9307
##  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91911
##  Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93437
##  Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93153
##  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94608
##  Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96651
##      Family          CCAvg          Education        Mortgage
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0
##  Median :2.000   Median : 1.500   Median :2.000   Median :  0.0
##  Mean   :2.396   Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :635.0
##  Personal.Loan   Securities.Account   CD.Account         Online
##  Min.   :0.000   Min.   :0.0000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000     Median :0.0000   Median :1.0000
##  Mean   :0.096   Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
##  3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
##    CreditCard
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.294
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
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), 0.7*dim(bank)[1])
test.index <- setdiff(row.names(bank), train.index)
train <- bank[train.index, ]
test <- bank[test.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. In R use functions melt() and cast(), or function table(). In Python, use panda dataframe methods melt() and pivot().

```
table("CC"=bank$CreditCard,"PL"=bank$Personal.Loan,"O/L"=bank$Online)
```

```
## , , O/L = 0
##
##    PL
## CC     0    1
##   0 1300  128
##   1  527   61
##
## , , O/L = 1
##
##    PL
## CC     0    1
##   0 1893  209
##   1  800   82
```

```
t1= recast(bank,bank$CreditCard+bank$Personal.Loan~bank$Online)
```

```
## Using Personal.Loan, Online, CreditCard as id variables
```

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

```
t1
```

bank$CreditCard <fct>	**bank$Personal.Loan** <fct>	**0** <int>	**1** <int>
0	0	1300	1893
0	1	128	209
1	0	527	800
1	1	61	82

4 rows

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)].

> Probability of Loan acceptance given having a bank credit card and user of online services is 82/882 = 0.09297

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.

```
t2= recast(bank,bank$Personal.Loan~bank$Online)
```

```
## Using Personal.Loan, Online, CreditCard as id variables
```

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

```
t2
```

bank$Personal.Loan <fct>	0 <int>	1 <int>
0	1827	2693
1	189	291

2 rows

```
t3= recast(bank,bank$CreditCard~bank$Online)
```

```
## Using Personal.Loan, Online, CreditCard as id variables
```

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

```
t3
```

bank$CreditCard <fct>	0 <int>	1 <int>
0	1428	2102
1	588	882

2 rows

D. Compute the following quantities [P(A | B) means “the probability of A 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)

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

```
##            Personal.Loan
## CreditCard    0    1
##          0 2241  232
##          1  933   94
```

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

```
##       Personal.Loan
## Online    0    1
##      0 1304  132
##      1 1870  194
```

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

```
##
##    0    1
## 3174  326
```

P(Cc|Pl) = 94/(94+232) = 0.28834 P(Ol|Pl) = 194/(194+132) = 0.59509 P(Pl) = 326/(326+3174) = 0.09314
P(Cc|Pl')= 933/(933+2241) = 0.29395 P(Ol|Pl')= 1870/(1870+1304)= 0.58916 P(Pl') = 3174/(3174+326) = 0.90685

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

```
((94/(94+232))*(194/(194+132))*(326/(326+3174)))/(((94/(94+232))*(194/(194+132))*(326/(326+3174
)))+((933/(933+2241))*(1870/(1870+1304))*3174/(3174+326)))
```

```
## [1] 0.09236489
```

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

> 9.23% are very similar to the 9.297% the difference between the exact method and the naive-bayes method is the exact method would need the the exact same independent variable classifications to predict, where the naive bayes method does not.

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).

```
library('e1071')
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

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

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90685714 0.09314286
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4108381 0.5891619
##   1 0.4049080 0.5950920
##
##    CreditCard
## Y           0         1
##   0 0.7060491 0.2939509
##   1 0.7116564 0.2883436
```

The naive bayes is the exact same output we retrieved in the previous methods. (0.288)*(0.595)*(0.093)/((0.288)*(0.595)*(0.093) + (0.293)*(0.589)*(0.906) = .0089 which is almost the same response provided as above.