Assignment 3

```
library(readr)
UniversalBank <- read_csv("UniversalBank.csv")
```

```
## Rows: 5000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education, M...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(UniversalBank)
```

Set up train and test data frames

```
UniversalBank$'Personal Loan' = as.factor(UniversalBank$'Personal Loan')
UniversalBank$Online = as.factor(UniversalBank$Online)
UniversalBank$CreditCard = as.factor(UniversalBank$CreditCard)
set.seed(1)
train_index <- sample(row.names(UniversalBank), 0.6*dim(UniversalBank)[1])
test_index <- setdiff(row.names(UniversalBank), train_index)
train.df <- UniversalBank[train_index, ]
test.df <- UniversalBank[test_index, ]
train <- UniversalBank[train_index, ]
test = UniversalBank[train_index,]
```

A. Create a pivot table for the training data with Online as a column variable, Credit Card as a row variable, and Personal 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().

```
library(reshape2)
melted<- melt(train, id.vars = c("CreditCard", "Personal Loan"), variable.name = "Online")
```

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

```
recast <- dcast(melted, CreditCard + `Personal Loan` ~ Online)
```

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

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

```
##   CreditCard Personal Loan Online
## 1          0             0   1924
## 2          0             1    198
## 3          1             0    801
## 4          1             1     77
```

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 (Personal Loan = 1) conditional on having a bank credit card (CreditCard = 1) and being an active user of online banking services (Online = 1)].

```r
# The probability that a customer that owns a credit card and actively uses online banking services is
77/(77+801+198+1924)*100
```

```
## [1] 2.566667
```

```r
# 2.6%
```

C. Create two separate pivot tables for the training data. One will have Personal Loan (rows) as a function of Online (columns) and the other will have Personal Loan (rows) as a function of CreditCard.

```r
# Pivot table with training data. Personal Loan as a function of Online
melted_1 <- melt(train, id.vars = c("Personal Loan"), variable.name = "Online")
```

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

```r
recast_1=dcast(melted_1,'Personal Loan'~Online)
```

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

```r
print(recast_1[,c(1,13)])
```

```
##   "Personal Loan" Online
## 1   Personal Loan   3000
```

```r
# Pivot table with training data. Personal Loan as a function of CreditCard
melted_2 = melt(train,id=c("CreditCard"),variable = "Online")
```

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

```r
recast_2=dcast(melted_2,CreditCard~Online)
```

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

```r
print(recast_2[,c(1,14)])
```

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

D. Compute the following quantities [P(A | B) means "the probability of A given B"]: i. P(CreditCard = 1 | Personal Loan = 1)

```r
# Proportion of credit card holders among the loan acceptors
table(train[,c("CreditCard",'Personal Loan')])
```

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

```r
77/(77+198)*100
```

```
## [1] 28
```

```r
# probability of 28% that credit card users accept personal loan
```

ii. P(Online = 1 | Personal Loan = 1)

```r
# Probability of Online users given personal loan acceptors
table(train[,c("Online","Personal Loan")])
```

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

```r
166/(166+109)*100 # = 60.36%
```

```
## [1] 60.36364
```

iii. P(Personal Loan = 1) (the proportion of loan acceptors)
iv. P(Personal Loan = 0)

```r
table(train[,c("Personal Loan")])
```

```
## Personal Loan
##    0    1
## 2725  275
```

```r
275/3000*100 # proportion of loan acceptors = 9.17%
```

```
## [1] 9.166667
```

```r
2725/3000*100 # proportiong of non loan acceptros = 90.83%
```

```
## [1] 90.83333
```

iv. P(CreditCard = 1 | Personal Loan = 0)

```r
table(train[,c("CreditCard",'Personal Loan')])
```

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

```r
801/(1924+801)*100 # = 29.40%
```

```
## [1] 29.3945
```

v. P(Online = 1 | Personal Loan = 0)

```r
table(train[,c("Online","Personal Loan")])
```

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

```r
1588/(1588+1137)*100 # = 58.27%
```

```
## [1] 58.27523
```

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

```r
((77/(77+198))*(166/(166+109))*(275/(275+2725)))/(((77/(77+198))*(166/(166+109))*(275/(275+2725)))+((80
```

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

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

```r
# 9.05 vs 2.57

# the vaule obtained from the pivot table is less 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).

```r
# Personal Loan, Online and CreditCard are needed to compute P(Loan = 1 | CC = 1, Online = 1)
# Run naive Bayes on the data
library(e1071)
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
```

```r
nb_train <- select(train.df, "Personal Loan", Online:CreditCard)
nb_test <- select(test.df, "Personal Loan", Online:CreditCard)
nb_model <- naiveBayes(`Personal Loan` ~ ., data = nb_train)
print(nb_model)
```

```
##
## 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
```

```r
# Entry that corresponds to P(Loan = 1 | CC = 1, Online = 1)
# 0.091667 = 9.17%

# Compare to number obtained in E = 9.05%
# 9.05% vs 9.17% both numbers are really close which means our model is accurate
```