Customer Intelligence and Big Data

Alan Rijnders and Lorenzo Severi

11/4/2021

1. Introduction

We, Alan Rijnders and Lorenzo Severi have been appointed from Alpha Bank to help tackling one of the biggest problems physical banks are facing now, being the significant number of customers shifting towards online financial institutions, such as N26 and illimity Bank.

If Alpha Bank aims to survive in the financial market in the long run, a proactive way of predicting the customers willing to leave their service is essential. In fact, by discovering beforehand those customers, Alpha Bank will be in the position to find solutions aimed at better suiting their needs by eventually persuading them to remain.

One might argue that some scholars have recently suggested companies avoiding focusing just on retaining customers; long-term customers seem indeed to be extremely costly, as they tend to expect lower prices in exchange for their loyalty (Anderson & Jap, 2005). Nonetheless, it is common knowledge that, if feasible, companies should find ways to persuade them to remain, but without forgetting to keep an eye on their profitability.

2. Methodology

The database provided consist of labeled data, meaning that it represents a case of supervised learning. As we will be dealing with a prediction of a binary category (i.e., "0 à remain" or "1 à leave"), the problem involves classification, as the outcome variable will not be continuous. During our classes, we have seen several statistical methods suitable for this kind of data, namely Logistic Regression, Decision Tree and Random Forest.

Logistic regression is a predictive analysis used to explain the relationship between one non-metric dependent variable and one (or several) nominal, ordinal, interval or ratio independent variables. Instead of predicting the dependent variable, this technique predicts the probability of the dependent variable to be true (Statistics Solutions, n.d.; Edgar & Manz, 2017).

Decision trees (also known as CART trees) are graphic representation of different alternative solutions that are suitable to tackle a problem. It is usually used to determine the most effective courses of action. All decision trees start with a single node that is connected through branches into possible outcomes. All the different outcomes lead to additional nodes which branch off into other possibilities (OmniSci, n.d.; Lucidchart, n.d.).

Random forest is a collection of decision trees trained with a bootstrapped technique, so that the ensemble boosts the accuracy of the decision trees outcome (Donges, 2021).

3. Data preparation

To be able to analyze the dataset, we first need to set the working directory where the file is located, and subsequently to open the .csv file. Then, we need to install and call in several free libraries, as they are required for conducting our analysis. With regards to the library "caret", it is important to install it by means of the dependencies, as we need them in order to calculate sensitivity and specificity.

```
#read in data
data <- read.csv("ch.csv")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```
library(ggplot2)
#install.packages('caret', dependencies = TRUE)
#install with dependencies = TRUE is important for the calculation of sensitivity and specificity
library(caret)
```

```
## Loading required package: lattice
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(repr)
library(caTools)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(rpart)
library(rpart.plot)
library(ggpubr)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
```

```
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

By briefly looking at the variables, we see how some need to be re-recorded. What is more, we considered the first two columns of our database (i.e., "X" and "CLIENTUM") as neglectable and thus we decided to crop them to obtain a cleaner dataset. Then, we proceed with the transformation of several variables (i.e., "Gender", "Education_Level", "Marital_Status", "Income_Category" and "Card_Category") into factor variables, so that we will be able to analyze them accurately. The last step is to record the variable "Attrition_Flag", which contains the two levels "Existing Customer" and "Attrited Customer", into a binary variable, with the value of "0" and "1" respectively. As there seems to be no possible detrimental errors in the dataset (e.g., non-response error, out-of-range error, ...), we will be further proceeding by assuming that the quantitative inferences we will take on the outcome will not be unvalidated.

```
#we drop the X and client number column
data <- subset(data, select=-c(X,CLIENTNUM))
```

```
#summary of the dataset
summary(data)
```

```
##  Attrition_Flag      Customer_Age      Gender           Dependent_count
##  Length:10127       Min.   :26.00   Length:10127       Min.   :0.000
##  Class :character   1st Qu.:41.00   Class :character   1st Qu.:1.000
##  Mode  :character   Median :46.00   Mode  :character   Median :2.000
##                     Mean   :46.33                      Mean   :2.346
##                     3rd Qu.:52.00                      3rd Qu.:3.000
##                     Max.   :73.00                      Max.   :5.000
##  Education_Level    Marital_Status     Income_Category    Card_Category
##  Length:10127       Length:10127       Length:10127       Length:10127
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Months_on_book  Total_Relationship_Count Months_Inactive_12_mon
##  Min.   :13.00   Min.   :1.000           Min.   :0.000
##  1st Qu.:31.00   1st Qu.:3.000           1st Qu.:2.000
##  Median :36.00   Median :4.000           Median :2.000
##  Mean   :35.93   Mean   :3.813           Mean   :2.341
##  3rd Qu.:40.00   3rd Qu.:5.000           3rd Qu.:3.000
##  Max.   :56.00   Max.   :6.000           Max.   :6.000
##  Contacts_Count_12_mon  Credit_Limit   Total_Trans_Amt Total_Trans_Ct
##  Min.   :0.000         Min.   : 1438   Min.   :  510   Min.   : 10.00
##  1st Qu.:2.000         1st Qu.: 2555   1st Qu.: 2156   1st Qu.: 45.00
##  Median :2.000         Median : 4549   Median : 3899   Median : 67.00
##  Mean   :2.455         Mean   : 8632   Mean   : 4404   Mean   : 64.86
##  3rd Qu.:3.000         3rd Qu.:11068   3rd Qu.: 4741   3rd Qu.: 81.00
##  Max.   :6.000         Max.   :34516   Max.   :18484   Max.   :139.00
##  Avg_Utilization_Ratio
##  Min.   :0.0000
##  1st Qu.:0.0230
##  Median :0.1760
##  Mean   :0.2749
##  3rd Qu.:0.5030
```

```
##  Max.   :0.9990
```

Data exploration

The first step of data exploration consists of reading and visualizing the data: there were 10,127 observations with 18 variables, ranging from non-metric to metric ones.

With regards to the socio-demographic characteristics, the average age was forty-six, with a minimum age of twenty-six and a maximum age of seventy-three, and the database consists of 52.9% female (5358) and 47.1% males (4769). With regards to the number of dependents, the average is slightly more than two people, with a maximum of five. 70.3% of the customers obtained at least an high-school diploma, but a significant 14.7% is not educated, and 15% have an unknown status. Concerning the marital status, 46.3% of people are married, 38.9% are single and the remaining 14.8% are either divorced or with an unknown status. The income category is considerable diversified: 35.2% has an annual income that is less than $40k, 17.7% $40k - $60k, 13.8% $60k - $80k, 15.2% $80 $120k, 7.2% more than $120k, and 11% have an unknown income status.

Concerning the variables related to the product, the data depicts that more than 93% of the customers have blue card, the average time of bank-customer relationship is 35 months with an average of almost four active products. In the graphs below, we show the different distribution of the aforementioned variables by means of their status with the Alpha Bank (i.e., "Attrited customer" vs "Existing customer").

```
table(data$Attrition_Flag)
```

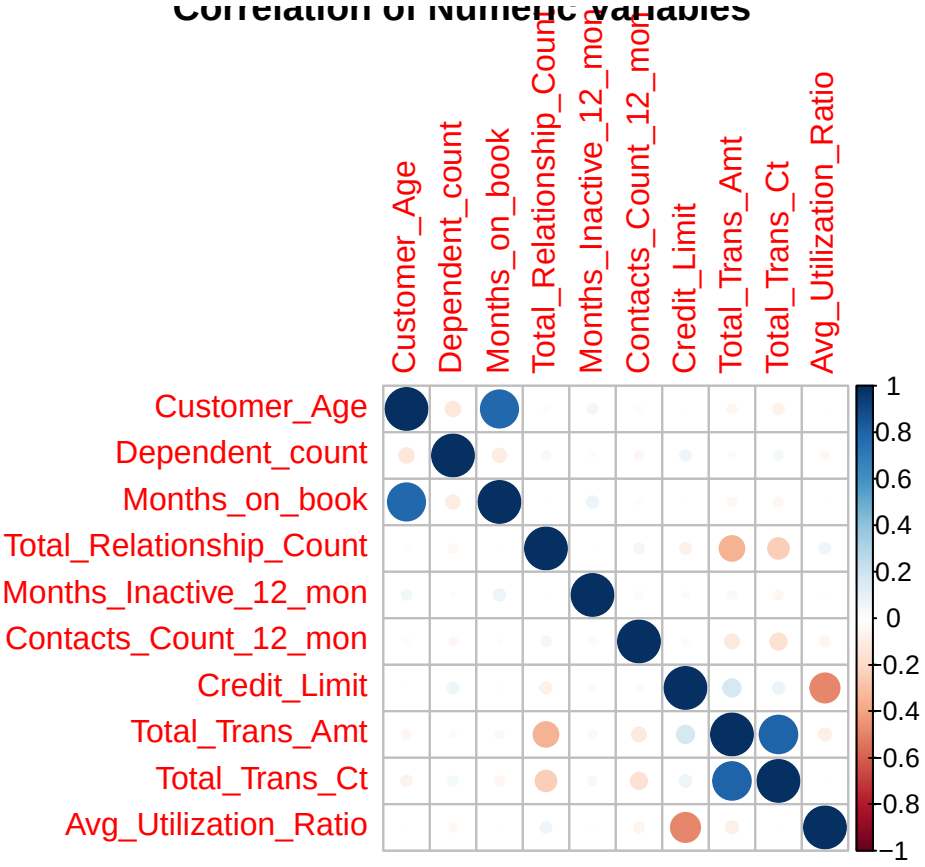
```
##
## Attrited Customer Existing Customer
##              1627              8500
```

We see that our dependent variable shows that out of the 10,127 customers, 83.9% (8,500 customers) have remained, whereas 16.1% (1,627 customers) have left.

Similarly we transform the variables Gender, Education Level, Marital Status, Income Category and Card Category to factor variables.

```
data <- transform(
  data,
  Attrition_Flag = as.factor(Attrition_Flag),
  Gender = as.factor(Gender),
  Education_Level = as.factor(Education_Level),
  Marital_Status = as.factor(Marital_Status),
  Income_Category = as.factor(Income_Category),
  Card_Category = as.factor(Card_Category))
```

```
nv <- sapply(data, is.numeric)
cormat <- cor(data[,nv])
corrplot::corrplot(cormat, title = "Correlation of Numeric Variables")
```

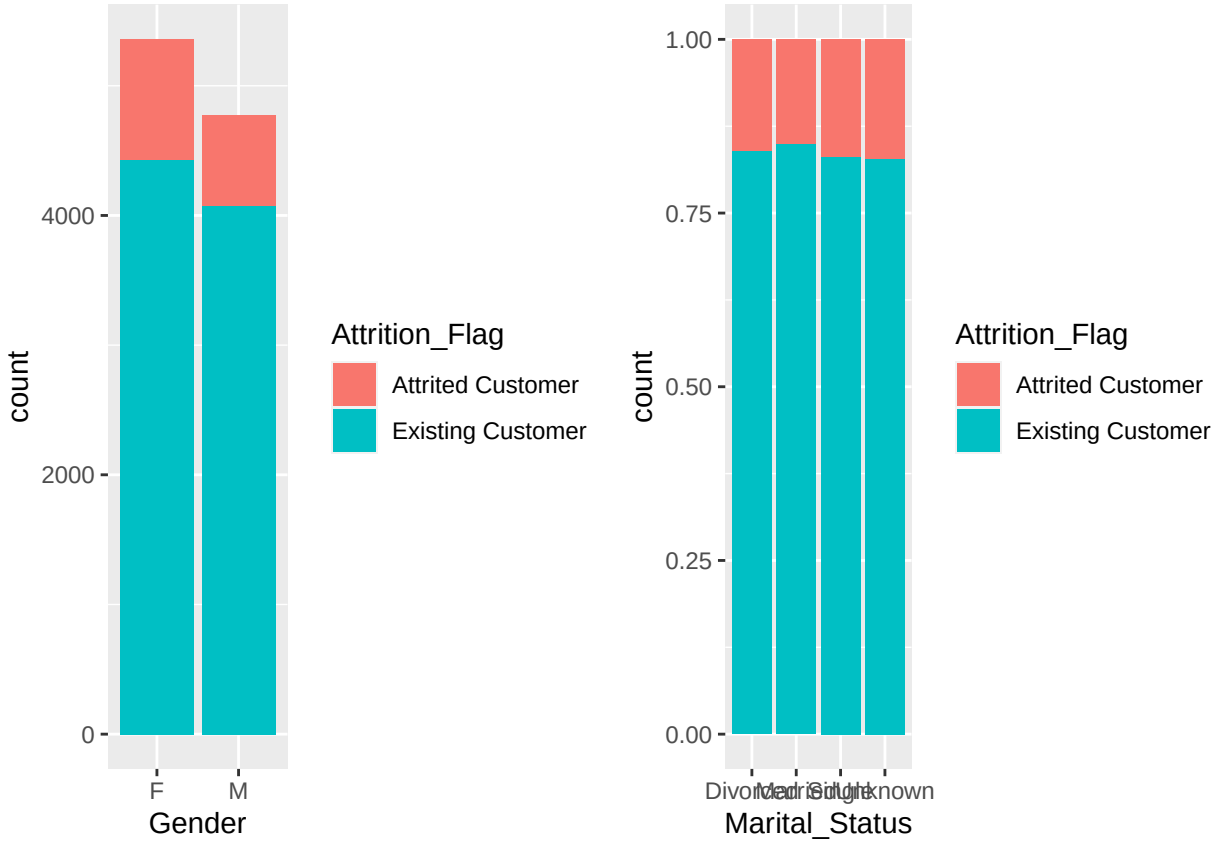


With regards to the correlation matrix, we see in Figure "Correlation of Numeric Variables" how some variables seem to be highly correlated. However, that was highly expected due to the fact that having a high number of variables increases the likelihood of finding overlaps among them. The variables involved are: "Months_on_book" with "Customer_Age" (positively correlated), "Total_Trans_Amt" with "Total_Trans_Ct" (positively correlated) and "Avg_Utilization_Ratio" with "Credit_Limit" (negatively correlated). Since they do not measure the same aspects, we proceed by considering all of them as adding significant information into our model, and thus we are not excluding one of them.

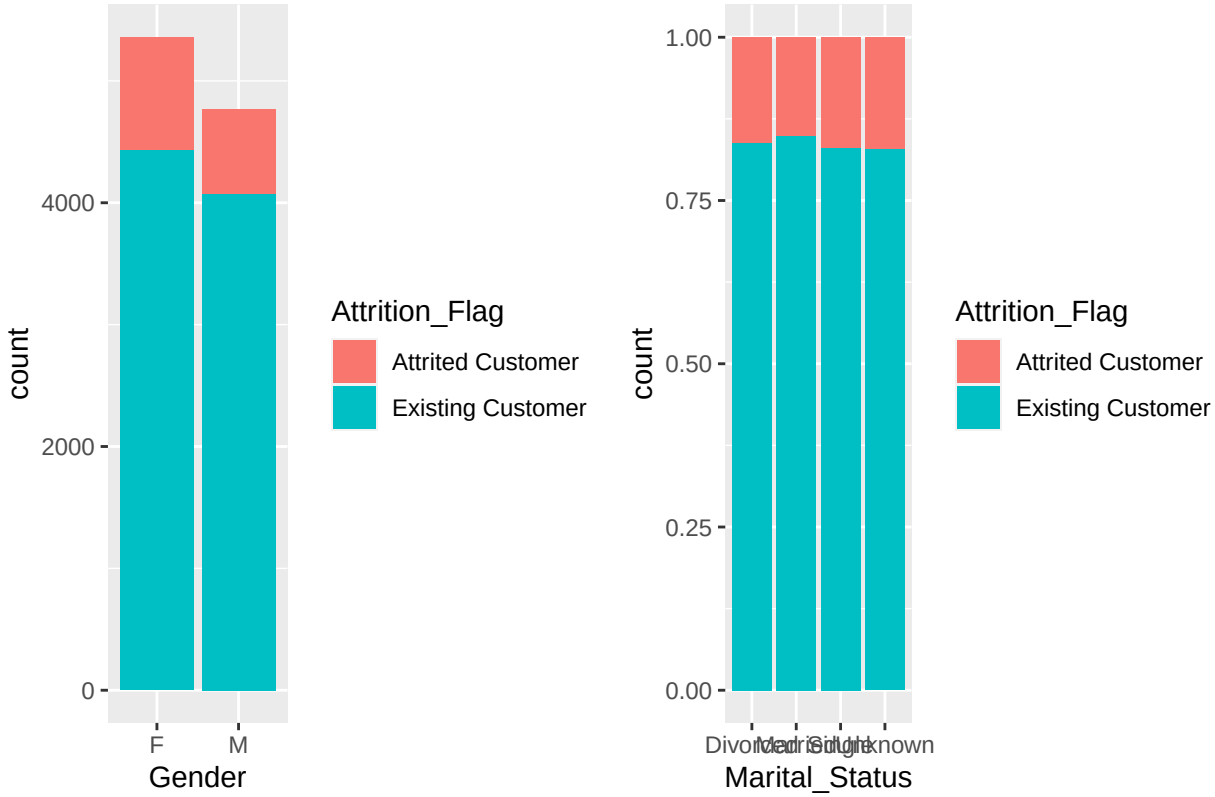
In the graphs below, we show the different distribution of the aforementioned variables by means of their status with the Alpha Bank (i.e., "Attrited customer" vs "Existing customer").

```
fig1 <-   ggarrange(ggplot(data, aes(x=Gender,fill=Attrition_Flag))+ geom_bar() ,
          ggplot(data, aes(x=Marital_Status,fill=Attrition_Flag))+ geom_bar(position = 'fill'))

print(fig1)
```

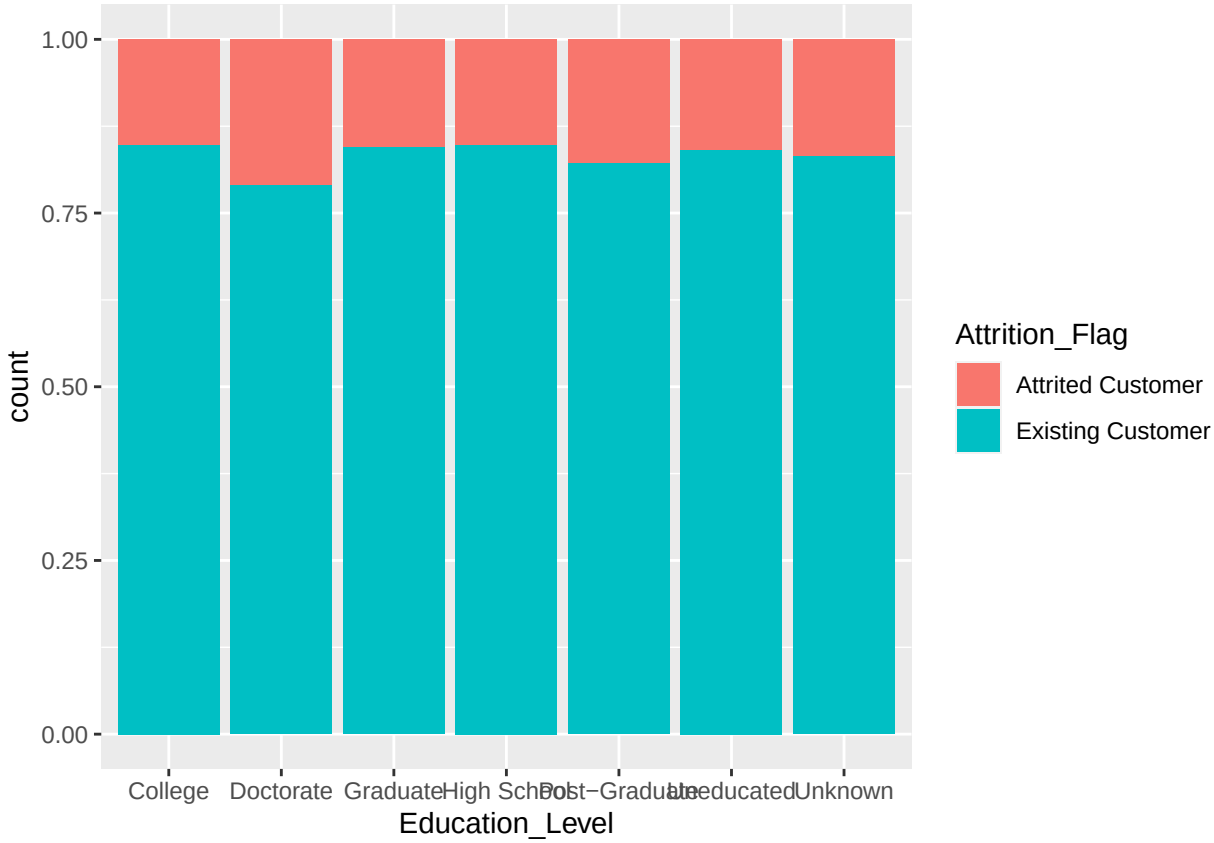



```
annotate_figure(fig1, bottom  = text_grob("Attrition Percentage in Gender, Marital Status and Card Categ
```

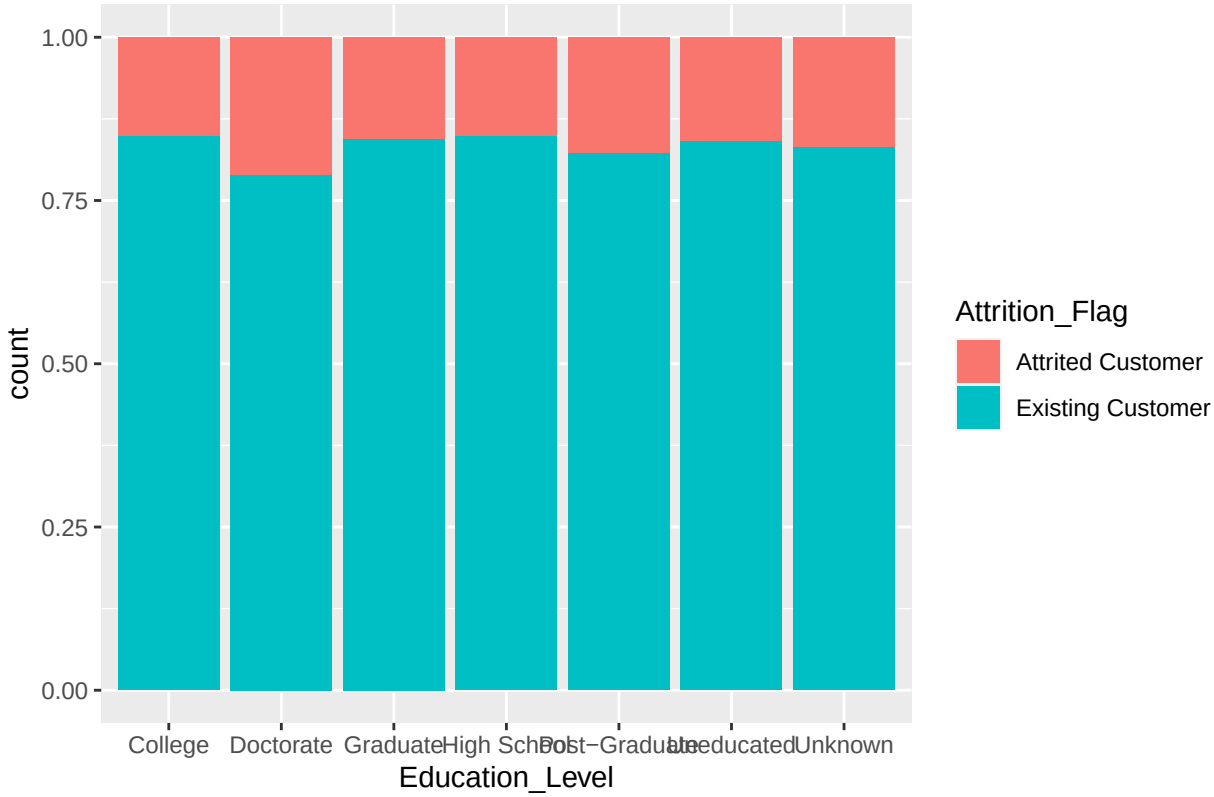


Attrition Percentage in Gender, Marital Status and Card Category

```
fig2 <- ggplot(data, aes(x=Education_Level,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig2)
```

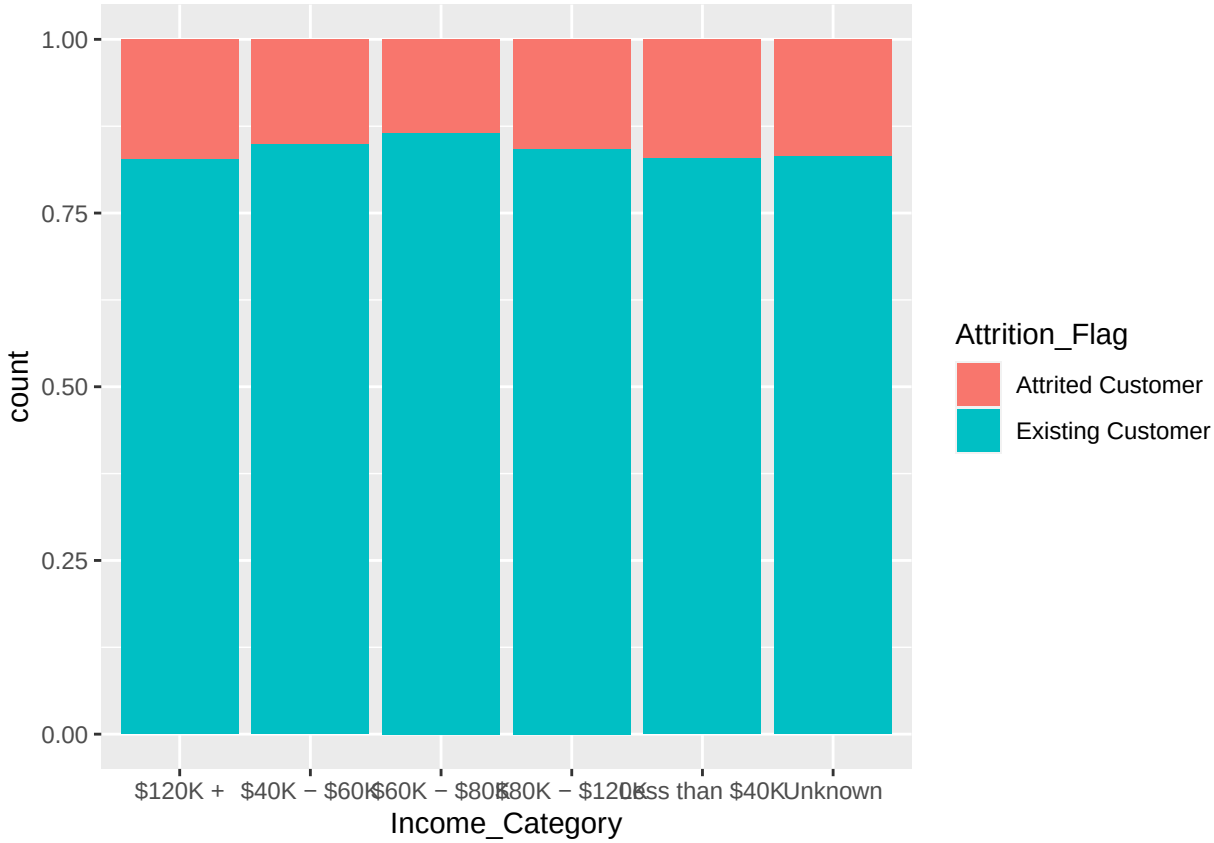



```
annotate_figure(fig2, bottom  = text_grob("Attrition Percentage for different levels of education", col
```

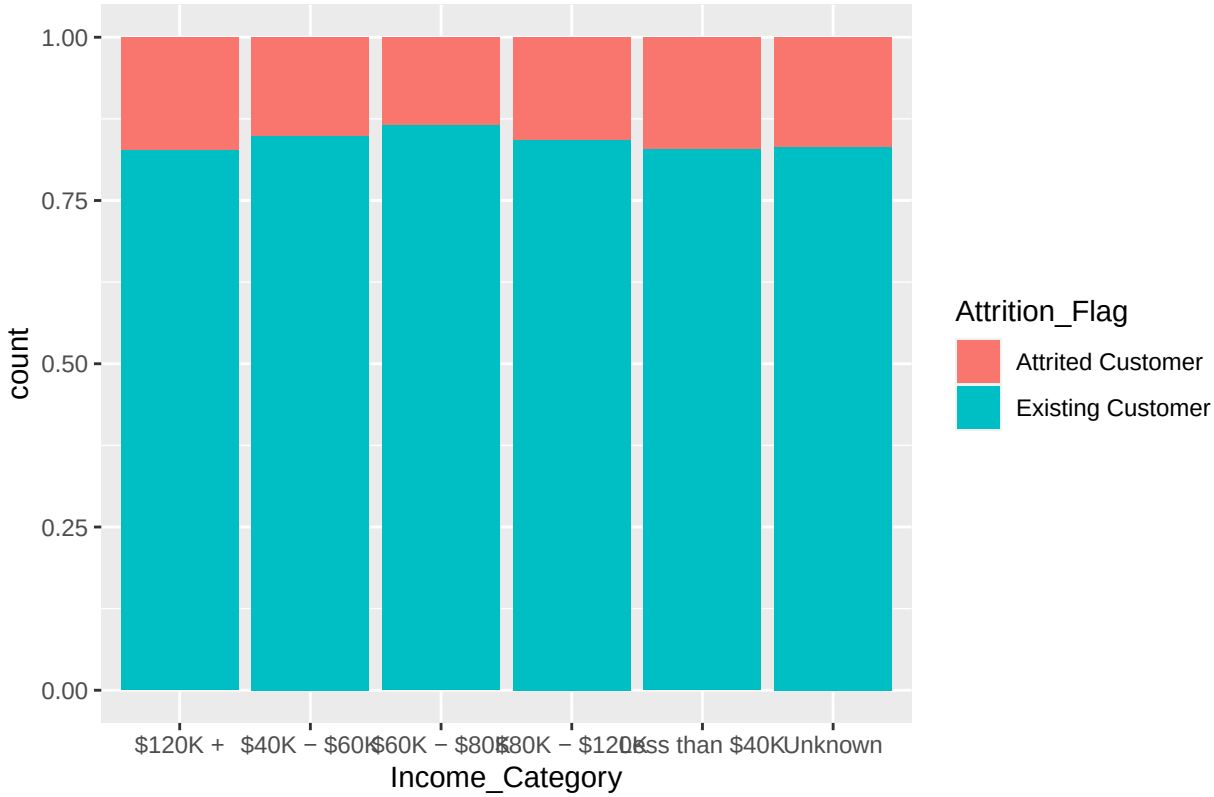


Attrition Percentage for different levels of education

```
fig3 <- ggplot(data, aes(x=Income_Category,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig3)
```

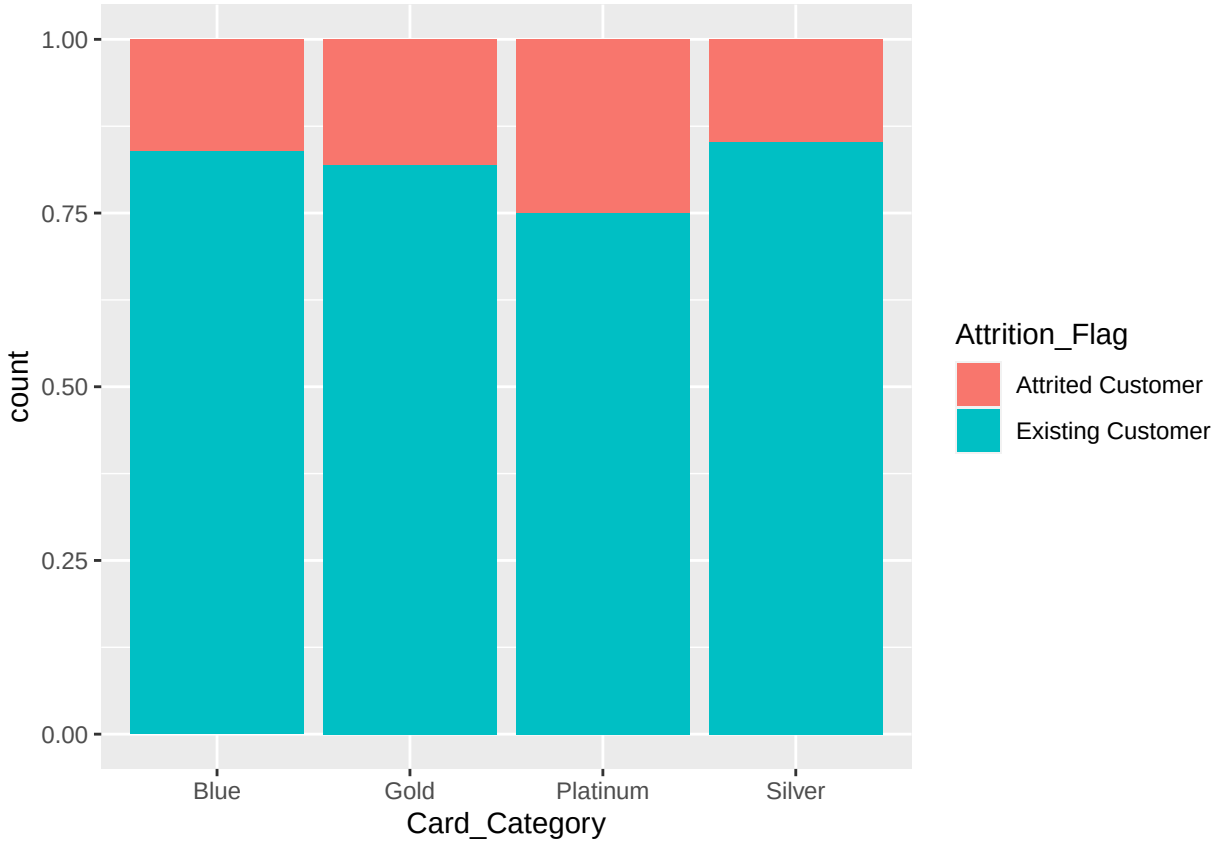



```
annotate_figure(fig3, bottom  = text_grob("Attrition Percentage for different income levels", col = "blu
```

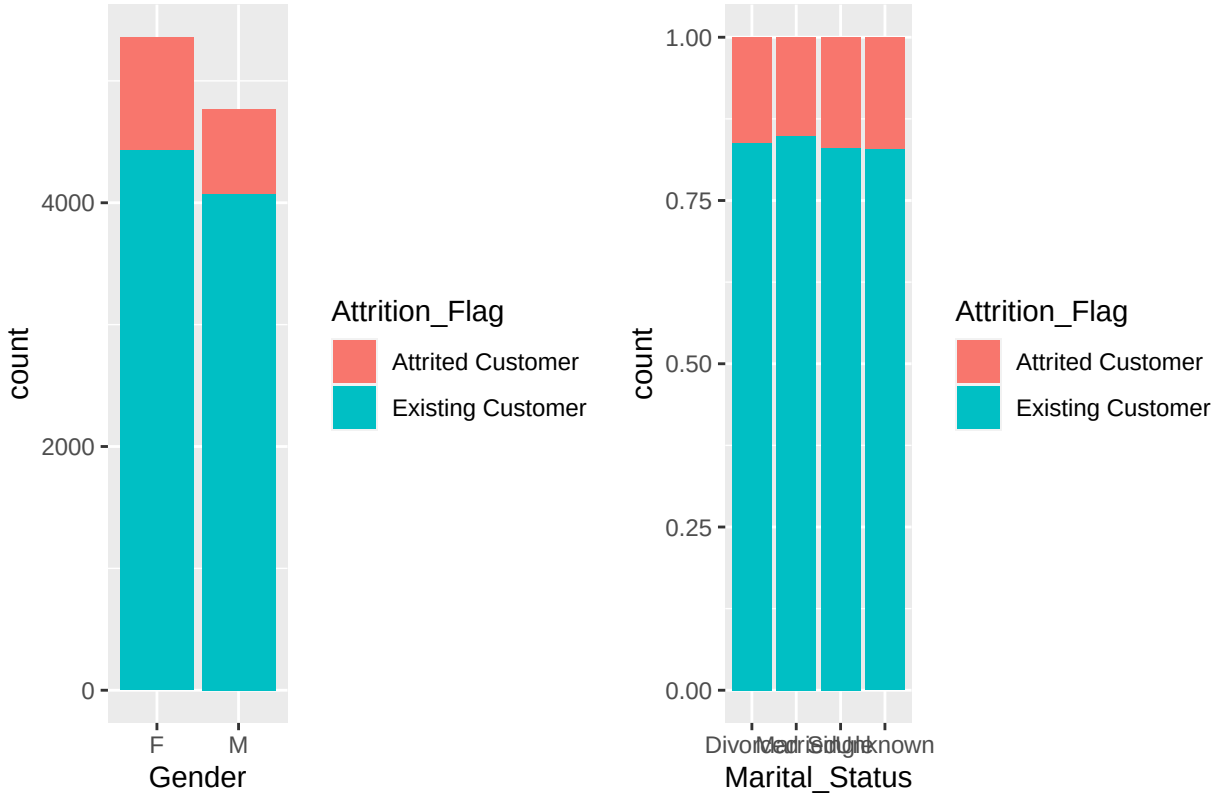


Attrition Percentage for different income levels

```
fig4 <- ggplot(data, aes(x=Card_Category,fill=Attrition_Flag))+ geom_bar(position = 'fill')
print(fig4)
```



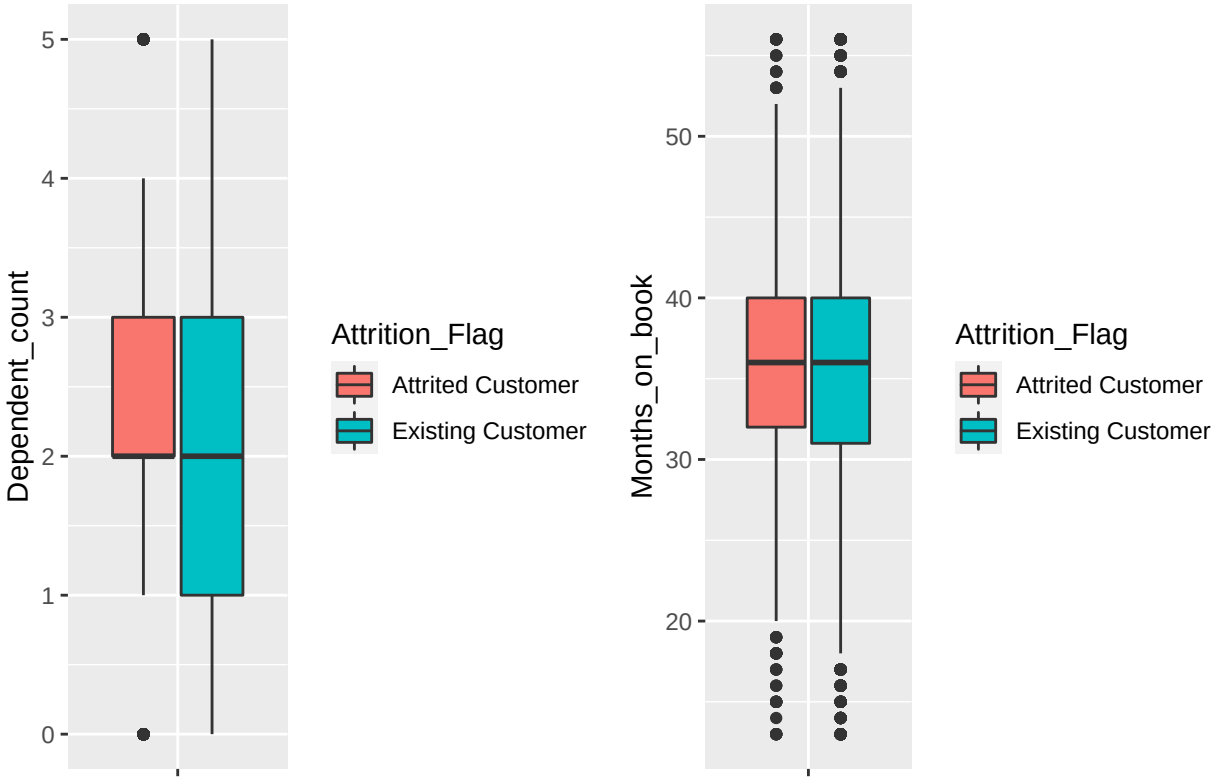

```
annotate_figure(fig1, bottom  = text_grob("Attrition Percentage for different cardholder categories", c
```



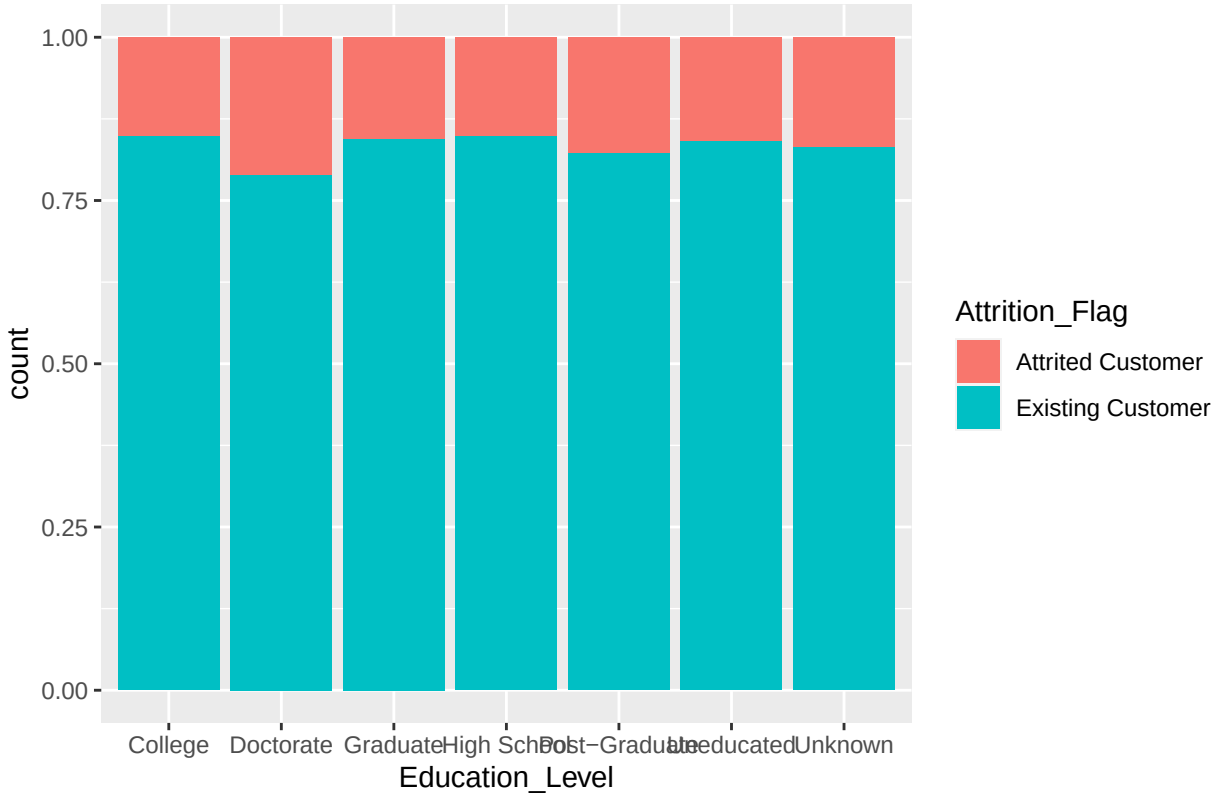
Attrition Percentage for different cardholder categories

```
fig5 <-   ggarrange(
          ggplot(data, aes(y= Dependent_count, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Months_on_book, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig5)
```



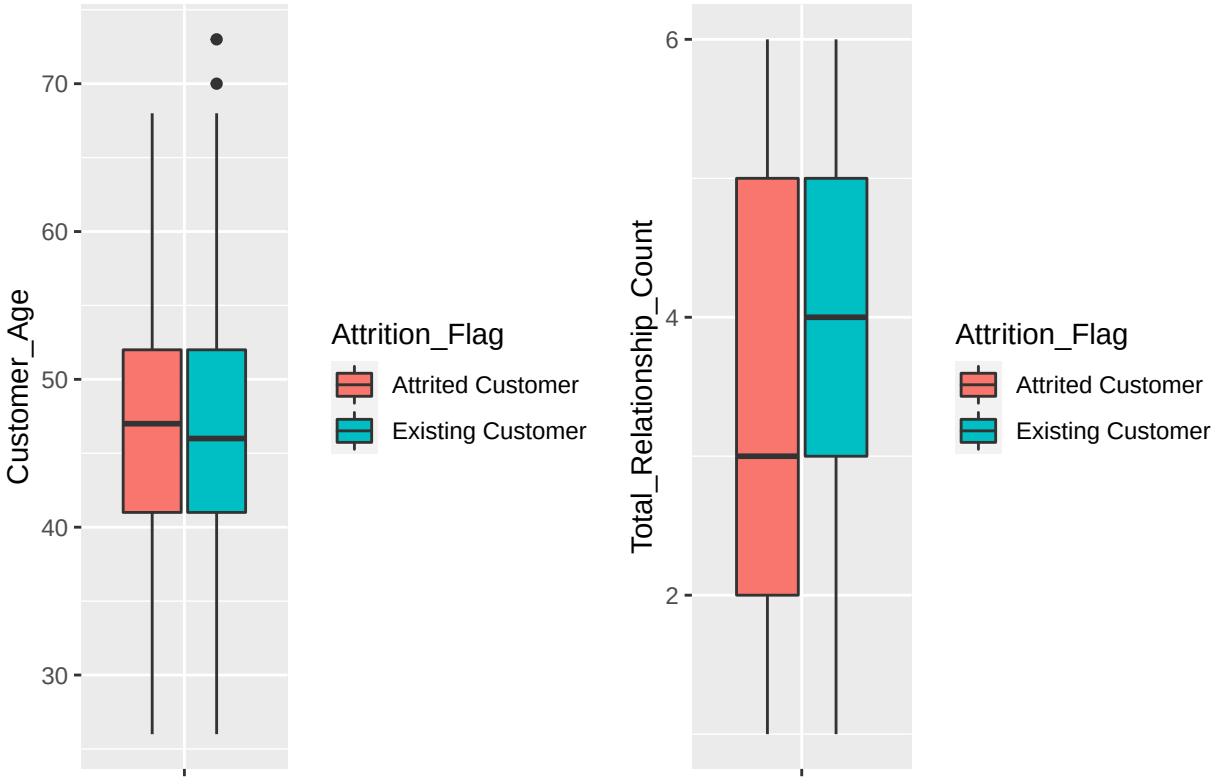

```
annotate_figure(fig2, bottom  = text_grob("Attrition Percentage for different number of dependents in fa
```



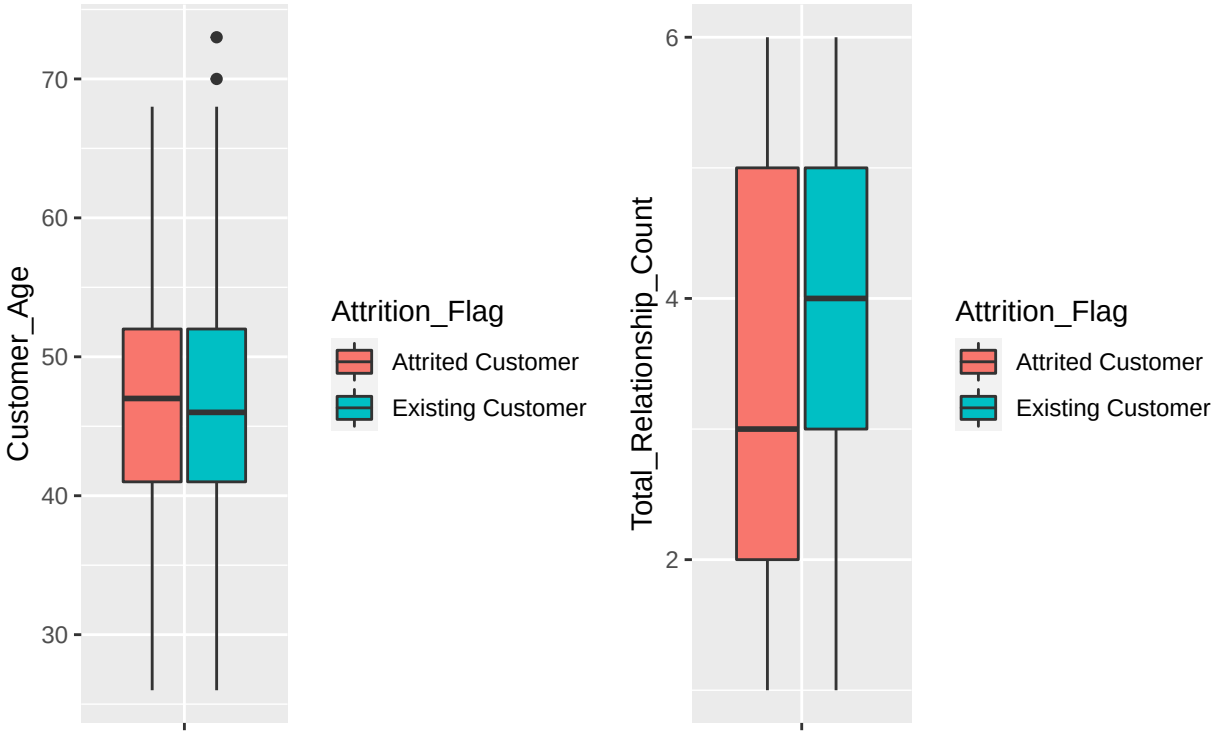
ge for different number of dependents in family and different number o

```
fig6 <-   ggarrange(
          ggplot(data, aes(y= Customer_Age, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Total_Relationship_Count, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig6)
```



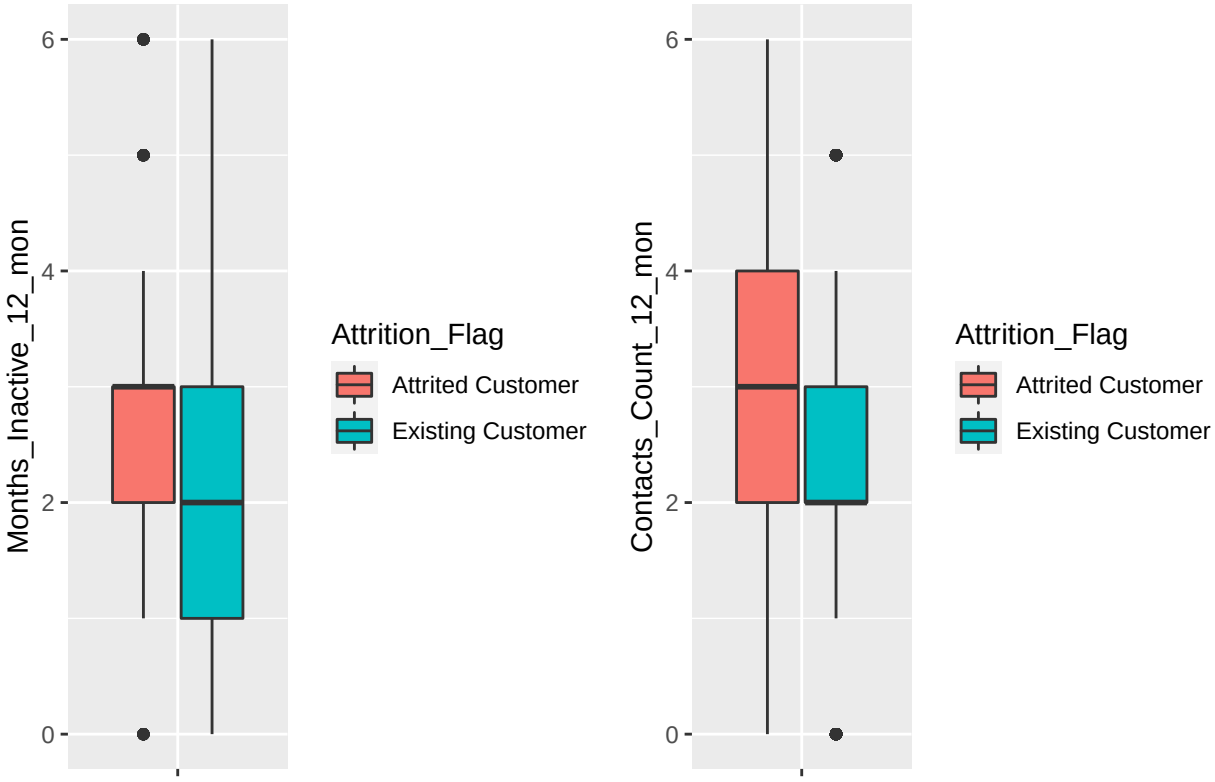

```
annotate_figure(fig6, bottom  = text_grob("Attrition Percentage in Age and Relationship counts", col =
```



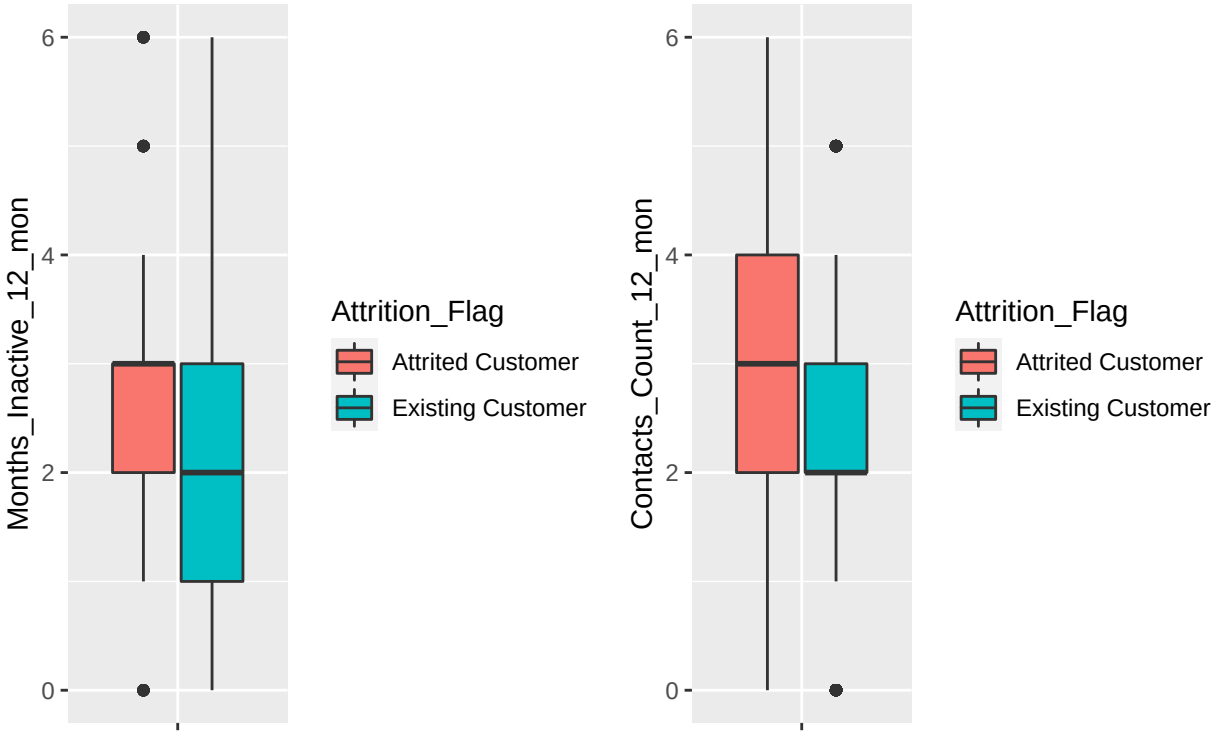
Attrition Percentage in Age and Relationship counts

```
fig7 <-   ggarrange(
          ggplot(data, aes(y= Months_Inactive_12_mon, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Contacts_Count_12_mon, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig7)
```



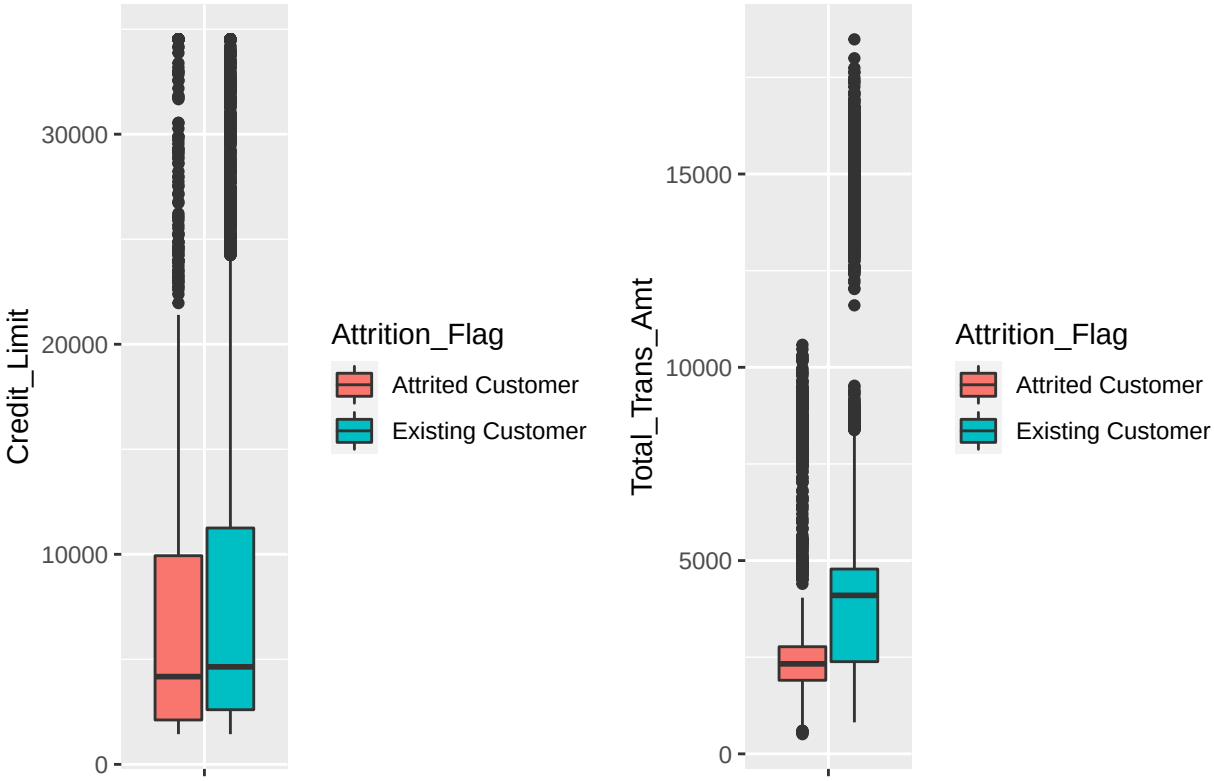

```
annotate_figure(fig7, bottom  = text_grob("Attrition Percentage in inactivity and number of contracts",
```



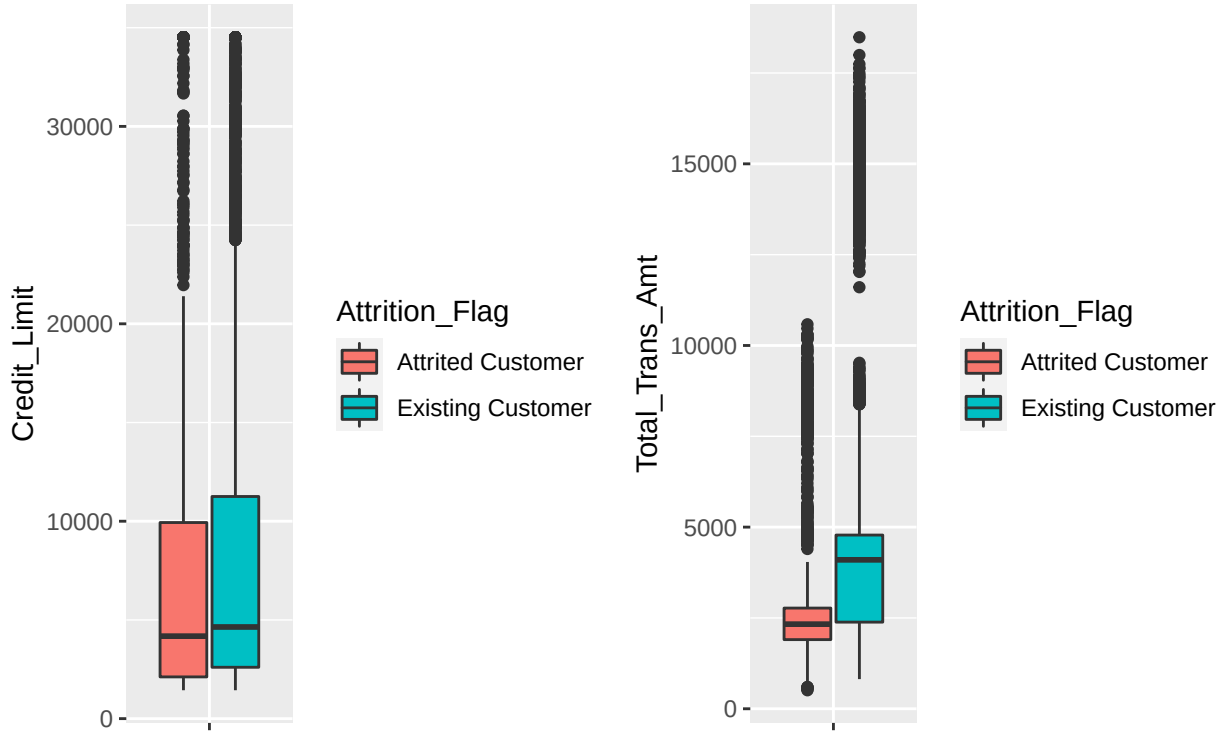
Attrition Percentage in inactivity and number of contracts

```
fig8 <-   ggarrange(
          ggplot(data, aes(y= Credit_Limit, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Total_Trans_Amt, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig8)
```



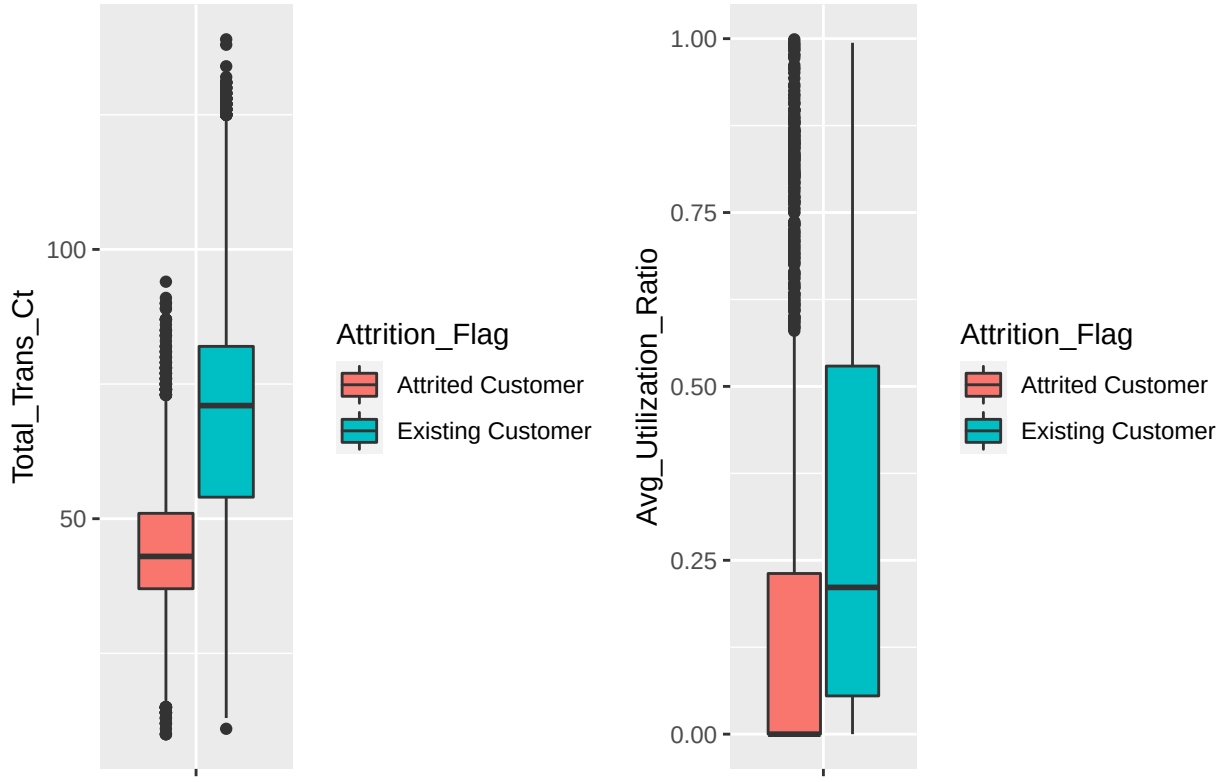

```
annotate_figure(fig8, bottom  = text_grob("Attrition Percentage in different levels of credit limit tra
```



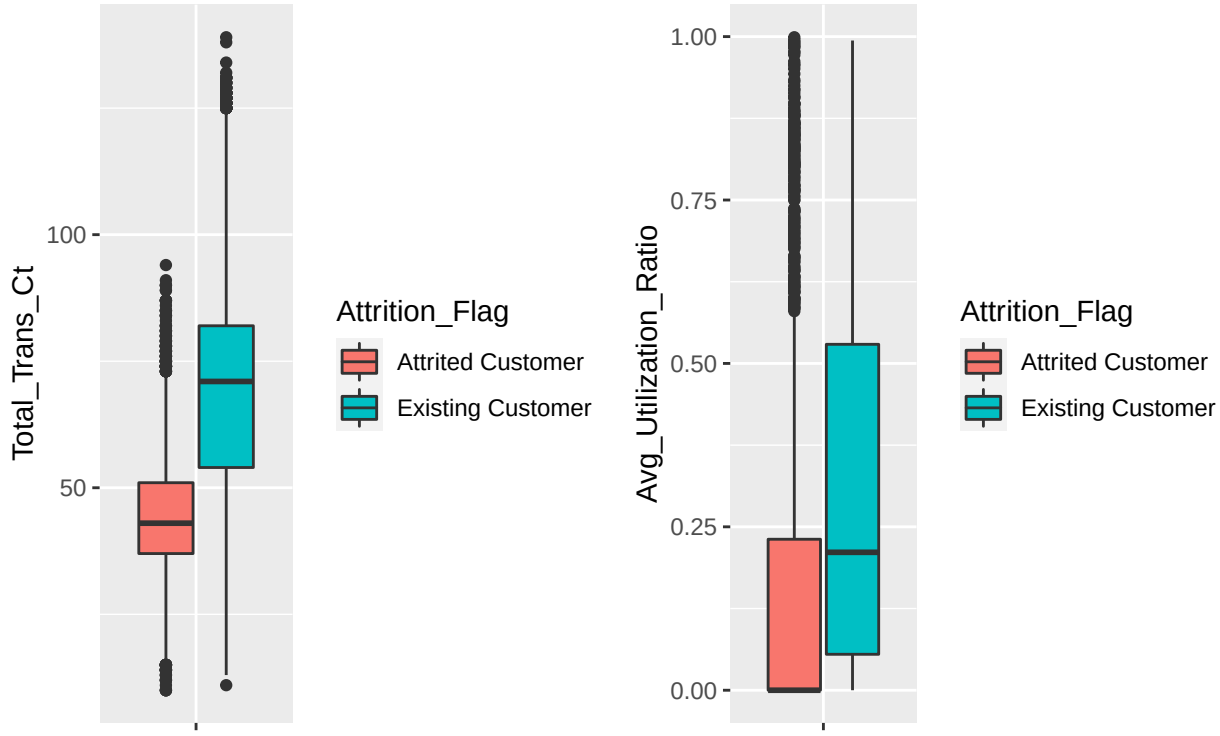
Attrition Percentage in different levels of credit limit transaction levels

```
fig9 <-   ggarrange(
          ggplot(data, aes(y= Total_Trans_Ct, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "),
          ggplot(data, aes(y= Avg_Utilization_Ratio, x = "", fill = Attrition_Flag))
+geom_boxplot() + xlab(" "))

print(fig9)
```




```
annotate_figure(fig9, bottom  = text_grob("Attrition Percentage in number of transactions and utilizati
```



Attrition Percentage in number of transactions and utilization ratio

```
##           Attrition_Flag              Customer_Age                    Gender
##                 "factor"                 "integer"                  "factor"
##          Dependent_count           Education_Level            Marital_Status
##                 "integer"                  "factor"                  "factor"
##          Income_Category             Card_Category            Months_on_book
##                 "factor"                  "factor"                 "integer"
## Total_Relationship_Count    Months_Inactive_12_mon     Contacts_Count_12_mon
##                 "integer"                 "integer"                 "integer"
##             Credit_Limit           Total_Trans_Amt           Total_Trans_Ct
##                 "numeric"                 "integer"                 "integer"
##    Avg_Utilization_Ratio
##                 "numeric"
```

It is essential to split the data into training and test sets. Indeed, we will fist use the training data to build our model and then, when we satisfied with the achievements, the remaining test data will deal to test our model, and thus to assess the power of our predictions. The training set is composed by 75% of the data, whereas the test set 25%.

```
smp_size <- floor(0.75 * nrow(data))

## set the seed to make your partition reproducible
set.seed(12345)
train_ind <- sample(seq_len(nrow(data)), size = smp_size)

train <- data[train_ind, ]
test <- data[-train_ind, ]
```

```
prop.table(table(train$Attrition_Flag))
```

```
##
## Attrited Customer Existing Customer
##          0.158262          0.841738
```

Since Attrition_Flag is a character variable with two possible values: either "Existing Customer" or "Attrited Customer" for modeling purposes we recode this variable into a factor with levels 0 and 1, where 1 represents a customer that has left the company when the person is still a customer at the company.

```
#recoding attrition_flag
train$Attrition_Flag <-ifelse(train$Attrition_Flag=="Attrited Customer",1,0)
test$Attrition_Flag <- ifelse(test$Attrition_Flag=="Attrited Customer", 1,0)
```

We now can proceed with our first technique, namely Logistic regression. As we don't have any preference between the two errors (i.e., Type 1 or Type 2), we proceed by selecting the threshold of t to be 0.5, as it will predict the most likely outcome. Indeed, as explained at the beginning of the report, companies should be advised that spending too much money on retaining customers is often the wrong action to take (even though it is important to find ways to retain). Hence, we decided that false positive and false negative would have the same impact, as it would be costly to lose a customer if not considered as willing to go (false positive?), but at the same time it is costly to find ways to retain them so considering a customer that wants to go even though is not costly as well (I said he goes but he remained -> false negative).

```
#logistic regression
glm <- glm(Attrition_Flag ~., data = train, family = "binomial")
summary(glm)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6806  -0.4127  -0.1936  -0.0800   3.3684
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     4.916e+00  4.913e-01  10.007  < 2e-16 ***
## Customer_Age                   -5.264e-04  8.325e-03  -0.063 0.949579
## GenderM                        -1.009e+00  1.652e-01  -6.105 1.03e-09 ***
## Dependent_count                 1.094e-01  3.241e-02   3.375 0.000737 ***
## Education_LevelDoctorate        3.667e-01  2.200e-01   1.667 0.095536 .
## Education_LevelGraduate         1.094e-01  1.520e-01   0.720 0.471788
## Education_LevelHigh School     -3.988e-02  1.620e-01  -0.246 0.805586
## Education_LevelPost-Graduate    2.768e-01  2.222e-01   1.246 0.212839
## Education_LevelUneducated       9.045e-02  1.703e-01   0.531 0.595397
## Education_LevelUnknown          1.984e-01  1.683e-01   1.179 0.238356
## Marital_StatusMarried          -5.234e-01  1.698e-01  -3.082 0.002054 **
## Marital_StatusSingle            1.005e-01  1.705e-01   0.589 0.555614
## Marital_StatusUnknown           1.198e-01  2.132e-01   0.562 0.574139
## Income_Category$40K - $60K     -1.214e+00  2.208e-01  -5.499 3.82e-08 ***
## Income_Category$60K - $80K     -9.105e-01  1.905e-01  -4.780 1.75e-06 ***
## Income_Category$80K - $120K    -5.444e-01  1.776e-01  -3.065 0.002178 **
## Income_CategoryLess than $40K -1.048e+00  2.401e-01  -4.365 1.27e-05 ***
## Income_CategoryUnknown        -1.166e+00  2.517e-01  -4.634 3.58e-06 ***
## Card_CategoryGold              1.087e+00  3.964e-01   2.742 0.006111 **
```

```
## Card_CategoryPlatinum          2.162e+00  7.218e-01   2.996 0.002740 **
## Card_CategorySilver            5.485e-01  2.070e-01   2.650 0.008055 **
## Months_on_book                -1.004e-02  8.315e-03  -1.208 0.227239
## Total_Relationship_Count      -4.734e-01  3.007e-02 -15.746  < 2e-16 ***
## Months_Inactive_12_mon         4.953e-01  4.134e-02  11.982  < 2e-16 ***
## Contacts_Count_12_mon          5.425e-01  3.968e-02  13.672  < 2e-16 ***
## Credit_Limit                  -6.085e-05  7.126e-06  -8.539  < 2e-16 ***
## Total_Trans_Amt                4.121e-04  2.497e-05  16.504  < 2e-16 ***
## Total_Trans_Ct                -1.139e-01  3.964e-03 -28.738  < 2e-16 ***
## Avg_Utilization_Ratio         -2.959e+00  1.844e-01 -16.048  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6634.6  on 7594  degrees of freedom
## Residual deviance: 3958.3  on 7566  degrees of freedom
## AIC: 4016.3
##
## Number of Fisher Scoring iterations: 6
```

We first look at accuracy (i.e., number of corrected guesses), specificity (i.e., true negative rate), sensitivity (i.e., true positive rate), and precision on our training set, so that we can analyze how well the model was at predicting the outcome.

```
pred <- predict(glm, data = train, type = "response")
# confusion matrix on training set
conmat <- table(train$Attrition_Flag, pred >= 0.5)
#show confusion matrix
print(conmat)
```

```
##
##     FALSE TRUE
##   0  6149  244
##   1   590  612
```

```
accuracy <- (6149+612)/nrow(train)
specificity <- 6149/(6149+244)
sensitivity <- 612/(612+590)
precision <- 612/(612+244)
```

The accuracy seems to be pretty .....

Secondly we deploy the model on our test set to see how well our model predicts on new data. We again calculate the Accuracy, Specificity, Sensitivity and Precision of our model.

```
# observations on the test set
predtest <- predict(glm, newdata = test, type = "response")
conMattest <- table(test$Attrition_Flag, predtest >= 0.5)
#show confusion matrix
print(conMattest)
```

```
##
##     FALSE TRUE
##   0  2029   78
##   1   189  236
```

```
accuracytest <- (2029+236)/nrow(test)
specificitytest <- 2029/(2029+78)
sensitivitytest <- 236/(236+189)
precisiontest <- 236/(236+78)
```

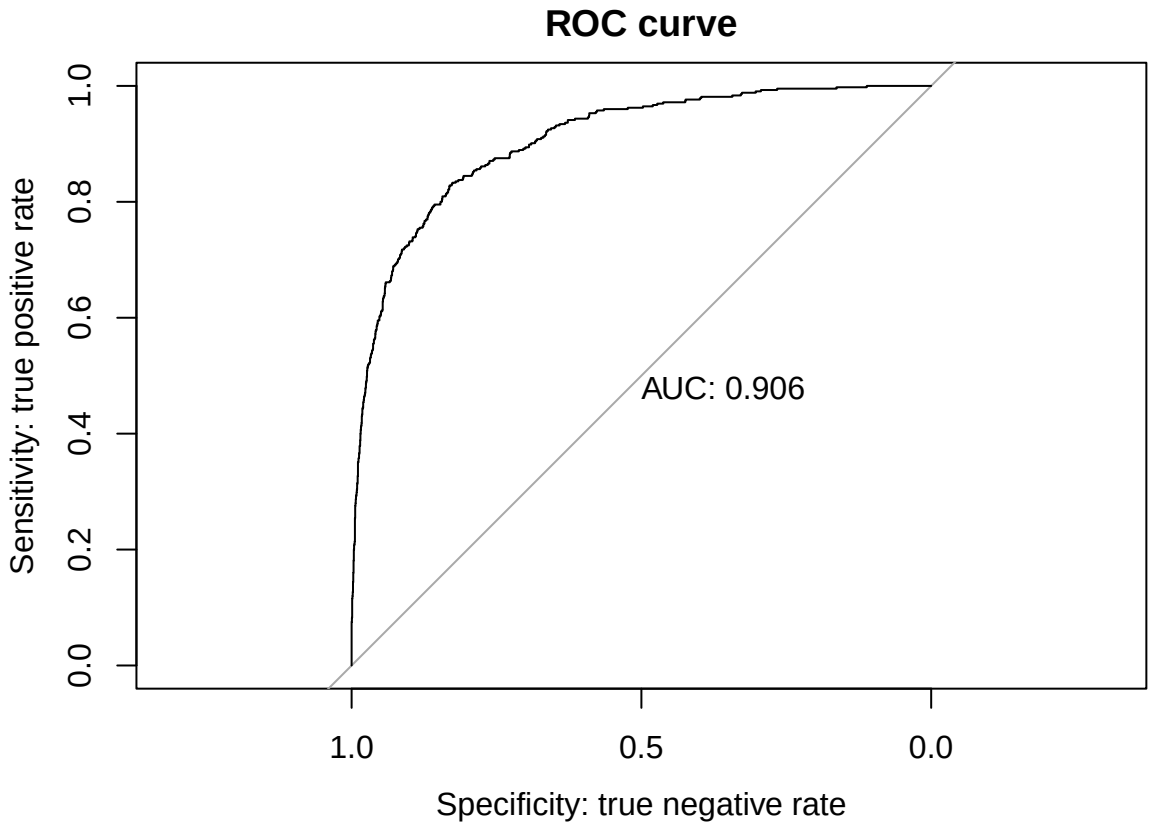
Again, the accuracy seems to be pretty .....

We now plot the ROC curve of our model. The ROC curve measures the .. the larger the area the more accurate the model, with an area of 1 signifying a perfect model and an area of 0.5 or lower as a situation in which the model does not perform better than a random prediction.

```
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, predtest), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



The area under the curve equals to ... this is a very high value and is an indication that our model is performing quite well.

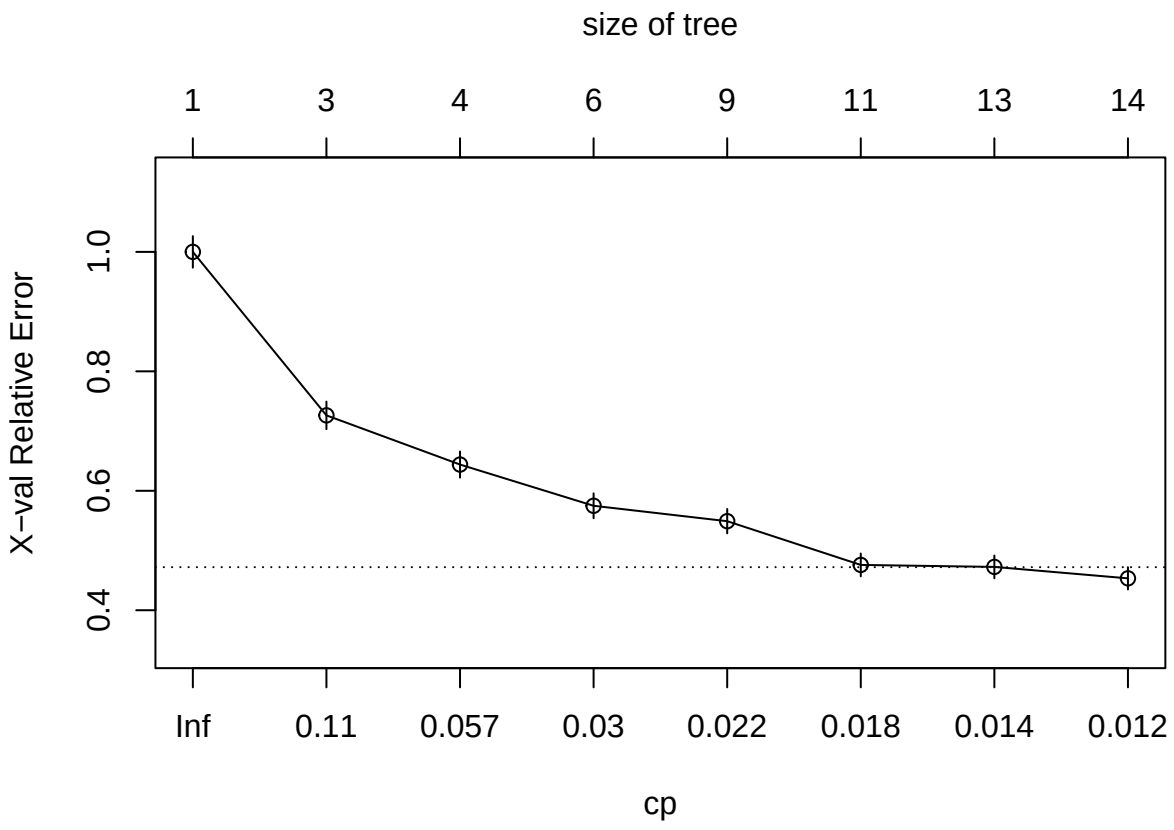
Next we calculate the variable importance for all of the variables we include in our model. The variable importance measures which variables are the main predictors of the loyalty of our customers. Insight into the main predictors could help give the company focus in developing marketing strategies to prevent attrition.

```
logisticvariableimportance <- varImp(glm, scale = FALSE)
print(logisticvariableimportance)
```

```
##                                   Overall
## Customer_Age                    0.0632359
## GenderM                         6.1052300
## Dependent_count                 3.3753243
## Education_LevelDoctorate        1.6668892
## Education_LevelGraduate         0.7195736
## Education_LevelHigh School      0.2461248
## Education_LevelPost-Graduate    1.2457979
## Education_LevelUneducated       0.5310317
## Education_LevelUnknown          1.1791066
## Marital_StatusMarried           3.0822941
## Marital_StatusSingle            0.5893686
## Marital_StatusUnknown           0.5619667
## Income_Category$40K - $60K      5.4989659
## Income_Category$60K - $80K      4.7802932
## Income_Category$80K - $120K     3.0648737
## Income_CategoryLess than $40K   4.3652086
## Income_CategoryUnknown          4.6344535
## Card_CategoryGold               2.7417825
## Card_CategoryPlatinum           2.9955012
## Card_CategorySilver             2.6497523
## Months_on_book                  1.2075007
## Total_Relationship_Count       15.7457192
## Months_Inactive_12_mon        11.9818320
## Contacts_Count_12_mon         13.6722308
## Credit_Limit                   8.5389789
## Total_Trans_Amt               16.5035498
## Total_Trans_Ct                28.7383614
## Avg_Utilization_Ratio         16.0475677
```

Of course we want to compare multiple supervised learning methods to compare the quality of different models and choose the model that is the best at predicting customer attrition. The 2nd model we use is a basic CART decision tree.

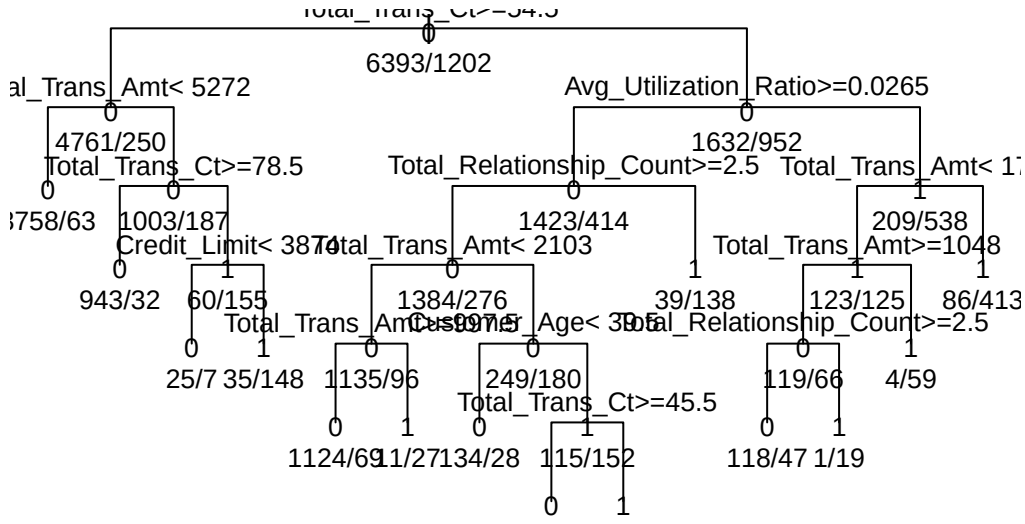
```
tree <- rpart(Attrition_Flag ~., method = "class", data = train)
printcp(tree)
plotcp(tree)
```




```
summary(tree)
```

The cp shows ....

```
# plot tree
plot(tree, uniform=TRUE,
   main="Classification Tree for Attrition")
text(tree, use.n=TRUE, all=TRUE, cex=.8)
```



This shows an overview of our decision tree. For instance a customer with .. and .. will have a .. percent chance of leaving the company.

We test the quality of our model by firstly predicting attrition on our training dataset.

```
#options(digits=4)
# assess the model's accuracy with train dataset by make a prediction on the train data.
Predict_model1_train <- predict(tree, train, type = "class")
#build a confusion matrix to make comparison
conMat <- confusionMatrix(as.factor(Predict_model1_train), as.factor(train$Attrition_Flag))
#show confusion matrix
conMat$table
```

```
##            Reference
## Prediction    0    1
##          0 6190  291
##          1  203  911
```

This shows the confusion matrix for our decision tree model. Our predictions on the training set seem to be doing pretty decent, but to give more insight we proceed by calculating the Accuracy, Sensitivity, Specificity and Precision of our model.

```
sensitivity(conMat$table)
```

```
## [1] 0.9682465
```

```
specificity(conMat$table)
```

```
## [1] 0.7579035
```

```
print(accuracy <- (6190+911)/(6190+911+291+203))
```

```
## [1] 0.9349572
```

```
print(precision <- 911/(911+203))
```

```
## [1] 0.8177738
```

The model looks to do a decent job, our sensitivity seems to be quite higher than our specificity, which implies that our model is better at correctly classifying clients that left than at finding true loyal customers. This could be because of ...

Now that we have constructed the model we proceed by predicting the values in the test set in order to assess the suitability of the model.

```
Predict_model1_test <- predict(tree, test, type = "class")

conMattest <- confusionMatrix(as.factor(Predict_model1_test), as.factor(test$Attrition_Flag))

conMattest$table
```

```
##            Reference
## Prediction    0    1
##          0 2032  110
##          1   75  315
```

This shows the confusion matrix for our decision tree model. Our predictions on the test set seem to be of similar quality as they were on the training data. Again we proceed by calculating the Accuracy, Sensitivity, Specificity and Precision of our model.

```
sensitivity(conMattest$table)
```

```
## [1] 0.9644044
```

```
specificity(conMattest$table)
```

```
## [1] 0.7411765
```

```
print(accuracy <- (2032+315)/(2032+315+110+75))
```

```
## [1] 0.9269352
```

```
print(precision <- 315/(315+75))
```

```
## [1] 0.8076923
```

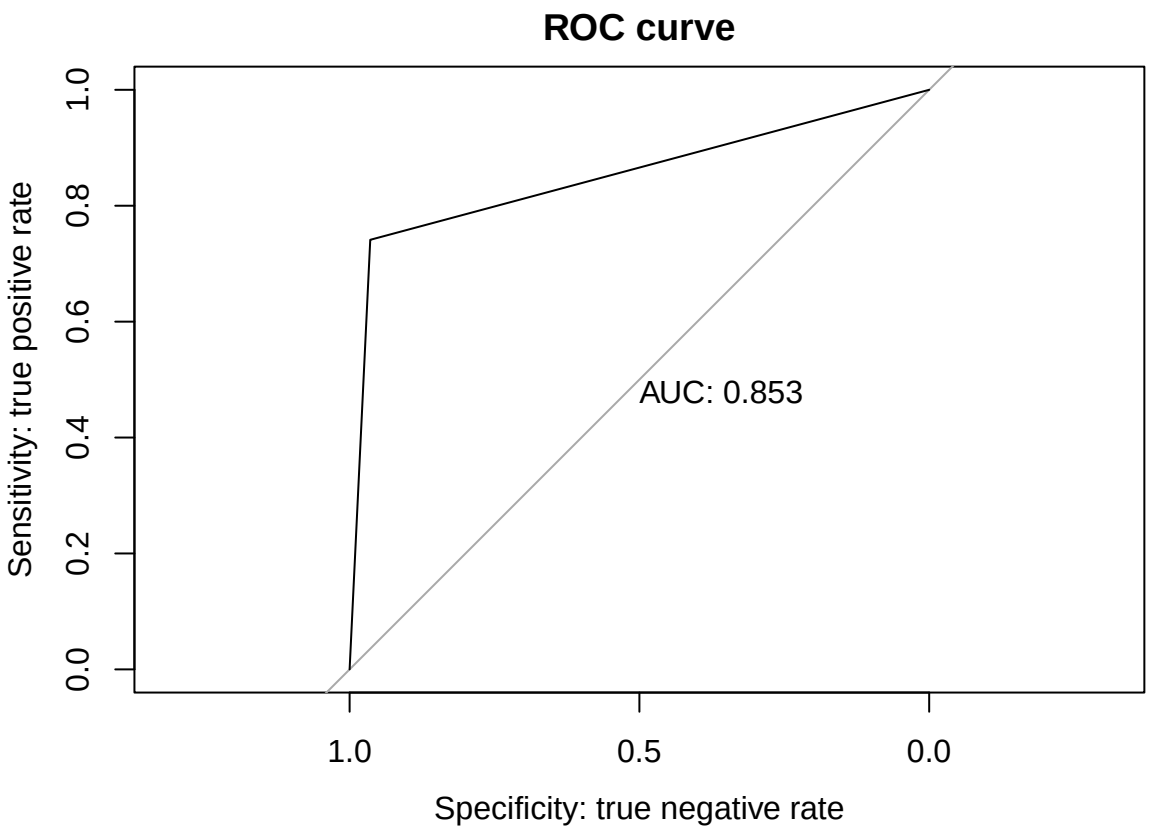
There is not much difference between the accuracy for our model when comparing for the test and training set. The Sensitivity is slightly higher(0.01) and the specificity slightly lower(0.01). The accuracy is slightly lower than when predicting on the training set, however the difference is marginal

We construct the ROC curve to compare how well our model predicts Attrition compared to a completely random guessing strategy.

```
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, as.numeric(Predict_model1_test)), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



The area under the curve of ... implies that our model is doing a good job in predicting customer attrition.

We calculate the variable importance for our decision tree to see which variables were decisive in classifying the customers and predicting their behaviour.

```
treevariableimportance <- varImp(tree, scale = FALSE)
print(treevariableimportance)
```

```
##                            Overall
## Avg_Utilization_Ratio    586.32904
## Contacts_Count_12_mon    137.52696
## Credit_Limit             205.61775
## Customer_Age              31.69373
## Gender                    80.31005
## Months_Inactive_12_mon   203.82831
## Months_on_book            17.75329
## Total_Relationship_Count 413.91122
## Total_Trans_Amt          872.06459
## Total_Trans_Ct           625.78974
## Dependent_count            0.00000
## Education_Level            0.00000
## Marital_Status             0.00000
## Income_Category            0.00000
## Card_Category              0.00000
```

A 3rd type of model that we could implement is a random forest. CART decision trees are easily interpretable, however, output can be ... because of ... Therefore we implement a random forest model that uses a bagging procedure producing ... regression trees and takes the average of each regression tree to improve

the .. of the model results.

```
train$Attrition_Flag <- as.character(train$Attrition_Flag)
train$Attrition_Flag <- as.factor(train$Attrition_Flag)
rf <- randomForest(Attrition_Flag ~ ., , data = train, proximity=FALSE,importance = FALSE)
print(rf)
```

```
##
## Call:
##  randomForest(formula = Attrition_Flag ~ ., data = train, , proximity = FALSE,      importance = FALS
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 5.29%
## Confusion matrix:
##      0   1 class.error
## 0 6297  96  0.01501642
## 1  306 896  0.25457571
```

```
summary(rf)
```

```
##                 Length Class  Mode
## call               6  -none- call
## type               1  -none- character
## predicted        7595  factor numeric
## err.rate         1500  -none- numeric
## confusion           6  -none- numeric
## votes           15190  matrix numeric
## oob.times        7595  -none- numeric
## classes             2  -none- character
## importance         15  -none- numeric
## importanceSD        0  -none- NULL
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                7595  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

This shows the summary of our random forest model. We can see that .....

Just like with the 2 previous models we continue by predicting customer attrition in our training dataset to assess how well our model fits the data.

```
predrf <- predict(rf, data = "train", type = "response")
print(rftab <- table(predrf, train$Attrition_Flag))
```

```
##
## predrf    0    1
##      0 6297  306
##      1   96  896
```

```
print(accuracyrf <- (6290+914)/nrow(train))
```

```
## [1] 0.9485188
```

```
print(sensitivityrf <- 914/(914+103))
```

```
## [1] 0.8987217
```

```
print(precisionrf <- 914/(914+288))
```

```
## [1] 0.7603993
```

```
print(specificityrf <- 6290/(6290 + 288))
```

```
## [1] 0.9562177
```

The accuracy of ... implies that our model is quite accurate in its predictions. The sensitivity and specificty signal that ... whereas the precision of .. tells us ..

To verify the fit of our model we also predict customer attrition in the test dataset.

```
predrftest <- predict(rf, newdata = test, type = "response")
print(rftabtest <- table(predrftest, test$Attrition_Flag))
```

```
##
## predrftest    0    1
##          0 2076   92
##          1   31  333
```

```
print(accuracyrftest <- (2073+335)/nrow(test))
```

```
## [1] 0.9510269
```

```
print(sensitivityrftest <- 335/(335+34))
```

```
## [1] 0.9078591
```

```
print(precisionrftest <- 335/(335+90))
```

```
## [1] 0.7882353
```

```
print(specificityrftest <- 2073/(2073 + 90))
```

```
## [1] 0.9583911
```

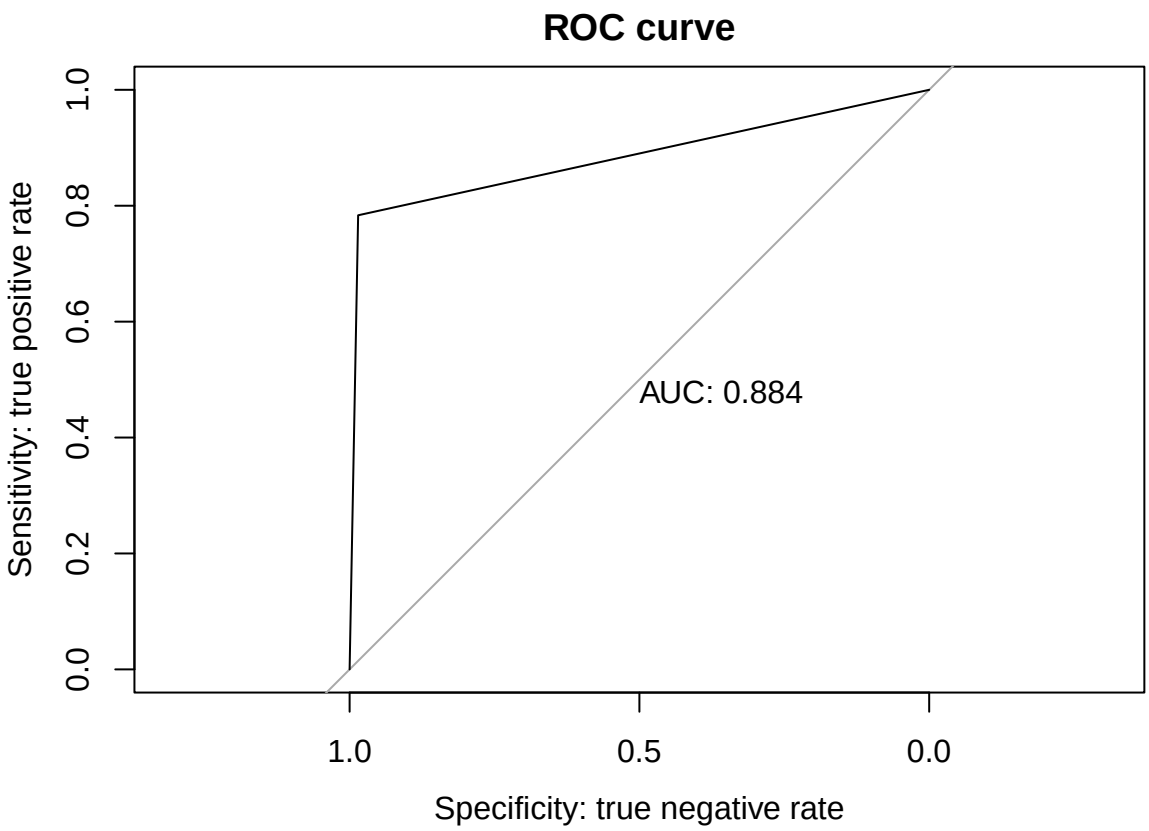
The accuracy of ... tells us that our model is quite accurate in its predictions. The sensitivity and specificty show that ... whereas the precision of .. gives us an indication of how ..

We also plot an ROC curve for our random forest model to see how well our predictions are doing compared to random guessing.

```
par(mai=c(.9,.8,.2,.2))
plot(roc(test$Attrition_Flag, as.numeric(predrftest)), print.auc=TRUE,
     col="black", lwd=1, main="ROC curve", xlab="Specificity: true negative rate", ylab="Sensitivity: t
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



The area under the curve of ... indicates that our model is doing a good job in predicting customer attrition.

In the following graph we compare the results of all our 3 models at the same time.

```
glm.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(predtest))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

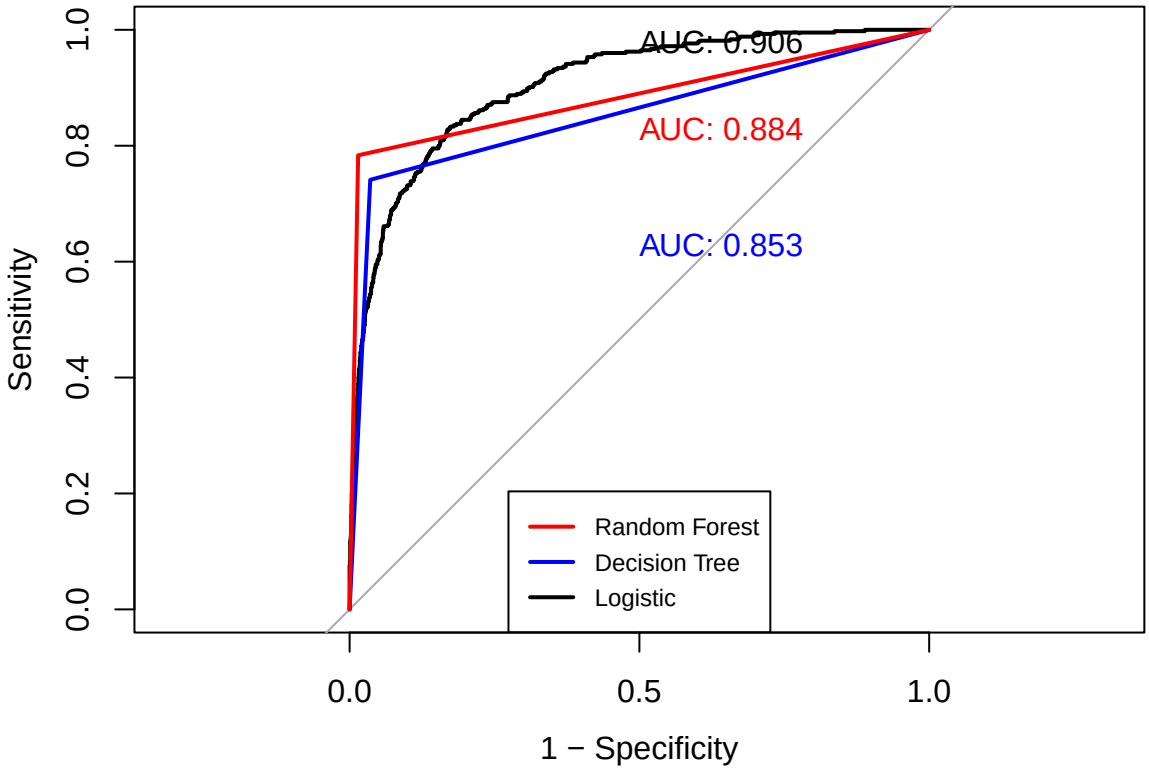
```
rpart.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(Predict_model1_test))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
rf.roc <- roc(response = test$Attrition_Flag, predictor = as.numeric(predrftest))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(glm.roc,      legacy.axes = TRUE, print.auc.y = 1.0, print.auc = TRUE)
plot(rpart.roc, col = "blue", add = TRUE, print.auc.y = 0.65, print.auc = TRUE)
plot(rf.roc, col = "red" , add = TRUE, print.auc.y = 0.85, print.auc = TRUE)
legend("bottom", c("Random Forest", "Decision Tree", "Logistic"),
       lty = c(1,1), lwd = c(2, 2), col = c("red", "blue", "black"), cex = 0.75)
```



The graph indicates that the Logistic regression model that we constructed seems to be the best model for predicting customer attrition. The area under the curve of .. indicates that in ... of the cases we would predict correctly.

Given that the logistic regression model came out as the best model we shortly highlight the results of the logistic regression to give insights into the drivers of customer attrition. According to our regression the .... , ... , ... and .... where the most important predictors of customer attrition. ... with a coefficient of ... would on average increase the likelihood of attrition by ... percent.

... with a coefficient of ... would on average increase the likelihood of churning by ... percent.

... with a coefficient of ... would on average increase the likelihood of churning by ... percent.

... with a coefficient of ... would on average increase the likelihood of churning by ... percent.