My Own Project - Bank Customer Churn Analysis

#Executive summary

This document is an data-science analysis of customer churn by using a bank data-set. I will use different simple machine learning model to predict if a customer is more likely to churn or not. Most of the method and approach are based on the knowledge acquired during the online course.

I will use accuracy of confusion matrix to evaluate/compare the models.

Library

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

```
## Warning: le package 'tidyr' a été compilé avec la version R 4.1.2
```

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

```
library(caret)
```

```
## Warning: le package 'caret' a été compilé avec la version R 4.1.2
```

```
## Le chargement a nécessité le package : lattice
```

```
##
## Attachement du package : 'caret'
```

```
## L'objet suivant est masqué depuis 'package:purrr':
##
##     lift
```

```
library(data.table)
```

```
## Warning: le package 'data.table' a été compilé avec la version R 4.1.2
```

```
##
## Attachement du package : 'data.table'

## Les objets suivants sont masqués depuis 'package:dplyr':
##
##     between, first, last

## L'objet suivant est masqué depuis 'package:purrr':
##
##     transpose
```

```
library(caTools)
```

```
## Warning: le package 'caTools' a été compilé avec la version R 4.1.2
```

```
library(rpart)# Decision tree modeling
library(rpart.plot) # Decision tree ploting
```

```
## Warning: le package 'rpart.plot' a été compilé avec la version R 4.1.2
```

```
library(randomForest)
```

```
## Warning: le package 'randomForest' a été compilé avec la version R 4.1.2

## randomForest 4.6-14

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

##
## Attachement du package : 'randomForest'

## L'objet suivant est masqué depuis 'package:dplyr':
##
##     combine

## L'objet suivant est masqué depuis 'package:ggplot2':
##
##     margin
```

```
# Formating, Visualizations and tables
library(knitr) # Table
```

```
## Warning: le package 'knitr' a été compilé avec la version R 4.1.2
```

```
# Data handling Packages
library(tidyverse) # Data handling/ Graphics
library(data.table) # Data handling
```

Data loading

```r
set.seed(1987)
df_raw <- data.table::fread("Churn_Modelling.csv")
```

Data exploration

```r
## To get data structure
str(df_raw)
```

```
## Classes 'data.table' and 'data.frame':   10000 obs. of  14 variables:
##  $ RowNumber      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CustomerId     : int  15634602 15647311 15619304 15701354 15737888 15574012 15592531 15656148 1579
##  $ Surname        : chr  "Hargrave" "Hill" "Onio" "Boni" ...
##  $ CreditScore    : int  619 608 502 699 850 645 822 376 501 684 ...
##  $ Geography      : chr  "France" "Spain" "France" "France" ...
##  $ Gender         : chr  "Female" "Female" "Female" "Female" ...
##  $ Age            : int  42 41 42 39 43 44 50 29 44 27 ...
##  $ Tenure         : int  2 1 8 1 2 8 7 4 4 2 ...
##  $ Balance        : num  0 83808 159661 0 125511 ...
##  $ NumOfProducts  : int  1 1 3 2 1 2 2 4 2 1 ...
##  $ HasCrCard      : int  1 0 1 0 1 1 1 1 0 1 ...
##  $ IsActiveMember : int  1 1 0 0 1 0 1 0 1 1 ...
##  $ EstimatedSalary: num  101349 112543 113932 93827 79084 ...
##  $ Exited         : int  1 0 1 0 0 1 0 1 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

You can include R code in the document as follows:

```r
## To get an understanding of data
summary(df_raw)
```

```
##    RowNumber       CustomerId         Surname           CreditScore
##  Min.   :    1   Min.   :15565701   Length:10000       Min.   :350.0
##  1st Qu.: 2501   1st Qu.:15628528   Class :character   1st Qu.:584.0
##  Median : 5000   Median :15690738   Mode  :character   Median :652.0
##  Mean   : 5000   Mean   :15690941                      Mean   :650.5
##  3rd Qu.: 7500   3rd Qu.:15753234                      3rd Qu.:718.0
##  Max.   :10000   Max.   :15815690                      Max.   :850.0
##   Geography            Gender               Age            Tenure
##  Length:10000       Length:10000       Min.   :18.00   Min.   : 0.000
##  Class :character   Class :character   1st Qu.:32.00   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Median :37.00   Median : 5.000
##                                        Mean   :38.92   Mean   : 5.013
##                                        3rd Qu.:44.00   3rd Qu.: 7.000
##                                        Max.   :92.00   Max.   :10.000
##     Balance       NumOfProducts    HasCrCard      IsActiveMember
##  Min.   :     0   Min.   :1.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:     0   1st Qu.:1.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 97199   Median :1.00   Median :1.0000   Median :1.0000
##  Mean   : 76486   Mean   :1.53   Mean   :0.7055   Mean   :0.5151
##  3rd Qu.:127644   3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:1.0000
```

```
##  Max.   :250898   Max.   :4.00   Max.   :1.0000   Max.   :1.0000
##  EstimatedSalary         Exited
##  Min.   :     11.58   Min.   :0.0000
##  1st Qu.: 51002.11   1st Qu.:0.0000
##  Median :100193.91   Median :0.0000
##  Mean   :100090.24   Mean   :0.2037
##  3rd Qu.:149388.25   3rd Qu.:0.0000
##  Max.   :199992.48   Max.   :1.0000
```

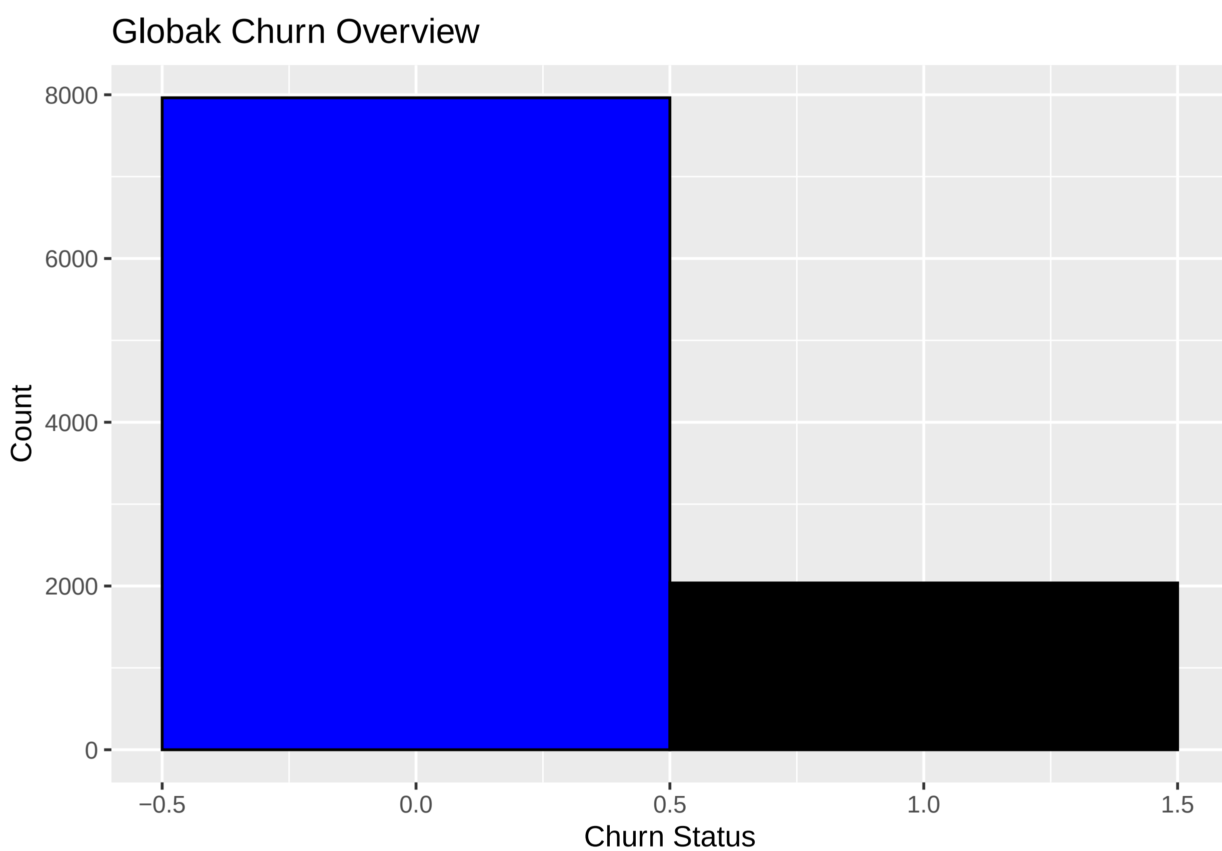
The churn status is in the column "Exited" 0 = Not Churn 1 = Churn

```
## Check if there is NA value in "Exited" column
df_raw%>%filter(is.na(Exited))%>%summarise(n())
```

```
##   n()
## 1   0
```

Global Churn overview

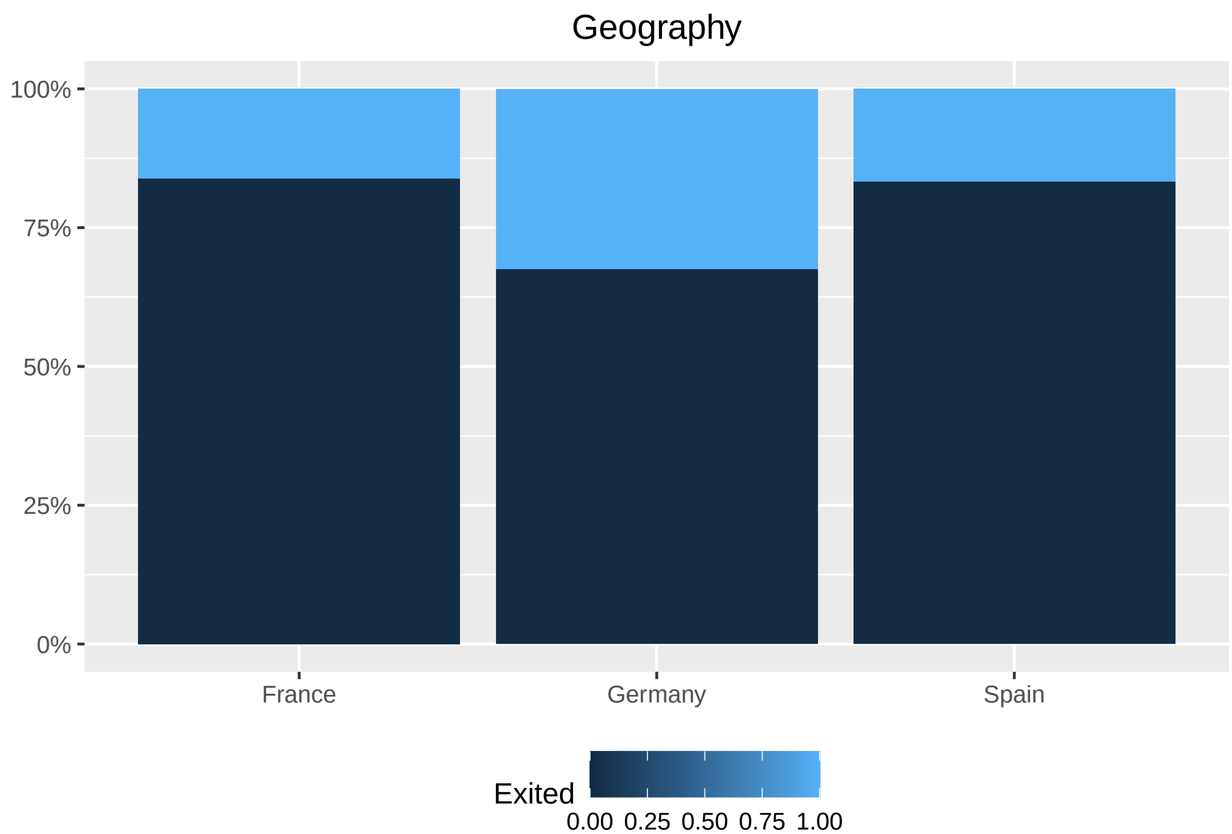
```
df_raw%>%ggplot(aes(Exited))+
  geom_histogram(binwidth = 1, fill = c("Blue", "Black"), col="black")+
  labs(title = "Globak Churn Overview" , x= "Churn Status", y= "Count")
```



Explore correlation between churn and other variables

Churn by geography

```
df_raw %>%
group_by(Geography, Exited) %>%count() %>%
  ggplot(aes(x = Geography, y = n, fill = Exited)) +
  geom_col(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(y = NULL, x = NULL) +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "bottom") +
  ggtitle("Geography")
```

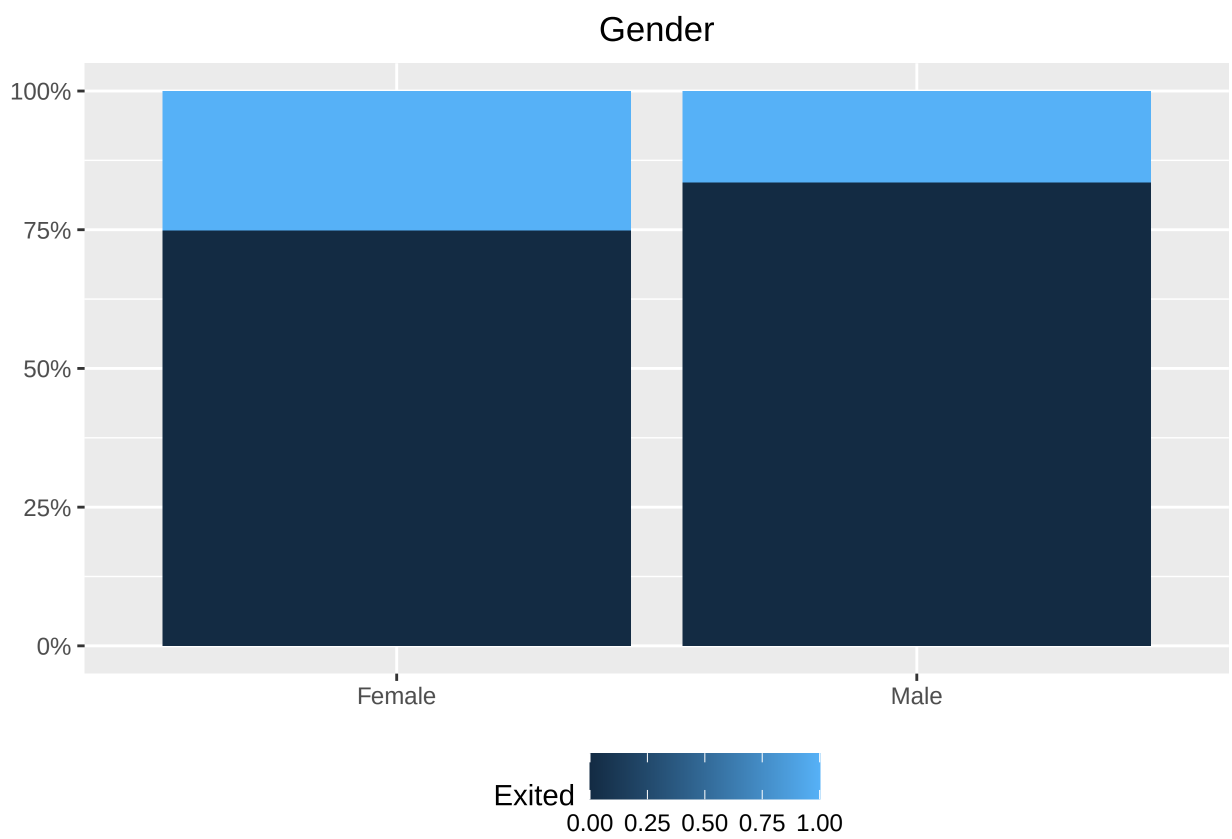


There is more churn in "Germany"

Churn by genre

```
df_raw %>%group_by(Gender, Exited) %>%count() %>%
  ggplot(aes(x = Gender, y = n, fill = Exited)) +
  geom_col(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(y = NULL, x = NULL) +
```

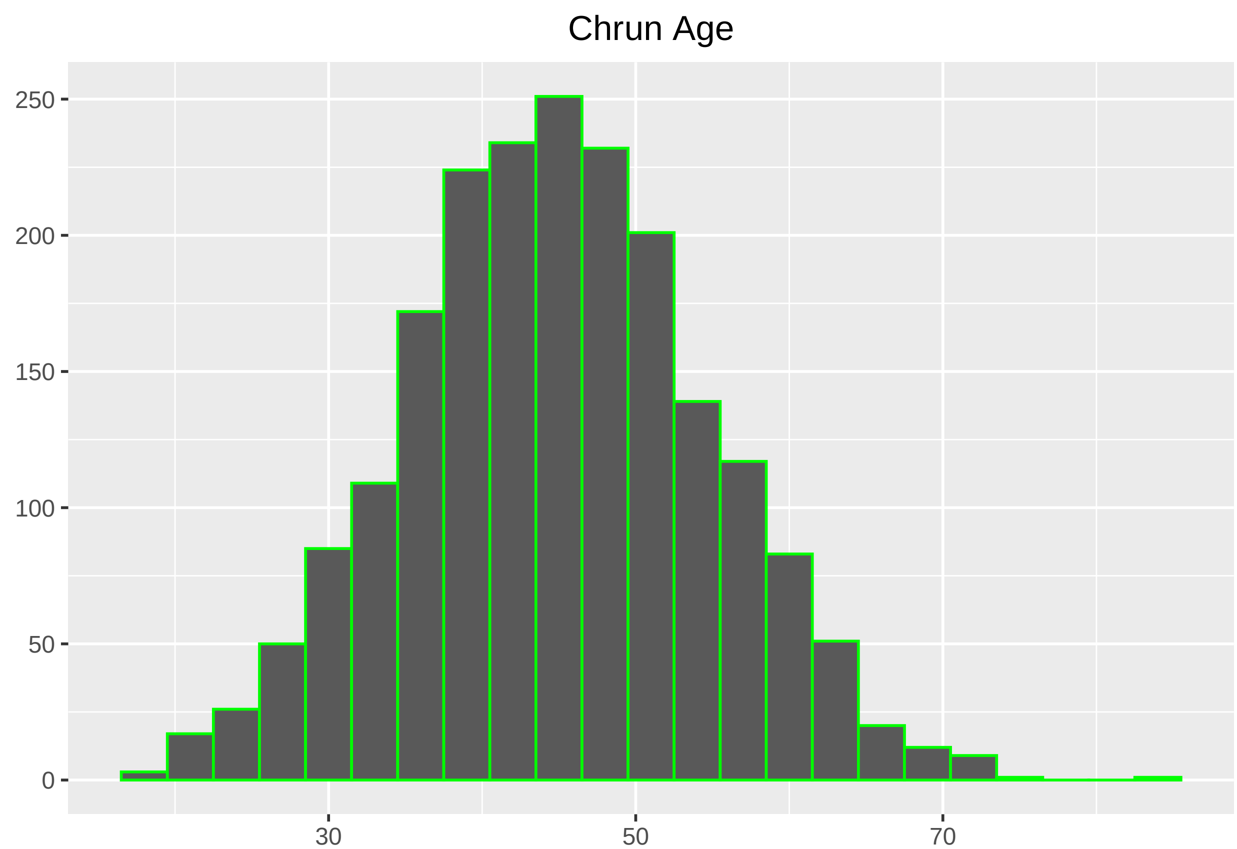
```
theme(plot.title = element_text(hjust = 0.5),
      legend.position = "bottom") +
ggtitle("Gender")
```



There also a slight effect of genre. Women churn more than men.

Churn distribution by age

```
df_raw %>%filter(Exited==1)%>%
group_by(Age) %>%
ggplot(aes(x = Age)) +
geom_histogram(color="green", binwidth = 3) +
labs(y = NULL, x = NULL) +
theme(plot.title = element_text(hjust = 0.5),
legend.position = "bottom") +
ggtitle("Chrun Age")
```



This plot show a normal distribution of the churn age with the average between 38 and 55.

So there is definitively an effect of age.

Churn by Tenure

```
df_raw %>%filter(Exited==1)%>%
group_by(Tenure) %>%
ggplot(aes(x = Tenure)) +
geom_histogram(color="green", binwidth = 1) +
labs(y = NULL, x = NULL) +
theme(plot.title = element_text(hjust = 0.5),
legend.position = "bottom") +
ggtitle("Chrun Age")
```


```r
#Average tenure before churned

avg_tenure<-df_raw %>%filter(Exited==1)%>%summarise(mean(Tenure))

round(avg_tenure)
```

What is the average tenure for exited customer

```
##   mean(Tenure)
## 1            5
```

The average is around 5 year. Mean the company shoud pay attention when a tenure year is near to 5.

Data Modeling

```r
## Keep only the variable needed for our models

df<-df_raw%>%select(-c(Surname,RowNumber,CustomerId))
head(df)
```

```
##    CreditScore Geography Gender Age Tenure   Balance NumOfProducts HasCrCard
## 1:         619    France Female  42      2      0.00             1         1
## 2:         608     Spain Female  41      1  83807.86             1         0
## 3:         502    France Female  42      8 159660.80             3         1
## 4:         699    France Female  39      1      0.00             2         0
## 5:         850     Spain Female  43      2 125510.82             1         1
## 6:         645     Spain   Male  44      8 113755.78             2         1
##    IsActiveMember EstimatedSalary Exited
## 1:              1       101348.88      1
## 2:              1       112542.58      0
## 3:              0       113931.57      1
## 4:              0        93826.63      0
## 5:              1        79084.10      0
## 6:              0       149756.71      1
```

```
#Create data partition into a training and testing dataset
set.seed(1987)
index<-createDataPartition(y=df$Exited, p=.75, list = FALSE)# partition indexes
train<-df[index] # Create training partition
test<-df[-index] # Create testing partition
head(train)
```

```
##    CreditScore Geography Gender Age Tenure  Balance NumOfProducts HasCrCard
## 1:         619    France Female  42      2      0.0             1         1
## 2:         502    France Female  42      8 159660.8             3         1
## 3:         699    France Female  39      1      0.0             2         0
## 4:         850     Spain Female  43      2 125510.8             1         1
## 5:         645     Spain   Male  44      8 113755.8             2         1
## 6:         501    France   Male  44      4 142051.1             2         0
##    IsActiveMember EstimatedSalary Exited
## 1:              1       101348.88      1
## 2:              0       113931.57      1
## 3:              0        93826.63      0
## 4:              1        79084.10      0
## 5:              0       149756.71      1
## 6:              1        74940.50      0
```

```
#Table to collect the models performance
table <- tibble(Model="Begin", Acc=0.0)
```

Model 1: Logistic regression

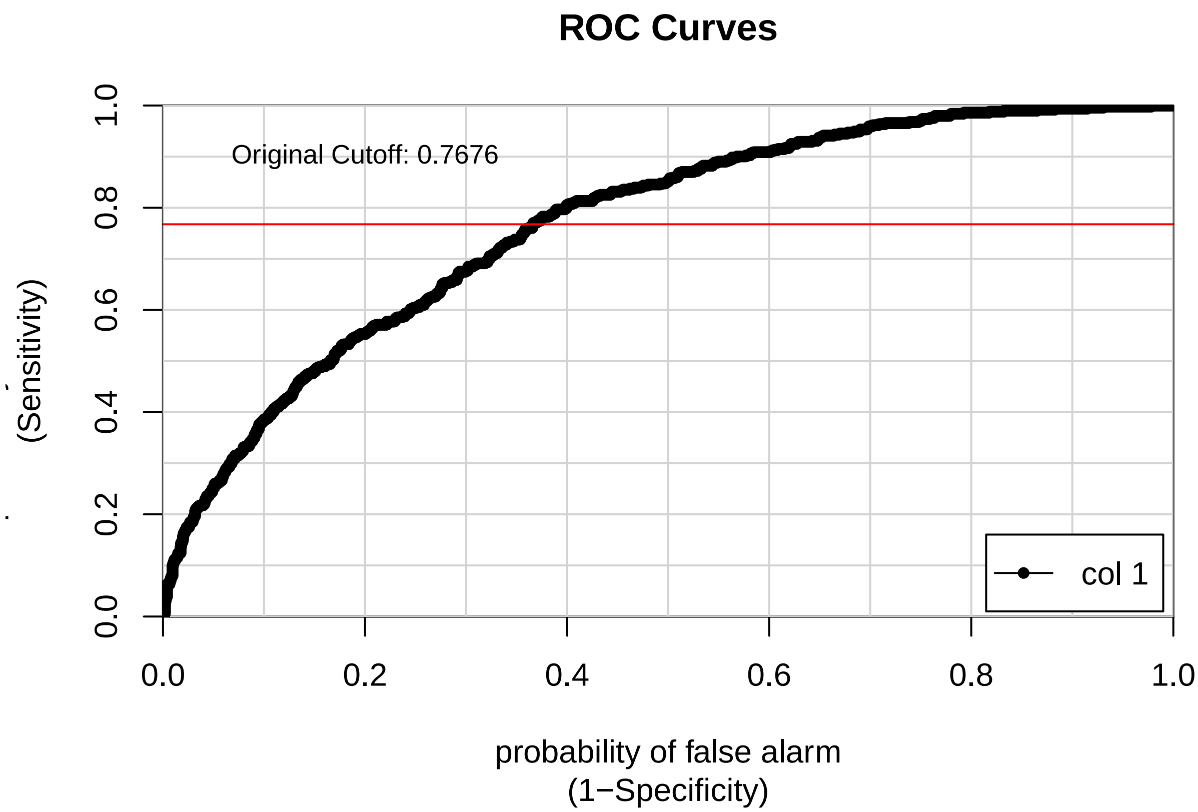
```
set.seed(1987)
# Modeling logistic regression
model1<-glm(train$Exited ~ . , family = "binomial", train)
# Model summary data
summary(model1)
```

```
##
## Call:
## glm(formula = train$Exited ~ ., family = "binomial", data = train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2633  -0.6649  -0.4564  -0.2694   2.9977
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.439e+00  2.799e-01 -12.284  < 2e-16 ***
## CreditScore      -5.923e-04  3.200e-04  -1.851   0.0642 .
## GeographyGermany  7.012e-01  7.791e-02   9.000  < 2e-16 ***
## GeographySpain   -4.579e-03  8.146e-02  -0.056   0.9552
## GenderMale       -5.477e-01  6.274e-02  -8.731  < 2e-16 ***
## Age               7.267e-02  2.975e-03  24.427  < 2e-16 ***
## Tenure           -1.582e-02  1.083e-02  -1.461   0.1441
## Balance           2.825e-06  5.891e-07   4.796 1.62e-06 ***
## NumOfProducts    -1.009e-01  5.391e-02  -1.872   0.0613 .
## HasCrCard        -4.519e-02  6.835e-02  -0.661   0.5085
## IsActiveMember   -1.096e+00  6.640e-02 -16.513  < 2e-16 ***
## EstimatedSalary   8.911e-07  5.425e-07   1.643   0.1005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7629.1  on 7499  degrees of freedom
## Residual deviance: 6460.4  on 7488  degrees of freedom
## AIC: 6484.4
##
## Number of Fisher Scoring iterations: 5
```

```r
# Now we will predict
# Make the prediction on testing data
pred1<-predict(model1, test, type="response")
```

```r
#Generate the ROC curve the determine the cut-off

model.AUC<-colAUC(pred1, test$Exited, plotROC=T)
abline(h = model.AUC, col="red")
text(.2, .9, cex=.8, labels=paste("Original Cutoff:", round(model.AUC,4)))
```



The cutoff value is : 0.7676

```r
# Now we can use conditional expression to make the prediction
classification<-ifelse(pred1>0.7676, 1, 0)
classification<-factor(classification)
```

```r
#Confusion Matrix to determine the Accuracy
confusionMatrix(classification,
                factor(test$Exited))$overall["Accuracy"]
```

```
## Accuracy
##   0.8056
```

```r
result1<-confusionMatrix(classification,
                         factor(test$Exited))$overall["Accuracy"]
```

We have got about 80.56% of Accuracy.

```r
#Update the result table

table <- bind_rows(table,
                   tibble(Model = "Logistic regression",
                          Acc = result1))
kable(table)
```

Model	Acc
Begin	0.0000
Logistic regression	0.8056

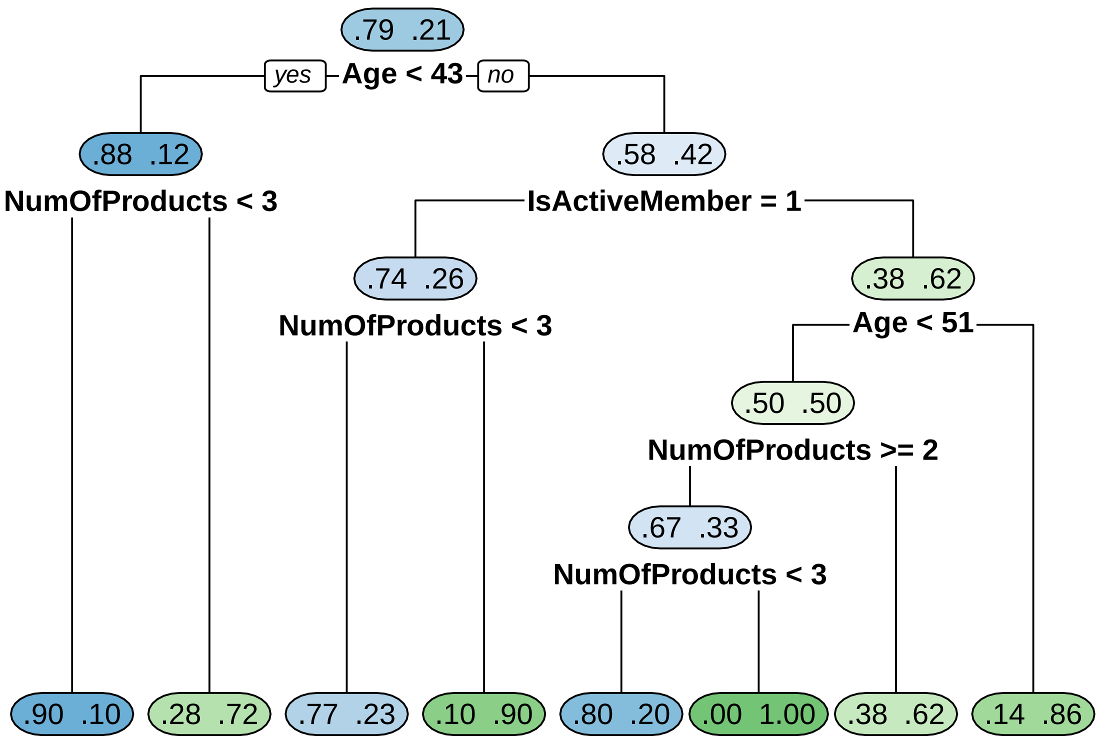
Model 2: Decision Tree matrix

```
## For the following models, we will update the train and test data
train<-train%>%mutate(Exited=factor(Exited))
test<-test%>%mutate(Exited=factor(Exited))
```

```
#Build decision tree model
set.seed(1987)

df_tree<-rpart(train$Exited ~ ., data = train)

# Check the variable importance
df_tree$variable.importance
```

```
##             Age   NumOfProducts  IsActiveMember         Balance     CreditScore
##      346.920989      217.888689      143.529357        4.033815        3.068414
## EstimatedSalary          Tenure
##        2.357311        1.034712
```

```
#Plot the decision tree

rpart.plot(df_tree, extra=5)
```




```
#Make a prediction using decision tree
pred2<-predict(df_tree, train, type="class")
```

```
#Accuracy on train set
confusionMatrix(pred2, train$Exited)$overall["Accuracy"]
```

```
## Accuracy
##   0.8564
```

```
#re-apply all decision tree steps on test data set
set.seed(1987)
df_tree<-rpart(test$Exited ~ ., data = test)

# Check summary information
summary(df_tree)
```

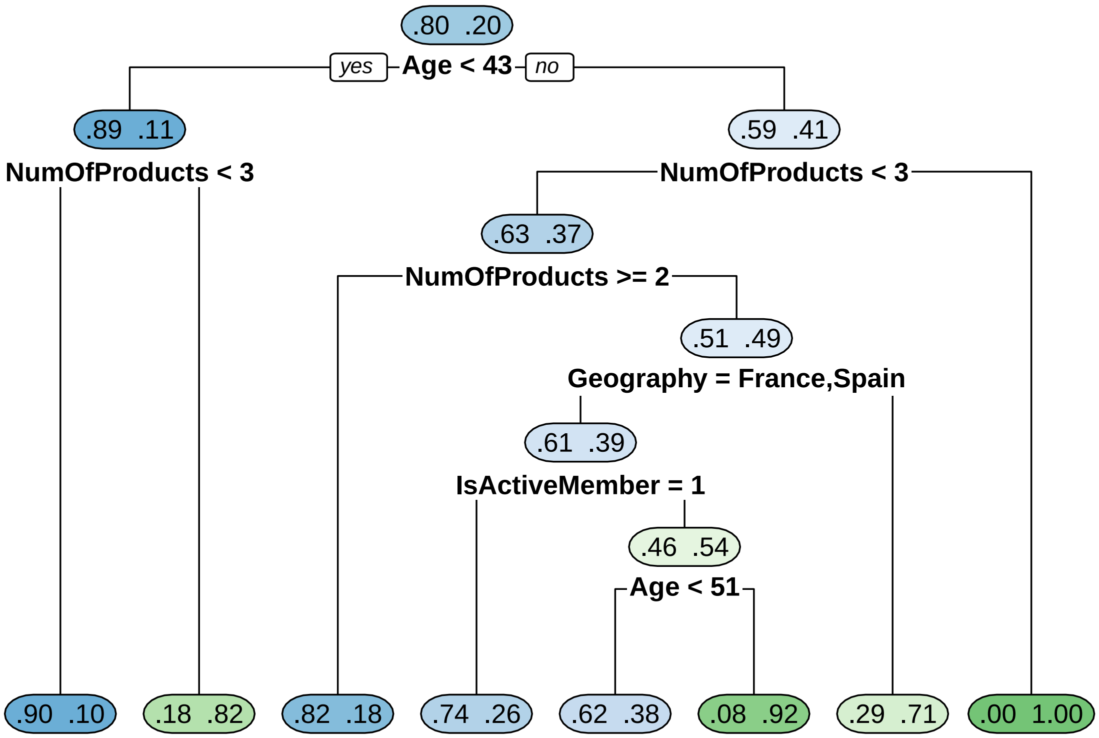
```
## Call:
## rpart(formula = test$Exited ~ ., data = test)
##   n= 2500
##
##           CP nsplit rel error    xerror       xstd
## 1 0.04132791      0 1.0000000 1.0000000 0.04040446
## 2 0.03252033      5 0.7642276 0.8292683 0.03755570
## 3 0.01000000      7 0.6991870 0.7276423 0.03559722
```

```
##
## Variable importance
##             Age   NumOfProducts       Geography  IsActiveMember         Balance
##              44              39               8               4               3
## EstimatedSalary
##               1
##
## Node number 1: 2500 observations,    complexity param=0.04132791
##   predicted class=0  expected loss=0.1968  P(node) =1
##     class counts:  2008   492
##    probabilities: 0.803 0.197
##   left son=2 (1781 obs) right son=3 (719 obs)
##   Primary splits:
##       Age            < 42.5     to the left,  improve=88.44124, (0 missing)
##       NumOfProducts  < 2.5      to the left,  improve=78.18141, (0 missing)
##       Geography      splits as  LRL,          improve=29.82222, (0 missing)
##       IsActiveMember < 0.5      to the right, improve=15.20159, (0 missing)
##       Balance        < 61626.58 to the left,  improve=11.67042, (0 missing)
##   Surrogate splits:
##       NumOfProducts   < 3.5      to the left,  agree=0.715, adj=0.010, (0 split)
##       CreditScore     < 367.5    to the right, agree=0.713, adj=0.001, (0 split)
##       EstimatedSalary < 150.63   to the right, agree=0.713, adj=0.001, (0 split)
##
## Node number 2: 1781 observations,    complexity param=0.04132791
##   predicted class=0  expected loss=0.1122965  P(node) =0.7124
##     class counts:  1581   200
##    probabilities: 0.888 0.112
##   left son=4 (1747 obs) right son=5 (34 obs)
##   Primary splits:
##       NumOfProducts  < 2.5      to the left,  improve=35.067410, (0 missing)
##       Geography      splits as  LRL,          improve= 8.524725, (0 missing)
##       Age            < 34.5     to the left,  improve= 6.679336, (0 missing)
##       IsActiveMember < 0.5      to the right, improve= 5.591198, (0 missing)
##       Balance        < 38523.81 to the left,  improve= 4.052402, (0 missing)
##
## Node number 3: 719 observations,    complexity param=0.04132791
##   predicted class=0  expected loss=0.4061196  P(node) =0.2876
##     class counts:   427   292
##    probabilities: 0.594 0.406
##   left son=6 (680 obs) right son=7 (39 obs)
##   Primary splits:
##       NumOfProducts  < 2.5      to the left,  improve=29.087910, (0 missing)
##       Geography      splits as  LRL,          improve=20.333670, (0 missing)
##       IsActiveMember < 0.5      to the right, improve=17.170350, (0 missing)
##       Balance        < 81002.18 to the left,  improve=12.796910, (0 missing)
##       Age            < 65.5     to the right, improve= 9.903962, (0 missing)
##
## Node number 4: 1747 observations
##   predicted class=0  expected loss=0.09845449  P(node) =0.6988
##     class counts:  1575   172
##    probabilities: 0.902 0.098
##
## Node number 5: 34 observations
##   predicted class=1  expected loss=0.1764706  P(node) =0.0136
```

```
##     class counts:     6    28
##    probabilities: 0.176 0.824
##
## Node number 6: 680 observations,    complexity param=0.04132791
##   predicted class=0  expected loss=0.3720588  P(node) =0.272
##     class counts:   427   253
##    probabilities: 0.628 0.372
##   left son=12 (262 obs) right son=13 (418 obs)
##   Primary splits:
##       NumOfProducts  < 1.5      to the right, improve=30.402580, (0 missing)
##       Geography      splits as  LRL,          improve=20.808350, (0 missing)
##       IsActiveMember < 0.5      to the right, improve=16.478970, (0 missing)
##       Balance        < 81002.18 to the left,  improve=13.417100, (0 missing)
##       Age            < 65.5     to the right, improve= 9.163541, (0 missing)
##   Surrogate splits:
##       Balance         < 11751.66 to the left,  agree=0.713, adj=0.256, (0 split)
##       Age             < 43.5     to the left,  agree=0.624, adj=0.023, (0 split)
##       EstimatedSalary < 192293.6 to the right, agree=0.619, adj=0.011, (0 split)
##
## Node number 7: 39 observations
##   predicted class=1  expected loss=0  P(node) =0.0156
##     class counts:     0    39
##    probabilities: 0.000 1.000
##
## Node number 12: 262 observations
##   predicted class=0  expected loss=0.1832061  P(node) =0.1048
##     class counts:   214    48
##    probabilities: 0.817 0.183
##
## Node number 13: 418 observations,    complexity param=0.04132791
##   predicted class=0  expected loss=0.4904306  P(node) =0.1672
##     class counts:   213   205
##    probabilities: 0.510 0.490
##   left son=26 (285 obs) right son=27 (133 obs)
##   Primary splits:
##       Geography       splits as  LRL,          improve=18.258780, (0 missing)
##       IsActiveMember  < 0.5      to the right, improve=15.484740, (0 missing)
##       Age             < 66.5     to the right, improve=10.007470, (0 missing)
##       EstimatedSalary < 143347.6 to the left,  improve= 7.160993, (0 missing)
##       Gender          splits as  RL,           improve= 3.973540, (0 missing)
##   Surrogate splits:
##       CreditScore     < 440.5    to the right, agree=0.687, adj=0.015, (0 split)
##       EstimatedSalary < 197250.8 to the left,  agree=0.687, adj=0.015, (0 split)
##
## Node number 26: 285 observations,    complexity param=0.03252033
##   predicted class=0  expected loss=0.3894737  P(node) =0.114
##     class counts:   174   111
##    probabilities: 0.611 0.389
##   left son=52 (155 obs) right son=53 (130 obs)
##   Primary splits:
##       IsActiveMember  < 0.5      to the right, improve=10.611780, (0 missing)
##       Balance         < 35589.1  to the right, improve= 6.194895, (0 missing)
##       Age             < 66.5     to the right, improve= 5.642960, (0 missing)
##       EstimatedSalary < 143347.6 to the left,  improve= 5.441737, (0 missing)
```

```
##       Gender          splits as  RL,          improve= 2.277895, (0 missing)
##   Surrogate splits:
##       Age             < 49.5     to the right, agree=0.656, adj=0.246, (0 split)
##       EstimatedSalary < 120182.4 to the left,  agree=0.582, adj=0.085, (0 split)
##       CreditScore     < 654      to the right, agree=0.579, adj=0.077, (0 split)
##       Geography       splits as  R-L,          agree=0.572, adj=0.062, (0 split)
##       Balance         < 161549.7 to the left,  agree=0.572, adj=0.062, (0 split)
##
## Node number 27: 133 observations
##   predicted class=1  expected loss=0.2932331  P(node) =0.0532
##     class counts:    39    94
##    probabilities: 0.293 0.707
##
## Node number 52: 155 observations
##   predicted class=0  expected loss=0.2645161  P(node) =0.062
##     class counts:   114    41
##    probabilities: 0.735 0.265
##
## Node number 53: 130 observations,    complexity param=0.03252033
##   predicted class=1  expected loss=0.4615385  P(node) =0.052
##     class counts:    60    70
##    probabilities: 0.462 0.538
##   left son=106 (92 obs) right son=107 (38 obs)
##   Primary splits:
##       Age             < 50.5     to the left,  improve=15.7195000, (0 missing)
##       EstimatedSalary < 159324.7 to the left,  improve= 3.3313540, (0 missing)
##       Balance         < 24556.88 to the right, improve= 2.2624430, (0 missing)
##       CreditScore     < 525      to the left,  improve= 1.5384620, (0 missing)
##       Gender          splits as  RL,           improve= 0.8060867, (0 missing)
##   Surrogate splits:
##       EstimatedSalary < 182498.4 to the left,  agree=0.723, adj=0.053, (0 split)
##
## Node number 106: 92 observations
##   predicted class=0  expected loss=0.3804348  P(node) =0.0368
##     class counts:    57    35
##    probabilities: 0.620 0.380
##
## Node number 107: 38 observations
##   predicted class=1  expected loss=0.07894737  P(node) =0.0152
##     class counts:     3    35
##    probabilities: 0.079 0.921
```

```r
rpart.plot(df_tree, extra=5)
```




```
#Make a prediction using decision tree
pred2<-predict(df_tree, test, type="class")


result2<-confusionMatrix(pred2, test$Exited)$overall["Accuracy"]


#Insert in the resut table
table <- bind_rows(table,
                   tibble(Model = "Decision Tree",
                          Acc = result2))
kable(table)
```

Model	Acc
Begin	0.0000
Logistic regression	0.8056
Decision Tree	0.8624

Decision tree give a better prediction with 86,24% accuracy. That is correct estimation.

Model 3: Random forest

```
set.seed(1987)
control <- trainControl(method="cv", number = 5)
grid <- data.frame(mtry = c(1, 5, 10, 25, 50, 100))

pred3<- train(Exited ~ ., method = "rf",
                   data = train,
                   tuneGrid = grid,
      ntree = 150,
                   trControl = control,
                 )

result3<-confusionMatrix(predict(pred3,test , type = "raw"),
                test$Exited)$overall["Accuracy"]

table <- bind_rows(table,
                   tibble(Model = "Random Forest",
                          Acc = result3))

kable(table)
```

Model	Acc
Begin	0.0000
Logistic regression	0.8056
Decision Tree	0.8624
Random Forest	0.8552

Accuracy of 85,52% for random forest model

Doing great, but still under decision tree model

Conclusion - Model comparison

The final table of accuracy is here :

```
kable(table)
```

Model	Acc
Begin	0.0000
Logistic regression	0.8056
Decision Tree	0.8624
Random Forest	0.8552

By looking at this table, we can conclude that the best model for customer chrun prediction is "Decision Tree"