Previsão de Churn com Regressão Logistica

Análise exploratória e construção de modelo usando o R

Céasar Lemos, B.Sc in Matemática e Cientsita de Dados

02/02/2020

Abstract

A regressão logistica é uma variação da regressão linear que busca explicar a probabilidade de algo acontecer dado determinadas variáveis. A variável dependente assume valores entre 0 e 1 onde, quanto mais próximo de 1, maior a probabilidade do evento acontecer. A regressão logística é muito utilizada no campo da ciência de dados para predizer eventos como churn, fraudes em transações financeiras, filtros de spams, evasão de alunos, dentre outras situações. Nesta publicação, o foco é em prever churn e vamos utilizar um conjunto de dados de telecomunicaçõs disponível no Kaggle.

Pacotes

```r
## Pacotes usados
library(tidyverse)
library(plyr)
library(gridExtra)
library(GGally)
library(caret)
library(MASS)
library(forecast)
library(ROCR)
library(caret)
library(cowplot)
```

Coleta de dados

Os dados foram coletados da web no seguinte endereço: https://www.kaggle.com/blastchar/telco-customer-churn.

```r
## Carregando os dados
dtChurn <- read.csv(choose.files())
str(dtChurn)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 553
##  $ 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))

Tratamento dos dados

Os dados contém 7043 linhas e 21 colunas. O nosso alvo é a coluna *Churn*, que contém as saídas que queremos prever (o cliente cancelou ou não o serviço). Usamos todas as outras colunas como variáveis explicativas do nosso modelo. Antes, precisamos verificar se há valores ausentes nas colunas e tratar esta anomalia.

```
## Verificando quantidade de miss value
sapply(dtChurn, 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
```

Como mostrado acima, existem 11 valores ausentes na coluna *TotalCharges*. Vamos remover estes dados, vitso que representa apenas 0,16% da base total.

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

As colunas *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *StreamingTV* e *StreamingMovies* possui uma categoria chamada *No internet service*. Vamos trocar para *No* com o objetivo de categorizar de forma mais correta.

```
range_cols <- c(10:15)

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

Vamos trocar a categoria *No phone service* da coluna *MultipleLines* para *No*.

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

A coluna *tenure* possui muitos elementos para lidar em uma regressão logística. Vamos verificar o máximo de meses dos clientes e criar intervalos de tempo.

```
max(dtChurn$tenure)
```

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

O máximo de tempo de um cliente permanece no plano é de 72 meses (ou 6 anos). Vamos criar uma função para fazer essa categorização.

```
tgroup <- 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')
  }
}
```

```
## Aplicando a função sobre a coluna tenure
dtChurn$tenure_group <- sapply(dtChurn$tenure,tgroup)
dtChurn$tenure_group <- as.factor(dtChurn$tenure_group)
```

Mudaremos os valores da coluna *SeniorCitizen* de "0 ou 1" para "Yes ou No" para padronizar com as demais colunas.

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

Agora vamos remover as colunas que são desnecessárias.

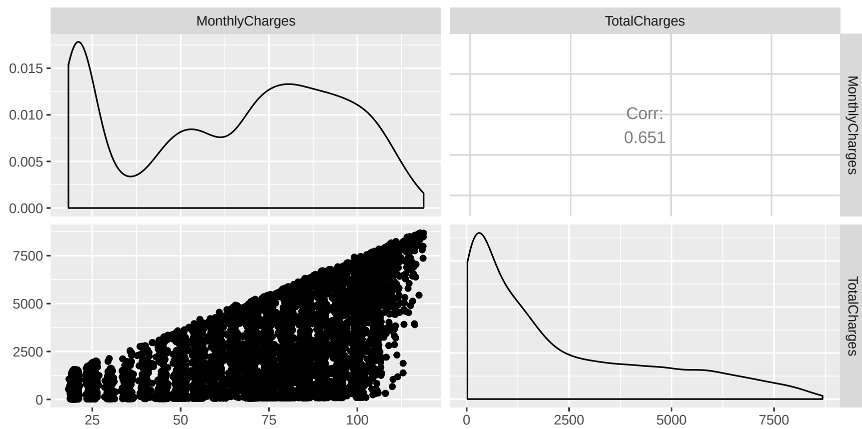
```
dtChurn$customerID <- NULL
dtChurn$tenure <- NULL
```

Análise Exploratória dos dados

Autocorrelação

As colunas *MonthlyCharges* e *TotalCharges* parecem que possuem autocorrelação. Vamos verificar e, se for o caso, usar apenas uma delas.

```
ggpairs(dtChurn[,17:18])
```

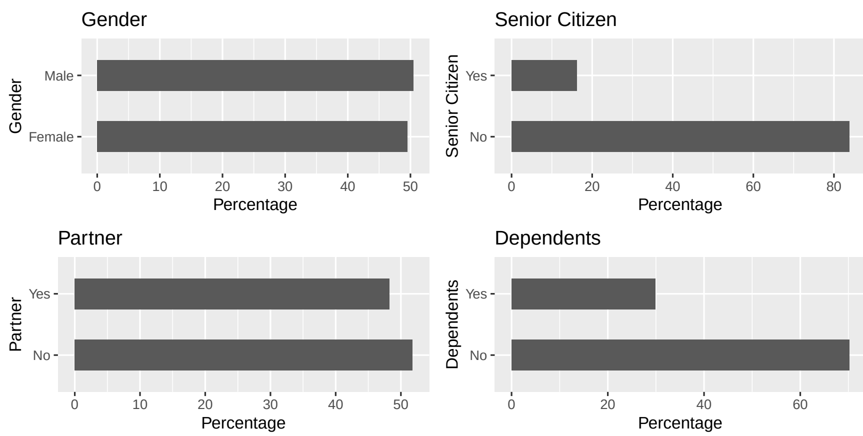


De fato existe uma correlação de 0,651 entre as duas variáveis. Vamos ficar com a MonthlyCharges.

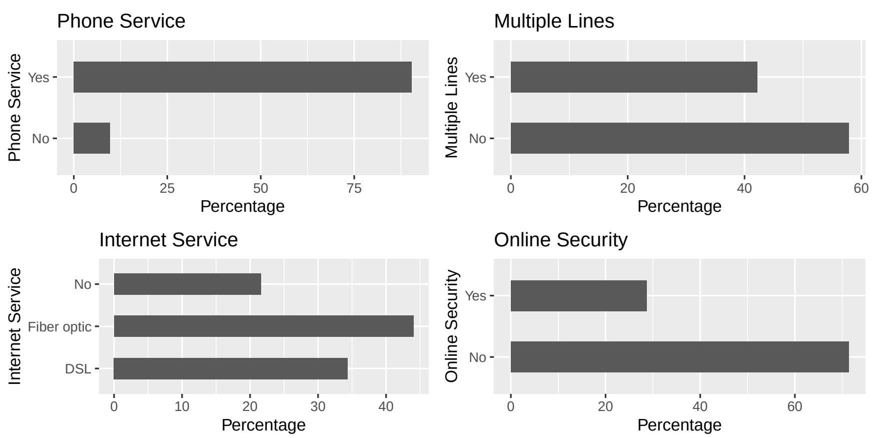
```
dtChurn$TotalCharges <- NULL
```

Plot das variáveis

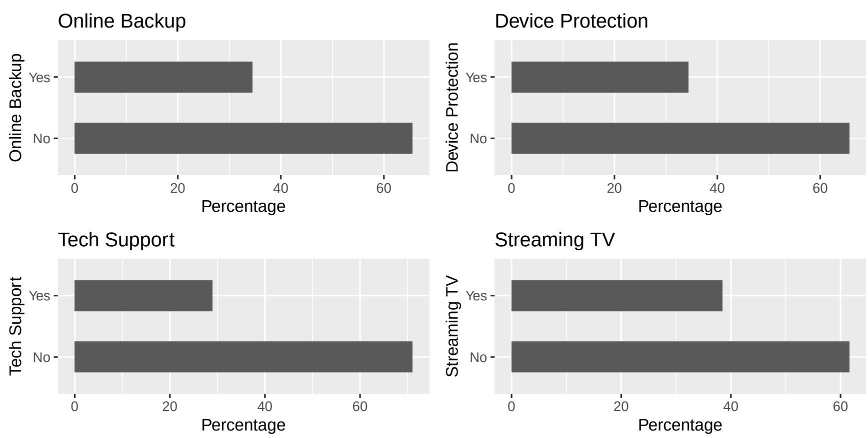
```
p1 <- ggplot(dtChurn, aes(x=gender)) + ggtitle("Gender") + xlab("Gender") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

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

p3 <- ggplot(dtChurn, aes(x=Partner)) + ggtitle("Partner") + xlab("Partner") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p4 <- ggplot(dtChurn, aes(x=Dependents)) + ggtitle("Dependents") +
  xlab("Dependents") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

grid.arrange(p1, p2, p3, p4, ncol=2)
```



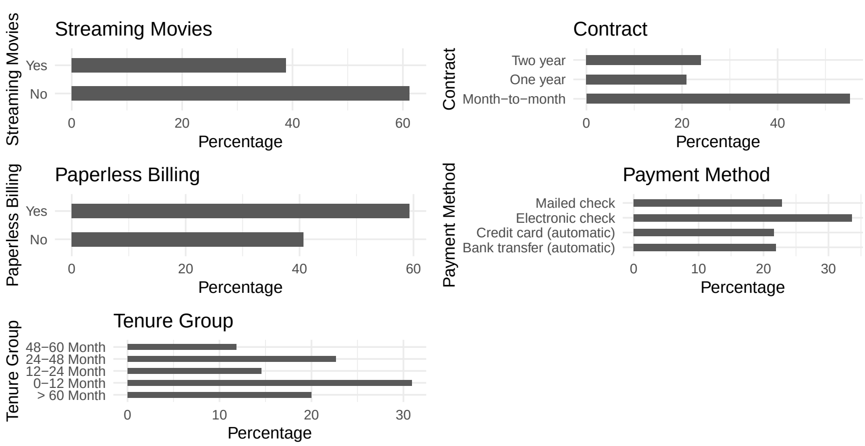
```
p5 <- ggplot(dtChurn, aes(x=PhoneService)) + ggtitle("Phone Service") +
  xlab("Phone Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p6 <- ggplot(dtChurn, aes(x=MultipleLines)) + ggtitle("Multiple Lines") +
  xlab("Multiple Lines") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p7 <- ggplot(dtChurn, aes(x=InternetService)) + ggtitle("Internet Service") +
  xlab("Internet Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p8 <- ggplot(dtChurn, aes(x=OnlineSecurity)) + ggtitle("Online Security") +
  xlab("Online Security") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

grid.arrange(p5, p6, p7, p8, ncol=2)
```



```r
p9 <- ggplot(dtChurn, aes(x=OnlineBackup)) + ggtitle("Online Backup") +
  xlab("Online Backup") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p10 <- ggplot(dtChurn, aes(x=DeviceProtection)) + ggtitle("Device Protection") +
  xlab("Device Protection") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p11 <- ggplot(dtChurn, aes(x=TechSupport)) + ggtitle("Tech Support") +
  xlab("Tech Support") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

p12 <- ggplot(dtChurn, aes(x=StreamingTV)) + ggtitle("Streaming TV") +
  xlab("Streaming TV") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) +
  ylab("Percentage") + coord_flip()

grid.arrange(p9, p10, p11, p12, ncol=2)
```



```r
p13 <- ggplot(dtChurn, 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(dtChurn, 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(dtChurn, 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(dtChurn, 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(dtChurn, 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)
```



Vamos manter todas as variáveis e usar a função *stepAIC()* para selecionar apenas as variáveis com significância.

Regressão Logística

Criando o modelo

```
## Criando o conjunto de treino e teste
intrain <- createDataPartition(dtChurn$Churn,p=0.7,list=FALSE)
set.seed(1)
training <- dtChurn[intrain,]
testing <- dtChurn[-intrain,]
```

```
## Confirmando se o particionamento está correto
dim(training); dim(testing)
```

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

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

Criando o modelo logístico

```
ChurnModel <- stepAIC(glm(Churn ~ .,
                          family=binomial(link="logit"),data=training),direction = "both")
```

```
## Start:  AIC=4147.75
## Churn ~ gender + SeniorCitizen + Partner + Dependents + PhoneService +
##     MultipleLines + InternetService + OnlineSecurity + OnlineBackup +
##     DeviceProtection + TechSupport + StreamingTV + StreamingMovies +
##     Contract + PaperlessBilling + PaymentMethod + MonthlyCharges +
##     tenure_group
##
##                    Df Deviance    AIC
## - OnlineBackup      1   4095.7 4145.7
## - DeviceProtection  1   4095.8 4145.8
## - InternetService   2   4097.8 4145.8
## - MonthlyCharges    1   4095.8 4145.8
## - PhoneService      1   4096.0 4146.0
## - Dependents        1   4096.5 4146.5
## - StreamingMovies   1   4096.6 4146.6
## - StreamingTV       1   4096.8 4146.8
## - Partner           1   4097.2 4147.2
## - OnlineSecurity    1   4097.4 4147.4
## <none>                  4095.7 4147.7
## - TechSupport       1   4097.9 4147.9
## - gender            1   4098.0 4148.0
## - MultipleLines     1   4099.5 4149.5
## - SeniorCitizen     1   4100.9 4150.9
```

```
## - PaymentMethod     3   4112.7 4158.7
## - PaperlessBilling  1   4109.9 4159.9
## - Contract          2   4168.5 4216.5
## - tenure_group      4   4269.1 4313.1
##
## Step:  AIC=4145.75
## Churn ~ gender + SeniorCitizen + Partner + Dependents + PhoneService +
##     MultipleLines + InternetService + OnlineSecurity + DeviceProtection +
##     TechSupport + StreamingTV + StreamingMovies + Contract +
##     PaperlessBilling + PaymentMethod + MonthlyCharges + tenure_group
##
##                    Df Deviance    AIC
## - DeviceProtection  1   4095.8 4143.8
## - MonthlyCharges    1   4096.3 4144.3
## - Dependents        1   4096.5 4144.5
## - PhoneService      1   4096.6 4144.6
## - Partner           1   4097.2 4145.2
## <none>                  4095.7 4145.7
## - gender            1   4098.0 4146.0
## - StreamingMovies   1   4099.3 4147.3
## + OnlineBackup      1   4095.7 4147.7
## - OnlineSecurity    1   4100.0 4148.0
## - StreamingTV       1   4100.1 4148.1
## - SeniorCitizen     1   4100.9 4148.9
## - TechSupport       1   4101.4 4149.4
## - InternetService   2   4104.6 4150.6
## - MultipleLines     1   4106.3 4154.3
## - PaymentMethod     3   4112.7 4156.7
## - PaperlessBilling  1   4109.9 4157.9
## - Contract          2   4168.6 4214.6
## - tenure_group      4   4270.1 4312.1
##
## Step:  AIC=4143.77
## Churn ~ gender + SeniorCitizen + Partner + Dependents + PhoneService +
##     MultipleLines + InternetService + OnlineSecurity + TechSupport +
##     StreamingTV + StreamingMovies + Contract + PaperlessBilling +
##     PaymentMethod + MonthlyCharges + tenure_group
##
##                    Df Deviance    AIC
## - Dependents        1   4096.5 4142.5
## - PhoneService      1   4096.9 4142.9
## - MonthlyCharges    1   4097.0 4143.0
## - Partner           1   4097.3 4143.3
## <none>                  4095.8 4143.8
## - gender            1   4098.0 4144.0
## + DeviceProtection  1   4095.7 4145.7
## + OnlineBackup      1   4095.8 4145.8
## - OnlineSecurity    1   4100.8 4146.8
```

```
## - SeniorCitizen     1   4101.0 4147.0
## - StreamingMovies   1   4101.2 4147.2
## - TechSupport       1   4102.1 4148.1
## - StreamingTV       1   4102.5 4148.5
## - PaymentMethod     3   4112.8 4154.8
## - InternetService   2   4111.5 4155.5
## - MultipleLines     1   4109.7 4155.7
## - PaperlessBilling  1   4110.0 4156.0
## - Contract          2   4169.1 4213.1
## - tenure_group      4   4270.3 4310.3
##
## Step:  AIC=4142.52
## Churn ~ gender + SeniorCitizen + Partner + PhoneService + MultipleLines +
##     InternetService + OnlineSecurity + TechSupport + StreamingTV +
##     StreamingMovies + Contract + PaperlessBilling + PaymentMethod +
##     MonthlyCharges + tenure_group
##
##                    Df Deviance    AIC
## - PhoneService      1   4097.6 4141.6
## - MonthlyCharges    1   4097.8 4141.8
## <none>                  4096.5 4142.5
## - gender            1   4098.7 4142.7
## - Partner           1   4099.5 4143.5
## + Dependents        1   4095.8 4143.8
## + DeviceProtection  1   4096.5 4144.5
## + OnlineBackup      1   4096.5 4144.5
## - OnlineSecurity    1   4101.5 4145.5
## - StreamingMovies   1   4102.1 4146.1
## - SeniorCitizen     1   4102.8 4146.8
## - TechSupport       1   4102.8 4146.8
## - StreamingTV       1   4103.3 4147.3
## - PaymentMethod     3   4113.9 4153.9
## - InternetService   2   4112.5 4154.5
## - MultipleLines     1   4110.6 4154.6
## - PaperlessBilling  1   4110.7 4154.7
## - Contract          2   4171.1 4213.1
## - tenure_group      4   4271.0 4309.0
##
## Step:  AIC=4141.59
## Churn ~ gender + SeniorCitizen + Partner + MultipleLines + InternetService +
##     OnlineSecurity + TechSupport + StreamingTV + StreamingMovies +
##     Contract + PaperlessBilling + PaymentMethod + MonthlyCharges +
##     tenure_group
##
##                    Df Deviance    AIC
## <none>                  4097.6 4141.6
## - gender            1   4099.8 4141.8
## - Partner           1   4100.5 4142.5
```

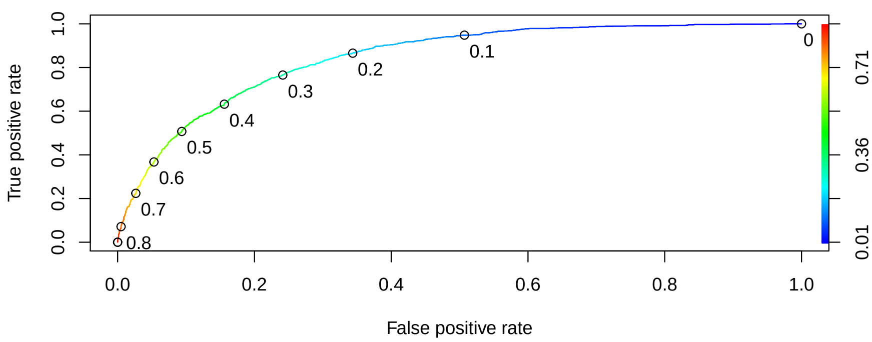
```
## + PhoneService      1   4096.5 4142.5
## + Dependents        1   4096.9 4142.9
## + OnlineBackup      1   4097.1 4143.1
## + DeviceProtection  1   4097.3 4143.3
## - OnlineSecurity    1   4101.5 4143.5
## - TechSupport       1   4102.8 4144.8
## - SeniorCitizen     1   4104.1 4146.1
## - PaymentMethod     3   4115.0 4153.0
## - MonthlyCharges    1   4111.6 4153.6
## - PaperlessBilling  1   4111.8 4153.8
## - StreamingMovies   1   4116.3 4158.3
## - MultipleLines     1   4117.5 4159.5
## - StreamingTV       1   4118.9 4160.9
## - InternetService   2   4159.9 4199.9
## - Contract          2   4171.5 4211.5
## - tenure_group      4   4274.8 4310.8
```

```r
anova(ChurnModel, 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     2.93      4922     5699.8 0.0866995 .
## SeniorCitizen     1   107.74      4921     5592.1 < 2.2e-16 ***
## Partner           1   126.97      4920     5465.1 < 2.2e-16 ***
## MultipleLines     1    14.22      4919     5450.9 0.0001629 ***
## InternetService   2   487.63      4917     4963.3 < 2.2e-16 ***
## OnlineSecurity    1   160.78      4916     4802.5 < 2.2e-16 ***
## TechSupport       1   126.83      4915     4675.7 < 2.2e-16 ***
## StreamingTV       1     0.05      4914     4675.6 0.8226481
## StreamingMovies   1     1.05      4913     4674.5 0.3049375
## Contract          2   328.11      4911     4346.4 < 2.2e-16 ***
## PaperlessBilling  1    12.19      4910     4334.3 0.0004814 ***
## PaymentMethod     3    39.87      4907     4294.4 1.136e-08 ***
## MonthlyCharges    1    19.56      4906     4274.8 9.738e-06 ***
## tenure_group      4   177.24      4902     4097.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plotando a curva ROC do modelo de Treino

```
predictTrain <- predict(ChurnModel, type = "response")
ROCRpred <- prediction(predictTrain, training$Churn)
ROCRperf <- performance(ROCRpred, "tpr","fpr")
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```



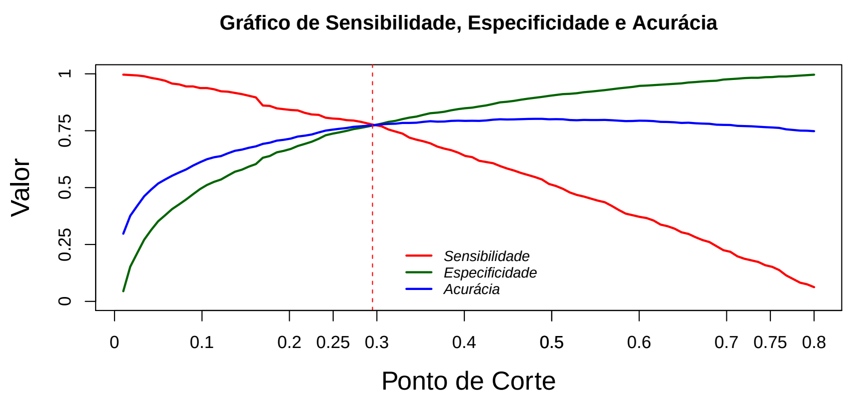
A taxa de *True Positives* do modelo é mostrada no eixo y, enquanto a taxa de *False Positive* é dada no eixo x. A linha mostra como essas duas medidas variam com diferentes valores.

A curva ROC sempre começa no ponto (0, 0), ou seja, o ponto de corte (Threshold) com valor >=1. Isso significa que nesse valor de Threshold não capturaremos nenhum caso para variavel dependente igual a "Yes",no nosso caso, mas rotularemos corretamente todos os casos que variável dependente for "No".

Mas, como escolher o melhor valor de corte? Este é um trade-off que depende do negócio. As vezes é necessário focar na otimização da sensibilidade do modelo (como em uma identificação de possível transação fraudulenta). No nosso caso, vamos verificar qual o ponto ótimo entre a sensibilidade, especificidade e acurácia.

```
## Criando variável com o resultado do modelo logístico
result <- predict(ChurnModel,newdata=testing,type='response')

## Criando variável com os valores reais de Churn para função
actual_churn <- factor(testing$Churn)

## Função para performance do modelo
perform_fn <- function(cutoff)
{
  predicted_churn <- factor(ifelse(result >= cutoff, "Yes", "No"))
  conf <- confusionMatrix(predicted_churn, actual_churn, positive = "Yes")
  accuray <- conf$overall[1]
  sensitivity <- conf$byClass[1]
  specificity <- conf$byClass[2]
```

```r
  out <- t(as.matrix(c(sensitivity, specificity, accuray)))
  colnames(out) <- c("sensitivity", "specificity", "accuracy")
  return(out)
}

## Criando gráfico de sensibilidade, especificidade e acurácia
set.seed(1)
s = seq(0.01,0.80,length=100)
OUT = matrix(0,100,3)

for(i in 1:100)
{
  OUT[i,] = perform_fn(s[i])
}

plot(s, OUT[,1],xlab="Ponto de Corte",ylab="Valor",cex.lab=1.5,cex.axis=1.5,ylim=c(0,1),
     type="l",lwd=2,axes=FALSE,col=2,
     main = "Gráfico de Sensibilidade, Especificidade e Acurácia")
axis(1,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)
axis(2,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)
lines(s,OUT[,2],col="darkgreen",lwd=2)
lines(s,OUT[,3],col=4,lwd=2)
box()
legend("bottom",col=c(2,"darkgreen",4,"darkred"),text.font =3,inset = 0.02,
       box.lty=0,cex = 0.8,
       lwd=c(2,2,2,2),c("Sensibilidade","Especificidade","Acurácia"))
abline(v = 0.295, col="red", lwd=1, lty=2)
axis(1, at = seq(0.1, 1, by = 0.1))
```



Como é mostrado no gráfico acima, o ponto ótimo de corte ocorre em, aproximadamente, 0.295. Com este ponto, maximizamos os três indicadores.

```r
previsao <- factor(ifelse(result >= 0.295, "Yes", "No"))
confusionMatrix(previsao, actual_churn, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1195  124
##        Yes  353  436
##
##                Accuracy : 0.7737
##                  95% CI : (0.7552, 0.7914)
##     No Information Rate : 0.7343
##     P-Value [Acc > NIR] : 1.786e-05
##
##                   Kappa : 0.487
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7786
##             Specificity : 0.7720
##          Pos Pred Value : 0.5526
##          Neg Pred Value : 0.9060
##              Prevalence : 0.2657
##          Detection Rate : 0.2068
##    Detection Prevalence : 0.3743
##       Balanced Accuracy : 0.7753
##
##        'Positive' Class : Yes
##
```

Com isso temos um modelo de regressão logística bem ajustado, conforme as informações de acurácia, sensibilidade e especificidade demonstrados na tabela acima.