Clube do Código 56

Usando regressão logística e Árvore de Decisão para fazer previsão de *Churn* em uma operadora de telecomunicações*

Vítor Wilher, MSc in Economics

24 de março, 2019

Abstract

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 simplesmente *churn*. Um setor no qual as taxas de cancelamento são particularmente úteis é o setor de telecomunicações. Vamos prever a rotatividade de clientes usando um conjunto de dados de telecomunicações disponível no site da IBM, com base em modelos de regressão logística e Árvore de Decisão.

Contents

*Esse exercício foi publicado originalmente em towardsdatascience.com, sendo de autoria de Susan Li.

1 Pacotes utilizados

```
## Carregar pacotes necessários
library(plyr)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(MASS)
library(caret)
library(randomForest)
library(party)
library(stargazer)
```

2 Coleta de Dados

Os dados foram transferidos por download do IBM Sample Data Sets. Cada linha representa um cliente, cada coluna contém os atributos desse cliente:

```
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
##  $ 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))

3 Tratamento dos dados

Os dados brutos contém 7043 linhas (clientes) e 21 colunas (recursos). A coluna *Churn* é o nosso alvo. Usamos todas as outras colunas como recursos do nosso modelo. Usamos `sapply` para verificar o número, se houver valores ausentes em cada coluna. Descobrimos que há 11 valores ausentes nas colunas *TotalCharges*. Então, vamos remover essas linhas com valores ausentes.

```
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), ]
```

Retirados os *missing values*, agora nós trocamos *No internet service* para *No* em seis colunas: *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *streamingTV* e *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")))
}
```

Agora nós trocamos *No phone service* para *No* na coluna *MultipleLines*.

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

A posse (*tenure*) mínima de uma linha nessa empresa é de um mês e a máxima de é de 72 meses. Nós podemos agrupar essa posse em cinco categorias: "0–12 Month", "12–24 Month", "24–48 Months", "48–60 Month" e "> 60 Month".

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

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

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

```
# Criar função
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')
    }
}
```

```
# Aplicar função sobre coluna tenure
churn$tenure_group <- sapply(churn$tenure,group_tenure)
churn$tenure_group <- as.factor(churn$tenure_group)
```

Agora, mudamos os valores na coluna "SeniorCitizen" de 0 e 1 para "No" ou "Yes".

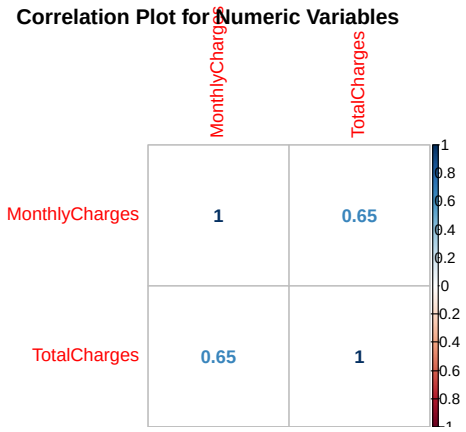
```
churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,
                                           from=c("0","1"),
                                           to=c("No", "Yes")))
```

Por fim, removemos as colunas que não iremos utilizar.

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

4 Análise Exploratória dos Dados

```
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")
```

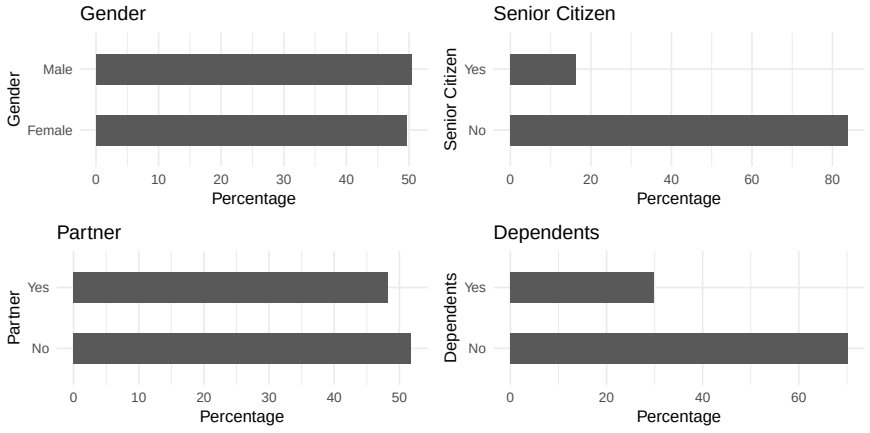


AS variáveis *Monthly Charges* e *Total Charges* são correlacionadas. Vamos utilizar apenas uma delas.

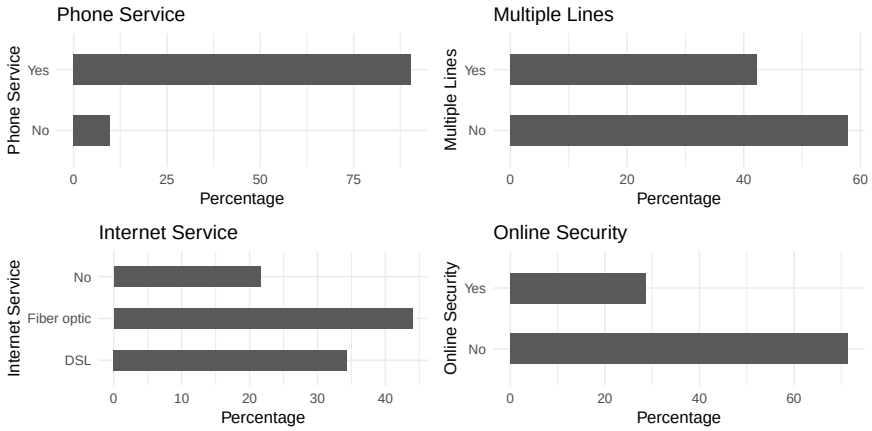
```
churn$TotalCharges <- NULL
```

4.1 Bar plots das variáveis categóricas

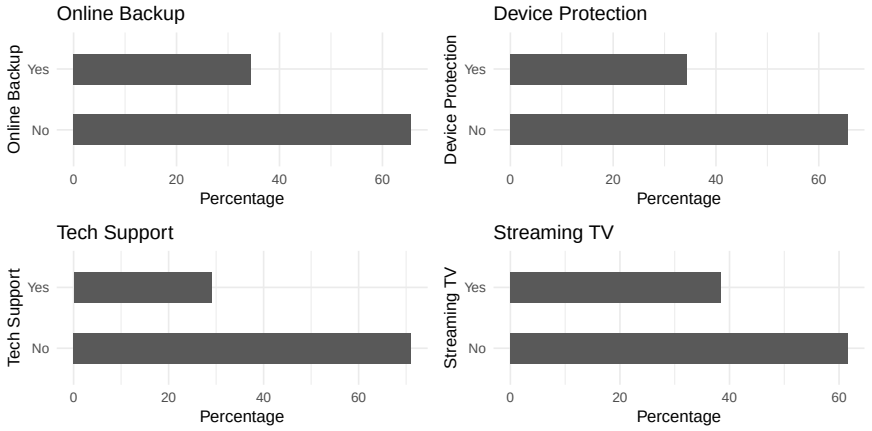
```
p1 <- ggplot(churn, aes(x=gender)) + ggtitle("Gender") + xlab("Gender") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p2 <- ggplot(churn, aes(x=SeniorCitizen)) + ggtitle("Senior Citizen") + xlab("Senior Citizen") +
  geom_bar(aes(y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



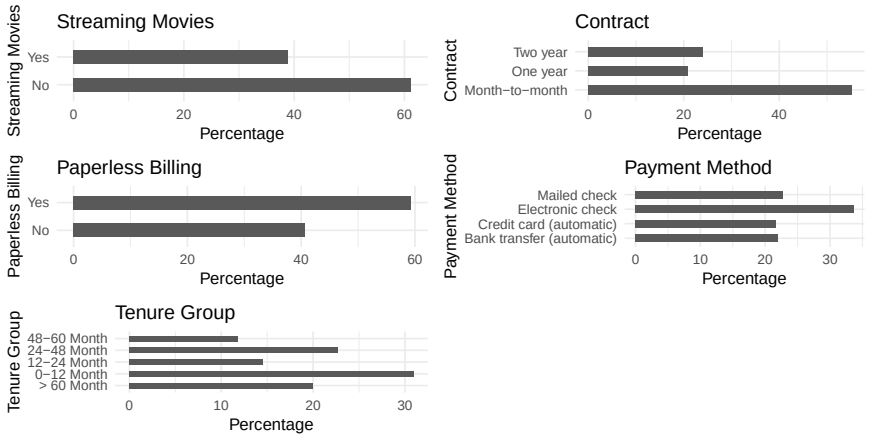
```
p5 <- ggplot(churn, aes(x=PhoneService)) + ggtitle("Phone Service") + xlab("Phone Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord_flip() + theme_minimal()
p6 <- ggplot(churn, aes(x=MultipleLines)) + ggtitle("Multiple Lines") + xlab("Multiple 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)
```



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

5 Regressão Logística

Criar os conjuntos de treino (*training*) e de teste (*test*).

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

Confirmamos se a divisão está correta.

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

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

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

Estimamos, então, o modelo logístico.

```
LogModel <- glm(Churn ~ ., family=binomial(link="logit"),data=training)
stargazer(LogModel, font.size = 'tiny', title='Regressão Logística',
          header=FALSE)
```

A tabela 1 ilustra o modelo. As três variáveis mais relevantes para explicar *Churn* são: Contract, Paperless Billing e tenure group. A seguir, analisando a tabela de desvio, podemos ver a queda no desvio ao adicionar cada variável uma de cada vez. Adicionar InternetService, Contract e tenure_group reduz significativamente o desvio residual. As outras variáveis, como PaymentMethod e Dependents, parecem melhorar menos o modelo, embora todos tenham p-valores baixos.

```
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     1.06      4922     5701.7  0.303643
## SeniorCitizen     1   114.81      4921     5586.9 < 2.2e-16 ***
## Partner           1   115.37      4920     5471.5 < 2.2e-16 ***
## Dependents        1    42.56      4919     5429.0 6.848e-11 ***
## PhoneService      1     0.27      4918     5428.7  0.603499
## MultipleLines     1     4.61      4917     5424.1  0.031725 *
## InternetService   2   432.39      4915     4991.7 < 2.2e-16 ***
## OnlineSecurity    1   168.82      4914     4822.9 < 2.2e-16 ***
## OnlineBackup      1    96.95      4913     4725.9 < 2.2e-16 ***
## DeviceProtection  1    34.65      4912     4691.3 3.937e-09 ***
## TechSupport       1    82.35      4911     4608.9 < 2.2e-16 ***
## StreamingTV       1     3.18      4910     4605.7  0.074326 .
## StreamingMovies   1     0.71      4909     4605.0  0.400239
## Contract          2   303.20      4907     4301.8 < 2.2e-16 ***
## PaperlessBilling  1    14.48      4906     4287.4  0.000142 ***
## PaymentMethod     3    32.46      4903     4254.9 4.178e-07 ***
## MonthlyCharges    1     0.26      4902     4254.6  0.611173
## tenure_group      4   131.94      4898     4122.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 1: Regressão Logística

	Dependent variable:
	Churn
genderMale	−0.047
	(0.077)
SeniorCitizenYes	0.255**
	(0.100)
PartnerYes	0.004
	(0.093)
DependentsYes	−0.193*
	(0.108)
PhoneServiceYes	−0.024
	(0.776)
MultipleLinesYes	0.335
	(0.212)
InternetServiceFiber optic	1.298
	(0.954)
InternetServiceNo	−1.390
	(0.965)
OnlineSecurityYes	−0.308
	(0.214)
OnlineBackupYes	−0.149
	(0.210)
DeviceProtectionYes	0.131
	(0.211)
TechSupportYes	−0.300
	(0.215)
StreamingTVYes	0.471
	(0.390)
StreamingMoviesYes	0.503
	(0.392)
ContractOne year	−0.887***
	(0.128)
ContractTwo year	−1.632***
	(0.215)
PaperlessBillingYes	0.343***
	(0.089)
PaymentMethodCredit card (automatic)	−0.033
	(0.135)
PaymentMethodElectronic check	0.348***
	(0.113)
PaymentMethodMailed check	0.071
	(0.136)
MonthlyCharges	−0.020
	(0.038)
tenure_group0-12 Month	1.603***
	(0.203)
tenure_group12-24 Month	0.797***
	(0.200)
tenure_group24-48 Month	0.464**
	(0.183)
tenure_group48-60 Month	0.212
	(0.198)
Constant	−1.352
	(0.986)
Observations	4,924
Log Likelihood	−2,061.348
Akaike Inf. Crit.	4,174.696
Note:	*p<0.1; **p<0.05; ***p<0.01

5.1 Verificando a acurácia do modelo

```
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 0.807400379506641"
```

Ou

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

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

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

```
## [1] 0.8074004
```

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

```
## [1] 0.1925996
```

5.2 Odds Ratio

Uma das medidas de desempenho interessantes na regressão logística é a *Odds Ratio*. Basicamente, *odds ratios* mede a chance de um evento acontecer.

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

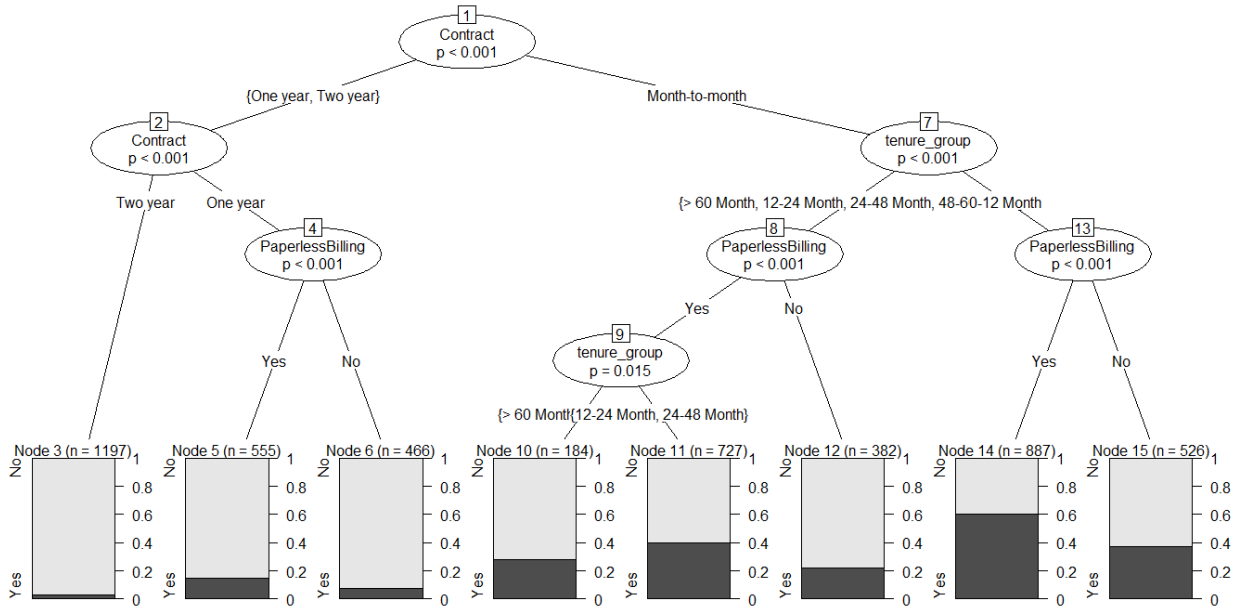
```
##                                         OR      2.5 %     97.5 %
## (Intercept)                      0.2586441 0.03737586  1.7833053
## genderMale                       0.9540391 0.81992423  1.1100818
## SeniorCitizenYes                 1.2906366 1.06058618  1.5702342
## PartnerYes                       1.0041697 0.83621026  1.2061523
## DependentsYes                    0.8243165 0.66617553  1.0184503
## PhoneServiceYes                  0.9761702 0.21339012  4.4670227
## MultipleLinesYes                 1.3984298 0.92338444  2.1190571
## InternetServiceFiber optic       3.6612206 0.56538994 23.7894029
## InternetServiceNo                0.2489861 0.03749698  1.6511065
## OnlineSecurityYes                0.7350131 0.48258405  1.1182200
## OnlineBackupYes                  0.8612526 0.57099332  1.2983662
## DeviceProtectionYes              1.1404905 0.75362426  1.7262372
## TechSupportYes                   0.7406619 0.48603590  1.1274520
## StreamingTVYes                   1.6017804 0.74589569  3.4440282
## StreamingMoviesYes               1.6532587 0.76686861  3.5691593
## ContractOne year                 0.4120808 0.31966551  0.5280499
## ContractTwo year                 0.1955223 0.12670344  0.2943287
## PaperlessBillingYes              1.4090157 1.18358171  1.6785320
## PaymentMethodCredit card (automatic) 0.9675713 0.74186156  1.2614868
## PaymentMethodElectronic check    1.4164223 1.13699061  1.7677353
## PaymentMethodMailed check        1.0733129 0.82180252  1.4032228
## MonthlyCharges                   0.9801122 0.90979090  1.0557887
## tenure_group0-12 Month           4.9693898 3.34732845  7.4341532
## tenure_group12-24 Month          2.2189526 1.50297585  3.2958867
## tenure_group24-48 Month          1.5901481 1.11424385  2.2845891
## tenure_group48-60 Month          1.2359940 0.83826217  1.8252005
```

6 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 nosso *Churn*. Para ilustrarmos, vamos utilizar apenas três variáveis: "Contract", "tenure_group" e "PaperlessBilling".

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

```
plot(tree)
```



Das três variáveis que usamos, *Contract* é a variável mais importante para prever a rotatividade de clientes ou não. Se um cliente em um contrato de um ano ou dois anos, não importa se ele (ela) tem ou não a PapelessBilling, ele (ela) tem menos probabilidade de sair. Por outro lado, se um cliente estiver em um contrato mensal, e no grupo de posse de 0 a 12 meses, e usando o PaperlessBilling, esse cliente terá maior probabilidade de sair.[1]

6.1 Avaliando a acurácia da Árvore de Decisão

```
pred_tree <- predict(tree, testing)
tab <- table(Predicted = pred_tree, Actual = testing$Churn)
print(paste('Decision Tree Accuracy',sum(diag(tab))/sum(tab)))
```

```
## [1] "Decision Tree Accuracy 0.776091081593928"
```

[1]Na hora de rodar o código, coloque `plot(tree)` e dê um zoom para ver a árvore completa. A interpretação fica mais fácil.