Stat_Markdown

2023-06-25

Import Libraries We Need

```
library("dplyr")
library("corrplot")
library("caTools")
library("ggpubr")
library("ROSE")
library("correlation")
library("moments") #to calculate skewness
library("olsrr") #to use ols_step_backward_p
library("MASS")
library("knitr")
library("forecast")
library("ggplot2")
library("PCAmixdata")
library("purrr")
library("corpcor")
library("car")
library("e1071")
library("ppcor")
library("pROC")
library("interactions")
library("glmnet")
library("formattable") # for giving a variable dictionary a better look
library("RColorBrewer")# for the visualization colorings
library("yarrr") #to make colors transparent
library("regclass")
```

Introduction to Data

In this project we are going to predict the probability of a customer attrition. We have a data containing demographic information about customers and their spending behavior. We downloaded the publicly available dataset from Kaggle which can be found in the link below. https://www.kaggle.com/datasets/thedevastator/predicting-credit-card-customer-attrition-with-m

```
bank_data_origin <- read.csv('~/GitHub/Stats_23_Project/BankChurners.csv')
head(bank_data_origin)
```

```
##   CLIENTNUM    Attrition_Flag Customer_Age Gender Dependent_count
## 1 768805383 Existing Customer           45      M               3
```

```
## 2 818770008 Existing Customer           49      F              5
## 3 713982108 Existing Customer           51      M              3
## 4 769911858 Existing Customer           40      F              4
## 5 709106358 Existing Customer           40      M              3
## 6 713061558 Existing Customer           44      M              2
##   Education_Level Marital_Status Income_Category Card_Category Months_on_book
## 1     High School        Married     $60K - $80K          Blue             39
## 2        Graduate         Single  Less than $40K          Blue             44
## 3        Graduate        Married    $80K - $120K          Blue             36
## 4     High School        Unknown  Less than $40K          Blue             34
## 5      Uneducated        Married     $60K - $80K          Blue             21
## 6        Graduate        Married     $40K - $60K          Blue             36
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
## 1                        5                      1                     3
## 2                        6                      1                     2
## 3                        4                      1                     0
## 4                        3                      4                     1
## 5                        5                      1                     0
## 6                        3                      1                     2
##   Credit_Limit Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1
## 1        12691                 777           11914                1.335
## 2         8256                 864            7392                1.541
## 3         3418                   0            3418                2.594
## 4         3313                2517             796                1.405
## 5         4716                   0            4716                2.175
## 6         4010                1247            2763                1.376
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 1            1144             42               1.625                 0.061
## 2            1291             33               3.714                 0.105
## 3            1887             20               2.333                 0.000
## 4            1171             20               2.333                 0.760
## 5             816             28               2.500                 0.000
## 6            1088             24               0.846                 0.311
##   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatio
## 1
## 2
## 3
## 4
## 5
## 6
##   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educatio
## 1
## 2
## 3
## 4
## 5
## 6
```

Before diving into modelling, it is important to understand the nature of the set. Here is the summary of the dataset:

```
summary(bank_data_origin)
```

```
##    CLIENTNUM         Attrition_Flag      Customer_Age      Gender
```

```
##  Min.   :708082083   Length:10127       Min.   :26.00   Length:10127
##  1st Qu.:713036770   Class :character   1st Qu.:41.00   Class :character
##  Median :717926358   Mode  :character   Median :46.00   Mode  :character
##  Mean   :739177606                      Mean   :46.33
##  3rd Qu.:773143533                      3rd Qu.:52.00
##  Max.   :828343083                      Max.   :73.00
##  Dependent_count Education_Level    Marital_Status     Income_Category
##  Min.   :0.000   Length:10127       Length:10127       Length:10127
##  1st Qu.:1.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2.346
##  3rd Qu.:3.000
##  Max.   :5.000
##  Card_Category      Months_on_book  Total_Relationship_Count
##  Length:10127       Min.   :13.00   Min.   :1.000
##  Class :character   1st Qu.:31.00   1st Qu.:3.000
##  Mode  :character   Median :36.00   Median :4.000
##                     Mean   :35.93   Mean   :3.813
##                     3rd Qu.:40.00   3rd Qu.:5.000
##                     Max.   :56.00   Max.   :6.000
##  Months_Inactive_12_mon Contacts_Count_12_mon  Credit_Limit
##  Min.   :0.000          Min.   :0.000         Min.   : 1438
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555
##  Median :2.000          Median :2.000         Median : 4549
##  Mean   :2.341          Mean   :2.455         Mean   : 8632
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068
##  Max.   :6.000          Max.   :6.000         Max.   :34516
##  Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
##  Min.   :   0        Min.   :    3   Min.   :0.0000       Min.   :  510
##  1st Qu.: 359        1st Qu.: 1324   1st Qu.:0.6310       1st Qu.: 2156
##  Median :1276        Median : 3474   Median :0.7360       Median : 3899
##  Mean   :1163        Mean   : 7469   Mean   :0.7599       Mean   : 4404
##  3rd Qu.:1784        3rd Qu.: 9859   3rd Qu.:0.8590       3rd Qu.: 4741
##  Max.   :2517        Max.   :34516   Max.   :3.3970       Max.   :18484
##  Total_Trans_Ct   Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
##  Min.   : 10.00   Min.   :0.0000      Min.   :0.0000
##  1st Qu.: 45.00   1st Qu.:0.5820      1st Qu.:0.0230
##  Median : 67.00   Median :0.7020      Median :0.1760
##  Mean   : 64.86   Mean   :0.7122      Mean   :0.2749
##  3rd Qu.: 81.00   3rd Qu.:0.8180      3rd Qu.:0.5030
##  Max.   :139.00   Max.   :3.7140      Max.   :0.9990
##  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
##  Min.   :0.0000077
##  1st Qu.:0.0000990
##  Median :0.0001815
##  Mean   :0.1599975
##  3rd Qu.:0.0003373
##  Max.   :0.9995800
##  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
##  Min.   :0.00042
##  1st Qu.:0.99966
##  Median :0.99982
##  Mean   :0.84000
##  3rd Qu.:0.99990
```

```
## Max.   :0.99999
```

As expected, the set we have contains both numerical and categorical variables. Hence, a direct use without cleaning the set would lead us to an inappropriate model. In the beginning we have 23 columns and 10127 rows. However, this dataset was part of a study that had the same aim with a different approach.This approach was Naive Bayes Classification and 2 columns named as "Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Lev and "Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_ were the results of their work. Thus, removing them will not only make the dataset more clear, but also will increase the originality of the work done in this project. We will start our project by removing these 2 columns which is independent from others. Finally, we have 21 columns and 10127 rows in the beginning.

```
dim(bank_data_origin)
```

```
## [1] 10127    23
```

```
bank_data <- subset(bank_data_origin, select = -c(Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Co
final_dim <- dim(bank_data)
final_dim
```

```
## [1] 10127    21
```

The columns that is going to be present in the study can be found in the table below next to their descriptions.

descriptions

Clientnum

refers to the distinct identification numbers assigned to customers, consisting of a unique sequence of 9 digits. The datasets contain a total of 10,127 customers with unique IDs.

Attrition_Flag

refers to the current status of customers, indicating whether they are Existing Customers (current customers) or Attrited Customers (churned customers). There are two distinct values for this target/output variable.

Customer_Age

represents the age of customers, with a range between 27 and 73.

Gender

is encoded as 'F' for Female and 'M' for Male.

Dependent_count

represents the number of dependents associated with a customer.

Education_Level

represents the educational qualification of a customer. It encompasses seven distinct values: High School, Graduate, Uneducated, College, Post-graduate, Doctorate, and Unknown. The Unknown category includes 1519 customers.

Marital_Status

represents the marital status of customers, with four unique values: Married, Single, Unknown, and Divorced. The Unknown category includes 749 customers.

Income_Category

represents the annual income category of cardholders: Less than 40K, 40K-60K, 60K-80K, 80K-120K, $120+, and Unknown. The Unknown category includes 1112 customers.

Card_Category

refers to a product variable that indicates the type of credit card held by customers. It includes four unique values: Blue, Gold, Silver, and Platinum.

Months_on_book

represents the duration, in months, that an account holder has been a customer at the bank.

Total_Relationship_Count

represents the number of products held by a customer.

Months_Inactive_12_mon

represents the number of months during which a customer has been inactive in the last 12 months (1 year).

Contacts_Count_12_mon

represents the number of times a customer has contacted the bank.

Credit_Limit

represents the credit limit on the customer's credit card.

Total_Revolving_Bal

represents the total revolving balance on the customer's credit card.

Avg_Open_To_Buy

represents the average Open to Buy Credit Line for the last 12 months.

Total_Amt_Chng_Q4_Q1

represents the change in transaction amount from the fourth quarter (Q4) to the first quarter (Q1).

Total_Trans_Amt

represents the total transaction amount in the last 12 months.

Total_Trans_Ct

represents the total transaction count in the last 12 months.

Total_Ct_Chng_Q4_Q1

represents the change in transaction count from the fourth quarter (Q4) to the first quarter (Q1).

Avg_Utilization_Ratio

represents the average card utilization ratio.

Data Exploration

After eliminating the columns, our data has 14 numerical and 7 categorical columns. Below is the display of the categorical variables.

```
#display of the categorical variables
table(bank_data$Attrition_Flag)
```

```
## 
## Attrited Customer Existing Customer 
##              1627              8500
```

```
table(bank_data$Gender)
```

```
## 
##    F    M 
## 5358 4769
```

```
table(bank_data$Education_Level)
```

```
## 
##       College     Doctorate      Graduate   High School Post-Graduate 
##          1013           451          3128          2013           516 
##    Uneducated       Unknown 
##          1487          1519
```

```
table(bank_data$Marital_Status)
```

```
## 
## Divorced  Married   Single  Unknown 
##      748     4687     3943      749
```

```
table(bank_data$Income_Category)
```

```
## 
##        $120K +    $40K - $60K    $60K - $80K   $80K - $120K Less than $40K 
##            727           1790           1402           1535           3561 
##        Unknown 
##           1112
```

```
table(bank_data$Card_Category)
```

```
## 
##     Blue     Gold Platinum   Silver 
##     9436      116       20      555
```

It has been noticed that some of the categorical columns (Education Level, Marital Status and Income Category) have unknown values but it is not in a format that can be detectable by R. In order to fix this, all the "Unknown" values are turned into NA.

```
#Change Unknown value to NA
bank_data_NA <- data.frame(bank_data)
bank_data_NA[bank_data_NA=='Unknown'] <- NA
```

After the transformation, removing these unknown data is easier.

```r
#Build a dataset without missing values
bank_data_withoutNA <- na.omit(bank_data_NA)
```

In order to show that the cleaning was successful and there is no null value inside our data we added a confirmation step here.

```r
colSums(is.na(bank_data_withoutNA))
```

```
##                CLIENTNUM           Attrition_Flag             Customer_Age
##                        0                        0                        0
##                   Gender          Dependent_count          Education_Level
##                        0                        0                        0
##           Marital_Status          Income_Category            Card_Category
##                        0                        0                        0
##           Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                        0                        0                        0
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                        0                        0                        0
##          Avg_Open_To_Buy     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt
##                        0                        0                        0
##           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                        0                        0                        0
```

For the 6 categorical columns, we change their data form to numeric to be able to implement a model. However, at this point is beneficiary to remind that these variables will stay as categorical type.

```r
bank_data_withoutNA_quan <- bank_data_withoutNA

bank_data_withoutNA_quan$Attrition_Flag <- as.numeric(bank_data_withoutNA_quan$Attrition_Flag == "Attrit

bank_data_withoutNA_quan$Gender <- as.numeric(bank_data_withoutNA_quan$Gender == "F")
bank_data_withoutNA_quan <- bank_data_withoutNA_quan %>% rename("Is_Female" = "Gender")

order_education_level <- list("Uneducated" = 1,
                              "High School" = 2,
                              "College" = 3,
                              "Graduate" = 4,
                              "Post-Graduate" = 5,
                              "Doctorate" = 6)
bank_data_withoutNA_quan$Education_Level <- unlist(order_education_level[as.character(bank_data_without

order_Marital_Status <- list("Single" = 1,
                             "Married" = 2,
                             "Divorced" = 3)
bank_data_withoutNA_quan$Marital_Status <- unlist(order_Marital_Status[as.character(bank_data_withoutNA_

order_Income_Category <- list("Less than $40K" = 1,
                              "$40K - $60K" = 2,
                              "$60K - $80K" = 3,
                              "$80K - $120K" = 4,
                              "$120K +" = 5)
bank_data_withoutNA_quan$Income_Category <- unlist(order_Income_Category[as.character(bank_data_without
```

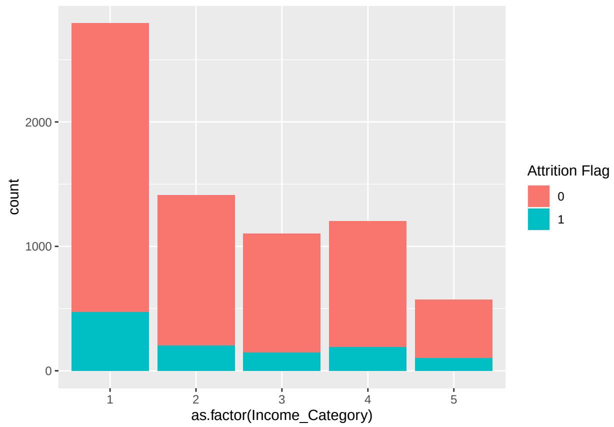
```r
order_Card_Category <- list("Blue" = 1,
                            "Silver" = 2,
                            "Gold" = 3,
                            "Platinum" = 4)
bank_data_withoutNA_quan$Card_Category <- unlist(order_Card_Category[as.character(bank_data_withoutNA_qu
```

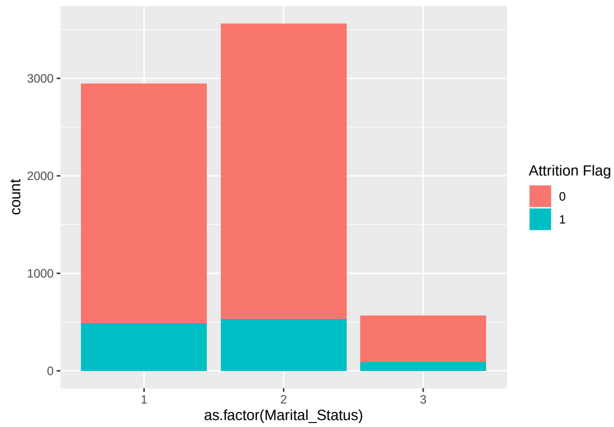
```r
colSums(is.na(bank_data_withoutNA_quan))
```

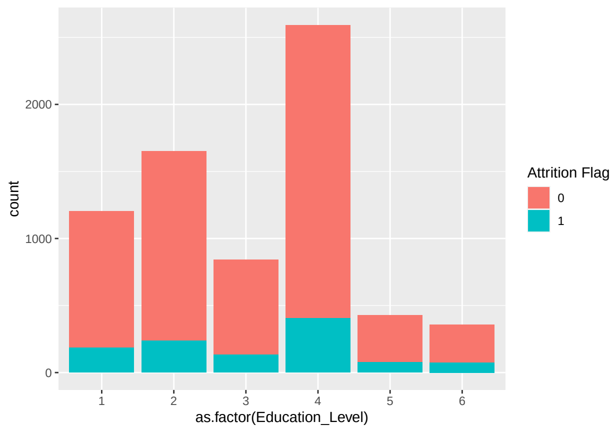
```
##                CLIENTNUM           Attrition_Flag             Customer_Age
##                        0                        0                        0
##                Is_Female          Dependent_count          Education_Level
##                        0                        0                        0
##           Marital_Status          Income_Category            Card_Category
##                        0                        0                        0
##           Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                        0                        0                        0
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                        0                        0                        0
##          Avg_Open_To_Buy      Total_Amt_Chng_Q4_Q1          Total_Trans_Amt
##                        0                        0                        0
##           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                        0                        0                        0
```

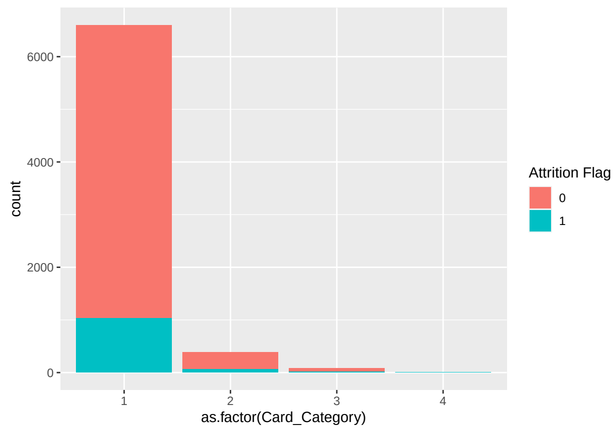
Here is the visualizations of categorical values regarding our target value, attrition flag.

```r
par(mar = c(4, 4, .1, .1))
#Categorical Value Visualizations

#Bar plots
par(mfrow=c(2,2))
myPalette <- brewer.pal(6, "Set2")
ggplot(bank_data_withoutNA_quan, aes(x = as.factor(Income_Category), fill = factor(Attrition_Flag))) + g
ggplot(bank_data_withoutNA_quan, aes(x = as.factor(Marital_Status), fill = factor(Attrition_Flag))) + ge
ggplot(bank_data_withoutNA_quan, aes(x = as.factor(Education_Level), fill = factor(Attrition_Flag))) + g
ggplot(bank_data_withoutNA_quan, aes(x = as.factor(Card_Category), fill = factor(Attrition_Flag))) + geo
```



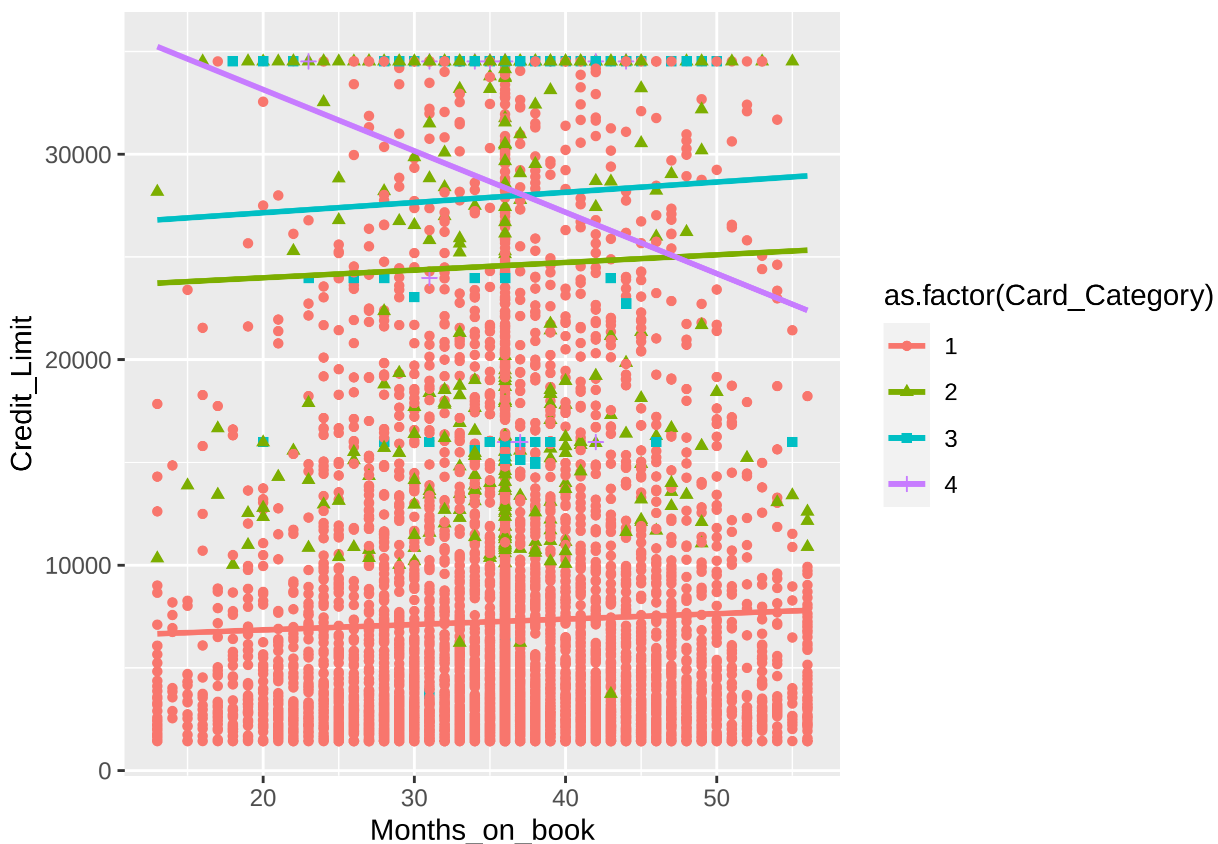






There seems to be a great imbalance on card category, while the others distributions seem to be reasonably similar. Before moving further, card categories' relationship with few other numerical values might be important to understand if card category is going to affect the performance of the model. Below, the relationship between card category and numerical columns that is expected to be decisive in churn out behavior (credit limit and months on book) is presented.

```
# Card Category

ggplot(bank_data_withoutNA_quan, aes(x=Months_on_book, y= Credit_Limit, shape = as.factor(Card_Category)
  geom_point() + geom_smooth(method=lm, se=FALSE, fullrange=TRUE)
```

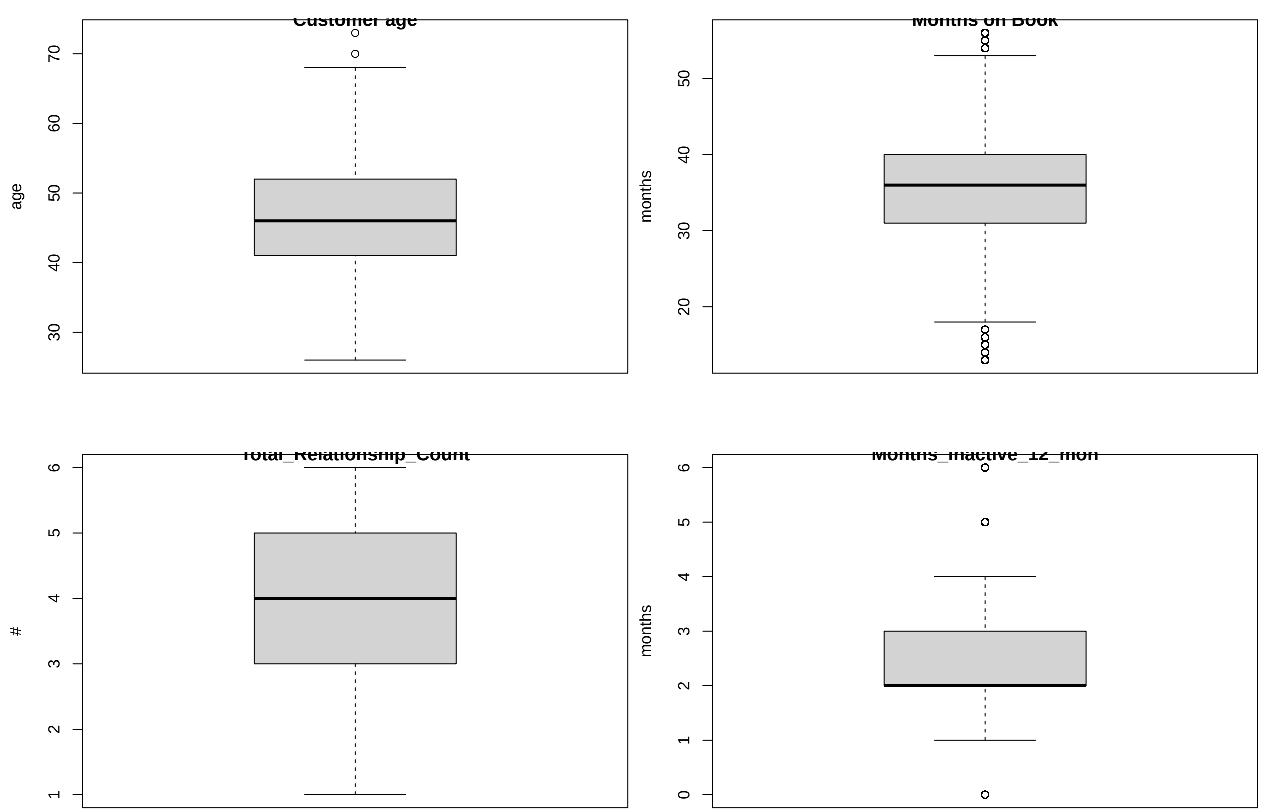


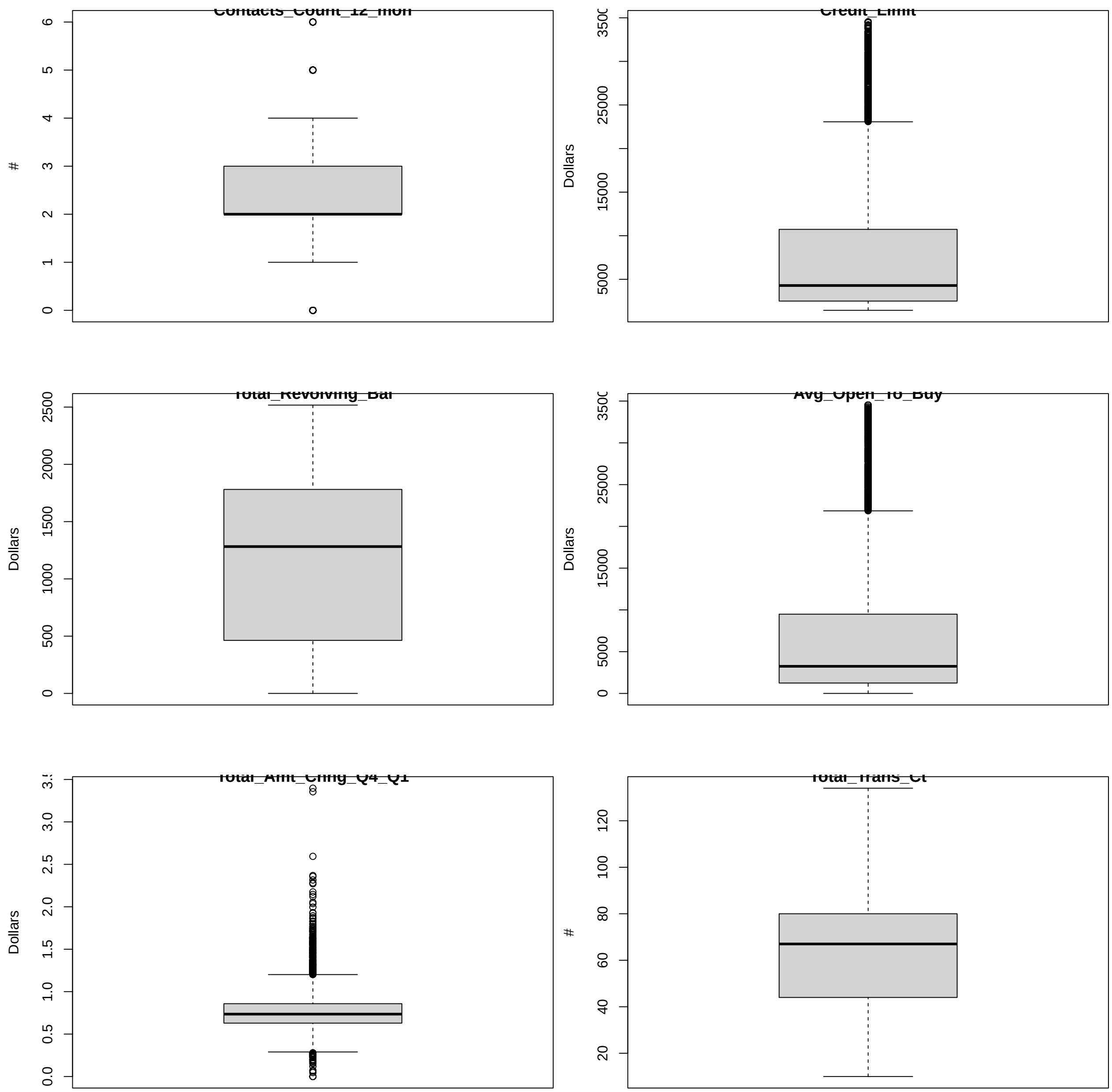
It is clear that blue card holders have different attributes than the others while the other card holders seem to be acting similar. Thus, we decided to rescope our study only to focus on blue card holders. By doing so, we believe that numerical values fed into the model will have more accurate effect on the model.

```
#Since most of the data is coming from the Blue cards and there is a visible difference on many paramet
card.exc.list <- c(2, 3, 4)
```

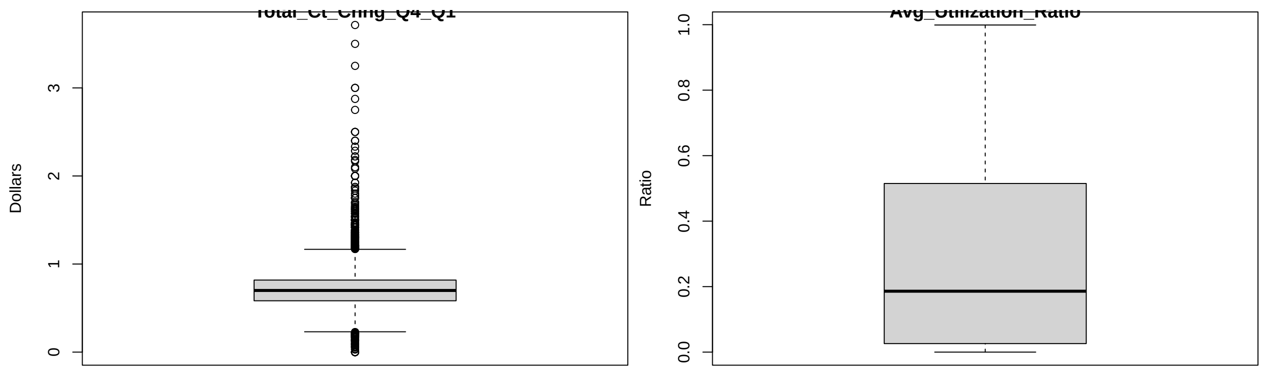
At this point, the numerical columns are decided to be checked whether they have any

```
par(mar = c(4, 4, .1, .1))
#Numerical columns analysis
attach(bank_data_withoutNA_quan)

cust.age.boxplot <- boxplot(Customer_Age, main="Customer age", ylab = "age")
months.onbook.boxplot <- boxplot(Months_on_book, main="Months on Book", ylab = "months")
reltn.cnt.boxplot <- boxplot(Total_Relationship_Count, main="Total_Relationship_Count", ylab = "#")
mnths.inact.boxplot <- boxplot(Months_Inactive_12_mon,  main="Months_Inactive_12_mon", ylab = "months")
cntc.cnt.boxplot <- boxplot(Contacts_Count_12_mon, main="Contacts_Count_12_mon", ylab = "#")
credit.limit.boxplot <- boxplot(Credit_Limit,  main="Credit_Limit", ylab = "Dollars")
ttl.revbal.boxplot <- boxplot(Total_Revolving_Bal, main="Total_Revolving_Bal", ylab = "Dollars")
avg.opnbuy.boxplot <- boxplot(Avg_Open_To_Buy, main="Avg_Open_To_Buy", ylab = "Dollars")
total.amtchg.boxplot <- boxplot(Total_Amt_Chng_Q4_Q1, main="Total_Amt_Chng_Q4_Q1", ylab = "Dollars")
ttl.transct.boxplot <- boxplot(Total_Trans_Ct, main="Total_Trans_Ct", ylab = "#")
total.cntchg.boxplot <- boxplot(Total_Ct_Chng_Q4_Q1, main="Total_Ct_Chng_Q4_Q1", ylab = "Dollars")
avg.utilrate.boxplot <- boxplot(Avg_Utilization_Ratio, main="Avg_Utilization_Ratio", ylab = "Ratio")
```





Contacts_Count_12_mon
#
Credit_Limit
Dollars
Total_Revolving_Bal
Dollars
Avg_Open_To_Buy
Dollars
Total_Amt_Chng_Q4_Q1
Dollars
Total_Trans_Ct
#



It is clear that many of the numerical columns have too many outliers. However, as a result of the nature of our study, eliminating the outliers might significantly affect the sake of study.

Data Preparation

So, we only selected the outliers in the age since there are only 2 people who is over 70 in addition to card holders other than the blue category.

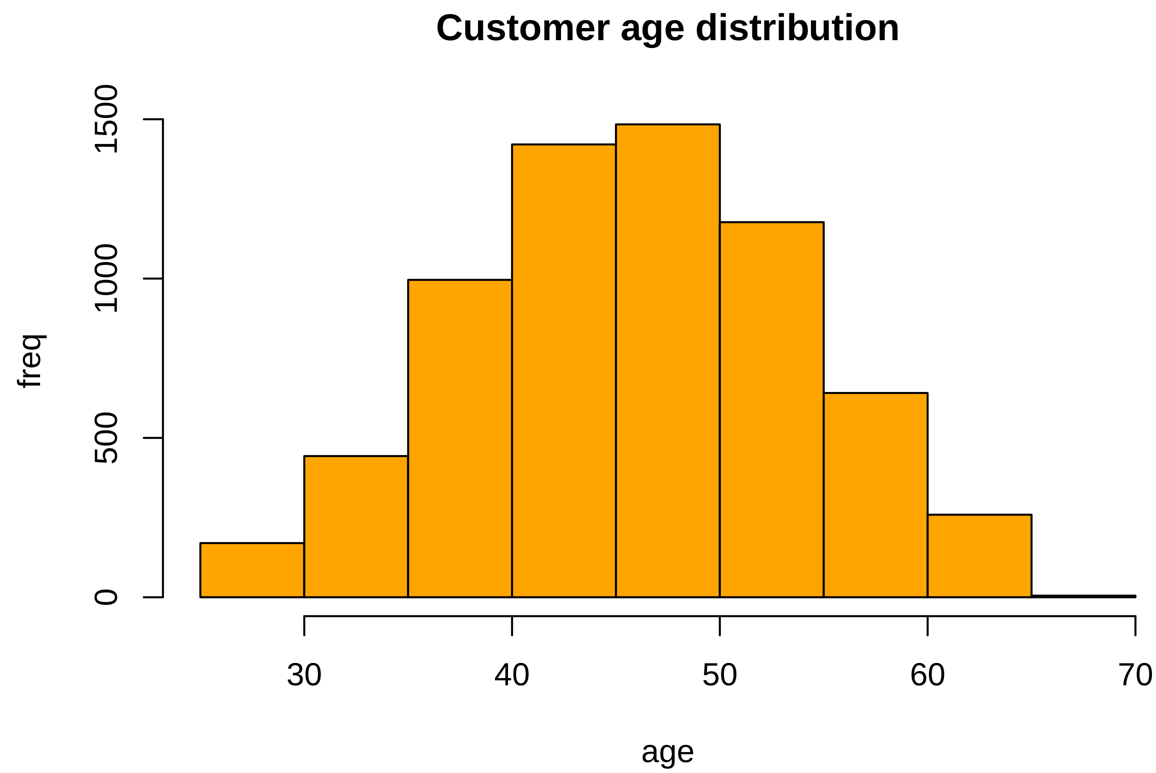
```
cleaned_bank_data_withoutNA_quan <- bank_data_withoutNA_quan
#Extracting Outliers from age and Rescoping the study to only focus on Blue Cards

#using the 1st quartile-1.5*IQR and 3rd quartile+1.5*IQR rule,
#it is seen that customers over the age of 70 are outliers
age.exc.list <- boxplot.stats(cleaned_bank_data_withoutNA_quan$Customer_Age)$out

#Since most of the data is coming from the Blue cards and there is a visible difference on many paramet
card.exc.list <- c(2, 3, 4)


cleaned_bank_data_withoutNA_quan <- subset(cleaned_bank_data_withoutNA_quan,!((Customer_Age %in% age.exc
cleaned_bank_data_withoutNA_quan<- subset(cleaned_bank_data_withoutNA_quan, select = -c(Card_Category))
head(cleaned_bank_data_withoutNA_quan)
```

```
##    CLIENTNUM Attrition_Flag Customer_Age Is_Female Dependent_count
## 1 768805383              0           45         0               3
## 2 818770008              0           49         1               5
## 3 713982108              0           51         0               3
## 5 709106358              0           40         0               3
## 6 713061558              0           44         0               2
## 9 710930508              0           37         0               3
##   Education_Level Marital_Status Income_Category Months_on_book
## 1               2              2               3             39
## 2               4              1               1             44
## 3               4              2               4             36
## 5               1              2               3             21
## 6               4              2               2             36
## 9               1              1               3             36
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
```

```
## 1                         5                      1                      3
## 2                         6                      1                      2
## 3                         4                      1                      0
## 5                         5                      1                      0
## 6                         3                      1                      2
## 9                         5                      2                      0
##   Credit_Limit Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1
## 1        12691                 777           11914                1.335
## 2         8256                 864            7392                1.541
## 3         3418                   0            3418                2.594
## 5         4716                   0            4716                2.175
## 6         4010                1247            2763                1.376
## 9        22352                2517           19835                3.355
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 1            1144             42               1.625                 0.061
## 2            1291             33               3.714                 0.105
## 3            1887             20               2.333                 0.000
## 5             816             28               2.500                 0.000
## 6            1088             24               0.846                 0.311
## 9            1350             24               1.182                 0.113
```

After the cleaning, lets examine the current situation of the dataset with more graphs.

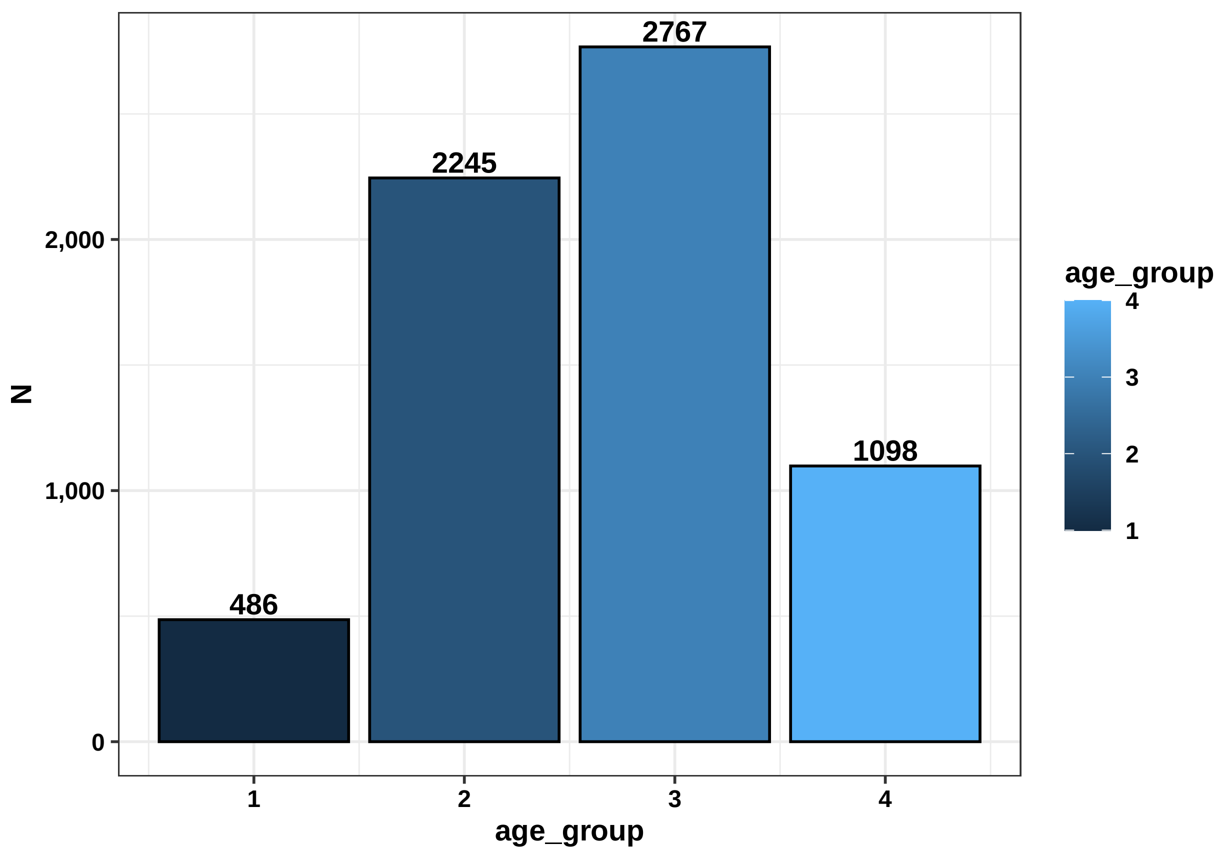
```r
# Descriptive Graphs
#histogram
Cust.age.hist <- hist(cleaned_bank_data_withoutNA_quan$Customer_Age, xlab="age", ylab="freq",
                      main="Customer age distribution", col="orange")
```



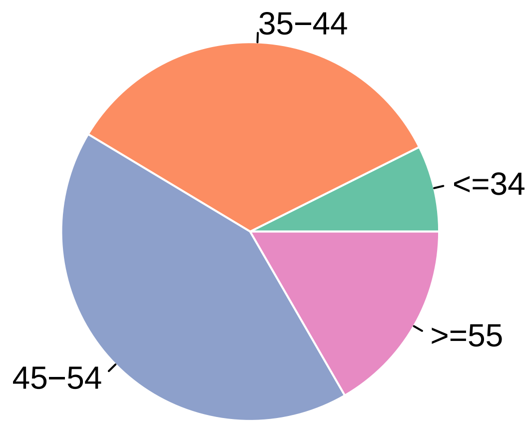

```
Cust.age.hist
```

```
## $breaks
##  [1] 25 30 35 40 45 50 55 60 65 70
##
## $counts
## [1]  170  443  996 1421 1484 1177  641  259    5
##
## $density
## [1] 0.005154639 0.013432383 0.030200121 0.043086719 0.044996968 0.035688296
## [7] 0.019436022 0.007853244 0.000151607
##
## $mids
## [1] 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5 67.5
##
## $xname
## [1] "cleaned_bank_data_withoutNA_quan$Customer_Age"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

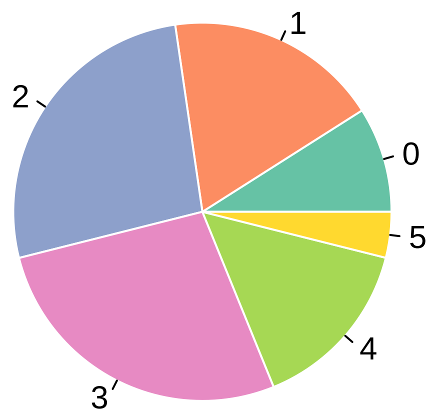
```r
#using the histogram, dividing ages into 4 groups seems satisfying

#Creating age groups
cleaned_bank_data_withoutNA_quan[cleaned_bank_data_withoutNA_quan$Customer_Age <= 34, "age_group"] <- 1
cleaned_bank_data_withoutNA_quan[cleaned_bank_data_withoutNA_quan$Customer_Age > 34 & cleaned_bank_data_
cleaned_bank_data_withoutNA_quan[cleaned_bank_data_withoutNA_quan$Customer_Age > 44 & cleaned_bank_data_
cleaned_bank_data_withoutNA_quan[cleaned_bank_data_withoutNA_quan$Customer_Age > 54, "age_group"] <- 4

#grouped age histogram
par(mfrow=c(1,1))
cleaned_bank_data_withoutNA_quan %>%
  group_by(age_group) %>% summarise(N=n()) %>%
  ggplot(aes(x=age_group,y=N,fill=age_group))+
  geom_bar(stat = 'identity',color='black')+
  scale_y_continuous(labels = scales::comma_format(accuracy = 2))+
  geom_text(aes(label=N),vjust=-0.25,fontface='bold')+
  theme_bw()+
  theme(axis.text = element_text(color='black',face='bold'),
        axis.title = element_text(color='black',face='bold'),
        legend.text = element_text(color='black',face='bold'),
        legend.title = element_text(color='black',face='bold'))
```



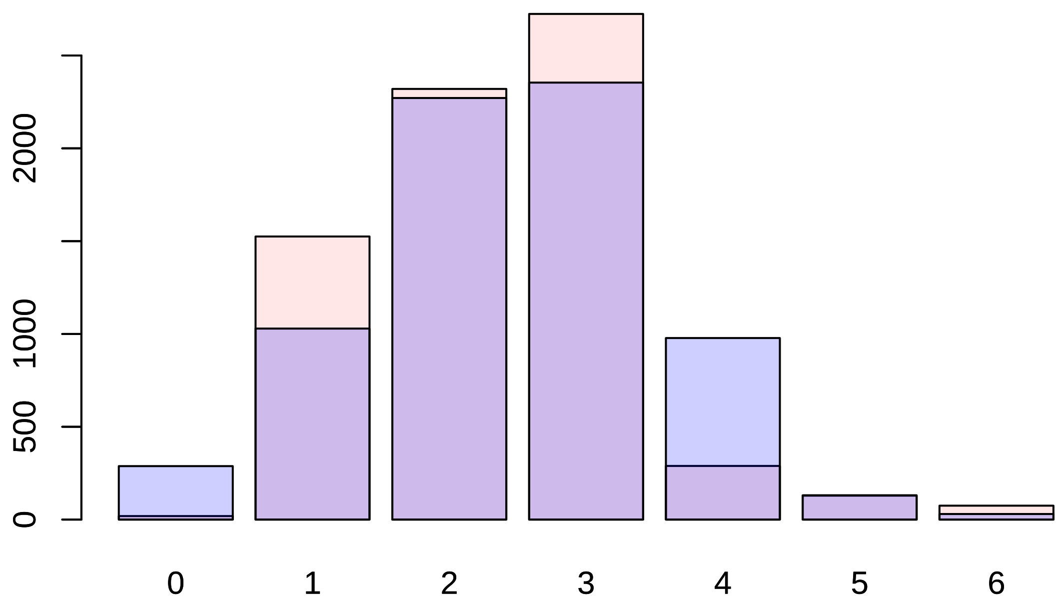

```r
# grouped age piechart
age.labels <- c("<=34", "35-44", "45-54", ">=55")
cust.age.piechart <- pie(count(cleaned_bank_data_withoutNA_quan, age_group)$n, border="white", col=myPal
```



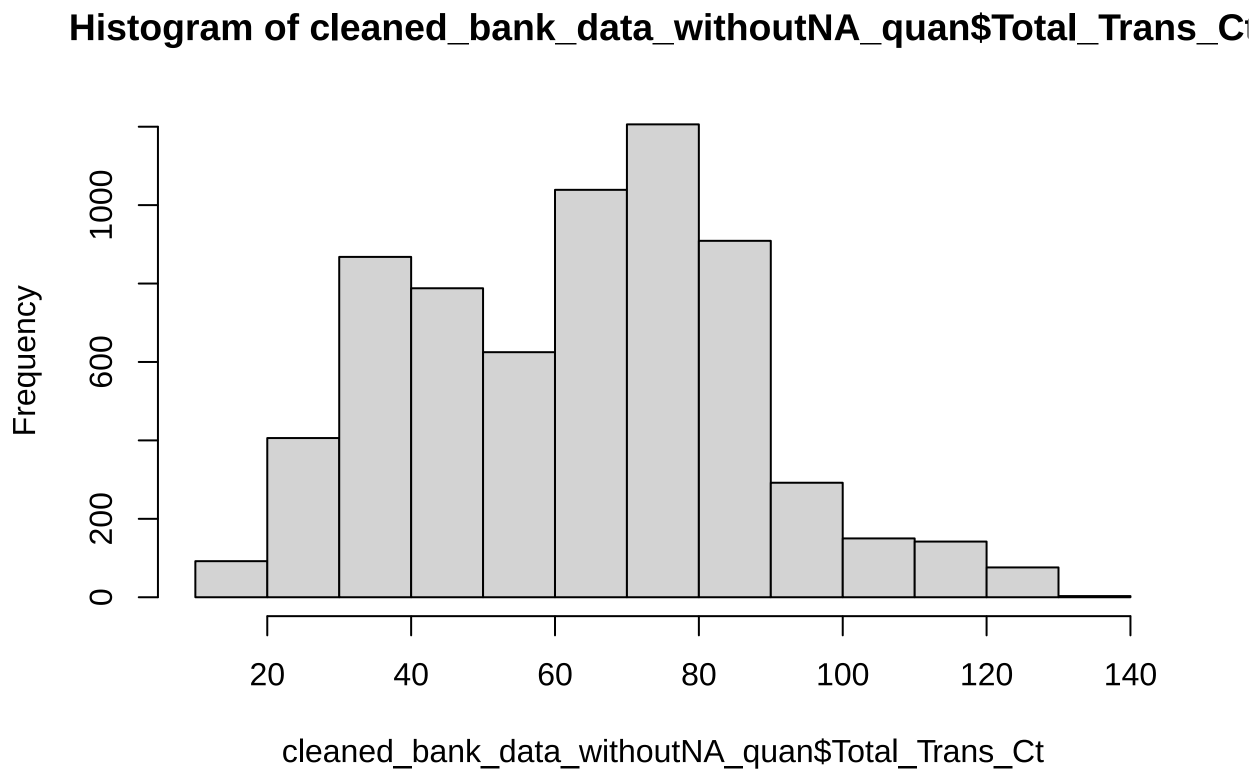

```
#Dependent Count

depcount.labels <- c(0, 1, 2, 3, 4, 5)
dependent.count.piechart <- pie(count(cleaned_bank_data_withoutNA_quan, Dependent_count)$n, border="whi
```




```r
#Months inactive
barplot(table(factor(Months_Inactive_12_mon,levels=min(Months_Inactive_12_mon):max(Months_Inactive_12_mo
barplot(table(factor(Contacts_Count_12_mon,levels=min(Contacts_Count_12_mon):max(Contacts_Count_12_mon))
```

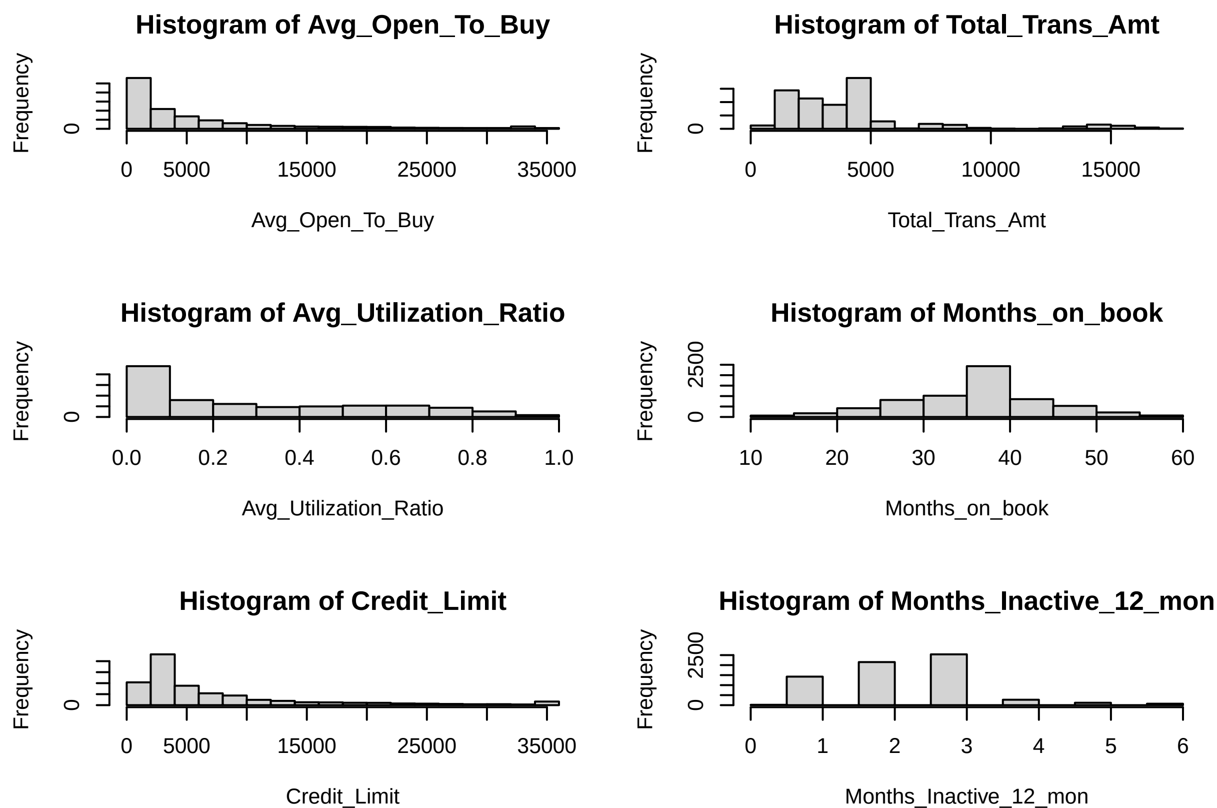



```
hist(cleaned_bank_data_withoutNA_quan$Total_Trans_Ct)
```




```
int.hist = function(x,ylab="Frequency",...) {
  barplot(table(factor(x,levels=min(x):max(x))),space=0,xaxt="n",ylab=ylab,...);axis(1)
}
```

```
#Histograms
attach(cleaned_bank_data_withoutNA_quan)
par(mfrow=c(3,2))
hist(Avg_Open_To_Buy)
hist(Total_Trans_Amt)
hist(Avg_Utilization_Ratio)
hist(Months_on_book)
hist(Credit_Limit)
hist(Months_Inactive_12_mon)
```



Splitting into Training and Test Sets

To be able to build and evaluate a model that predicts if a Customer is likely to churn, we divide the dataset so that 75% of the data will be the *Training set* and the remaining 25% the *Test set*. These two new datasets will be used respectively to train our models and to evaluate the performances of the models. We will then concentrate on the properties of the training set, while the test set will remain "unknown" until the evaluation of the models.

```
set.seed(0237)

sample <- sample.split(cleaned_bank_data_withoutNA_quan[,2:20]$Attrition_Flag,
                       SplitRatio = 0.75)
train <- subset(cleaned_bank_data_withoutNA_quan[,2:20],sample == TRUE)
test <- subset(cleaned_bank_data_withoutNA_quan[,2:20],sample == FALSE)
```

We also check that the proportion of the response variable is approximately the same in the two new datasets.

```
#Proportion of Attrited and Existing Customer in Training set
prop.table(table(train$Attrition_Flag))
```

```
##
##        0        1
## 0.843169 0.156831
```

```r
#Proportion of Attrited and Existing Customer in Test set
prop.table(table(test$Attrition_Flag))
```

```
##
##         0         1
## 0.8434466 0.1565534
```

It can be seen that the dataset is unbalanced. We decide then to create a balanced training set. By using this dataset the models will be able to represent better the Attrited Customers. To define this new Training set we utilized a mix of Oversampling and Undersampling to obtain a dataset with the same length of the original Training dataset.

```r
# ovun.sample is a function form the "Rose" library
train_bal <- ovun.sample(Attrition_Flag~.,data = train, method = "both", p = 0.5,
                         N =4948)$data
```

We can see that this new Training set is indeed balanced

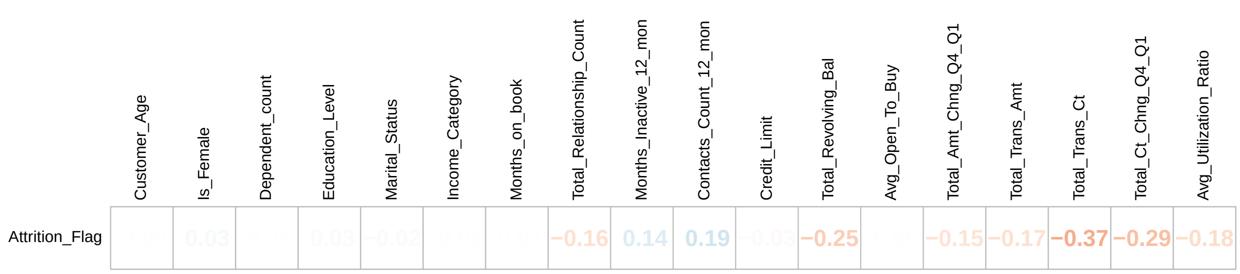
```r
#Proportion of Attrited and Existing Customer
prop.table(table(train_bal$Attrition_Flag))
```

```
##
##         0         1
## 0.4931285 0.5068715
```

Correlations

We will now focus on the unbalanced training set and we will investigate the correlation between variables. We will then try to understand which features are connected to attrited customer. First of all we compute the correlations between the response variable and the predictor variables using the *corrplot* function, from the homonym library.

```r
# Changing Attrition_Flag from factor to numeric
train$Attrition_Flag <- as.numeric(train$Attrition_Flag)

#Correlation matrix
cor_mat <- cor(train)

#Correlation with Attrition_Flag
corrplot(cor_mat[1,,drop=FALSE],method = "number",number.cex = 1.3,
         cl.pos = "n",tl.col = "black" ,tl.cex=0.9,diag = FALSE)
```

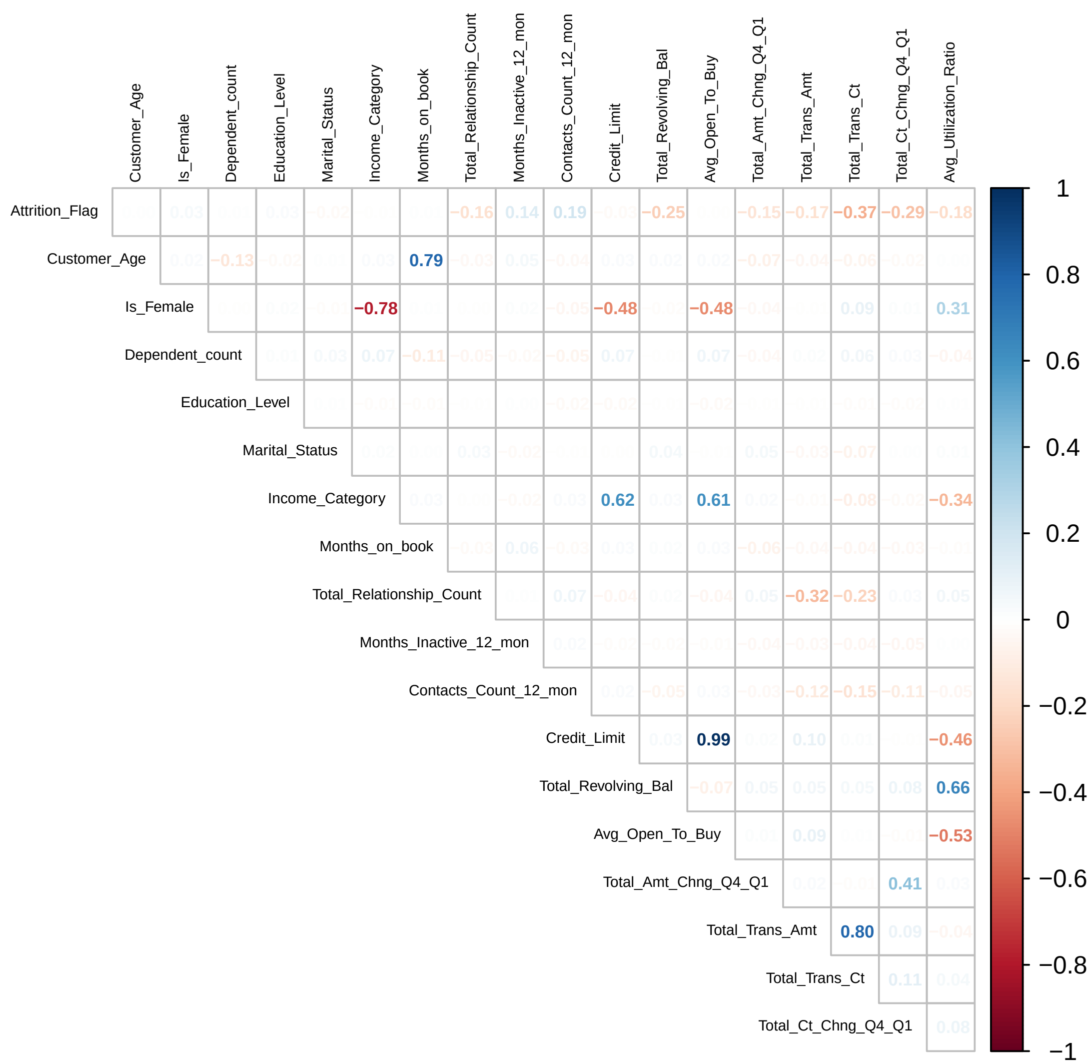


We report in the following table the highest values observed

Variable	Correlation with Attrition_Flag
Total_Trans_Ct	-0.37
Total_Ct_Chng_Q4_Q1	-0.29
Total_Revolving_Bal	-0.25
Contacts_Count_12_mon	0.19
Avg_Utilization_ratio	-0.18
Total_Trans_Amt	-0.17
Total_Relantioship_Count	-0.16

Focusing now on the other correlations, we create the following figure

```r
#Correlations
corrplot(cor_mat[1:19,1:19],method = "number",type = "upper",number.cex = 0.6,
         tl.pos = "td",tl.cex=0.5, tl.col = "black" ,diag = FALSE)
```



We then show the highest correlations in the following table

Variable 1	Variable 2	Correlation
Customer_Age	Months_on_book	0.79
Is_Female	Income_Category	-0.78
Is_Female	Credit_Limit	-0.48
Is_Female	Avg_Open_To_Buy	-0.48
Income_Category	Credit_Limit	0.62
Income_Category	Avg_Open_To_Buy	0.61
Credit_Limit	Avg_Open_To_Buy	0.99
Credit_Limit	Avg_Utilization_Ratio	-0.46
Total_Revolving_Bal	Avg_Utilization_Ratio	0.66
Avg_Open_To_Buy	Avg_Utilization_Ratio	-0.53
Total_Amt_Chng_Q4_Q1	Total_Ct_Chng_Q4_Q1	0.41
Total_Trans_Amt	Total_Trans_Ct	0.80

Since the correlation between *Credit_Limit* and *Avg_Open_To_Buy* is nearly 1, removing one of the two variable will not remove significant from our model. For this reason we decide to remove the variable *Avg_Open_To_Buy*. We can notice that even whitout considering the correlation we just removed, there are really high value of correlations. This might led to a case of collinearity, as we will see and discuss in the models' definition.

Partial Correlation

In the last figure, we analyzed the correlations between the variables in such a way that takes into account the relationships between the two variables involved in the correlation and the other variables present in the training set. To investigate the relationship between only the two variables we use the *correlation* function from the *correlation* library to calculate the partial correlations.

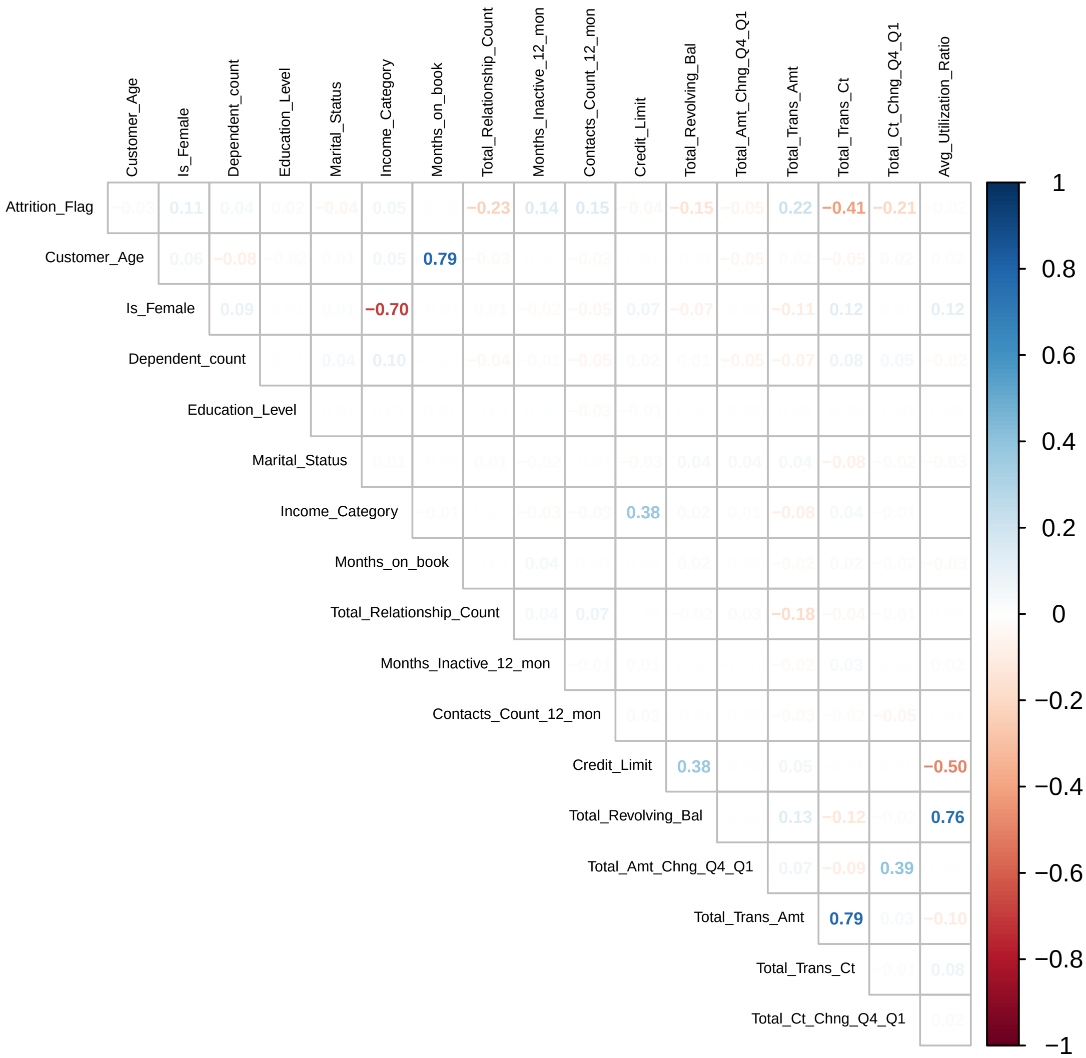
```
#Partial correlation (We will not show the output, since it takes too much space)
correlation(train[,-14],partial = TRUE)
```

For the response variable the most significant partial correlations are the following

Variable	Correlation with Attrition_Flag
Total_Trans_Ct	-0.41
Total_Relantioship_Count	-0.23
Total_Trans_Amt	0.22
Total_Ct_Chng_Q4_Q1	-0.21
Contacts_Count_12_mon	0.15
Total_Revolving_Bal	-0.15
Months_Inactive_12_mon	0.14
Is_Female	0.11

We also point out that many variables that had significant correlations have lower partial correlations values. Now we visualize the full results by using the function *corrplot* from the hononym library.

```r
#Partial Correlation matrix
part_cor_mat <- pcor(train[,-14])$estimate
corrplot(part_cor_mat, method = "number",type = "upper",number.cex = 0.6,
         tl.pos = "td",tl.cex=0.5, tl.col = "black" ,diag = FALSE)
```

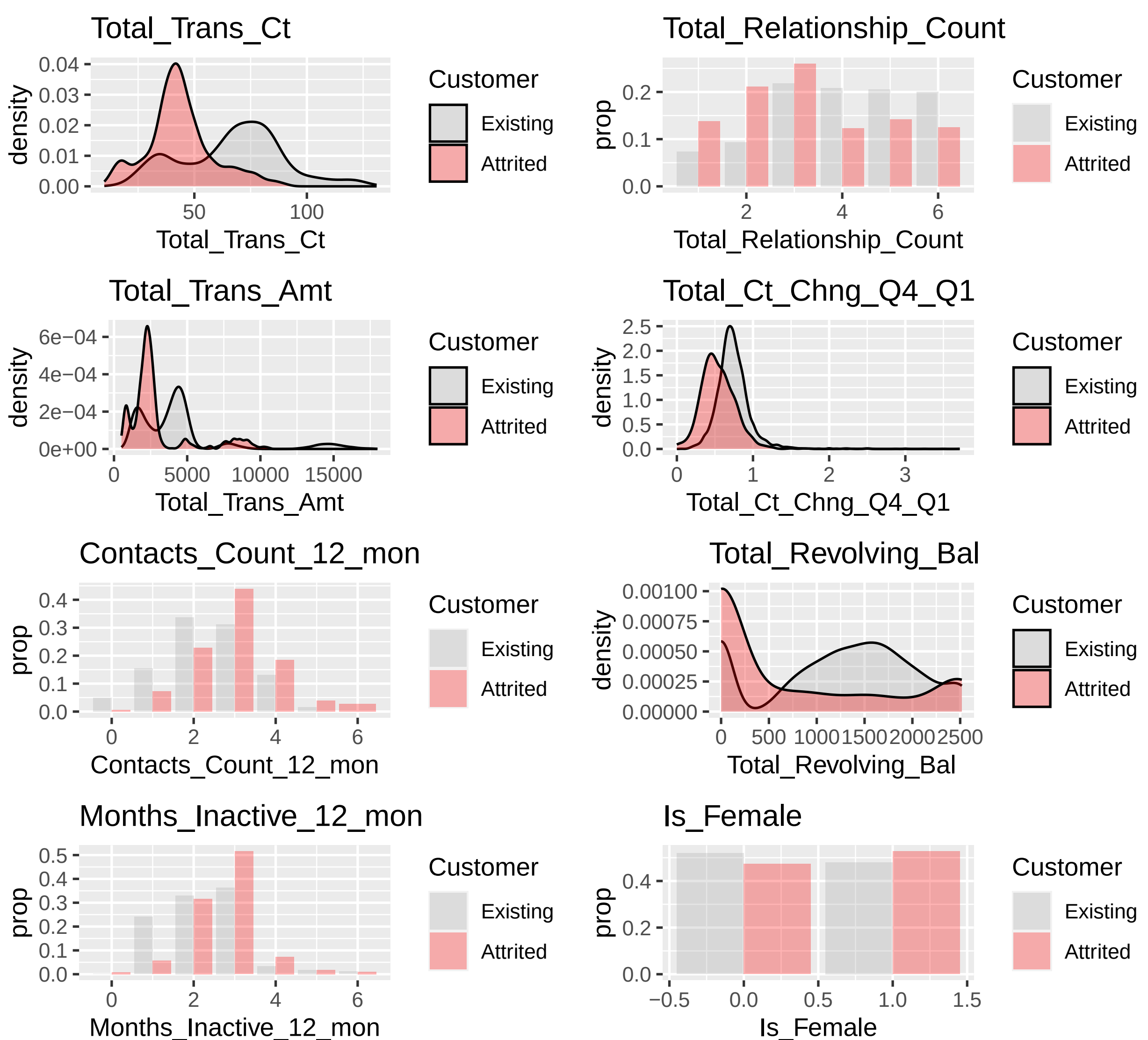


As done previously, we will show the highest partial correlations in a table:

Variable 1	Variable 2	Correlation
Customer_Age	Months_on_book	0.79
Is_Female	Income_Category	-0.70
Income_Category	Credit_Limit	0.38
Credit_Limit	Total_Revolving_Bal	0.38
Credit_Limit	Avg_Utilization_Ratio	-0.50
Total_Revolving_Bal	Avg_Utilization_Ratio	0.76
Total_Amt_Chng_Q4_Q1	Total_Ct_Chng_Q4_Q1	0.39
Total_Trans_Amt	Total_Trans_Ct	0.79

Next, we will investigate the relationship between the response variable and the other variables. To do so we separate the most correlated variables into the two classes given by *Attrition_Flag* and we plot the results.

```
Customer <- as.factor(train$Attrition_Flag)
a <- ggplot(train, aes(x = Total_Trans_Ct, fill = Customer)) +
  geom_density(alpha=0.3) + ggtitle("Total_Trans_Ct") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

b <- ggplot(train, aes(x = Total_Relationship_Count, fill = Customer)) +
  geom_bar(aes(y = after_stat(prop) ),alpha=0.3, position = "dodge") +
  ggtitle("Total_Relationship_Count") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

c <- ggplot(train, aes(x = Total_Trans_Amt, fill = Customer)) +
  geom_density(alpha=0.3) + ggtitle("Total_Trans_Amt") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

d <- ggplot(train, aes(x = Total_Ct_Chng_Q4_Q1, fill = Customer)) +
  geom_density(alpha=0.3) + ggtitle("Total_Ct_Chng_Q4_Q1")+
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

e <- ggplot(train, aes(x = Contacts_Count_12_mon, fill = Customer)) +
  geom_bar(aes(y = after_stat(prop) ),alpha=0.3, position = "dodge") +
  ggtitle("Contacts_Count_12_mon") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

f <- ggplot(train, aes(x = Total_Revolving_Bal, fill = Customer)) +
  geom_density(alpha=0.3) + ggtitle("Total_Revolving_Bal") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

g <- ggplot(train, aes(x = Months_Inactive_12_mon, fill = Customer)) +
  geom_bar(aes(y = after_stat(prop) ),alpha=0.3, position = "dodge") +
  ggtitle("Months_Inactive_12_mon") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))

h <- ggplot(train, aes(x = Is_Female, fill = Customer)) +
  geom_bar(aes(y = after_stat(prop) ),alpha=0.3, position = "dodge") +
  ggtitle("Is_Female") +
  scale_fill_manual(values = c("darkgrey","red"),labels = c("Existing","Attrited"))


ggarrange(a,b,c,d,e,f,g,h,nrow=4,ncol=2)
```



From these plots we can notice that a customer is more likely to churn if:

- In the last year he didn't make many transactions or the total amount of the transactions is low.
- In the 1st quarter he made less transactions in comparison to the last quarter.
- He holds a small amount of the bank's products.
- In the last year he contacted many times the bank.
- In the last year he has been inactive for many months.
- He didn't use much his card.

For *Total_Revolving_Bal* we can see that who has 0 as his Revolving Balance is more likely to churn, while customer who have a Revolving balance not excessively high are likely to remain with the bank. From the variable instead *Is_Female* we notice that Females are slightly more likely to churn in comparison with male customers, however it is not enough to infer a clear relationship between Attriting customers and the gender of the customer.

Model Definition

We are interested in finding the customers who are likely to churn, without penalizing too much the other customers. To do so we are going to define different models and compare how well they can identify Attriting customers. We are then searching for models with high values of *Recall*, which tells us how many Attriting customers there are, and *Precision*, which tells us how many of our predictions did churn. Because of these choices, we select the *F1 score* as our principal performance's test since it is the Harmonic mean of *Precision* and *Recall*.

To define the models we will use three different techniques: *Stepwise selection*, *Discriminant analysis* and *Regularized Regression* and for each of these we will compare the models obtained from the unbalanced and the balanced training set.

Simple logistic regression

As the first model we try the binomial generalized linear model, also known as *logistic regression*. It is a modelling technique used to solve binary classification problems by estimating the probability of an event happening based on the value of the predictor variables. In our analysis we will are trying to guess the value of the binary response variable *Attrition_Flag*, in other words we are estimating the probability of a customer's attrition. We will initially define the model containing all the predictor variables and, after addressing the problem of *Collinearity* we choose a subset of predictor variables that gives us a better model through *stepwise selection*.

Unbalanced dataset

We will start by defining the complete model and assess its performance.

```r
glm_1 <- glm(data = train,Attrition_Flag~ . - Avg_Open_To_Buy ,family = "binomial")
summary(glm_1)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ . - Avg_Open_To_Buy, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2100  -0.3654  -0.1740  -0.0704   3.3792
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               5.411e+00  5.878e-01   9.204  < 2e-16 ***
## Customer_Age             -1.994e-02  1.115e-02  -1.789 0.073569 .
## Is_Female                 1.108e+00  1.810e-01   6.122 9.23e-10 ***
## Dependent_count           1.023e-01  4.283e-02   2.388 0.016963 *
## Education_Level           6.842e-02  3.880e-02   1.763 0.077856 .
## Marital_Status           -2.817e-01  9.447e-02  -2.982 0.002866 **
## Income_Category           2.394e-01  7.186e-02   3.332 0.000862 ***
## Months_on_book           -1.760e-03  1.109e-02  -0.159 0.873882
## Total_Relationship_Count -5.241e-01  4.017e-02 -13.047  < 2e-16 ***
## Months_Inactive_12_mon    5.258e-01  5.564e-02   9.451  < 2e-16 ***
## Contacts_Count_12_mon     4.877e-01  5.174e-02   9.426  < 2e-16 ***
## Credit_Limit             -1.841e-05  9.753e-06  -1.887 0.059120 .
## Total_Revolving_Bal      -8.416e-04  1.065e-04  -7.899 2.80e-15 ***
```

```
## Total_Amt_Chng_Q4_Q1     -5.923e-01  2.740e-01  -2.162 0.030602 *
## Total_Trans_Amt           4.587e-04  3.253e-05  14.099  < 2e-16 ***
## Total_Trans_Ct           -1.127e-01  5.153e-03 -21.867  < 2e-16 ***
## Total_Ct_Chng_Q4_Q1      -2.903e+00  2.731e-01 -10.628  < 2e-16 ***
## Avg_Utilization_Ratio    -2.646e-01  3.527e-01  -0.750 0.453128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4298.6  on 4947  degrees of freedom
## Residual deviance: 2302.1  on 4930  degrees of freedom
## AIC: 2338.1
##
## Number of Fisher Scoring iterations: 6
```

For every model we will take $1/n$, $n = 1, ..., 10$ as different thresholds for testing the *F1 score* and we will then consider only the one that maximize it. Here we show the *F1 score* and other significant accuracy's tests obtained for that value of the threshold. As you can expect from an unbalanced dataset we have that the specificity is close to 1 while the recall is much lower.

```
#Threshold
Threshold <- 0.3
pred_glm_i <- predict(glm_1,test,type="response")
pred_i <- ifelse(pred_glm_i >= Threshold , 1,0)

#Confusion matrix

c_mat_i <- table(test$Attrition_Flag,pred_i)
```

	Predicted Values		
Real Values	0	1	Total
0	1284	106	1390
1	76	182	258
Total	1360	288	1648

```
#Accuracy

mean(pred_i==test$Attrition_Flag)*100
```

```
## [1] 88.95631
```

```
#True Negative Rate / Specificity

Spec_i <- c_mat_i[1,1]/sum(c_mat_i[1,])
Spec_i
```

```
## [1] 0.923741
```

```r
#Precision / Positive Predicted Value

Prec_i <- c_mat_i[2,2]/sum(c_mat_i[,2])
Prec_i
```

```
## [1] 0.6319444
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_i <- c_mat_i[2,2]/sum(c_mat_i[2,])
Rec_i
```

```
## [1] 0.7054264
```

```r
#F1 Score

F1_i <- 2 * (Prec_i * Rec_i)/(Prec_i + Rec_i)
F1_i
```

```
## [1] 0.6666667
```

Before starting with the stepwise selection we check for collinearity by checking the *Variance Inflation Function* of the predictor variables.

```r
vif(glm_1)
```

```
##            Customer_Age                Is_Female          Dependent_count
##                2.938622                 2.819186                 1.052154
##         Education_Level           Marital_Status          Income_Category
##                1.005029                 1.042643                 3.413455
##          Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                2.920568                 1.221905                 1.055363
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                1.043456                 1.967461                 2.765501
##     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt           Total_Trans_Ct
##                1.165563                 3.870338                 4.066751
##      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                1.197603                 3.177089
```

We can notice that there are a lot of values much higher than 1. That means that there might be a problem of collinearity which happens because there are variables highly correlated. Collinearity could be a problem in the model definition because it makes it difficult to determine the effects of the highly correlated variables on the response variable. We notice that the indipendent variables *Avg_Utilization_Ratio* and *Months_on_book* have high correlation coefficients with other variables and, from the p-values, it seems that they are not significant for the model. However we decide to keep *Avg_Utilization_Ratio* because of an interaction it has with *Total_Revolving_Bal* which is statistically significant for the model.

```r
glm_2 <- update(glm_1, . ~ . - Months_on_book)

vif(glm_2)
```

```
##            Customer_Age                  Is_Female           Dependent_count
##                1.066449                   2.818972                  1.051786
##         Education_Level             Marital_Status           Income_Category
##                1.004770                   1.041841                  3.412964
## Total_Relationship_Count  Months_Inactive_12_mon    Contacts_Count_12_mon
##                1.221888                   1.046814                  1.043434
##            Credit_Limit        Total_Revolving_Bal      Total_Amt_Chng_Q4_Q1
##                1.967450                   2.761569                  1.164990
##          Total_Trans_Amt             Total_Trans_Ct       Total_Ct_Chng_Q4_Q1
##                3.867367                   4.066665                  1.197285
##    Avg_Utilization_Ratio
##                3.171929
```

We notice that the value of the *VIF* for *Customer_Age* is smaller but the others didn't change much. We decide to keep the other variables even if there are two values close to 5, because they are statistically significant in face of that collinearity. In fact removing them leads to a much worse performance of the model.

Stepwise selection

Stepwise selection consists in a refinement of the model by iteratively removing (Backward selection) or adding (forward selection) variables based on their significance to the model's performance. By using this method we aim to find the most important variables and interactions, so that the model can fit accurately the data without being too complex. We will show the final choice of variables and interactions based on the p-value and the *Akaike information criterion* (AIC), which is an estimator of the quality of the model on the training set. Moreover AIC penalize the models with a big number of estimated parameters, reducing the possibility of overfitting.

```r
glm_3 <- update(glm_2, . ~ . + Avg_Utilization_Ratio*Total_Revolving_Bal )

glm_4 <- update(glm_3, . ~ . + Total_Trans_Amt*Total_Trans_Ct
                + Total_Trans_Amt*Total_Amt_Chng_Q4_Q1
                + Total_Trans_Amt*Is_Female
                + Total_Trans_Amt*Total_Relationship_Count)

glm_5 <- update(glm_4, . ~ . + Total_Ct_Chng_Q4_Q1*Dependent_count
                + Total_Ct_Chng_Q4_Q1*Is_Female)

glm_6 <- update(glm_5, . ~ . + Customer_Age*Marital_Status)

glm_7 <- update(glm_6, . ~ . - Education_Level)
```

```r
summary(glm_7)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Marital_Status + Income_Category + Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Credit_Limit +
##     Total_Revolving_Bal + Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt +
##     Total_Trans_Ct + Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio +
##     Total_Revolving_Bal:Avg_Utilization_Ratio + Total_Trans_Amt:Total_Trans_Ct +
##     Total_Amt_Chng_Q4_Q1:Total_Trans_Amt + Is_Female:Total_Trans_Amt +
##     Total_Relationship_Count:Total_Trans_Amt + Dependent_count:Total_Ct_Chng_Q4_Q1 +
```

```
##     Is_Female:Total_Ct_Chng_Q4_Q1 + Customer_Age:Marital_Status,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0313  -0.2703  -0.0969  -0.0213   3.3674
##
## Coefficients:
##                                             Estimate Std. Error z value
## (Intercept)                                2.877e+00  1.253e+00   2.296
## Customer_Age                               4.419e-02  2.260e-02   1.955
## Is_Female                                  3.016e+00  4.320e-01   6.981
## Dependent_count                            5.599e-01  1.516e-01   3.693
## Marital_Status                             1.557e+00  6.127e-01   2.541
## Income_Category                            2.792e-01  8.131e-02   3.433
## Total_Relationship_Count                  -8.299e-01  7.928e-02 -10.469
## Months_Inactive_12_mon                     5.331e-01  6.359e-02   8.384
## Contacts_Count_12_mon                      4.575e-01  5.861e-02   7.806
## Credit_Limit                              -1.632e-05  1.073e-05  -1.521
## Total_Revolving_Bal                       -1.260e-03  1.312e-04  -9.604
## Total_Amt_Chng_Q4_Q1                      -5.969e+00  6.475e-01  -9.219
## Total_Trans_Amt                            1.108e-03  2.394e-04   4.630
## Total_Trans_Ct                            -9.339e-02  7.224e-03 -12.928
## Total_Ct_Chng_Q4_Q1                       -3.592e-01  6.543e-01  -0.549
## Avg_Utilization_Ratio                     -5.498e+00  6.818e-01  -8.064
## Total_Revolving_Bal:Avg_Utilization_Ratio  3.297e-03  3.369e-04   9.788
## Total_Trans_Amt:Total_Trans_Ct            -2.543e-05  2.356e-06 -10.795
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt       1.789e-03  2.144e-04   8.344
## Is_Female:Total_Trans_Amt                 -2.339e-04  7.415e-05  -3.154
## Total_Relationship_Count:Total_Trans_Amt   9.232e-05  2.074e-05   4.450
## Dependent_count:Total_Ct_Chng_Q4_Q1       -7.779e-01  2.321e-01  -3.352
## Is_Female:Total_Ct_Chng_Q4_Q1             -1.983e+00  6.122e-01  -3.239
## Customer_Age:Marital_Status               -3.818e-02  1.321e-02  -2.891
##                                           Pr(>|z|)
## (Intercept)                               0.021691 *
## Customer_Age                              0.050590 .
## Is_Female                                 2.94e-12 ***
## Dependent_count                           0.000222 ***
## Marital_Status                            0.011044 *
## Income_Category                           0.000597 ***
## Total_Relationship_Count                   < 2e-16 ***
## Months_Inactive_12_mon                     < 2e-16 ***
## Contacts_Count_12_mon                     5.89e-15 ***
## Credit_Limit                              0.128305
## Total_Revolving_Bal                        < 2e-16 ***
## Total_Amt_Chng_Q4_Q1                       < 2e-16 ***
## Total_Trans_Amt                           3.65e-06 ***
## Total_Trans_Ct                             < 2e-16 ***
## Total_Ct_Chng_Q4_Q1                       0.583021
## Avg_Utilization_Ratio                     7.38e-16 ***
## Total_Revolving_Bal:Avg_Utilization_Ratio  < 2e-16 ***
## Total_Trans_Amt:Total_Trans_Ct             < 2e-16 ***
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt       < 2e-16 ***
## Is_Female:Total_Trans_Amt                 0.001608 **
```

```
## Total_Relationship_Count:Total_Trans_Amt  8.59e-06 ***
## Dependent_count:Total_Ct_Chng_Q4_Q1       0.000802 ***
## Is_Female:Total_Ct_Chng_Q4_Q1             0.001199 **
## Customer_Age:Marital_Status               0.003844 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4298.6  on 4947  degrees of freedom
## Residual deviance: 1799.3  on 4924  degrees of freedom
## AIC: 1847.3
##
## Number of Fisher Scoring iterations: 8
```

We decided to not remove *Credit_Limit* since removing it actually increases the *AIC*. Moreover, we decided to keep *Total_Ct_Chng_Q4_Q1* since it's been used in a interaction.

We now check the AIC values and we can notice that it gradually gets smaller.

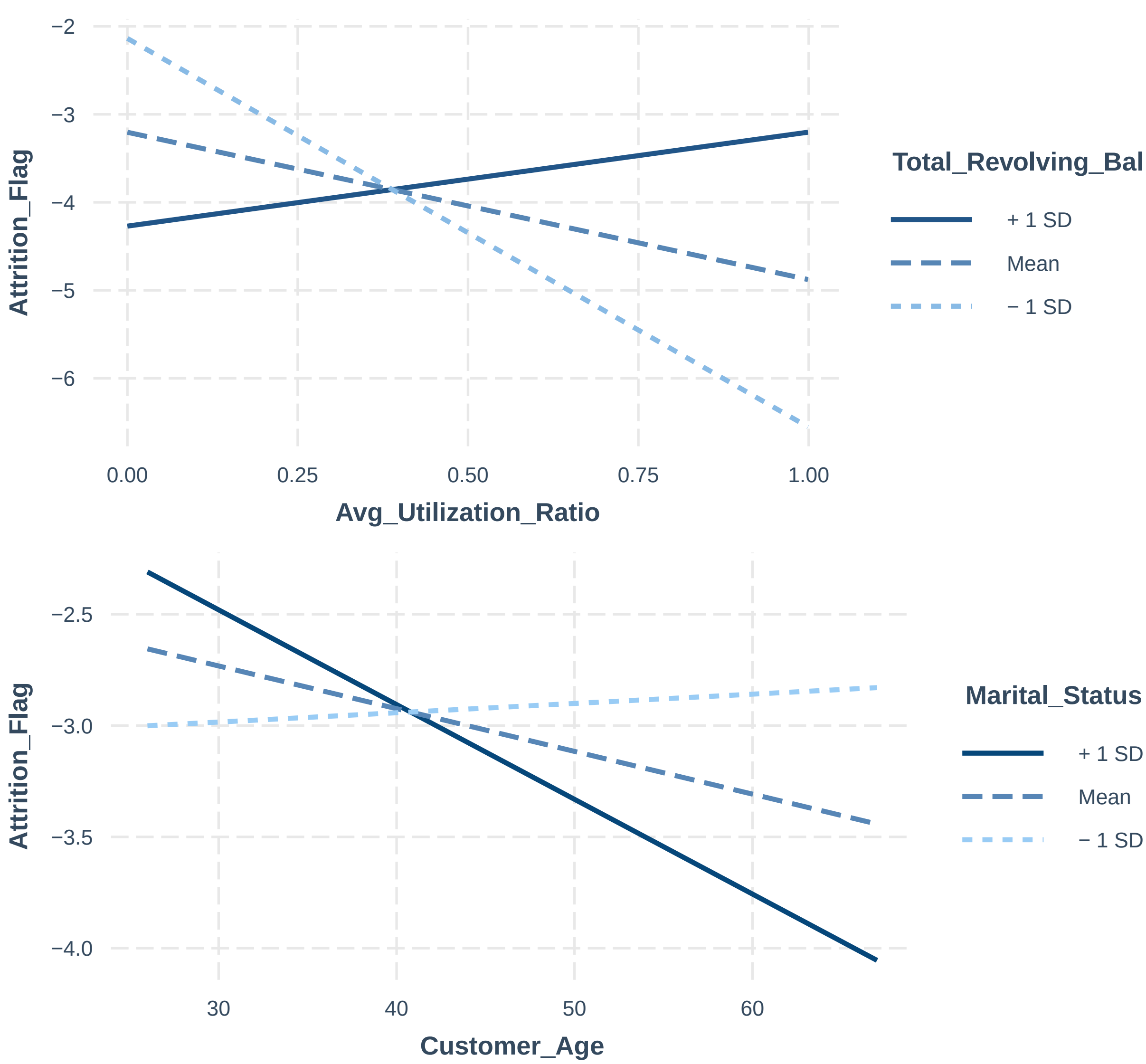
```
AIC(glm_1,glm_2,glm_3,glm_4,glm_5,glm_6,glm_7)
```

```
##       df      AIC
## glm_1 18 2338.129
## glm_2 17 2336.154
## glm_3 18 2212.688
## glm_4 22 1871.639
## glm_5 24 1853.617
## glm_6 25 1847.361
## glm_7 24 1847.350
```

We can see from the following figures the effect that one variable has on the response variable based on different values of the other variable of the interaction. We plotted the results on the models before applying the logit function, so that it's easier to verify the interactions by seeing the slopes of the lines. We decided to show only two of these plots since they are all pretty similar and we didn't want to use too much space.

```
i1 <- interact_plot(glm_3,pred = Avg_Utilization_Ratio,modx = Total_Revolving_Bal,
                    outcome.scale = "link")
i2 <- interact_plot(glm_4,pred = Total_Trans_Ct,modx = Total_Trans_Amt,
                    outcome.scale = "link")
i3 <- interact_plot(glm_4,pred = Total_Trans_Amt,modx = Total_Amt_Chng_Q4_Q1,
                    outcome.scale = "link")
i4 <- interact_plot(glm_4,pred = Total_Trans_Amt,modx = Is_Female,
                    outcome.scale = "link")
i5 <- interact_plot(glm_4,pred = Total_Trans_Amt,modx = Total_Relationship_Count,
                    outcome.scale = "link")
i6 <- interact_plot(glm_5,pred = Total_Ct_Chng_Q4_Q1,modx = Dependent_count,
                    outcome.scale = "link")
i7 <- interact_plot(glm_5,pred = Total_Ct_Chng_Q4_Q1,modx = Is_Female,
                    outcome.scale = "link")
i8 <- interact_plot(glm_6,pred = Customer_Age,modx = Marital_Status,
                    outcome.scale = "link")
```

```r
ggarrange(i1,i8,nrow=2,ncol=1)
```



As before, we check the best *F1 score* and the other significant values. We can notice that *glm_7* performs better than *glm_1* for every accuracy's test that we considered.

```r
#Threshold
Threshold <- 0.3
pred_glm_f <- predict(glm_7,test,type="response")
pred_f <- ifelse(pred_glm_f >= Threshold , 1,0)

#Confusion matrix
c_mat_f <- table(test$Attrition_Flag,pred_f)
```

	Predicted Values		
Real Values	0	1	Total
0	1299	91	1390
1	58	200	258
Total	1357	291	1648

```r
#Accuracy

mean(pred_f==test$Attrition_Flag)*100
```

```
## [1] 90.95874
```

```r
#True Negative Rate / Specificity

Spec_f <- c_mat_f[1,1]/sum(c_mat_f[1,])
Spec_f
```

```
## [1] 0.9345324
```

```r
#Precision / Positive Predicted Value

Prec_f <- c_mat_f[2,2]/sum(c_mat_f[,2])
Prec_f
```

```
## [1] 0.6872852
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_f <- c_mat_f[2,2]/sum(c_mat_f[2,])
Rec_f
```

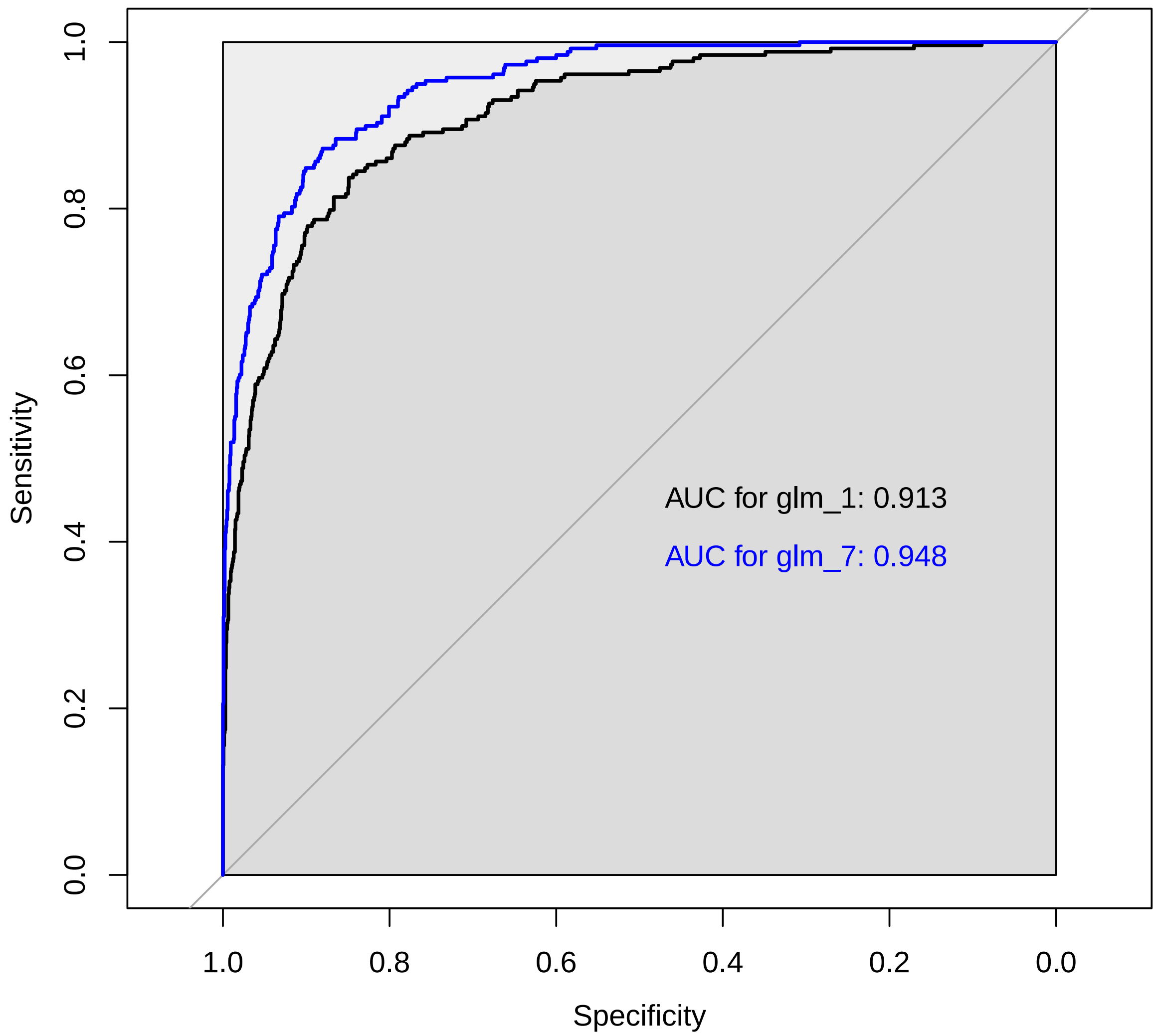
```
## [1] 0.7751938
```

```r
#F1 Score

F1_f <- 2 * (Prec_f * Rec_f)/(Prec_f + Rec_f)
F1_f
```

```
## [1] 0.7285974
```

We will now compute the ROC curve and the corrisponding AUC values of the initial and final models. The ROC curve is obtained by plotting the *Sensitivity* against the *Specificity* for various thresholds. The ROC curve and the corrispondent *Area under the curve* (AUC) are used to evaluate the performance of the models. In fact a ROC curve which is close to the top-left corner, or equivalently a AUC value close to 1, will indicate that the model predicts accurately the positive and the negative istances. We can see that the final model has a much higher AUC than the initial one. This, in conjunction with lower AIC and higher F1 score tells us that *glm_7* is a better choice then *glm_1*

```r
#ROC curves
roc_i <- roc(test$Attrition_Flag ~ pred_glm_i)
roc_f <- roc(test$Attrition_Flag ~ pred_glm_f)


plot(roc_i, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_f,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for glm_1:", round(roc_i$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for glm_7:", round(roc_f$auc, 3)), col = "blue")
```



Balanced dataset

We now define a logistic model trained on a balanced dataset. Fitting the model on the balanced data helps the model to recognize the minority class, at the cost of a higher *type I error*. So it's usually better to pick the model obtained from the balanced data if you prioritize finding customers who are likely to churn, or you can pick the unbalanced one if you don't want to penalize the existing customer who didn't plan to churn. We repeat the same procedure we used for the unbalanced dataset. We initially analyze the complete model and we check its performance

```
glm_1_bal <- glm(data = train_bal,Attrition_Flag~ . - Avg_Open_To_Buy ,family = "binomial")
summary(glm_1_bal)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ . - Avg_Open_To_Buy, family = "binomial",
##     data = train_bal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.97969  -0.47541   0.06384   0.50235   2.73670
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               7.862e+00  4.992e-01  15.750  < 2e-16 ***
## Customer_Age             -2.492e-02  8.601e-03  -2.898  0.00376 **
## Is_Female                 1.052e+00  1.455e-01   7.230 4.85e-13 ***
## Dependent_count           5.315e-02  3.384e-02   1.571  0.11627
## Education_Level           4.930e-02  3.053e-02   1.615  0.10634
## Marital_Status           -2.979e-01  7.185e-02  -4.145 3.39e-05 ***
## Income_Category           1.684e-01  5.760e-02   2.924  0.00346 **
## Months_on_book           -6.914e-03  8.712e-03  -0.794  0.42744
## Total_Relationship_Count -4.070e-01  2.961e-02 -13.745  < 2e-16 ***
## Months_Inactive_12_mon    5.699e-01  4.736e-02  12.034  < 2e-16 ***
## Contacts_Count_12_mon     4.291e-01  4.178e-02  10.271  < 2e-16 ***
## Credit_Limit             -1.843e-05  7.511e-06  -2.453  0.01415 *
## Total_Revolving_Bal      -6.365e-04  7.588e-05  -8.389  < 2e-16 ***
## Total_Amt_Chng_Q4_Q1     -7.973e-01  2.248e-01  -3.547  0.00039 ***
## Total_Trans_Amt           5.568e-04  2.793e-05  19.935  < 2e-16 ***
## Total_Trans_Ct           -1.293e-01  4.469e-03 -28.932  < 2e-16 ***
## Total_Ct_Chng_Q4_Q1      -2.523e+00  1.951e-01 -12.931  < 2e-16 ***
## Avg_Utilization_Ratio    -5.931e-01  2.622e-01  -2.262  0.02372 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6858.4  on 4947  degrees of freedom
## Residual deviance: 3503.8  on 4930  degrees of freedom
## AIC: 3539.8
##
## Number of Fisher Scoring iterations: 6
```

By comparing the *F1 score* we found that the best threshold is 0.7. Considering equal thresholds, it has a better *Recall* than the unbalanced one since it can represent better the Attriting customers. However it also has much worse Precision and this leads to an overall worse performance, as you can see from the *F1 score*.

```
#Threshold
Threshold <- 0.7
```

```r
pred_glm_i_bal <- predict(glm_1_bal,test,type="response")
pred_i_bal <- ifelse(pred_glm_i_bal >= Threshold , 1,0)

#Confusion matrix

c_mat_i_bal <- table(test$Attrition_Flag,pred_i_bal)
```

	Predicted Values		
Real Values	0	1	Total
0	1268	122	1390
1	75	183	258
Total	1343	305	1648

```r
#Accuracy

mean(pred_i_bal==test$Attrition_Flag)*100
```

```
## [1] 88.04612
```

```r
#True Negative Rate / Specificity

Spec_i_bal <- c_mat_i_bal[1,1]/sum(c_mat_i_bal[1,])
Spec_i_bal
```

```
## [1] 0.9122302
```

```r
#Precision / Positive Predicted Value

Prec_i_bal <- c_mat_i_bal[2,2]/sum(c_mat_i_bal[,2])
Prec_i_bal
```

```
## [1] 0.6
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_i_bal <- c_mat_i_bal[2,2]/sum(c_mat_i_bal[2,])
Rec_i_bal
```

```
## [1] 0.7093023
```

```r
#F1 Score

F1_i_bal <- 2 * (Prec_i_bal * Rec_i_bal)/(Prec_i_bal + Rec_i_bal)
F1_i_bal
```

```
## [1] 0.6500888
```

We check for collinearity and, as for the unbalanced case, we decide to remove the variable *Months_on_book*.

```r
vif(glm_1_bal)
```

```
##            Customer_Age                Is_Female          Dependent_count
##                2.665249                 2.920291                 1.075264
##         Education_Level           Marital_Status          Income_Category
##                1.016451                 1.027564                 3.538072
##          Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                2.696321                 1.193009                 1.083722
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                1.050338                 2.009319                 2.670097
##     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt           Total_Trans_Ct
##                1.177115                 4.202851                 4.315011
##      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                1.179883                 3.141681
```

```r
glm_2_bal <- update(glm_1_bal, . ~ . - Months_on_book)

vif(glm_2_bal)
```

```
##            Customer_Age                Is_Female          Dependent_count
##                1.093242                 2.920075                 1.073194
##         Education_Level           Marital_Status          Income_Category
##                1.016353                 1.023420                 3.528487
## Total_Relationship_Count   Months_Inactive_12_mon    Contacts_Count_12_mon
##                1.192499                 1.067066                 1.050284
##            Credit_Limit      Total_Revolving_Bal     Total_Amt_Chng_Q4_Q1
##                2.006464                 2.659067                 1.175743
##         Total_Trans_Amt           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1
##                4.203613                 4.310360                 1.179244
##    Avg_Utilization_Ratio
##                3.129420
```

Stepwise selection

We performed stepwise selection by comparing AIC values and p-values. We report the principal steps. In the final model we decided to keep *Education_level* even if not highly significant. We made this choice because removing it increase the value of the AIC.

```r
glm_3_bal <- update(glm_2_bal, . ~ . + Avg_Utilization_Ratio*Total_Revolving_Bal)
glm_4_bal <- update(glm_3_bal, . ~ . + Total_Trans_Amt*Total_Trans_Ct
                    + Total_Trans_Amt*Total_Amt_Chng_Q4_Q1
                    + Total_Trans_Amt*Is_Female
                    + Total_Trans_Amt*Total_Relationship_Count)
glm_5_bal <- update(glm_4_bal, . ~ . + Total_Ct_Chng_Q4_Q1*Dependent_count
                    + Total_Ct_Chng_Q4_Q1*Is_Female)
glm_6_bal <- update(glm_5_bal, . ~ . - Credit_Limit)
glm_7_bal <- update(glm_6_bal, . ~ . + Customer_Age*Marital_Status)
```

```r
summary(glm_7_bal)
```

```
##
```

```
## Call:
## glm(formula = Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Education_Level + Marital_Status + Income_Category + Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Total_Revolving_Bal +
##     Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt + Total_Trans_Ct +
##     Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio + Total_Revolving_Bal:Avg_Utilization_Ratio +
##     Total_Trans_Amt:Total_Trans_Ct + Total_Amt_Chng_Q4_Q1:Total_Trans_Amt +
##     Is_Female:Total_Trans_Amt + Total_Relationship_Count:Total_Trans_Amt +
##     Dependent_count:Total_Ct_Chng_Q4_Q1 + Is_Female:Total_Ct_Chng_Q4_Q1 +
##     Customer_Age:Marital_Status, family = "binomial", data = train_bal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.07161  -0.30679   0.01622   0.36208   2.80009
##
## Coefficients:
##                                             Estimate Std. Error z value
## (Intercept)                                6.764e+00  1.049e+00   6.448
## Customer_Age                               4.394e-02  1.778e-02   2.471
## Is_Female                                  2.850e+00  3.462e-01   8.232
## Dependent_count                            3.483e-01  1.208e-01   2.884
## Education_Level                            7.799e-02  3.531e-02   2.209
## Marital_Status                             1.593e+00  4.704e-01   3.388
## Income_Category                            1.600e-01  6.050e-02   2.644
## Total_Relationship_Count                  -8.114e-01  6.439e-02 -12.602
## Months_Inactive_12_mon                     5.903e-01  5.376e-02  10.979
## Contacts_Count_12_mon                      3.861e-01  4.756e-02   8.119
## Total_Revolving_Bal                       -1.180e-03  9.455e-05 -12.475
## Total_Amt_Chng_Q4_Q1                      -7.252e+00  5.371e-01 -13.503
## Total_Trans_Amt                            4.305e-04  1.888e-04   2.280
## Total_Trans_Ct                            -1.138e-01  5.742e-03 -19.814
## Total_Ct_Chng_Q4_Q1                       -8.507e-01  4.830e-01  -1.761
## Avg_Utilization_Ratio                     -5.299e+00  4.848e-01 -10.930
## Total_Revolving_Bal:Avg_Utilization_Ratio  3.113e-03  2.600e-04  11.972
## Total_Trans_Amt:Total_Trans_Ct            -2.014e-05  1.674e-06 -12.033
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt       2.143e-03  1.752e-04  12.233
## Is_Female:Total_Trans_Amt                 -2.058e-04  5.318e-05  -3.870
## Total_Relationship_Count:Total_Trans_Amt   1.010e-04  1.662e-05   6.076
## Dependent_count:Total_Ct_Chng_Q4_Q1       -3.636e-01  1.762e-01  -2.063
## Is_Female:Total_Ct_Chng_Q4_Q1             -1.614e+00  4.430e-01  -3.643
## Customer_Age:Marital_Status               -3.923e-02  1.018e-02  -3.855
##                                           Pr(>|z|)
## (Intercept)                               1.14e-10 ***
## Customer_Age                              0.013455 *
## Is_Female                                  < 2e-16 ***
## Dependent_count                           0.003921 **
## Education_Level                           0.027203 *
## Marital_Status                            0.000705 ***
## Income_Category                           0.008189 **
## Total_Relationship_Count                   < 2e-16 ***
## Months_Inactive_12_mon                     < 2e-16 ***
## Contacts_Count_12_mon                     4.69e-16 ***
## Total_Revolving_Bal                        < 2e-16 ***
## Total_Amt_Chng_Q4_Q1                       < 2e-16 ***
```

```
## Total_Trans_Amt                           0.022613 *
## Total_Trans_Ct                             < 2e-16 ***
## Total_Ct_Chng_Q4_Q1                       0.078222 .
## Avg_Utilization_Ratio                      < 2e-16 ***
## Total_Revolving_Bal:Avg_Utilization_Ratio  < 2e-16 ***
## Total_Trans_Amt:Total_Trans_Ct             < 2e-16 ***
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt       < 2e-16 ***
## Is_Female:Total_Trans_Amt                 0.000109 ***
## Total_Relationship_Count:Total_Trans_Amt  1.23e-09 ***
## Dependent_count:Total_Ct_Chng_Q4_Q1       0.039100 *
## Is_Female:Total_Ct_Chng_Q4_Q1             0.000269 ***
## Customer_Age:Marital_Status               0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6858.4  on 4947  degrees of freedom
## Residual deviance: 2718.7  on 4924  degrees of freedom
## AIC: 2766.7
##
## Number of Fisher Scoring iterations: 7
```

By checking the AIC we notice that the AIC gradually gets smaller.

```
AIC(glm_1_bal,glm_2_bal,glm_3_bal,glm_4_bal,glm_5_bal,glm_6_bal,glm_7_bal)
```

```
##           df      AIC
## glm_1_bal 18 3539.819
## glm_2_bal 17 3538.448
## glm_3_bal 18 3331.974
## glm_4_bal 22 2793.892
## glm_5_bal 24 2779.952
## glm_6_bal 23 2779.785
## glm_7_bal 24 2766.743
```

We analyze now the performance of the final model. The best threshold obtained by comparing the F1 score has changed from the one used in the inital model. Even with this change we obtain a similar Recall. However we have much better accuracy, specificity and precision.

```
#Threshold
Threshold <- 0.8
pred_glm_f_bal <- predict(glm_7_bal,test,type="response")
pred_f_bal <- ifelse(pred_glm_f_bal >= Threshold , 1,0)

#Confusion matrix

c_mat_f_bal <- table(test$Attrition_Flag,pred_f_bal)
```

	Predicted Values		
Real Values	0	1	Total

Predicted Values			
0	1319	71	1390
1	75	183	258
Total	1394	254	1648

```
#Accuracy

mean(pred_f_bal==test$Attrition_Flag)*100
```

```
## [1] 91.14078
```

```
#True Negative Rate / Specificity

Spec_f_bal <- c_mat_f_bal[1,1]/sum(c_mat_f_bal[1,])
Spec_f_bal
```

```
## [1] 0.9489209
```

```
#Precision / Positive Predicted Value

Prec_f_bal <- c_mat_f_bal[2,2]/sum(c_mat_f_bal[,2])
Prec_f_bal
```

```
## [1] 0.7204724
```

```
#Recall / True Positive Rate / Sensitivity

Rec_f_bal <- c_mat_f_bal[2,2]/sum(c_mat_f_bal[2,])
Rec_f_bal
```

```
## [1] 0.7093023
```

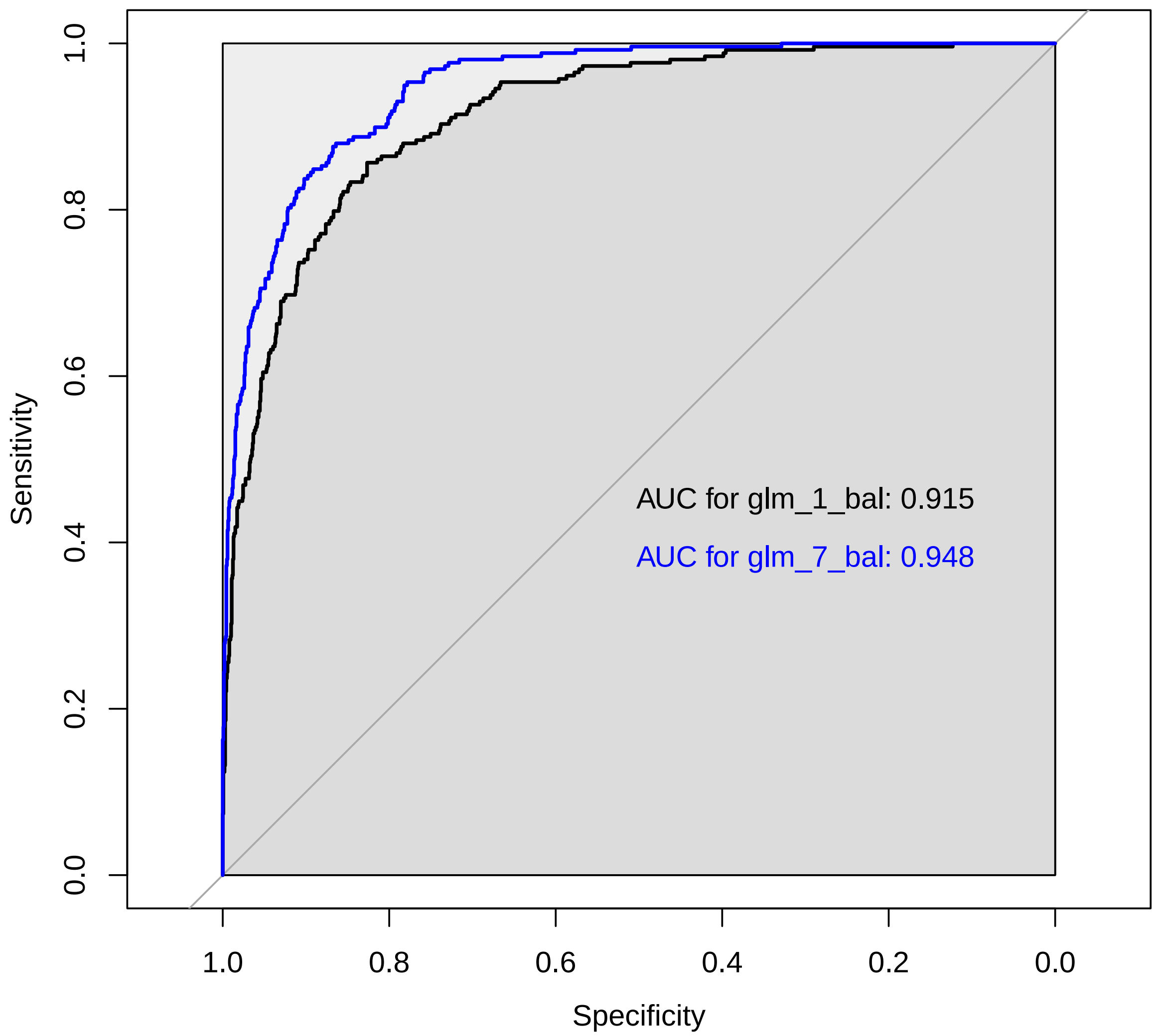
```
#F1 Score

F1_f_bal <- 2 * (Prec_f_bal * Rec_f_bal)/(Prec_f_bal + Rec_f_bal)
F1_f_bal
```

```
## [1] 0.7148438
```

We now plot the ROC curve and the corresponding AUC values. In this case, the *glm_7_bal* seems to perform better than *glm_1_bal*, in terms of AUC values and F1 score. However, between the unbalanced final model and the balanced one we prefer the model obtained with the unbalanced dataset since it has a higher F1 score, while having the same AUC values. Instead, if we aimed to have a better *Sensitivity*, we might had taken into consideration the balanced model.

```
#ROC curves
roc_i_bal <- roc(test$Attrition_Flag ~ pred_glm_i_bal)
roc_f_bal <- roc(test$Attrition_Flag ~ pred_glm_f_bal)
```

```r
plot(roc_i_bal, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_f_bal,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for glm_1_bal:", round(roc_i_bal$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for glm_7_bal:", round(roc_f_bal$auc, 3)), col = "blue")
```



Discriminant analysis

Discriminant analysis is a multivariate technique based on Bayes' rule, used to separate two or more groups of observations based on a set of predictor variables. It works by estimating the contribution that each predictor has in separating the groups. To do so it aims to minimize the variance within the classes and maximizing the variance between classes. we will consider the binary case, where it has to separate only two groups. As with regression, discriminant analysis can be linear, aiming to find a linear decision boundary for each class but it also can be polynomial. We will analyze the models obtained in the linear (*LDA*) and quadratic (*QDA*) case.

First of all we want to point out that discriminant analysis assumes that the variables are distributed normally on the two groups. In our case we can see with the *Shapiro-Wilk normality test* that the normality assumption is not satisfied. We still decide to apply the models to our data and compare theirs performances with the other models. We report only some of the *Shapiro-Wilk tests* to not take too much space.

```
# We can see that the continuous predictor variables don't follow a normal
# distribution on both classes
apply(train[train$Attrition_Flag == 1,][17:18],2,shapiro.test )
```

```
## $Total_Trans_Ct
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96628, p-value = 2.162e-12
##
##
## $Total_Ct_Chng_Q4_Q1
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.91941, p-value < 2.2e-16
```

```
apply(train[train$Attrition_Flag == 0,][17:18],2,shapiro.test )
```

```
## $Total_Trans_Ct
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.97923, p-value < 2.2e-16
##
##
## $Total_Ct_Chng_Q4_Q1
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.82955, p-value < 2.2e-16
```

Linear discriminant analysis

We start with the Linear discriminant analysis (LDA) which works by projecting the data onto a lower-dimensional space that maximizes the separation between the classes. It does this by finding the contributions that each predictive variable has in the classification of the response variable. This will results in discovering the best linear decision boundary. Other than the normality assumption, LDA also assumes that the covariance matrices are equal in all the classes.

Unbalanced dataset

We will consider the set of variables and interactions chosen in the final model of *Generalized Linear models*, since it performed better than the complete one and it mitigated the problem of collinearity.

```r
lda_u <- lda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count
             + Marital_Status + Income_Category + Total_Relationship_Count
             + Months_Inactive_12_mon + Contacts_Count_12_mon + Credit_Limit
             + Total_Revolving_Bal + Total_Amt_Chng_Q4_Q1
             + Total_Trans_Amt + Total_Trans_Ct + Total_Ct_Chng_Q4_Q1
             + Avg_Utilization_Ratio + Total_Revolving_Bal:Avg_Utilization_Ratio
             + Total_Trans_Amt:Total_Trans_Ct
             + Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
             + Is_Female:Total_Trans_Amt
             + Total_Relationship_Count:Total_Trans_Amt
             + Dependent_count:Total_Ct_Chng_Q4_Q1
             + Is_Female:Total_Ct_Chng_Q4_Q1
             + Customer_Age:Marital_Status ,
             data = train, family = "binomial")
lda_u
```

```
## Call:
## lda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Marital_Status + Income_Category + Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Credit_Limit +
##     Total_Revolving_Bal + Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt +
##     Total_Trans_Ct + Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio +
##     Total_Revolving_Bal:Avg_Utilization_Ratio + Total_Trans_Amt:Total_Trans_Ct +
##     Total_Amt_Chng_Q4_Q1:Total_Trans_Amt + Is_Female:Total_Trans_Amt +
##     Total_Relationship_Count:Total_Trans_Amt + Dependent_count:Total_Ct_Chng_Q4_Q1 +
##     Is_Female:Total_Ct_Chng_Q4_Q1 + Customer_Age:Marital_Status,
##     data = train, family = "binomial")
##
## Prior probabilities of groups:
##        0        1
## 0.843169 0.156831
##
## Group means:
##   Customer_Age Is_Female Dependent_count Marital_Status Income_Category
## 0     46.31807 0.4805849        2.333413       1.668744        2.311841
## 1     46.32474 0.5270619        2.364691       1.628866        2.291237
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
## 0                 3.979147               2.287152              2.366970
## 1                 3.295103               2.682990              2.960052
##   Credit_Limit Total_Revolving_Bal Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
## 0     7361.670           1255.4830            0.7750860        4411.586
## 1     6789.984            698.8531            0.6857912        2934.793
##   Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 0       67.17186           0.7425566             0.3211337
## 1       43.68299           0.5469420             0.1803750
##   Total_Revolving_Bal:Avg_Utilization_Ratio Total_Trans_Amt:Total_Trans_Ct
## 0                                  532.4306                       356678.3
## 1                                  335.0325                       155161.9
##   Total_Amt_Chng_Q4_Q1:Total_Trans_Amt Is_Female:Total_Trans_Amt
## 0                             3381.931                  2138.782
## 1                             2206.650                  1426.461
```

```
##   Total_Relationship_Count:Total_Trans_Amt Dependent_count:Total_Ct_Chng_Q4_Q1
## 0                                15523.772                            1.746506
## 1                                 9501.402                            1.276539
##   Is_Female:Total_Ct_Chng_Q4_Q1 Customer_Age:Marital_Status
## 0                     0.3618001                    77.34636
## 1                     0.2789472                    75.47423
##
## Coefficients of linear discriminants:
##                                                 LD1
## Customer_Age                           1.345086e-02
## Is_Female                              1.469362e+00
## Dependent_count                        1.422369e-01
## Marital_Status                         5.149516e-01
## Income_Category                        1.105290e-01
## Total_Relationship_Count              -4.716974e-01
## Months_Inactive_12_mon                 2.093839e-01
## Contacts_Count_12_mon                  1.930146e-01
## Credit_Limit                          -8.730643e-06
## Total_Revolving_Bal                   -6.384161e-04
## Total_Amt_Chng_Q4_Q1                  -1.725751e+00
## Total_Trans_Amt                       -7.078387e-05
## Total_Trans_Ct                        -5.240266e-02
## Total_Ct_Chng_Q4_Q1                   -6.273510e-01
## Avg_Utilization_Ratio                 -2.502870e+00
## Total_Revolving_Bal:Avg_Utilization_Ratio  1.484990e-03
## Total_Trans_Amt:Total_Trans_Ct        -1.804403e-06
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt   4.350297e-04
## Is_Female:Total_Trans_Amt             -6.617170e-05
## Total_Relationship_Count:Total_Trans_Amt  5.457884e-05
## Dependent_count:Total_Ct_Chng_Q4_Q1   -1.508857e-01
## Is_Female:Total_Ct_Chng_Q4_Q1         -9.958786e-01
## Customer_Age:Marital_Status           -1.335845e-02
```

We start by checking what is the best threshold in our model and the values of the F1 score and other significant performance's measures. The models seems to perform slightly worse than the final GLM in terms of F1 score and Recall, but it seems to predict better the negative response, since it has higher Specificity and Precision.

```r
#Threshold
Threshold <- 0.4
lda_u_predict <- predict(lda_u,test,type = "response")
lda_predict_u <- lda_u_predict$posterior
pred_lda_u <- ifelse(lda_predict_u[,2] >= Threshold , 1,0)

#Confusion matrix
c_mat_lda_u <- table(test$Attrition_Flag,pred_lda_u)
```

	Predicted Values		
Real Values	0	1	Total
0	1321	69	1390
1	80	178	258
Total	1401	247	1648

```r
#Accuracy

mean(pred_lda_u==test$Attrition_Flag)*100
```

```
## [1] 90.95874
```

```r
#True Negative Rate / Specificity

Spec_lda_u <- c_mat_lda_u[1,1]/sum(c_mat_lda_u[1,])
Spec_lda_u
```

```
## [1] 0.9503597
```

```r
#Precision / Positive Predicted Value

Prec_lda_u <- c_mat_lda_u[2,2]/sum(c_mat_lda_u[,2])
Prec_lda_u
```

```
## [1] 0.7206478
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_lda_u <- c_mat_lda_u[2,2]/sum(c_mat_lda_u[2,])
Rec_lda_u
```

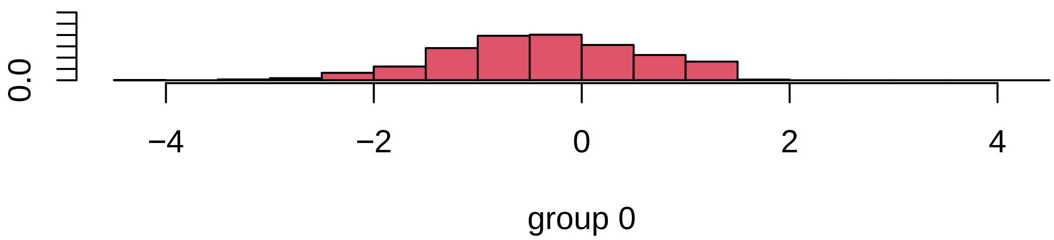
```
## [1] 0.6899225
```

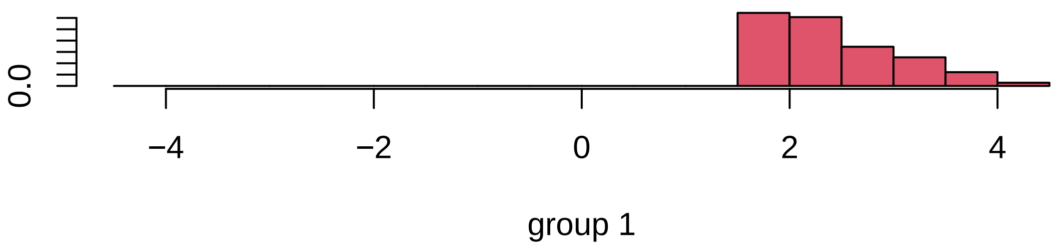
```r
#F1 Score

F1_lda_u <- 2 * (Prec_lda_u * Rec_lda_u)/(Prec_lda_u + Rec_lda_u)
F1_lda_u
```

```
## [1] 0.7049505
```

Finally we shows how the model separates the two categories. We can notice from the graph below that it manages to separate accurately the two classes, with only an overlap where there are mostly elements belonging to the *Attrition* class.

```r
# x indicates the linear combinations of the variables obtained by the model
# class indicates the two classes Existing and Attriting Customers.

ldahist(lda_u_predict$x[,1], g = lda_u_predict$class , col = 2)
```





Balanced dataset

We are now repeating the same steps done in the Unbalanced case. We consider the final set of variables used on the GLM.

```
lda_b <- lda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count
             + Education_Level + Marital_Status + Income_Category
             + Total_Relationship_Count + Months_Inactive_12_mon
             + Contacts_Count_12_mon + Total_Revolving_Bal
             + Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt + Total_Trans_Ct
             + Total_Ct_Chng_Q4_Q1  + Avg_Utilization_Ratio
             + Customer_Age:Total_Amt_Chng_Q4_Q1
             + Total_Revolving_Bal:Avg_Utilization_Ratio
             + Total_Trans_Amt:Total_Trans_Ct
             + Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
             + Is_Female:Total_Trans_Amt
             + Total_Relationship_Count:Total_Trans_Amt
             + Dependent_count:Total_Ct_Chng_Q4_Q1
             + Is_Female:Total_Ct_Chng_Q4_Q1
             + Customer_Age:Marital_Status ,
             data = train_bal, family = "binomial")
lda_b
```

```
## Call:
## lda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Education_Level + Marital_Status + Income_Category + Total_Relationship_Count +
```

```
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Total_Revolving_Bal +
##     Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt + Total_Trans_Ct +
##     Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio + Customer_Age:Total_Amt_Chng_Q4_Q1 +
##     Total_Revolving_Bal:Avg_Utilization_Ratio + Total_Trans_Amt:Total_Trans_Ct +
##     Total_Amt_Chng_Q4_Q1:Total_Trans_Amt + Is_Female:Total_Trans_Amt +
##     Total_Relationship_Count:Total_Trans_Amt + Dependent_count:Total_Ct_Chng_Q4_Q1 +
##     Is_Female:Total_Ct_Chng_Q4_Q1 + Customer_Age:Marital_Status,
##     data = train_bal, family = "binomial")
##
## Prior probabilities of groups:
##         0         1
## 0.4931285 0.5068715
##
## Group means:
##   Customer_Age Is_Female Dependent_count Education_Level Marital_Status
## 0     46.36189 0.4819672        2.320082        3.044672       1.665574
## 1     46.11164 0.5482456        2.319777        3.159888       1.615630
##   Income_Category Total_Relationship_Count Months_Inactive_12_mon
## 0        2.302869                 4.013115               2.311066
## 1        2.227273                 3.393142               2.681818
##   Contacts_Count_12_mon Total_Revolving_Bal Total_Amt_Chng_Q4_Q1
## 0              2.377459           1258.1189            0.7724857
## 1              2.930622            713.8373            0.6829537
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 0        4368.770       67.08115           0.7394672             0.3191320
## 1        2910.393       43.67624           0.5456471             0.1891061
##   Customer_Age:Total_Amt_Chng_Q4_Q1 Total_Revolving_Bal:Avg_Utilization_Ratio
## 0                          35.63106                                  532.3245
## 1                          31.58837                                  346.6438
##   Total_Trans_Amt:Total_Trans_Ct Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
## 0                       351514.9                             3346.995
## 1                       153719.1                             2189.020
##   Is_Female:Total_Trans_Amt Total_Relationship_Count:Total_Trans_Amt
## 0                  2173.387                                15402.223
## 1                  1502.860                                 9809.109
##   Dependent_count:Total_Ct_Chng_Q4_Q1 Is_Female:Total_Ct_Chng_Q4_Q1
## 0                            1.722951                     0.3582635
## 1                            1.250363                     0.2920618
##   Customer_Age:Marital_Status
## 0                    77.27131
## 1                    74.40670
##
## Coefficients of linear discriminants:
##                                                     LD1
## Customer_Age                                -1.096959e-02
## Is_Female                                    1.052223e+00
## Dependent_count                              1.114545e-01
## Education_Level                              2.226665e-02
## Marital_Status                               5.955706e-01
## Income_Category                              7.057002e-02
## Total_Relationship_Count                    -2.985882e-01
## Months_Inactive_12_mon                       2.400812e-01
## Contacts_Count_12_mon                        1.641084e-01
## Total_Revolving_Bal                         -5.836140e-04
```

```
## Total_Amt_Chng_Q4_Q1                          -3.675751e+00
## Total_Trans_Amt                                1.479013e-04
## Total_Trans_Ct                                -5.781119e-02
## Total_Ct_Chng_Q4_Q1                           -6.799209e-01
## Avg_Utilization_Ratio                         -2.300128e+00
## Customer_Age:Total_Amt_Chng_Q4_Q1              3.644249e-02
## Total_Revolving_Bal:Avg_Utilization_Ratio  1.423423e-03
## Total_Trans_Amt:Total_Trans_Ct                -3.507985e-06
## Total_Amt_Chng_Q4_Q1:Total_Trans_Amt           5.045910e-04
## Is_Female:Total_Trans_Amt                     -7.594590e-05
## Total_Relationship_Count:Total_Trans_Amt       3.270032e-05
## Dependent_count:Total_Ct_Chng_Q4_Q1           -1.237090e-01
## Is_Female:Total_Ct_Chng_Q4_Q1                 -4.632915e-01
## Customer_Age:Marital_Status                   -1.592532e-02
```

We now search for the threshold resulting in the best F1 score and we obtain 0.8. We have similar results to the balanced GLM with a slightly worse F1 score. We can also notice that it performs better in comparison to the LDA performed on the unbalanced dataset.

```
#Threshold
Threshold <- 0.8
lda_b_predict <- predict(lda_b,test,type = "response")
lda_predict_b <- lda_b_predict$posterior
pred_lda_b <- ifelse(lda_predict_b[,2] >= Threshold , 1,0)

#Confusion matrix
c_mat_lda_b <- table(test$Attrition_Flag,pred_lda_b)
```

	Predicted Values		
Real Values	0	1	Total
0	1317	73	1390
1	75	183	258
Total	1392	256	1648

```
#Accuracy

mean(pred_lda_b==test$Attrition_Flag)*100
```

```
## [1] 91.01942
```

```
#True Negative Rate / Specificity

Spec_lda_b <- c_mat_lda_b[1,1]/sum(c_mat_lda_b[1,])
Spec_lda_b
```

```
## [1] 0.947482
```

```
#Precision / Positive Predicted Value

Prec_lda_b <- c_mat_lda_b[2,2]/sum(c_mat_lda_b[,2])
Prec_lda_b
```

```
## [1] 0.7148438
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_lda_b <- c_mat_lda_b[2,2]/sum(c_mat_lda_b[2,])
Rec_lda_b
```

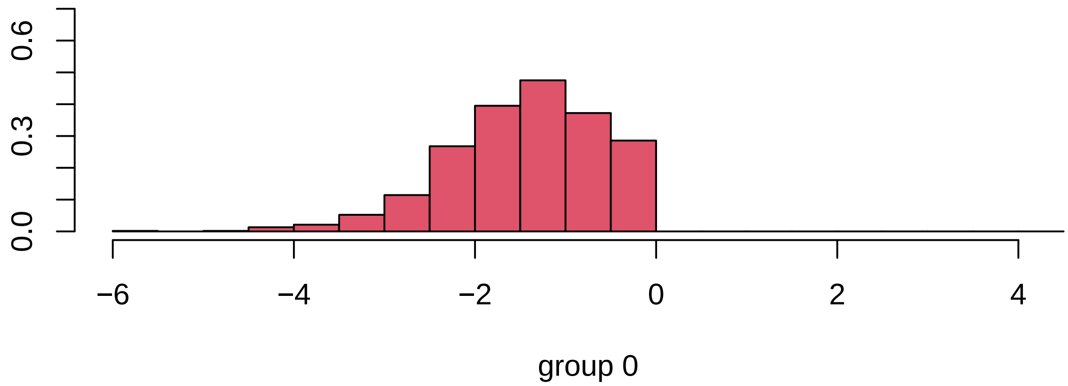
```
## [1] 0.7093023
```

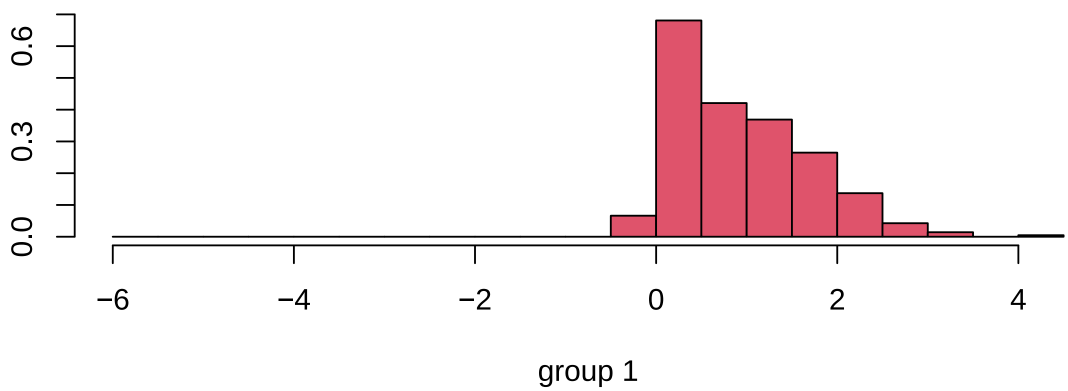
```r
#F1 Score

F1_lda_b <- 2 * (Prec_lda_b * Rec_lda_b)/(Prec_lda_b + Rec_lda_b)
F1_lda_b
```

```
## [1] 0.7120623
```

We can notice from the plots below how the model separate nicely the two classes but, in this case, the overlaps is more prominent than in the unbalanced model.

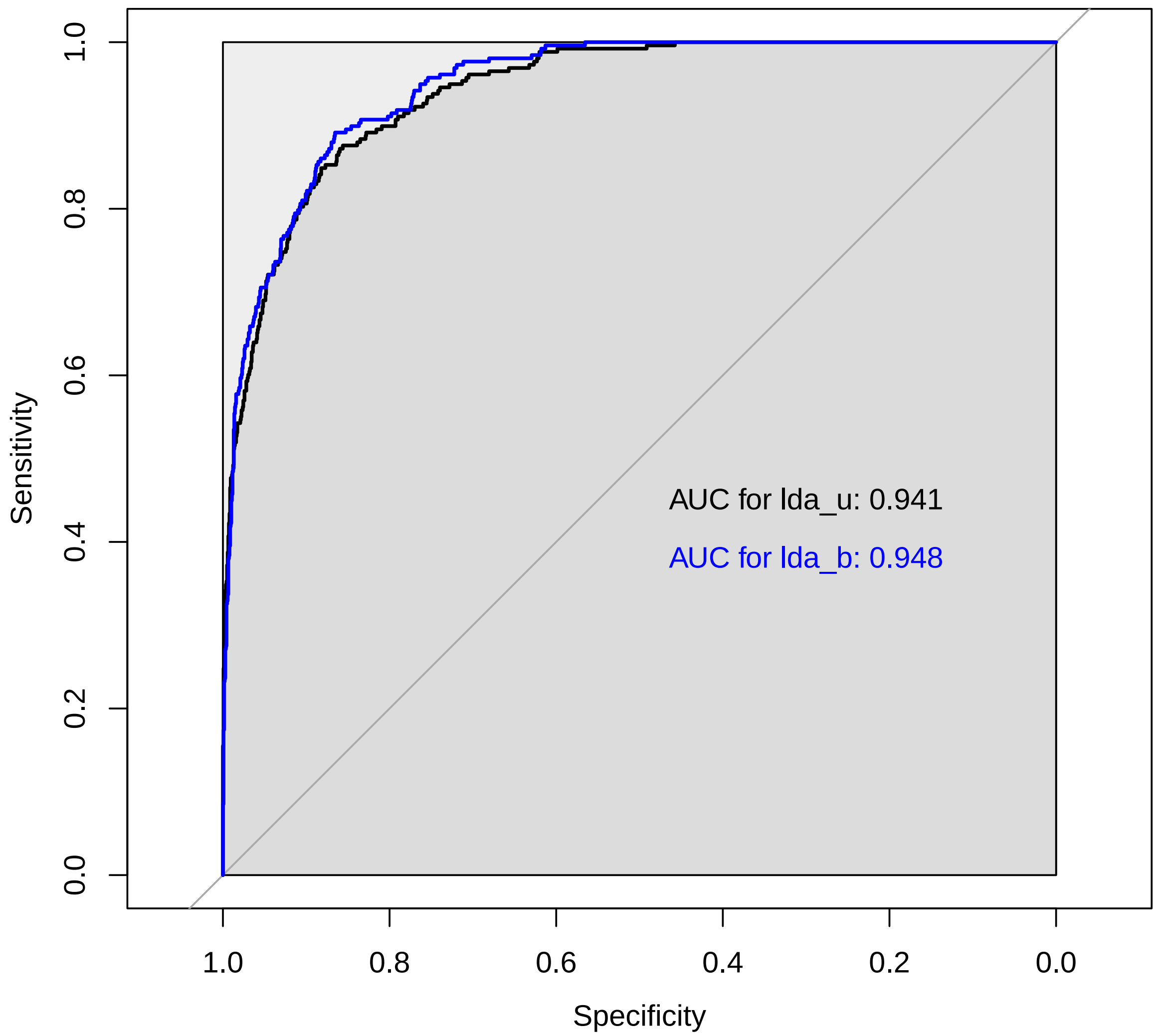
```r
# x indicates the linear combinations of the variables obtained by the model
# class indicates the two classes Existing and Attriting Customers.

ldahist(lda_b_predict$x[,1], g = lda_b_predict$class , col = 2)
```





We now shows the ROC curves and the AUC values corresponding to the LDA models trained on the unbalanced and balanced datasets. We can notice that the AUC value for the Balanced LDA is equal to best models obtained by GLM, even if the F1 score is lower.

```
#ROC curves
roc_lda_u <- roc(test$Attrition_Flag ~ lda_predict_u[,2])
roc_lda_b <- roc(test$Attrition_Flag ~ lda_predict_b[,2])

plot(roc_lda_u, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_lda_b,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for lda_u:", round(roc_lda_u$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for lda_b:", round(roc_lda_b$auc, 3)), col = "blue")
```



Quadratic discriminant analysis

Quadratic discriminant analysis is computationally more expensive than LDA but it is able to capture more complex relationship between predictors and the response variable by using quadratic decision boundaries for each class. It also doesn't assume that the covariance matrices are equal in all the classes.

Unbalanced dataset

Like in LDA, we consider the set of variables and interactions chosen in *glm_7*, beacuse of its performance and the problem of collinearity.

```
qda_u <- qda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count
             + Marital_Status + Income_Category + Total_Relationship_Count
             + Months_Inactive_12_mon + Contacts_Count_12_mon + Credit_Limit
             + Total_Revolving_Bal + Total_Amt_Chng_Q4_Q1
             + Total_Trans_Amt + Total_Trans_Ct + Total_Ct_Chng_Q4_Q1
             + Avg_Utilization_Ratio + Total_Revolving_Bal:Avg_Utilization_Ratio
             + Total_Trans_Amt:Total_Trans_Ct
```

```r
             + Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
             + Is_Female:Total_Trans_Amt
             + Total_Relationship_Count:Total_Trans_Amt
             + Dependent_count:Total_Ct_Chng_Q4_Q1
             + Is_Female:Total_Ct_Chng_Q4_Q1
             + Customer_Age:Marital_Status ,
             data = train, family = "binomial")
qda_u
```

```
## Call:
## qda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Marital_Status + Income_Category + Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Credit_Limit +
##     Total_Revolving_Bal + Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt +
##     Total_Trans_Ct + Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio +
##     Total_Revolving_Bal:Avg_Utilization_Ratio + Total_Trans_Amt:Total_Trans_Ct +
##     Total_Amt_Chng_Q4_Q1:Total_Trans_Amt + Is_Female:Total_Trans_Amt +
##     Total_Relationship_Count:Total_Trans_Amt + Dependent_count:Total_Ct_Chng_Q4_Q1 +
##     Is_Female:Total_Ct_Chng_Q4_Q1 + Customer_Age:Marital_Status,
##     data = train, family = "binomial")
##
## Prior probabilities of groups:
##        0        1
## 0.843169 0.156831
##
## Group means:
##   Customer_Age Is_Female Dependent_count Marital_Status Income_Category
## 0     46.31807 0.4805849        2.333413       1.668744        2.311841
## 1     46.32474 0.5270619        2.364691       1.628866        2.291237
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
## 0                 3.979147               2.287152              2.366970
## 1                 3.295103               2.682990              2.960052
##   Credit_Limit Total_Revolving_Bal Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
## 0     7361.670           1255.4830            0.7750860        4411.586
## 1     6789.984            698.8531            0.6857912        2934.793
##   Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 0       67.17186           0.7425566             0.3211337
## 1       43.68299           0.5469420             0.1803750
##   Total_Revolving_Bal:Avg_Utilization_Ratio Total_Trans_Amt:Total_Trans_Ct
## 0                                  532.4306                       356678.3
## 1                                  335.0325                       155161.9
##   Total_Amt_Chng_Q4_Q1:Total_Trans_Amt Is_Female:Total_Trans_Amt
## 0                             3381.931                  2138.782
## 1                             2206.650                  1426.461
##   Total_Relationship_Count:Total_Trans_Amt Dependent_count:Total_Ct_Chng_Q4_Q1
## 0                                15523.772                            1.746506
## 1                                 9501.402                            1.276539
##   Is_Female:Total_Ct_Chng_Q4_Q1 Customer_Age:Marital_Status
## 0                     0.3618001                    77.34636
## 1                     0.2789472                    75.47423
```

We have that 0.9 is the threshold resulting in the best F1 score and we use it to calculate other significant measures for the model's performance. We noticed that it has a really high Recall, considering the threshold chosen. It seems that the model can recognize with a good accuracy the *Attriting Customers*, at the cost of a small precision.

```
#Threshold
Threshold <- 0.9
qda_u_predict <- predict(qda_u,test,type = "response")
qda_predict_u <- qda_u_predict$posterior
pred_qda_u <- ifelse(qda_predict_u[,2] >= Threshold , 1,0)

#Confusion matrix
c_mat_qda_u <- table(test$Attrition_Flag,pred_qda_u)
```

	Predicted Values		
Real Values	0	1	Total
0	1310	80	1390
1	68	210	190
Total	1378	270	1648

```
#Accuracy

mean(pred_qda_u==test$Attrition_Flag)*100
```

```
## [1] 91.01942
```

```
#True Negative Rate / Specificity

Spec_qda_u <- c_mat_qda_u[1,1]/sum(c_mat_qda_u[1,])
Spec_qda_u
```

```
## [1] 0.942446
```

```
#Precision / Positive Predicted Value

Prec_qda_u <- c_mat_qda_u[2,2]/sum(c_mat_qda_u[,2])
Prec_qda_u
```

```
## [1] 0.7037037
```

```
#Recall / True Positive Rate / Sensitivity

Rec_qda_u <- c_mat_qda_u[2,2]/sum(c_mat_qda_u[2,])
Rec_qda_u
```

```
## [1] 0.7364341
```

```
#F1 Score

F1_qda_u <- 2 * (Prec_qda_u * Rec_qda_u)/(Prec_qda_u + Rec_qda_u)
F1_qda_u
```

```
## [1] 0.719697
```

Balanced dataset

We consider the final set of variables used on the balanced GLM and we define the corresponding qda models.

```
qda_b <- qda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count
             + Education_Level + Marital_Status + Income_Category
             + Total_Relationship_Count + Months_Inactive_12_mon
             + Contacts_Count_12_mon + Total_Revolving_Bal
             + Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt + Total_Trans_Ct
             + Total_Ct_Chng_Q4_Q1  + Avg_Utilization_Ratio
             + Customer_Age:Total_Amt_Chng_Q4_Q1
             + Total_Revolving_Bal:Avg_Utilization_Ratio
             + Total_Trans_Amt:Total_Trans_Ct
             + Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
             + Is_Female:Total_Trans_Amt
             + Total_Relationship_Count:Total_Trans_Amt
             + Dependent_count:Total_Ct_Chng_Q4_Q1
             + Is_Female:Total_Ct_Chng_Q4_Q1
             + Customer_Age:Marital_Status ,
             data = train_bal, family = "binomial")
qda_b
```

```
## Call:
## qda(Attrition_Flag ~ Customer_Age + Is_Female + Dependent_count +
##     Education_Level + Marital_Status + Income_Category + Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Total_Revolving_Bal +
##     Total_Amt_Chng_Q4_Q1 + Total_Trans_Amt + Total_Trans_Ct +
##     Total_Ct_Chng_Q4_Q1 + Avg_Utilization_Ratio + Customer_Age:Total_Amt_Chng_Q4_Q1 +
##     Total_Revolving_Bal:Avg_Utilization_Ratio + Total_Trans_Amt:Total_Trans_Ct +
##     Total_Amt_Chng_Q4_Q1:Total_Trans_Amt + Is_Female:Total_Trans_Amt +
##     Total_Relationship_Count:Total_Trans_Amt + Dependent_count:Total_Ct_Chng_Q4_Q1 +
##     Is_Female:Total_Ct_Chng_Q4_Q1 + Customer_Age:Marital_Status,
##     data = train_bal, family = "binomial")
##
## Prior probabilities of groups:
##         0         1
## 0.4931285 0.5068715
##
## Group means:
##   Customer_Age Is_Female Dependent_count Education_Level Marital_Status
## 0     46.36189 0.4819672        2.320082        3.044672       1.665574
## 1     46.11164 0.5482456        2.319777        3.159888       1.615630
##   Income_Category Total_Relationship_Count Months_Inactive_12_mon
## 0        2.302869                 4.013115               2.311066
## 1        2.227273                 3.393142               2.681818
##   Contacts_Count_12_mon Total_Revolving_Bal Total_Amt_Chng_Q4_Q1
## 0              2.377459           1258.1189            0.7724857
## 1              2.930622            713.8373            0.6829537
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 0        4368.770       67.08115           0.7394672             0.3191320
## 1        2910.393       43.67624           0.5456471             0.1891061
##   Customer_Age:Total_Amt_Chng_Q4_Q1 Total_Revolving_Bal:Avg_Utilization_Ratio
```

```
## 0                          35.63106                                         532.3245
## 1                          31.58837                                         346.6438
##   Total_Trans_Amt:Total_Trans_Ct Total_Amt_Chng_Q4_Q1:Total_Trans_Amt
## 0                      351514.9                                 3346.995
## 1                      153719.1                                 2189.020
##   Is_Female:Total_Trans_Amt Total_Relationship_Count:Total_Trans_Amt
## 0                 2173.387                                15402.223
## 1                 1502.860                                 9809.109
##   Dependent_count:Total_Ct_Chng_Q4_Q1 Is_Female:Total_Ct_Chng_Q4_Q1
## 0                            1.722951                     0.3582635
## 1                            1.250363                     0.2920618
##   Customer_Age:Marital_Status
## 0                    77.27131
## 1                    74.40670
```

The best threshold based on the F1 score is 0.9. We can see that it accentuates the behavior of the unbalanced QDA. In fact, even with threshold equal to 0.9, the Recall is extremly high while the Precision is low. The resulting F1 score is lower in comparison to the model built with the other techniques.

```r
#Threshold
Threshold <- 0.9
qda_b_predict <- predict(qda_b,test,type = "response")
qda_predict_b <- qda_b_predict$posterior
pred_qda_b <- ifelse(qda_predict_b[,2] >= Threshold , 1,0)

#Confusion matrix
c_mat_qda_b <- table(test$Attrition_Flag,pred_qda_b)
```

	Predicted Values		
Real Values	0	1	Total
0	1243	147	1390
1	40	218	258
Total	1283	365	1648

```r
#Accuracy

mean(pred_qda_b==test$Attrition_Flag)*100
```

```
## [1] 88.65291
```

```r
#True Negative Rate / Specificity

Spec_qda_b <- c_mat_qda_b[1,1]/sum(c_mat_qda_b[1,])
Spec_qda_b
```

```
## [1] 0.8942446
```

```r
#Precision / Positive Predicted Value

Prec_qda_b <- c_mat_qda_b[2,2]/sum(c_mat_qda_b[,2])
Prec_qda_b
```

```
## [1] 0.5972603
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_qda_b <- c_mat_qda_b[2,2]/sum(c_mat_qda_b[2,])
Rec_qda_b
```

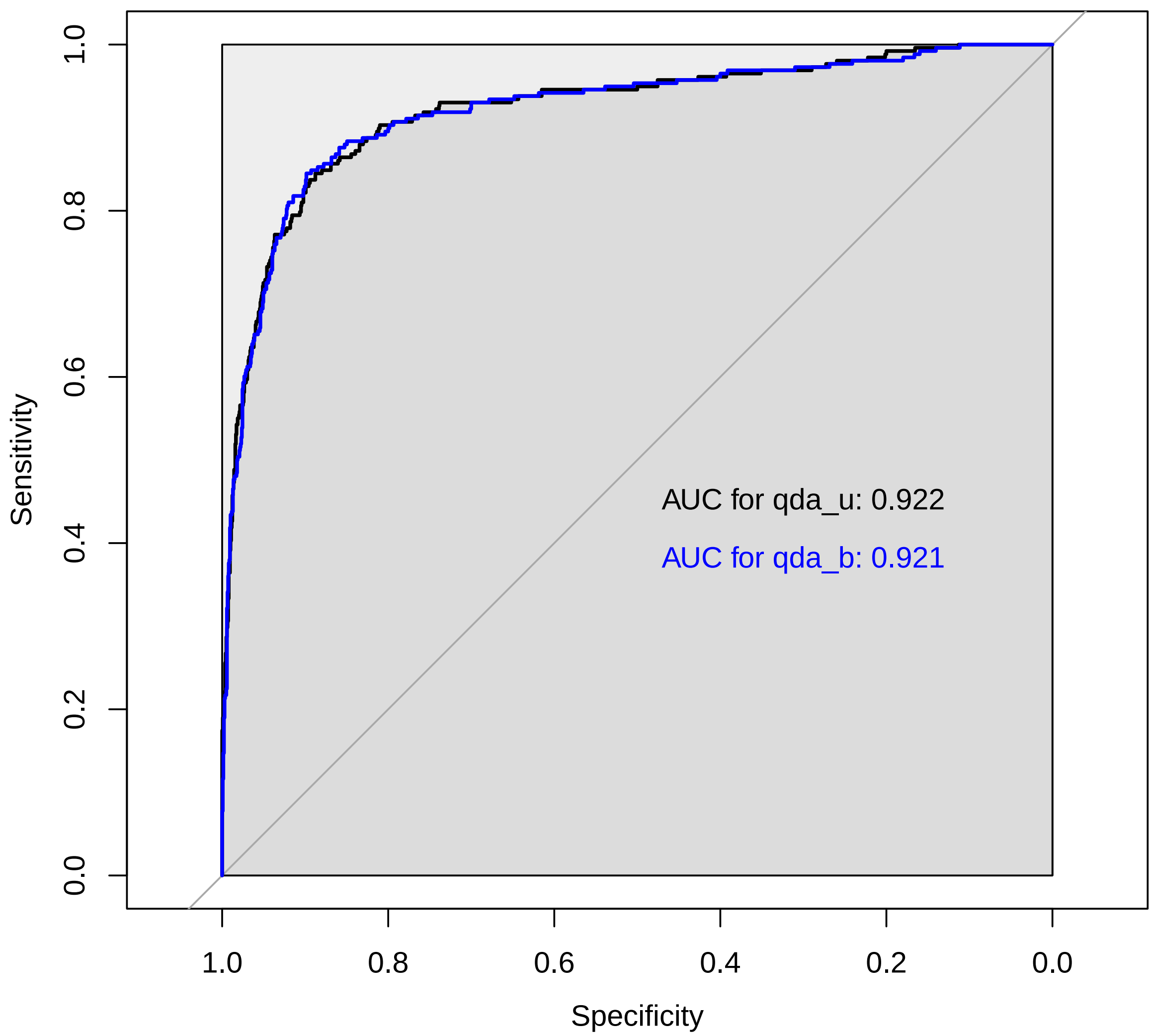
```
## [1] 0.8449612
```

```r
#F1 Score

F1_qda_b <- 2 * (Prec_qda_b * Rec_qda_b)/(Prec_qda_b + Rec_qda_b)
F1_qda_b
```

```
## [1] 0.6998395
```

We plot the ROC curves and corresponding AUC values of the two QDA models. We can notice that the two ROC curve are pretty similar and there is not a curve which is strictly above the other. The AUC values are also much lower in comparison to the other models, except the initial GLM.

```r
#ROC curves
roc_qda_u <- roc(test$Attrition_Flag ~ qda_predict_u[,2])
roc_qda_b <- roc(test$Attrition_Flag ~ qda_predict_b[,2])

plot(roc_qda_u, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_qda_b,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for qda_u:", round(roc_qda_u$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for qda_b:", round(roc_qda_b$auc, 3)), col = "blue")
```



Shrinkage methods

We now analyze the models obtained by using *Ridge* and *Lasso*, two famous shrinkage methods. A shrinkage method, also called Regularization, is used to train models that generalize better on unseen data, in our case in the test dataset. It do so by shrinking the coefficients estimates towards 0, preventing the model from overfitting. The two shrinking methods that we are considering, *Ridge regression* and *Lasso regression*, works by addind a penalization term to the ordinary least square (OLS) function, which is the objective function of linear regression. In this way they reduce the impact of collinearity and prevent overfitting. Since collinearity is a problem in our models, we explore the performances of these models.

To begin with, we define a new training and test set obtained by adding to the original datasets the interactions term as variables and we will later transform them into matrices. This is done since we will use the function *cv.glmnet*, from the *glmnet* library. In fact this function takes a matrix as an input and to include the interactions we will then need to create new variables.

```r
#Creation of train and test set with interactions
train_int <- data.frame(train)

train_int$Total_Revolving_Bal_Avg_Utilization_Ratio <-
    train$Total_Revolving_Bal * train$Avg_Utilization_Ratio
train_int$Total_Trans_Amt_Total_Trans_Ct <-
  train$Total_Trans_Amt * train$Total_Trans_Ct
train_int$Total_Trans_Amt_Total_Amt_Chng_Q4_Q1 <-
  train$Total_Trans_Amt * train$Total_Amt_Chng_Q4_Q1
train_int$Total_Trans_Amt_Is_Female <-
  train$Total_Trans_Amt * train$Is_Female
train_int$Total_Trans_Amt_Total_Relationship_Count <-
  train$Total_Trans_Amt * train$Total_Relationship_Count
train_int$Dependant_count_Total_Ct_Chng_Q4_Q1 <-
    train$Dependent_count * train$Total_Ct_Chng_Q4_Q1
train_int$Is_Female_Total_Ct_Chng_Q4_Q1 <-
    train$Is_Female * train$Total_Ct_Chng_Q4_Q1
train_int$Customer_Age_Marital_Status <-
    train$Customer_Age * train$Marital_Status

test_int <- data.frame(test)

test_int$Total_Revolving_Bal_Avg_Utilization_Ratio <-
    test$Total_Revolving_Bal * test$Avg_Utilization_Ratio
test_int$Total_Trans_Amt_Total_Trans_Ct <-
  test$Total_Trans_Amt * test$Total_Trans_Ct
test_int$Total_Trans_Amt_Total_Amt_Chng_Q4_Q1 <-
  test$Total_Trans_Amt * test$Total_Amt_Chng_Q4_Q1
test_int$Total_Trans_Amt_Is_Female <-
  test$Total_Trans_Amt * test$Is_Female
test_int$Total_Trans_Amt_Total_Relationship_Count <-
  test$Total_Trans_Amt * test$Total_Relationship_Count
test_int$Dependant_count_Total_Ct_Chng_Q4_Q1 <-
    test$Dependent_count * test$Total_Ct_Chng_Q4_Q1
test_int$Is_Female_Total_Ct_Chng_Q4_Q1 <-
    test$Is_Female * test$Total_Ct_Chng_Q4_Q1
test_int$Customer_Age_Marital_Status <-
    test$Customer_Age * test$Marital_Status
```

We do the same for the balanced training set. We don't need to do another matrix for the test set since the interactions are the same as in the unbalanced

```r
#Creation of train_bal with interactions
train_bal_int <- data.frame(train_bal)

train_bal_int$Total_Revolving_Bal_Avg_Utilization_Ratio <-
    train_bal$Total_Revolving_Bal * train_bal$Avg_Utilization_Ratio
train_bal_int$Total_Trans_Amt_Total_Trans_Ct <-
  train_bal$Total_Trans_Amt * train_bal$Total_Trans_Ct
train_bal_int$Total_Trans_Amt_Total_Amt_Chng_Q4_Q1 <-
  train_bal$Total_Trans_Amt * train_bal$Total_Amt_Chng_Q4_Q1
train_bal_int$Total_Trans_Amt_Is_Female <-
  train_bal$Total_Trans_Amt * train_bal$Is_Female
```

```
train_bal_int$Total_Trans_Amt_Total_Relationship_Count <-
  train_bal$Total_Trans_Amt * train_bal$Total_Relationship_Count
train_bal_int$Dependant_count_Total_Ct_Chng_Q4_Q1 <-
    train_bal$Dependent_count * train_bal$Total_Ct_Chng_Q4_Q1
train_bal_int$Is_Female_Total_Trans_Ct_Q4_Q1 <-
    train_bal$Is_Female * train_bal$Total_Ct_Chng_Q4_Q1
train_bal_int$Customer_Age_Marital_Status <-
    train_bal$Customer_Age * train_bal$Marital_Status
```

Ridge regression

Ridge regression adds a penalty term to the objective function that is proportional to the L2 norm of the vector of the coefficients, shrinking them towards 0. However, since it doesn't actually set them to zero, we will consider the set of variables which gave the best performing GLM.

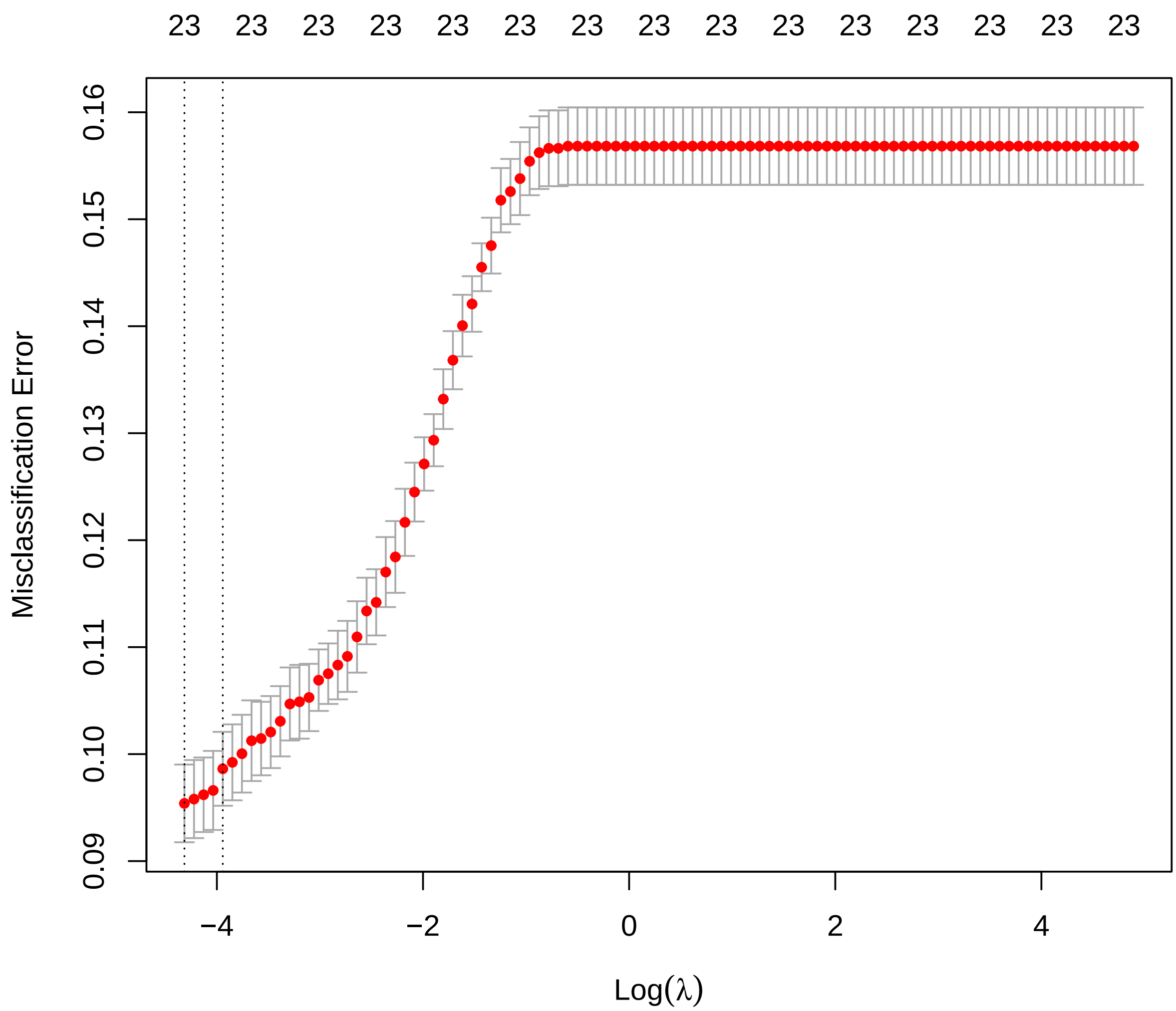
Unbalanced dataset

We create the matrix that we will use for the model

```
# Matrix without Attrition_Flag, Education_Level , Months_on_book and Avg_Open_To_Buy
train_mat <- data.matrix(train_int[,-c(1,5,8,14)])
test_mat <- data.matrix(test_int[,-c(1,5,8,14)])
```

We apply Ridge logistic regression to our matrix and we extract the optimal plot the misclassification error, which depends on the values of $\log(\lambda)$, where λ is the parameter that controls the amount of penalization. We can notice that the increase of the penalization leads to an increase of the misclassification error based on the minimum misclassification error. Finally we extract the value of λ that minimizes the misclassification error.

```
ridge_u <- cv.glmnet(train_mat,train$Attrition_Flag, alpha = 0,
                     family = "binomial", type.measure = "class")
plot(ridge_u)
```




```
lambda_ridge_u <- ridge_u$lambda.min
lambda_ridge_u
```

```
## [1] 0.01336669
```

The best threshold for the F1 score is 0.3. It seems that the model has a worse performance in comparison to the models previously analyzed.

```
#Threshold
Threshold <- 0.3
ridge_predict_u <- predict(ridge_u,test_mat,type = "response", s = lambda_ridge_u)
pred_ridge_u <- ifelse(ridge_predict_u >= Threshold , 1,0)

#Confusion matrix
c_mat_ridge_u <- table(test$Attrition_Flag,pred_ridge_u)
c_mat_ridge_u
```

```
##    pred_ridge_u
##        0    1
##   0 1296   94
##   1   77  181
```

Predicted Values			
Real Values	0	1	Total
0	1296	94	1390
1	77	181	258
Total	1373	275	1648

```
#Accuracy

mean(pred_ridge_u==test$Attrition_Flag)*100
```

```
## [1] 89.62379
```

```
#True Negative Rate / Specificity

Spec_ridge_u <- c_mat_ridge_u[1,1]/sum(c_mat_ridge_u[1,])
Spec_ridge_u
```

```
## [1] 0.9323741
```

```
#Precision / Positive Predicted Value

Prec_ridge_u <- c_mat_ridge_u[2,2]/sum(c_mat_ridge_u[,2])
Prec_ridge_u
```

```
## [1] 0.6581818
```

```
#Recall / True Positive Rate / Sensitivity

Rec_ridge_u <- c_mat_ridge_u[2,2]/sum(c_mat_ridge_u[2,])
Rec_ridge_u
```

```
## [1] 0.7015504
```

```
#F1 Score

F1_ridge_u <- 2 * (Prec_ridge_u * Rec_ridge_u)/(Prec_ridge_u + Rec_ridge_u)
F1_ridge_u
```

```
## [1] 0.6791745
```

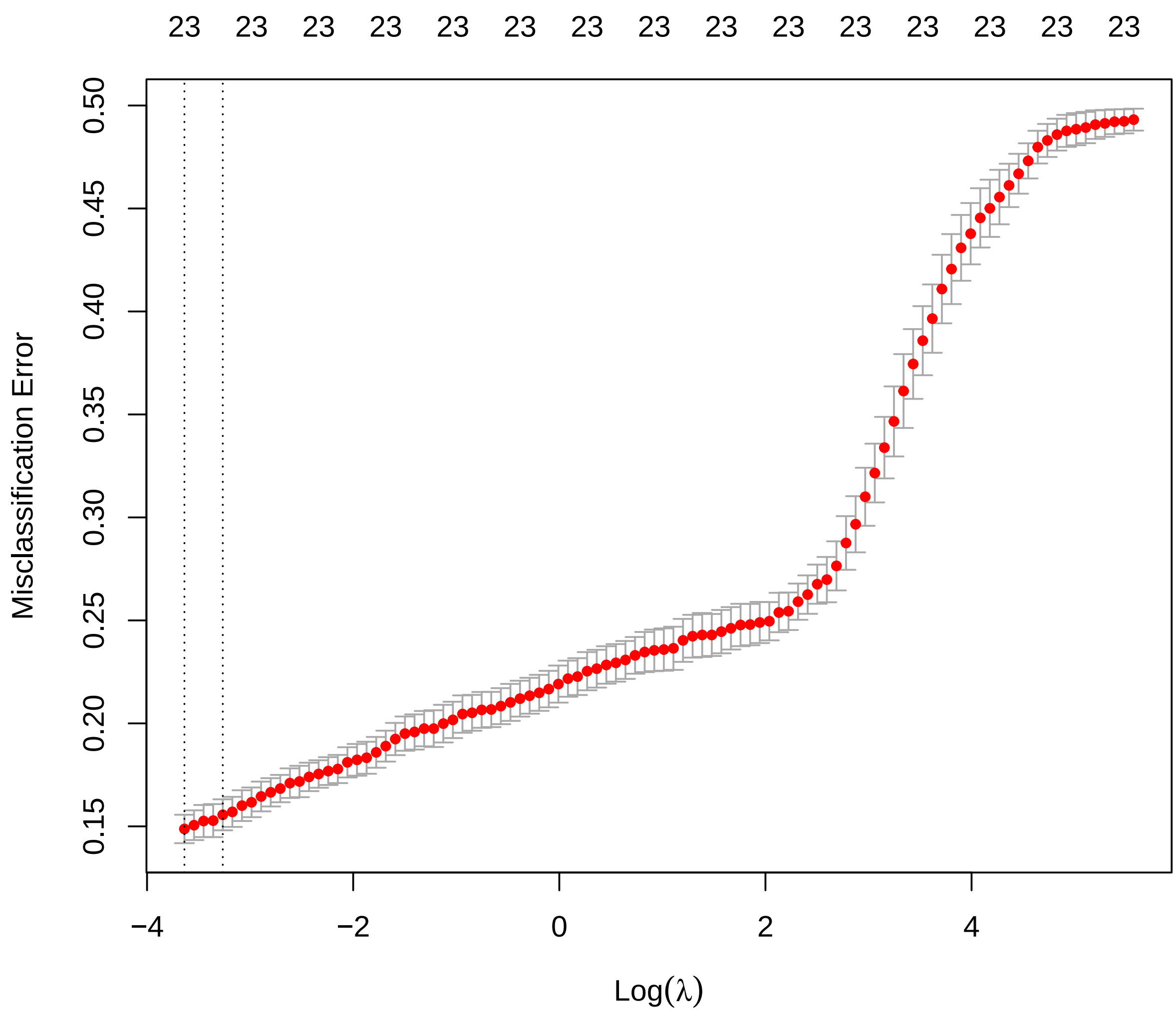
Balanced dataset

We create the matrix that we will use for the model

```r
# Matrix without Attrition_Flag, Months_on_book,Credit_Limit and Avg_Open_To_Buy
train_mat_b <- data.matrix(train_bal_int[,-c(1,8,12,14)])
test_mat_b <- data.matrix(test_int[,-c(1,8,12,14)])
```

We apply the same step done with the balanced step. We can notice that, like in the previous case the increase of the penalization leads to an increase of the misclassification error. Moreover for big values of λ the misclassification error gets close to 0.5.

```r
ridge_b <- cv.glmnet(train_mat_b,train_bal$Attrition_Flag, alpha = 0,
                     family = "binomial", type.measure = "class")
plot(ridge_b)
```



Finally we extract the optimal lambda based on the minimum misclassification error.

```r
lambda_ridge_b <- ridge_b$lambda.min
lambda_ridge_b
```

```
## [1] 0.0263209
```

The best threshold for the F1 score is 0.6. While it has a better Recall that the unbalanced model, it seems that its performance is much worse in comparison to the other models.

```r
#Threshold
Threshold <- 0.6
ridge_predict_b <- predict(ridge_b,test_mat_b,type = "response", s = lambda_ridge_b)
pred_ridge_b <- ifelse(ridge_predict_b >= Threshold , 1,0)

#Confusion matrix
c_mat_ridge_b <- table(test$Attrition_Flag,pred_ridge_b)
```

	Predicted Values		
Real Values	0	1	Total
0	1257	133	1390
1	62	196	258
Total	1319	329	1648

```r
#Accuracy

mean(pred_ridge_b==test$Attrition_Flag)*100
```

```
## [1] 88.16748
```

```r
#True Negative Rate / Specificity

Spec_ridge_b <- c_mat_ridge_b[1,1]/sum(c_mat_ridge_b[1,])
Spec_ridge_b
```

```
## [1] 0.9043165
```

```r
#Precision / Positive Predicted Value

Prec_ridge_b <- c_mat_ridge_b[2,2]/sum(c_mat_ridge_b[,2])
Prec_ridge_b
```

```
## [1] 0.5957447
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_ridge_b <- c_mat_ridge_b[2,2]/sum(c_mat_ridge_b[2,])
Rec_ridge_b
```

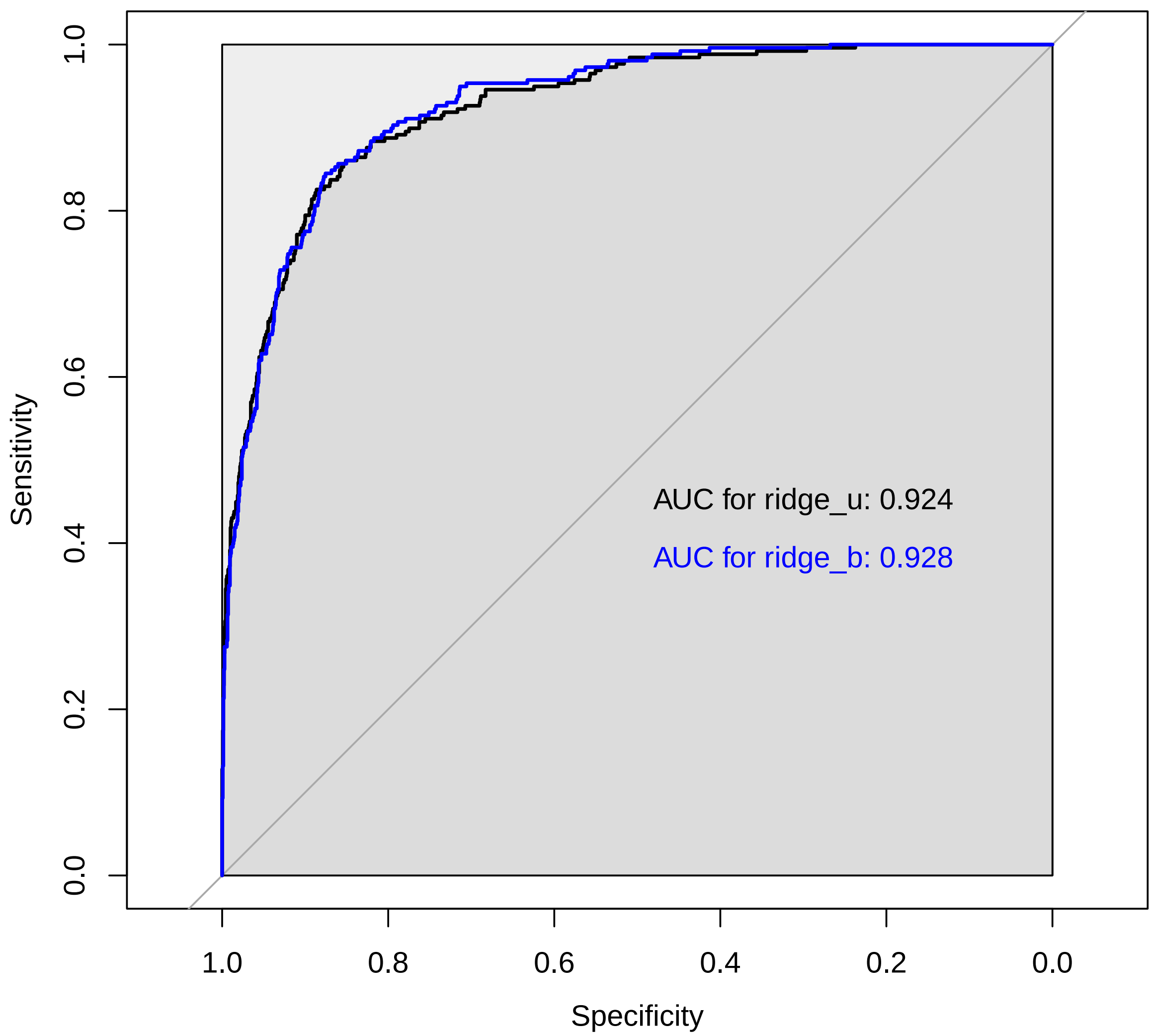
```
## [1] 0.7596899
```

```r
#F1 Score

F1_ridge_b <- 2 * (Prec_ridge_b * Rec_ridge_b)/(Prec_ridge_b + Rec_ridge_b)
F1_ridge_b
```

```
## [1] 0.6678024
```

We plot the ROC curves and corresponding AUC values of the two models. Like the F1 score, we have the AUC values are much lower compared to the models obtained with the other techniques, except for the QDA. Even if they have a better AUC than QDA, because of their poor F1 score, these two model are a worse choice for our problem. It seems that adding the L2 norm as a penalization term leads to a worse model in terms of performance.

```r
#ROC curves
roc_ridge_u <- roc(test$Attrition_Flag ~ as.numeric(ridge_predict_u))
roc_ridge_b <- roc(test$Attrition_Flag ~ as.numeric(ridge_predict_b))

plot(roc_ridge_u, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_ridge_b,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for ridge_u:", round(roc_ridge_u$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for ridge_b:", round(roc_ridge_b$auc, 3)), col = "blue")
```



Lasso regression

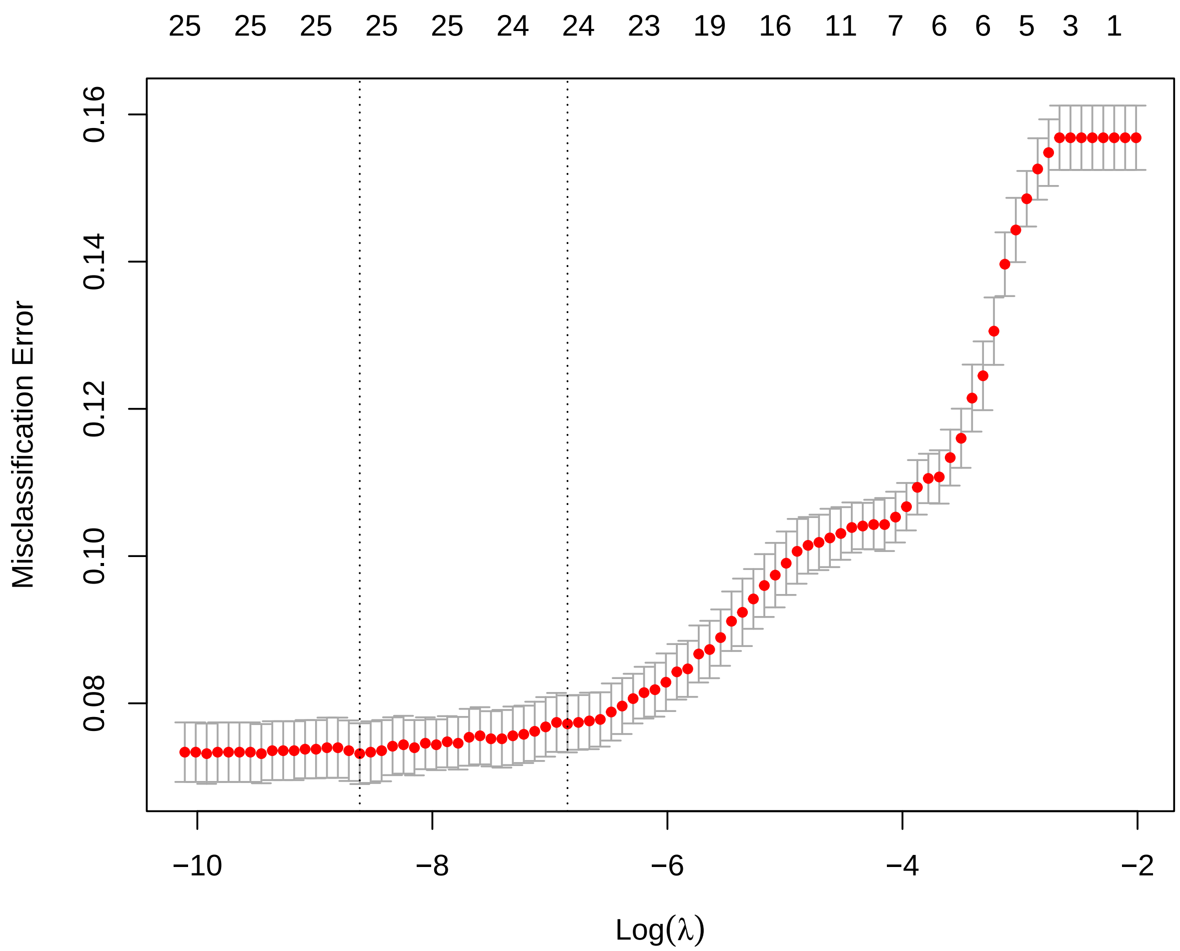
Lasso regression adds the L1 norm of the vector of the coefficients as a penalty term to the objective function. By doing so it shrinks the coefficients towards 0 and can set some coefficient exactly equal to 0. Thanks to this property, Lasso regression can identify and exclude the predictors which do not contribute to the predictions of the model. Because of this property we consider the set of all the predictor variables in addition to the interaction used in the GLM.

Unbalanced dataset

```
# Matrix without Attrition_Flag and Avg_Open_To_Buy
train_mat <- data.matrix(train_int[,-c(1)])
test_mat <- data.matrix(test_int[,-c(1)])
```

Like for the Ridge models, we plot the misclassification error in function of the values of $\log(\lambda)$.

```r
lasso_u <- cv.glmnet(train_mat,train$Attrition_Flag, alpha = 1,
                     family = "binomial", type.measure = "class")
plot(lasso_u)
```



We pick λ so that it minimizes the misclassification error.

```r
lambda_lasso_u <- lasso_u$lambda.min
lambda_lasso_u
```

```
## [1] 0.0001808576
```

The best threshold for the F1 score is 0.4. Unlike with Ridge, the model's performance seems comparable to the models obtained with stepwise selection and discriminant analyis.

```r
#Threshold
Threshold <- 0.4
```

```r
lasso_predict_u <- predict(lasso_u,test_mat,type = "response", s = lambda_lasso_u)
pred_lasso_u <- ifelse(lasso_predict_u >= Threshold , 1,0)

#Confusion matrix
c_mat_lasso_u <- table(test$Attrition_Flag,pred_lasso_u)
```

	Predicted Values		
Real Values	0	1	Total
0	1322	68	1390
1	73	185	258
Total	1395	253	1648

```r
#Accuracy

mean(pred_lasso_u==test$Attrition_Flag)*100
```

```
## [1] 91.44417
```

```r
#True Negative Rate / Specificity

Spec_lasso_u <- c_mat_lasso_u[1,1]/sum(c_mat_lasso_u[1,])
Spec_lasso_u
```

```
## [1] 0.9510791
```

```r
#Precision / Positive Predicted Value

Prec_lasso_u <- c_mat_lasso_u[2,2]/sum(c_mat_lasso_u[,2])
Prec_lasso_u
```

```
## [1] 0.7312253
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_lasso_u <- c_mat_lasso_u[2,2]/sum(c_mat_lasso_u[2,])
Rec_lasso_u
```

```
## [1] 0.7170543
```

```r
#F1 Score

F1_lasso_u <- 2 * (Prec_lasso_u * Rec_lasso_u)/(Prec_lasso_u + Rec_lasso_u)
F1_lasso_u
```

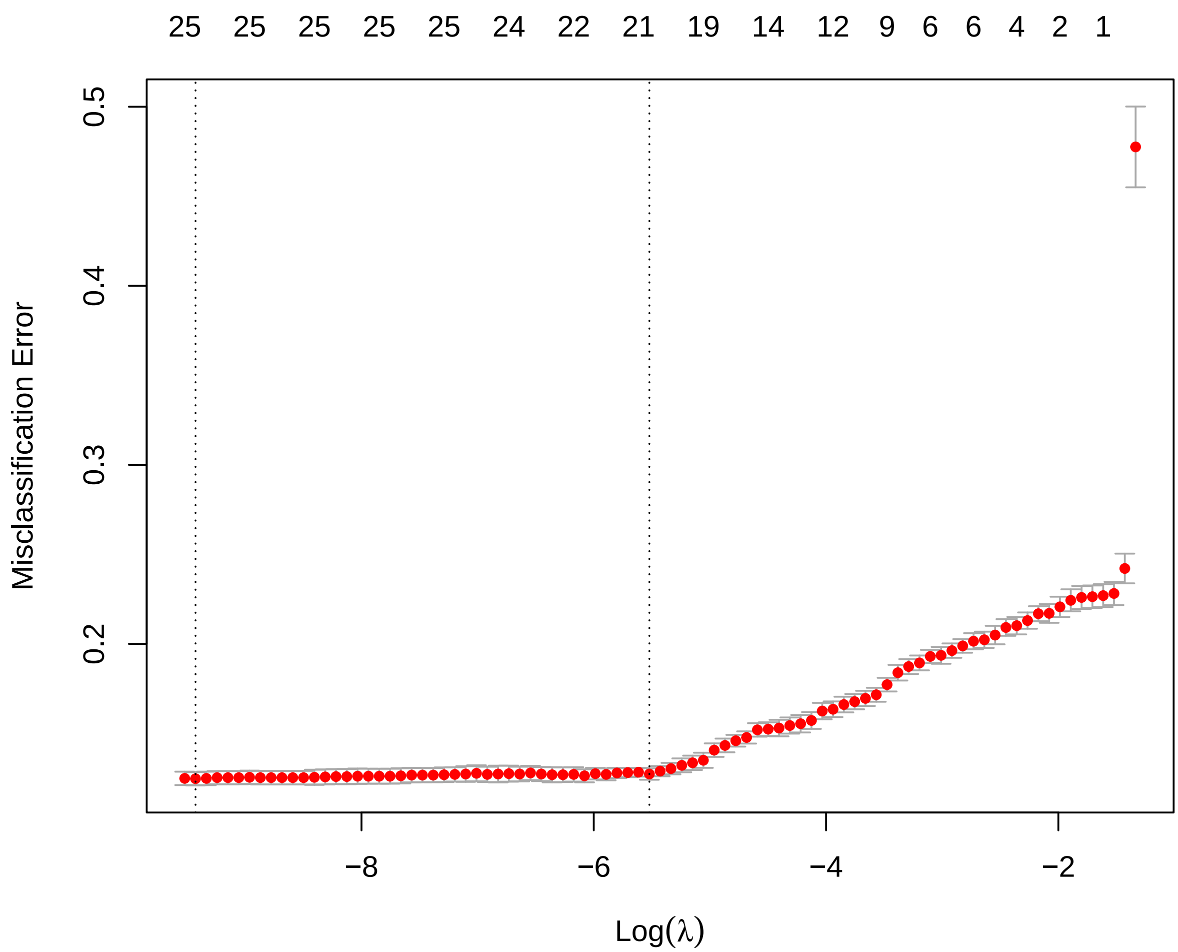
```
## [1] 0.7240705
```

Balanced dataset

```r
# Matrix without Attrition_Flag
train_mat_b <- data.matrix(train_bal_int[,-c(1)])
test_mat_b <- data.matrix(test_int[,-c(1)])
```

We shows the values of the misclassification error depending on $\log(\lambda)$.

```r
lasso_b <- cv.glmnet(train_mat_b,train_bal$Attrition_Flag, alpha = 1,
                     family = "binomial", type.measure = "class")
plot(lasso_b)
```



We report the optimal lambda based on the minimum misclassification error that we will use to evaluate the model performance.

```r
lambda_lasso_b <- lasso_b$lambda.min
lambda_lasso_b
```

```
## [1] 8.038022e-05
```

After choosing the best threshold we compute some useful test for the model's performance. We have that the F1 score is better than the ones obtained with Ridge but lower than the one obtained with the unbalanced set.

```r
#Threshold
Threshold <- 0.7
lasso_predict_b <- predict(lasso_b,test_mat_b,type = "response", s = lambda_lasso_b)
pred_lasso_b <- ifelse(lasso_predict_b >= Threshold , 1,0)

#Confusion matrix

c_mat_lasso_b <- table(test$Attrition_Flag,pred_lasso_b)
```

	Predicted Values		
Real Values	0	1	Total
0	1283	107	1390
1	56	202	258
Total	1339	309	1648

```r
#Accuracy

mean(pred_lasso_b==test$Attrition_Flag)*100
```

```
## [1] 90.10922
```

```r
#True Negative Rate / Specificity

Spec_lasso_b <- c_mat_lasso_b[1,1]/sum(c_mat_lasso_b[1,])
Spec_lasso_b
```

```
## [1] 0.9230216
```

```r
#Precision / Positive Predicted Value

Prec_lasso_b <- c_mat_lasso_b[2,2]/sum(c_mat_lasso_b[,2])
Prec_lasso_b
```

```
## [1] 0.6537217
```

```r
#Recall / True Positive Rate / Sensitivity

Rec_lasso_b <- c_mat_lasso_b[2,2]/sum(c_mat_lasso_b[2,])
Rec_lasso_b
```

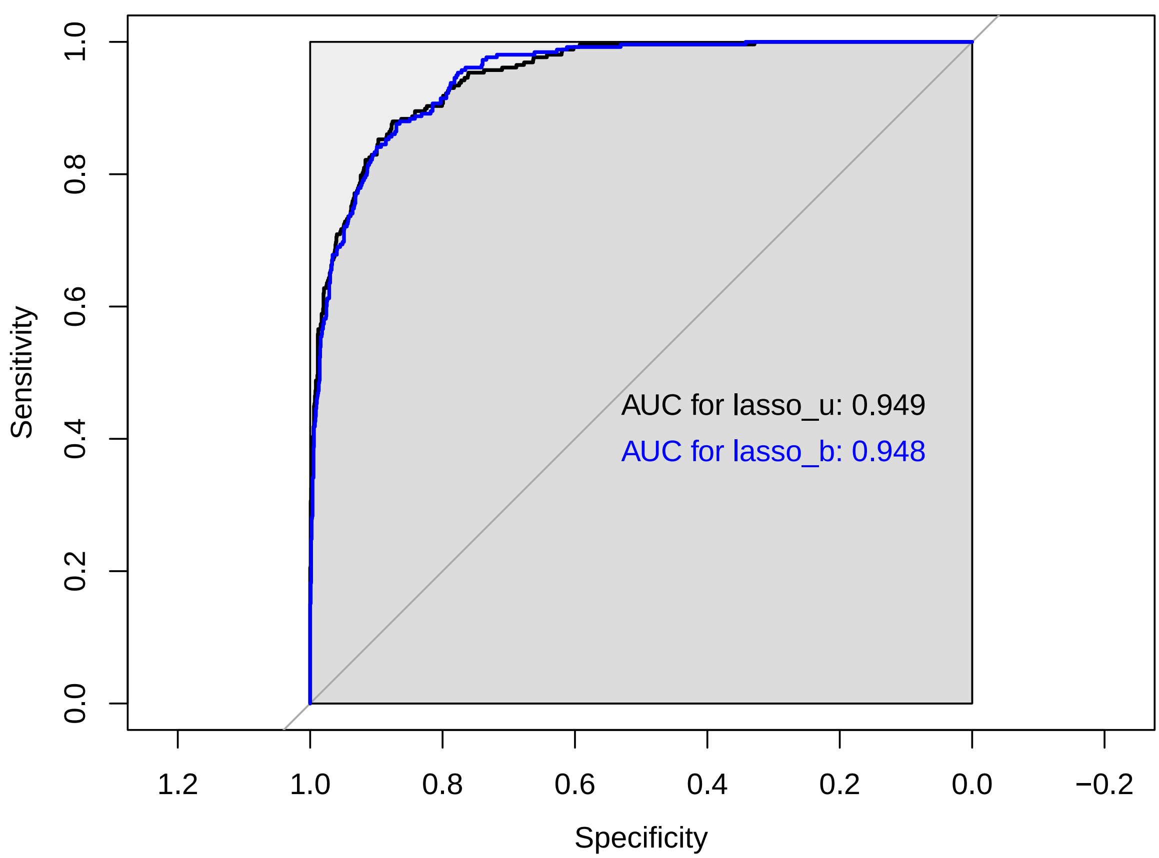
```
## [1] 0.7829457
```

```r
#F1 Score

F1_lasso_b <- 2 * (Prec_lasso_b * Rec_lasso_b)/(Prec_lasso_b + Rec_lasso_b)
F1_lasso_b
```

```
## [1] 0.712522
```

We plot the ROC curves and corresponding AUC values of the two models. We notice that the ROC curves and the AUC values are comparable to the best other models analyzed, however, since the balanced model had a higher F1 score it is a better choice for our problem.

```
#ROC curves
roc_lasso_u <- roc(test$Attrition_Flag ~ as.numeric(lasso_predict_u))
roc_lasso_b <- roc(test$Attrition_Flag ~ as.numeric(lasso_predict_b))

plot(roc_lasso_u, col = "black",print.auc = FALSE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, lwd=2)
plot(roc_lasso_b,add = TRUE,col = "blue", print.auc = FALSE, lwd=2)
text(0.3, 0.45, paste("AUC for lasso_u:", round(roc_lasso_u$auc, 3)), col = "black")
text(0.3, 0.38, paste("AUC for lasso_b:", round(roc_lasso_b$auc, 3)), col = "blue")
```



Models comparison and conclusions

We are now going to summarize the results obtained with the different models. We report in two different tables the best F1 score, the corresponding Recall and the the AUC value associated with the models fitted in the unbalanced and balanced datasets.

Unbalanced dataset

We denote with *GLM_i* the GLM with all the independent variables considered and with *GLM_f* the GLM with the addiction of the interaction and the use of stepwise selection. We report the Recall of each value since, in case of models with similar performance, we prioritize identifying the Attriting Customer over penalizing Customers who weren't going to churn.

Model	*F1*	*Recall*	*AUC*
GLM_i	0.667	0.705	0.913
GLM_f	0.726	0.775	0.948
LDA	0.705	0.690	0.941
QDA	0.720	0.736	0.921
RIDGE	0.679	0.702	0.924
LASSO	0.724	0.717	0.949

We will consider as the best model the one obtained with GLM, since it has the highest F1 score and Recall, and the second highest AUC values. We also notice how Lasso regression leads to a model with a similar AUC values and F1 score as the final GLM but lower accuracy. It could then be a good choice if we prefer to involve less customers who didn't have the intention to churn. Another interesting model is the one obtained with QDA that, while having low AUC, it has good values of F1 score and Recall. Moreover we can assert that Ridge regression is not a good choice for our problem since its performance is comparable to the complete GLM, the GLM with all the predictors.

Balanced dataset

We apply the same notation used in the Unbalanced models.

Model	*F1*	*Recall*	*AUC*
GLM_i	0.650	0.709	0.915
GLM_f	0.715	0.709	0.948
LDA	0.712	0.709	0.948
QDA	0.700	0.845	0.921
RIDGE	0.668	0.760	0.928
LASSO	0.712	0.783	0.948

As you can except from models fitted on balanced data, the Recall tends to be higher then the ones obtained from the unbalanced models. The AUC values obtained, instead, are similar or even greater than the corresponding models fitted on the unbalanced training set. However theirs F1 score are all lower of their counterpart, with the exception of the LDA, which is still lower than the best unbalanced models. For this reason, we pick the unbalanced *GLM_f* as the best overall model for our problem.

Conclusions

Like we said at the beginning of the model's definition, we aim to discover customers who are going to churn without bothering too many customer who didn't plan to leave the bank. To do so we have chosen the F1 score as the most significant measure of the model's performance. After analyzing different models in a balanced dataset and in a unbalanced dataset, we chose *glm_f* as the best model to address our problem. In fact *glm_f* had the best F1 score and one of the highest AUC value and Recall between all the models considered.