Prevendo a Rotatividade de Clientes em uma Operadora de Telecomunicações

A rotatividade de clientes ocorre quando clientes ou assinantes param de fazer negócios com uma empresa ou serviço, também conhecido como atrito com clientes. Também é referido como perda de clientes ou clientes. Um setor no qual as taxas de cancelamento são particularmente úteis é o setor de telecomunicações, porque a maioria dos clientes tem várias opções de escolha dentro de uma localização geográfica.

Vamos prever a rotatividade de clientes usando o conjunto de dados de telecomunicações. Introduziremos a regressão logística, a *Decision Tree* e a *Random Florest*.

```
library(plyr)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(MASS)
library(caret)
library(randomForest)
library(party)
```

The data was downloaded from IBM Sample Data Sets. Each row represents a customer, each column contains that customer's attributes:

```
churn <- read.csv('Telco-Customer-Churn.csv')
str(churn)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 6552 1003 4771 5605 4535 ..
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

As variáveis contidas no *dataset* são:

- customerID
- gender (female, male)
- SeniorCitizen (Whether the customer is a senior citizen or not (1, 0))
- Partner (Whether the customer has a partner or not (Yes, No))
- Dependents (Whether the customer has dependents or not (Yes, No))
- tenure (Number of months the customer has stayed with the company)
- PhoneService (Whether the customer has a phone service or not (Yes, No))
- MultipleLines (Whether the customer has multiple lines r not (Yes, No, No phone service)
- InternetService (Customer's internet service provider (DSL, Fiber optic, No)

- OnlineSecurity (Whether the customer has online security or not (Yes, No, No internet service)
- OnlineBackup (Whether the customer has online backup or not (Yes, No, No internet service)
- DeviceProtection (Whether the customer has device protection or not (Yes, No, No internet service)
- TechSupport (Whether the customer has tech support or not (Yes, No, No internet service)
- streamingTV (Whether the customer has streaming TV or not (Yes, No, No internet service)
- streamingMovies (Whether the customer has streaming movies or not (Yes, No, No internet service)
- Contract (The contract term of the customer (Month-to-month, One year, Two year)
- PaperlessBilling (Whether the customer has paperless billing or not (Yes, No))
- PaymentMethod (The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)))
- MonthlyCharges (The amount charged to the customer monthly)
- TotalCharges (The total amount charged to the customer)
- Churn (Whether the customer churned or not (Yes or No))

The raw data contains 7043 rows (customers) and 21 columns (features). The "Churn" column is our target. We'll use all other columns as features to our model.

We use sapply to check the number if missing values in each columns. We found that there are 11 missing values in "TotalCharges" columns. So, let's remove these rows with missing values.

```
sapply(churn, function(x) sum(is.na(x)))
```

```
##       customerID           gender    SeniorCitizen          Partner
##                0                0                0                0
##       Dependents           tenure     PhoneService    MultipleLines
##                0                0                0                0
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                0                0                0                0
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                0                0                0                0
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                0                0                0               11
##            Churn
##                0
```

```
churn <- churn[complete.cases(churn), ]
```

Change "No internet service" to "No" for six columns, they are: "OnlineSecurity", "OnlineBackup", "DeviceProtection", "TechSupport", "streamingTV", "streamingMovies".

```
cols_recode1 <- c(10:15)
for(i in 1:ncol(churn[,cols_recode1])) {
        churn[,cols_recode1][,i] <- as.factor(mapvalues
                                               (churn[,cols_recode1][,i], from =c("No internet service"),to=c("No")))
}
```

Change "No phone service" to "No" for column "MultipleLines"

```
churn$MultipleLines <- as.factor(mapvalues(churn$MultipleLines,
                                           from=c("No phone service"),
                                           to=c("No")))
```

The minimum tenure is 1 month and maximum tenure is 72 months, we can group them into five tenure groups: "0–12 Month", "12–24 Month", "24–48 Months", "48–60 Month", "> 60 Month".

```
min(churn$tenure); max(churn$tenure)
```

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

```
## [1] 72
```

```
group_tenure <- function(tenure){
    if (tenure >= 0 & tenure <= 12){
        return('0-12 Month')
    }else if(tenure > 12 & tenure <= 24){
        return('12-24 Month')
    }else if (tenure > 24 & tenure <= 48){
```

```
        return('24-48 Month')
    }else if (tenure > 48 & tenure <=60){
        return('48-60 Month')
    }else if (tenure > 60){
        return('> 60 Month')
    }
}
```

```
churn$tenure_group <- sapply(churn$tenure,group_tenure)
churn$tenure_group <- as.factor(churn$tenure_group)
```

Change the values in column “SeniorCitizen” from 0 or 1 to “No” or “Yes”.

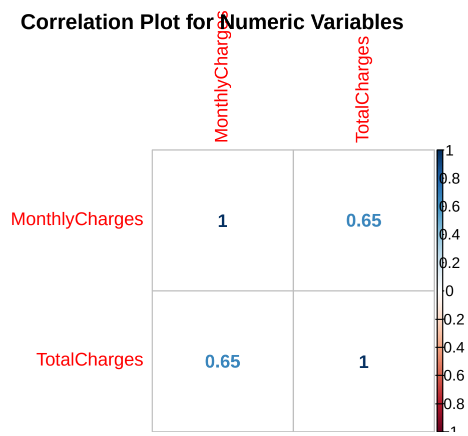
```
churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,
                                 from=c("0","1"),
                                 to=c("No", "Yes")))
```

Remove the columns we do not need for the analysis:

```
churn$customerID <- NULL
churn$tenure <- NULL
```

Exploratory data analysis and feature selection

```
numeric.var <- sapply(churn, is.numeric) ## Find numerical variables
corr.matrix <- cor(churn[,numeric.var])  ## Calculate the correlation matrix
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numeric Variables", method="number")
```

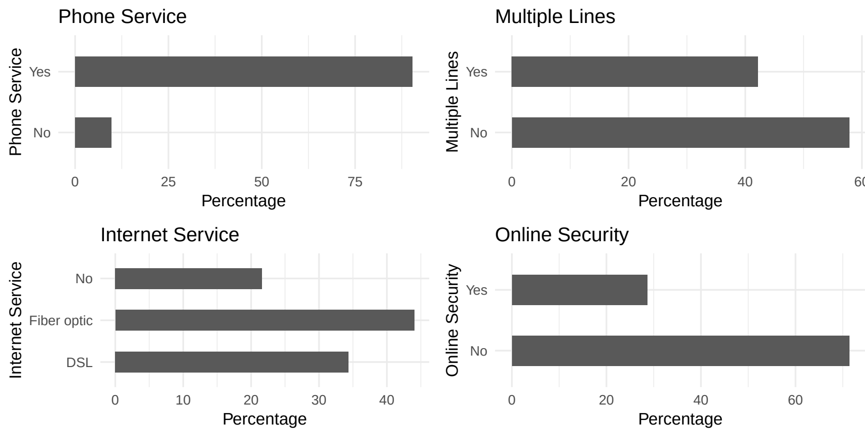


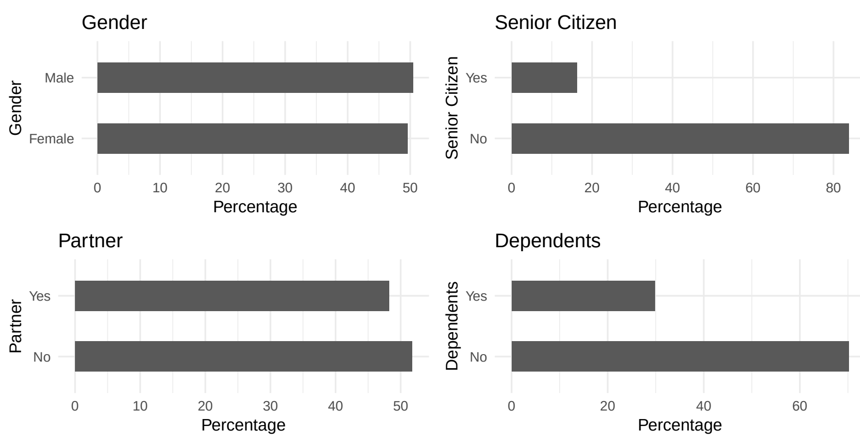
The Monthly Charges and Total Charges are correlated. So one of them will be removed from the model. We remove Total Charges.

```
churn$TotalCharges <- NULL
```

Bar plots of categorical variables

```
p1 <- ggplot(churn, aes(x=gender)) + ggtitle("Gender") + xlab("Gender") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p2 <- ggplot(churn, aes(x=SeniorCitizen)) + ggtitle("Senior Citizen") + xlab("Senior Citizen") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p3 <- ggplot(churn, aes(x=Partner)) + ggtitle("Partner") + xlab("Partner") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p4 <- ggplot(churn, aes(x=Dependents)) + ggtitle("Dependents") + xlab("Dependents") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
grid.arrange(p1, p2, p3, p4, ncol=2)
```

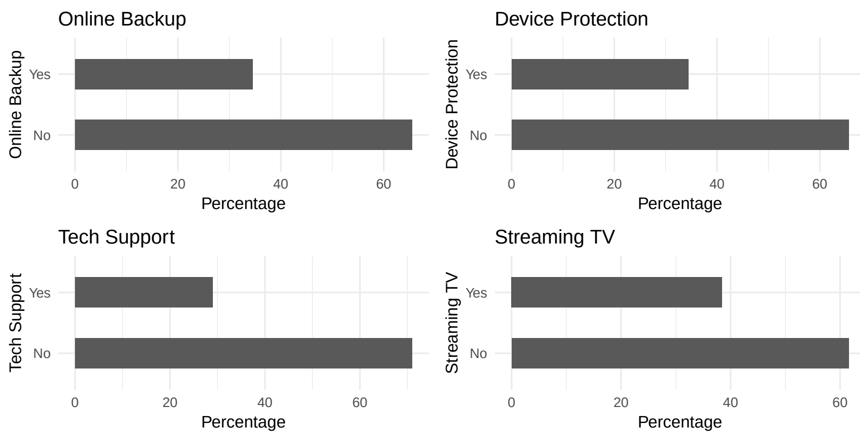


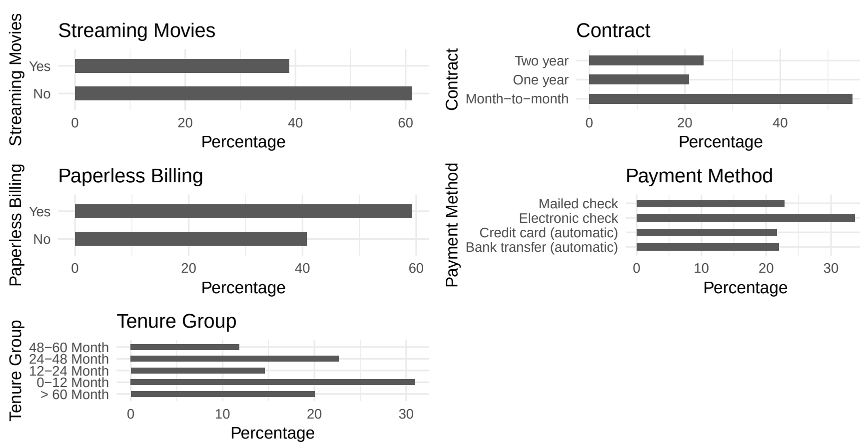



```
p5 <- ggplot(churn, aes(x=PhoneService)) + ggtitle("Phone Service") + xlab("Phone Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p6 <- ggplot(churn, aes(x=MultipleLines)) + ggtitle("Multiple Lines") + xlab("Multiple Lines") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p7 <- ggplot(churn, aes(x=InternetService)) + ggtitle("Internet Service") + xlab("Internet Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p8 <- ggplot(churn, aes(x=OnlineSecurity)) + ggtitle("Online Security") + xlab("Online Security") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
grid.arrange(p5, p6, p7, p8, ncol=2)
```

```
p9 <- ggplot(churn, aes(x=OnlineBackup)) + ggtitle("Online Backup") + xlab("Online Backup") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p10 <- ggplot(churn, aes(x=DeviceProtection)) + ggtitle("Device Protection") + xlab("Device Protection") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p11 <- ggplot(churn, aes(x=TechSupport)) + ggtitle("Tech Support") + xlab("Tech Support") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p12 <- ggplot(churn, aes(x=StreamingTV)) + ggtitle("Streaming TV") + xlab("Streaming TV") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
grid.arrange(p9, p10, p11, p12, ncol=2)
```

```
p13 <- ggplot(churn, aes(x=StreamingMovies)) + ggtitle("Streaming Movies") + xlab("Streaming Movies") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p14 <- ggplot(churn, aes(x=Contract)) + ggtitle("Contract") + xlab("Contract") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p15 <- ggplot(churn, aes(x=PaperlessBilling)) + ggtitle("Paperless Billing") + xlab("Paperless Billing") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p16 <- ggplot(churn, aes(x=PaymentMethod)) + ggtitle("Payment Method") + xlab("Payment Method") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p17 <- ggplot(churn, aes(x=tenure_group)) + ggtitle("Tenure Group") + xlab("Tenure Group") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
grid.arrange(p13, p14, p15, p16, p17, ncol=2)
```


Online Backup
Yes
No
0
20
40
60
Percentage
Device Protection
Tech Support
Streaming TV


Streaming Movies
Yes
No
Percentage
Contract
Two year
One year
Month−to−month
Paperless Billing
Payment Method
Mailed check
Electronic check
Credit card (automatic)
Bank transfer (automatic)
Tenure Group
48−60 Month
24−48 Month
12−24 Month
0−12 Month
> 60 Month

Todas as variáveis categóricas têm uma distribuição ampla razoável, portanto, todas elas serão mantidas para análise posterior.

Logistic Regression Model Fitting

Split the data into training and testing sets.

```
intrain<- createDataPartition(churn$Churn,p=0.7,list=FALSE)
set.seed(2017)
training<- churn[intrain,]
testing<- churn[-intrain,]
```

Confirm the splitting is correct.

```
dim(training); dim(testing)
```

```
## [1] 4924   19
```

```
## [1] 2108   19
```

Fitting the Model

```
LogModel <- glm(Churn ~ .,family=binomial(link="logit"),data=training)
print(summary(LogModel))
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial(link = "logit"), data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0190  -0.6698  -0.2962   0.6800   3.0992
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                         -2.0602044  0.9787545  -2.105
## genderMale                           0.0651685  0.0773433   0.843
## SeniorCitizenYes                     0.2632649  0.0997455   2.639
## PartnerYes                           0.0363204  0.0919794   0.395
## DependentsYes                       -0.1754284  0.1070587  -1.639
## PhoneServiceYes                     -0.5197606  0.7729906  -0.672
## MultipleLinesYes                     0.2977188  0.2102272   1.416
## InternetServiceFiber optic           0.8766620  0.9489050   0.924
## InternetServiceNo                   -1.0001763  0.9603425  -1.041
## OnlineSecurityYes                   -0.3172461  0.2117680  -1.498
## OnlineBackupYes                     -0.1843978  0.2096893  -0.879
## DeviceProtectionYes                 -0.0563110  0.2100341  -0.268
## TechSupportYes                      -0.3398058  0.2133462  -1.593
## StreamingTVYes                       0.2961443  0.3889339   0.761
## StreamingMoviesYes                   0.2420115  0.3883068   0.623
## ContractOne year                    -0.6628568  0.1267019  -5.232
## ContractTwo year                    -1.6102677  0.2183167  -7.376
## PaperlessBillingYes                  0.3245029  0.0886986   3.658
## PaymentMethodCredit card (automatic) -0.0006838  0.1364818  -0.005
## PaymentMethodElectronic check        0.3095810  0.1135705   2.726
## PaymentMethodMailed check            0.0850955  0.1372632   0.620
## MonthlyCharges                      -0.0007481  0.0377173  -0.020
## tenure_group0-12 Month               1.8743873  0.2067993   9.064
## tenure_group12-24 Month              0.9375341  0.2027863   4.623
## tenure_group24-48 Month              0.5607222  0.1865340   3.006
## tenure_group48-60 Month              0.3545120  0.2000965   1.772
##                                     Pr(>|z|)
## (Intercept)                         0.035298 *
## genderMale                          0.399459
## SeniorCitizenYes                    0.008306 **
## PartnerYes                          0.692935
## DependentsYes                       0.101293
## PhoneServiceYes                     0.501328
```

```
## MultipleLinesYes                    0.156724
## InternetServiceFiber optic          0.355556
## InternetServiceNo                   0.297653
## OnlineSecurityYes                   0.134112
## OnlineBackupYes                     0.379192
## DeviceProtectionYes                 0.788619
## TechSupportYes                      0.111218
## StreamingTVYes                      0.446403
## StreamingMoviesYes                  0.533121
## ContractOne year                    1.68e-07 ***
## ContractTwo year                    1.63e-13 ***
## PaperlessBillingYes                 0.000254 ***
## PaymentMethodCredit card (automatic) 0.996002
## PaymentMethodElectronic check       0.006413 **
## PaymentMethodMailed check           0.535295
## MonthlyCharges                      0.984176
## tenure_group0-12 Month               < 2e-16 ***
## tenure_group12-24 Month             3.78e-06 ***
## tenure_group24-48 Month             0.002647 **
## tenure_group48-60 Month             0.076444 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5702.8  on 4923  degrees of freedom
## Residual deviance: 4116.8  on 4898  degrees of freedom
## AIC: 4168.8
##
## Number of Fisher Scoring iterations: 6
```

Feature analysis:

1. The top three most-relevant features include Contract, Paperless Billing and tenure group, all of which are categorical variables.

```
anova(LogModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Churn
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                              4923     5702.8
## gender            1     0.32      4922     5702.4 0.5739264
## SeniorCitizen     1   111.80      4921     5590.6 < 2.2e-16 ***
## Partner           1   110.00      4920     5480.6 < 2.2e-16 ***
## Dependents        1    33.30      4919     5447.3 7.879e-09 ***
## PhoneService      1     0.03      4918     5447.3 0.8640550
## MultipleLines     1     8.17      4917     5439.1 0.0042597 **
## InternetService   2   457.10      4915     4982.0 < 2.2e-16 ***
## OnlineSecurity    1   150.41      4914     4831.6 < 2.2e-16 ***
## OnlineBackup      1    80.84      4913     4750.8 < 2.2e-16 ***
## DeviceProtection  1    50.92      4912     4699.9 9.644e-13 ***
## TechSupport       1    85.50      4911     4614.4 < 2.2e-16 ***
## StreamingTV       1     1.54      4910     4612.8 0.2140591
## StreamingMovies   1     0.02      4909     4612.8 0.8782387
## Contract          2   280.30      4907     4332.5 < 2.2e-16 ***
## PaperlessBilling  1    13.82      4906     4318.7 0.0002015 ***
## PaymentMethod     3    28.74      4903     4290.0 2.543e-06 ***
## MonthlyCharges    1     0.00      4902     4289.9 0.9734711
## tenure_group      4   173.17      4898     4116.8 < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analyzing the deviance table we can see the drop in deviance when adding each variable one at a time. Adding InternetService, Contract and tenure_group significantly reduces the residual deviance. The other variables such as PaymentMethod and Dependents seem to improve the model less even though they all have low p-values.

Assessing the predictive ability of the model

```
testing$Churn <- as.character(testing$Churn)
testing$Churn[testing$Churn=="No"] <- "0"
testing$Churn[testing$Churn=="Yes"] <- "1"
fitted.results <- predict(LogModel,newdata=testing,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != testing$Churn)
print(paste('Logistic Regression Accuracy',1-misClasificError))
```

```
## [1] "Logistic Regression Accuracy 0.802182163187856"
```

De outra forma

```
logit = cbind(as.numeric(testing$Churn),
              as.numeric(fitted.results))

teste = ifelse(logit[,1]==logit[,2], "Sim", "No")

sum(teste=="Sim")/nrow(logit)
```

```
## [1] 0.8021822
```

```
sum(teste=="No")/nrow(logit)
```

```
## [1] 0.1978178
```

Odds Ratio

One of the interesting perfomance measurements in logistic regression is Odds Ratio.Basically, Odds retios is what the odds of an event is happening?

```
exp(cbind(OR=coef(LogModel), confint(LogModel)))
```

```
##                                            OR      2.5 %     97.5 %
## (Intercept)                         0.1274279 0.01865337  0.8659269
## genderMale                          1.0673389 0.91724431  1.2421686
## SeniorCitizenYes                    1.3011714 1.06991049  1.5819748
## PartnerYes                          1.0369881 0.86600892  1.2420731
## DependentsYes                       0.8390974 0.67972781  1.0343284
## PhoneServiceYes                     0.5946629 0.13057896  2.7051041
## MultipleLinesYes                    1.3467830 0.89206479  2.0342056
## InternetServiceFiber optic          2.4028656 0.37437764 15.4591443
## InternetServiceNo                   0.3678146 0.05597392  2.4173338
## OnlineSecurityYes                   0.7281515 0.48050132  1.1023346
## OnlineBackupYes                     0.8316049 0.55116730  1.2541828
## DeviceProtectionYes                 0.9452451 0.62611838  1.4266761
## TechSupportYes                      0.7119086 0.46825824  1.0809195
## StreamingTVYes                      1.3446642 0.62749717  2.8835251
## StreamingMoviesYes                  1.2738088 0.59510439  2.7279441
## ContractOne year                    0.5153769 0.40101856  0.6591668
## ContractTwo year                    0.1998341 0.12843605  0.3028124
## PaperlessBillingYes                 1.3833428 1.16295340  1.6466621
## PaymentMethodCredit card (automatic) 0.9993164 0.76458358  1.3058433
## PaymentMethodElectronic check       1.3628539 1.09181799  1.7044374
```

```
## PaymentMethodMailed check           1.0888210 0.83239487  1.4259291
## MonthlyCharges                      0.9992522 0.92801769  1.0759317
## tenure_group0-12 Month              6.5168248 4.36398962  9.8223601
## tenure_group12-24 Month             2.5536766 1.72222256  3.8161607
## tenure_group24-48 Month             1.7519374 1.22019396  2.5372415
## tenure_group48-60 Month             1.4254849 0.96422045  2.1148309
```

For each unit increase in Monthly Charge, there is a 1.01% decrease in the likelihood of a customer's churning.

Decision Tree

Árvores de decisão são métodos de aprendizado de máquinas supervisionado não-paramétricos, muito utilizados em tarefas de classificação e regressão. Vamos utilizar uma para prever

```
churn <- read.csv('Telco-Customer-Churn.csv')
churn <- churn[complete.cases(churn), ]
```

```
cols_recode1 <- c(10:15)
for(i in 1:ncol(churn[,cols_recode1])) {
        churn[,cols_recode1][,i] <- as.factor(mapvalues
                                              (churn[,cols_recode1][,i], from =c("No internet service"),to=c("No")))
}
```

```
churn$MultipleLines <- as.factor(mapvalues(churn$MultipleLines,
                                           from=c("No phone service"),
                                           to=c("No")))
```

```
group_tenure <- function(tenure){
    if (tenure >= 0 & tenure <= 12){
        return('0-12 Month')
    }else if(tenure > 12 & tenure <= 24){
        return('12-24 Month')
    }else if (tenure > 24 & tenure <= 48){
        return('24-48 Month')
    }else if (tenure > 48 & tenure <=60){
        return('48-60 Month')
    }else if (tenure > 60){
        return('> 60 Month')
    }
}
```

```
churn$tenure_group <- sapply(churn$tenure,group_tenure)
churn$tenure_group <- as.factor(churn$tenure_group)
```

```
churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,
                                      from=c("0","1"),
                                      to=c("No", "Yes")))
```

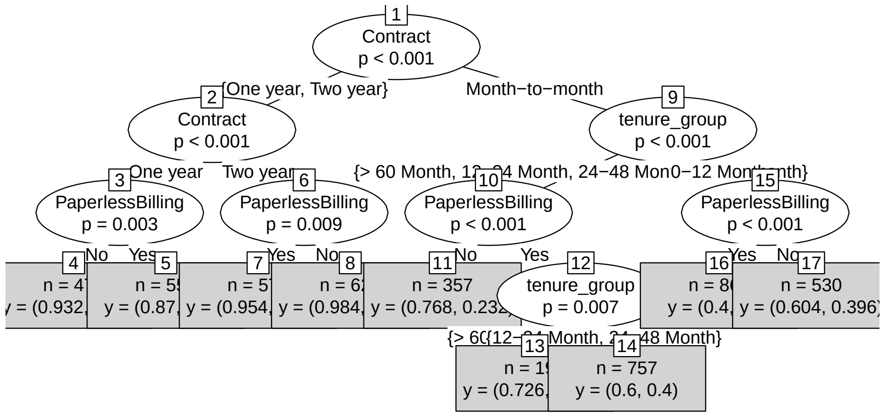
```
churn$customerID <- NULL
churn$tenure <- NULL
churn$TotalCharges <- NULL
```

```
intrain<- createDataPartition(churn$Churn,p=0.7,list=FALSE)
set.seed(2017)
training<- churn[intrain,]
testing<- churn[-intrain,]
```

For illustration purpose, we are going to use only three variables, they are "Contract", "tenure_group" and "PaperlessBilling".

```
tree <- ctree(Churn~Contract+tenure_group+PaperlessBilling, training)
```

```r
plot(tree, type='simple')
```



Out of three variables we use, Contract is the most important variable to predict customer churn or not churn.

If a customer in a one-year contract and not using PapelessBilling, then this customer is unlikely to churn.

On the other hand, if a customer is in a month-to-month contract, and in the tenure group of 0-12 months, and using PaperlessBilling, then this customer is more likely to churn.

```r
pred_tree <- predict(tree, testing)
print("Confusion Matrix for Decision Tree"); table(Predicted = pred_tree, Actual = testing$Churn)
```

```
## [1] "Confusion Matrix for Decision Tree"
```

```
##          Actual
## Predicted   No  Yes
##       No  1395  346
##       Yes  153  214
```

```r
p1 <- predict(tree, training)
tab1 <- table(Predicted = p1, Actual = training$Churn)
tab2 <- table(Predicted = pred_tree, Actual = testing$Churn)
```

```r
print(paste('Decision Tree Accuracy',sum(diag(tab2))/sum(tab2)))
```

```
## [1] "Decision Tree Accuracy 0.763282732447818"
```

Random Forest

Floresta Aleatória (random forest) é um algoritmo de aprendizagem supervisionada. Como você pode perceber pelo seu nome, ele cria uma floresta de um modo aleatório. A "floresta" que ele cria é uma combinação (ensemble) de árvores de decisão, na maioria dos casos treinados com o método de bagging. A idéia principal do método de bagging é que a combinação dos modelos de aprendizado aumenta o resultado geral.

Dizendo de modo simples: o algoritmo de florestas aleatórias cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estável. Abaixo fazemos um exemplo.

```r
set.seed(2017)
rfModel <- randomForest(Churn ~., data = training)
print(rfModel)
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
```

```
##
##         OOB estimate of  error rate: 20.92%
## Confusion matrix:
##       No Yes class.error
## No  3247 368   0.1017981
## Yes  662 647   0.5057296
```

Prediction is pretty good when predicting "No". Error rate is much higher when predicting "Yes".

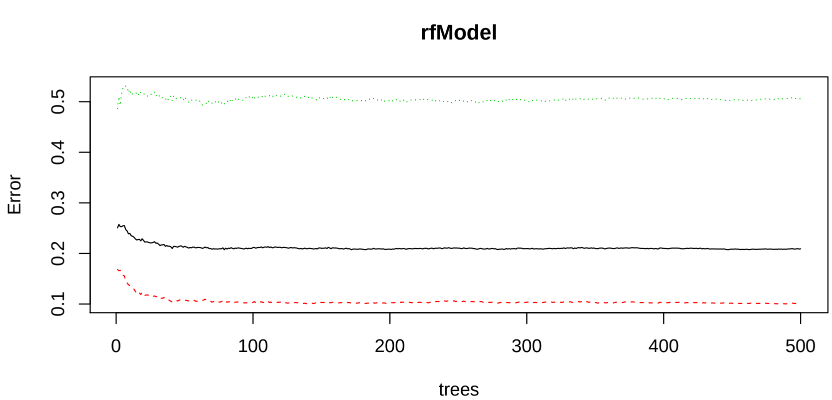
Prediction and confusion matrix

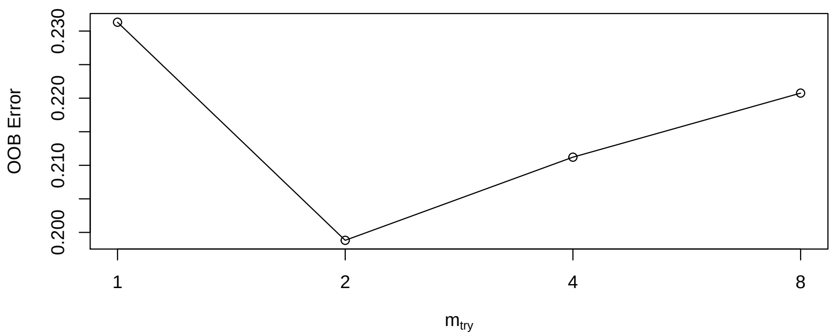
```
pred_rf <- predict(rfModel, testing)
caret::confusionMatrix(pred_rf, testing$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1385  285
##        Yes  163  275
##
##                Accuracy : 0.7875
##                  95% CI : (0.7694, 0.8048)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 9.284e-09
##
##                   Kappa : 0.4146
##  Mcnemar's Test P-Value : 1.086e-08
##
##             Sensitivity : 0.8947
##             Specificity : 0.4911
##          Pos Pred Value : 0.8293
##          Neg Pred Value : 0.6279
##              Prevalence : 0.7343
##          Detection Rate : 0.6570
##    Detection Prevalence : 0.7922
##       Balanced Accuracy : 0.6929
##
##        'Positive' Class : No
##
```

Error rate for Random Forest Model

```
plot(rfModel)
```






```r
t <- tuneRF(training[, -18], training[, 18], stepFactor = 0.5, plot = TRUE, ntreeTry = 200, trace = TRUE, improve = 0.05)
```

```
## mtry = 4  OOB error = 21.12%
## Searching left ...
## mtry = 8     OOB error = 22.08%
## -0.04519231 0.05
## Searching right ...
## mtry = 2     OOB error = 19.88%
## 0.05865385 0.05
## mtry = 1     OOB error = 23.13%
## -0.1634321 0.05
```

Fit the Random Forest Model again

```r
rfModel_new <- randomForest(Churn ~., data = training, ntree = 200, mtry = 2, importance = TRUE, proximity = TRUE)
print(rfModel_new)
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = training, ntree = 200,      mtry = 2, importance = TRUE, proximity = TRUE)
##               Type of random forest: classification
##                     Number of trees: 200
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 20.06%
## Confusion matrix:
##       No Yes class.error
## No  3300 315  0.08713693
## Yes  673 636  0.51413293
```

Make Predictions and Confusion Matrix again

```r
pred_rf_new <- predict(rfModel_new, testing)
caret::confusionMatrix(pred_rf_new, testing$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1410  306
##        Yes  138  254
##
##                Accuracy : 0.7894
##                  95% CI : (0.7713, 0.8066)
```

```
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 2.734e-09
##
##                   Kappa : 0.403
##  Mcnemar's Test P-Value : 2.273e-15
##
##             Sensitivity : 0.9109
##             Specificity : 0.4536
##          Pos Pred Value : 0.8217
##          Neg Pred Value : 0.6480
##              Prevalence : 0.7343
##          Detection Rate : 0.6689
##    Detection Prevalence : 0.8140
##       Balanced Accuracy : 0.6822
##
##        'Positive' Class : No
##
```

Random Forest Feature Importance

```
varImpPlot(rfModel_new, sort=T, n.var = 10, main = 'Top 10 Feature Importance')
```