Técnicas de Classificação

Adriano

15/12/2018

Análise dos Dados do Naufrágio do Titanic

Esse estudo tem por objetivo testar algumas técnicas de classificação supervionada e selecionar aquela que tiver o melhor desempenho segundo uma métrica escolhida, utilizando a base de dados do naufrágio do Titanic.

1. Definindo o problema

Segundo a wikipédia, o Titanic era um navio que partiu em sua primeira e única viagem com 1316 passageiros a bordo: 325 na primeira classe, 285 na segunda e 706 na terceira. Deles, 922 embarcaram em Southampton, 274 em Cherbourg-Octeville na França e 120 em Queenstown na Irlanda. A lista da Primeira Classe do Titanic era uma lista de pessoas ricas e proeminentes da alta classe em 1912. Os passageiros da segunda classe eram turistas à lazer, acadêmicos, membros do clero e famílias inglesas e americanas de classe média e os da terceira classe partiram esperando começar vida nova nos Estados Unidos e Canadá.

Na noite de 14 de abril de 1912 por volta de 23:40h, enquanto o Titanic navegava a cerca de 640 quilômetros ao Sul dos Grandes Bancos da Terra Nova, o navio atingiu um iceberg e começou a afundar. O estudo em questão tentará responder às seguintes questões:

```
O fato de uma pessoa ter sobrevivido ao desastre está relacionado, de alguma forma,
com sua situação sócio-econômica?
Caso esteja, poderíamos prever se uma pessoa sobreviveria,
a partir dos dados disponíveis?
```

A métrica utilizada como balizador para aferir a qualidade do modelo será a **sensibilidade** (ou **RECALL**), pois teremos como objetivo minimizar os falsos negativos, dado que o modelo tentará determinar quem sobrevive e quem não sobrevive, é melhor apontar alguém como possível sobrevivente de forma incorreta (FALSO POSITIVO) do que apontar um sobrevivente como não sobrevivente (FALSO NEGATIVO).

2. Conhecendo os dados

A base de dados sobre o desastre está disponível no **Kaggle**. Ela está separada entre dados de treinamento e dados de teste. A descrição das colunas encontra-se abaixo.

Descrição dos dados:

Variável	Descrição
PassengerId	Identificador do Passageiro
Survived	Variável de indicadora de sobrevivência (0 = Não Sobreviveu, 1 = Sobreviveu)
Pclass	Classe do passageiro
Name	Nome do passageiro
Sex	Sexo do passageiro
Age	Idade do passageiro
SibSp	Número de irmãos/cônjuge no navio

Variável	Descrição
Parch	Número de pais e filhos no navio
Ticket	Número da passagem
Fare	Preço da passagem
Cabin	Código da cabine
Embarked	Porto de embarque

3. Preparando os dados

Carregando as bibliotecas necessárias. Será necessário carregar a biblioteca **tidyverse** que possui as ferramentas necessária para a preparação do dados, mas a **titanic** que possui o *data set* e a **ModelMetrics** que possui métricas para avaliação de modelos.

Carregando o *data set* necessário para trabalhar. Serão dois *data set*, um para treino do modelo e outro para prova do modelo.

```
## [1] "DataSet de Treino"
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
## [1] "DataSet de Prova"
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Trabalhando no *data set* de treino, modificando o tipo de dado das colunas **Sex** e **Embarked**, passando elas para do tipo *factor*. Retirando as colunas **PassengerId**, **Ticket** e **Cabin** pois não serão utilizadas pelo modelo, pois não explicariam a pergunta do problema.

```
## 'data.frame':    891 obs. of  9 variables:
##  $ Survived : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass   : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name     : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "
##  $ Age      : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp    : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch    : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare     : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ fSex     : Factor w/ 2 levels "male","female": 1 2 2 2 1 1 1 1 2 2 ...
##  $ fEmbarked: Factor w/ 3 levels "S","C","Q": 1 2 1 1 1 3 1 1 1 2 ...
```

Verificando a existência de dados **N/A**. Como pode-se notar a coluna **Age** possui muitos valores **N/A** e enquanto que a coluna **fEmbarked** possui apenas 2.

```
## [1] "Total de Valores N/A por coluna"
```

```
##  Survived    Pclass      Name       Age     SibSp     Parch      Fare
##         0         0         0       177         0         0         0
##      fSex fEmbarked
##         0         2
```

Como a coluna **fEmbarked** possui apenas 2 valores **N/A**, optou-se por excluir essas linhas.

```
##     Survived Pclass                                      Name Age SibSp
## 62         1      1                    Icard, Miss. Amelie  38     0
## 830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)  62     0
##     Parch Fare   fSex fEmbarked
## 62      0   80 female      <NA>
## 830     0   80 female      <NA>
```

Devido a grande quantidade de valores **N/A** na coluna **Age**, optou-se por excluir a coluna inteira.

```
## 'data.frame':    889 obs. of  8 variables:
##  $ Survived : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass   : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name     : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "
##  $ SibSp    : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch    : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare     : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ fSex     : Factor w/ 2 levels "male","female": 1 2 2 2 1 1 1 1 2 2 ...
##  $ fEmbarked: Factor w/ 3 levels "S","C","Q": 1 2 1 1 1 3 1 1 1 2 ...
```

Ajustando os nomes das colunas do *data set* de treino.

```
## 'data.frame':    889 obs. of  8 variables:
##  $ Sobreviveu: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Classe    : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Nome      : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "
##  $ HFamilia  : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ VFamilia  : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Preco     : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Sexo      : Factor w/ 2 levels "male","female": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Embarque  : Factor w/ 3 levels "S","C","Q": 1 2 1 1 1 3 1 1 1 2 ...
```

Repetindo o mesmo processo para o *data set* de prova.

```
## [1] "Total de Valores N/A por coluna"
```

```
##    Pclass      Name       Age     SibSp     Parch      Fare      fSex
##         0         0        86         0         0         1         0
## fEmbarked
##         0
```

```
## [1] "DataSet de Prova"
```

```
## 'data.frame':    417 obs. of  7 variables:
##  $ Classe  : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Nome    : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis" "
##  $ HFamilia: int  0 1 0 0 1 0 0 1 0 2 ...
##  $ VFamilia: int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Preco   : num  7.83 7 9.69 8.66 12.29 ...
##  $ Sexo    : Factor w/ 2 levels "male","female": 1 2 1 1 2 1 2 1 2 1 ...
##  $ Embarque: Factor w/ 3 levels "S","C","Q": 3 1 3 1 1 1 3 1 2 1 ...
```

Dividindo o *data set* de treino em dois, o primeiro para treinar efetivamento o modelo e o segundo para validá-lo na proporção de 80% - 20%.

80% do *data set* para treinar o modelo e 20% do *data set* para validar o modelo.

```
## [1] "DataSet Validação"
```

```
## 'data.frame':    178 obs. of  8 variables:
##  $ Sobreviveu: int  1 0 0 0 1 1 0 0 0 1 ...
##  $ Classe    : int  1 1 3 3 3 1 3 3 3 2 ...
##  $ Nome      : chr  "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "McCarthy, Mr. Timothy J" "
##  $ HFamilia  : int  1 0 3 0 0 0 0 0 0 1 ...
##  $ VFamilia  : int  0 0 1 0 0 0 0 0 0 2 ...
##  $ Preco     : num  71.28 51.86 21.07 7.85 8.03 ...
##  $ Sexo      : Factor w/ 2 levels "male","female": 2 1 1 2 2 1 1 1 1 2 ...
##  $ Embarque  : Factor w/ 3 levels "S","C","Q": 2 1 1 1 3 1 2 1 2 2 ...
```

```
## [1] "Total da amostra = 178"
```

```
## [1] "DataSet Treino"
```

```
## 'data.frame':    711 obs. of  8 variables:
##  $ Sobreviveu: int  1 1 1 1 0 0 0 1 0 0 ...
##  $ Classe    : int  1 1 2 3 3 3 3 3 3 1 ...
##  $ Nome      : chr  "Hays, Miss. Margaret Bechstein" "Swift, Mrs. Frederick Joel (Margaret Welles Ba
##  $ HFamilia  : int  0 0 1 1 0 0 0 0 0 0 ...
##  $ VFamilia  : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Preco     : num  83.16 25.93 26 7.78 7.12 ...
##  $ Sexo      : Factor w/ 2 levels "male","female": 2 2 2 1 1 1 1 2 1 1 ...
##  $ Embarque  : Factor w/ 3 levels "S","C","Q": 2 1 1 1 1 1 3 1 1 1 ...
```

```
## [1] "Total da amostra = 711"
```

Transformando a coluna *Sobreviveu* em fator para otimizar a aplicação dos modelos de classificação supervisionada.

```
## 'data.frame':    711 obs. of  8 variables:
##  $ Sobreviveu: Factor w/ 2 levels "0","1": 2 2 2 2 1 1 1 2 1 1 ...
##  $ Classe    : int  1 1 2 3 3 3 3 3 3 1 ...
##  $ Nome      : chr  "Hays, Miss. Margaret Bechstein" "Swift, Mrs. Frederick Joel (Margaret Welles Ba
##  $ HFamilia  : int  0 0 1 1 0 0 0 0 0 0 ...
##  $ VFamilia  : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Preco     : num  83.16 25.93 26 7.78 7.12 ...
##  $ Sexo      : Factor w/ 2 levels "male","female": 2 2 2 1 1 1 1 2 1 1 ...
##  $ Embarque  : Factor w/ 3 levels "S","C","Q": 2 1 1 1 1 1 3 1 1 1 ...
```

```
## 'data.frame':    178 obs. of  8 variables:
##  $ Sobreviveu: Factor w/ 2 levels "0","1": 2 1 1 1 2 2 1 1 1 2 ...
##  $ Classe    : int  1 1 3 3 3 1 3 3 3 2 ...
##  $ Nome      : chr  "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "McCarthy, Mr. Timothy J" "
##  $ HFamilia  : int  1 0 3 0 0 0 0 0 0 1 ...
##  $ VFamilia  : int  0 0 1 0 0 0 0 0 0 2 ...
##  $ Preco     : num  71.28 51.86 21.07 7.85 8.03 ...
##  $ Sexo      : Factor w/ 2 levels "male","female": 2 1 1 2 2 1 1 1 1 2 ...
##  $ Embarque  : Factor w/ 3 levels "S","C","Q": 2 1 1 1 3 1 2 1 2 2 ...
```

4. Modelagem

4.1 Regressão Logística

Primeira técnica a ser usada para classificar os dados será **Regressão Logística**.

Primeiro ajuste foi considerando somente as variáveis *Classe*, *Preço* e *Embarque*, pois são as variáveis que possuem alguma relação com critérios sócio-econômicos dos passageiros.

```
##
## Call:
## glm(formula = Sobreviveu ~ Classe + Preco + Embarque + 1, family = binomial(),
##     data = treino)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8504  -0.9474  -0.6880   1.0686   1.7653
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.920539   0.326389   2.820  0.00480 **
## Classe      -0.755367   0.122147  -6.184 6.25e-10 ***
## Preco        0.003181   0.002420   1.314  0.18873
## EmbarqueC    0.560484   0.217565   2.576  0.00999 **
## EmbarqueQ    0.755537   0.291104   2.595  0.00945 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 943.14  on 710  degrees of freedom
## Residual deviance: 847.44  on 706  degrees of freedom
## AIC: 857.44
##
## Number of Fisher Scoring iterations: 4
```

No segundo ajuste retirou-se a variável *Embarque* devido ao não atendimento do nível de confiança. Ou seja, existia um probabilidade maior que 95% do coeficiente dessa variável se igual a **zero**.

```
##
## Call:
## glm(formula = Sobreviveu ~ Classe + Preco + 1, family = binomial(),
##     data = treino)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8041  -0.7750  -0.7442   1.0165   1.6858
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.979396   0.319999   3.061  0.00221 **
## Classe      -0.718530   0.118003  -6.089 1.14e-09 ***
## Preco        0.004364   0.002413   1.809  0.07052 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 943.14  on 710  degrees of freedom
## Residual deviance: 859.19  on 708  degrees of freedom
## AIC: 865.19
##
## Number of Fisher Scoring iterations: 4
```

No terceiro ajuste foi mantido somente a variável *Classe* no modelo.

```
##
## Call:
## glm(formula = Sobreviveu ~ Classe + 1, family = binomial(), data = treino)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4210  -0.7462  -0.7462   0.9520   1.6820
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.40266    0.23142   6.061 1.35e-09 ***
## Classe      -0.84625    0.09802  -8.634  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 943.14  on 710  degrees of freedom
## Residual deviance: 863.18  on 709  degrees of freedom
## AIC: 867.18
##
## Number of Fisher Scoring iterations: 4
```

O ajuste escolhido foi o segundo que possui a equação:

$$Sobreviveu = Classe + Preco + \beta$$

Sendo β o intercepto.

Como o primeiro ajuste possui variáveis que não atendem aos testes estatísticos, a escolha reside entre o segundo ajuste e o terceiro ajuste. Pelo Critério de Informação de Akaike - **AIC**, o valor do segundo ajuste é menor que o terceiro ajuste, portanto possui uma qualidade melhor. Posto isso, a melhor opçao é o segundo ajuste.

```
## [1] "Segundo Ajuste - AIC:  865.190353478141"
```

```
## [1] "Terceiro Ajuste - AIC:  867.182249207522"
```

A escolha pelo segundo ajuste também é confirmada comparando a **Raiz do Erro Quadrático Médio** dos três ajustes.

```
## [1] "Segundo ajuste - RMSE:  0.455766889238112"
```

```
## [1] "Terceiro ajuste - RMSE:  0.457134092169539"
```

Validando o ajuste escolhido.

A **Matriz de Confusão** da validação mostra que o modelo é capaz de prever somente cerca de 40% das pessoas que sobreviveram segundo a métrica **Sensibilidade (RECALL)**, considerando o limite de corte com probabilidade de 50%.

```
## [1] "Limite de corte:  50 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            93            43
## Previsto Positivo            14            28
```

```
## [1] "RECALL:  39.4366 %"
```

A **Precisão** e **F1 Score** confirmam a baixa capacidade de previsão do modelo.

```
## [1] "Precisão:  66.6667 %"
```

```
## [1] "F1 Score:  49.5575 %"
```

A Área Sob a Curva - **AUC** mede a qualidade de modelo, quanto maior o valor do do **AUC** melhor é o modelo. Esse ajuste de **Regressão Logística** mostra um nível um pouco superior a 50%. Que é o percentual que retrata quando não utilizamos modelo algum, deixando a escolha ao acaso.

```
## [1] "AUC:  69.6854 %"
```

O passo seguinte é obter um **Limite de Corte** otimizado que represente a melhor escolha e que pontencialize a métrica **F1 Score**.

```
## [1] "Limite de Corte Otimizado:  0.361264877159018"
```

```
## [1] "Valor Máximo alcançado pela métrica F1 Score:  0.591549295774648"
```

Utilizando o novo **Limite de Corte** otimizado para obter os valores da **Precisão**, **Recall** e **F1 Score**, tem-se as medidas de qualidade para o ajuste utilizando a técnica de **Regressão Logística**.

```
## [1] "Precisão:  59.1549 %"
```

```
## [1] "RECALL:  59.1549 %"
```

```
## [1] "F1 Score:  59.1549 %"
```

```
## [1] "Matriz de Confusão:"
```

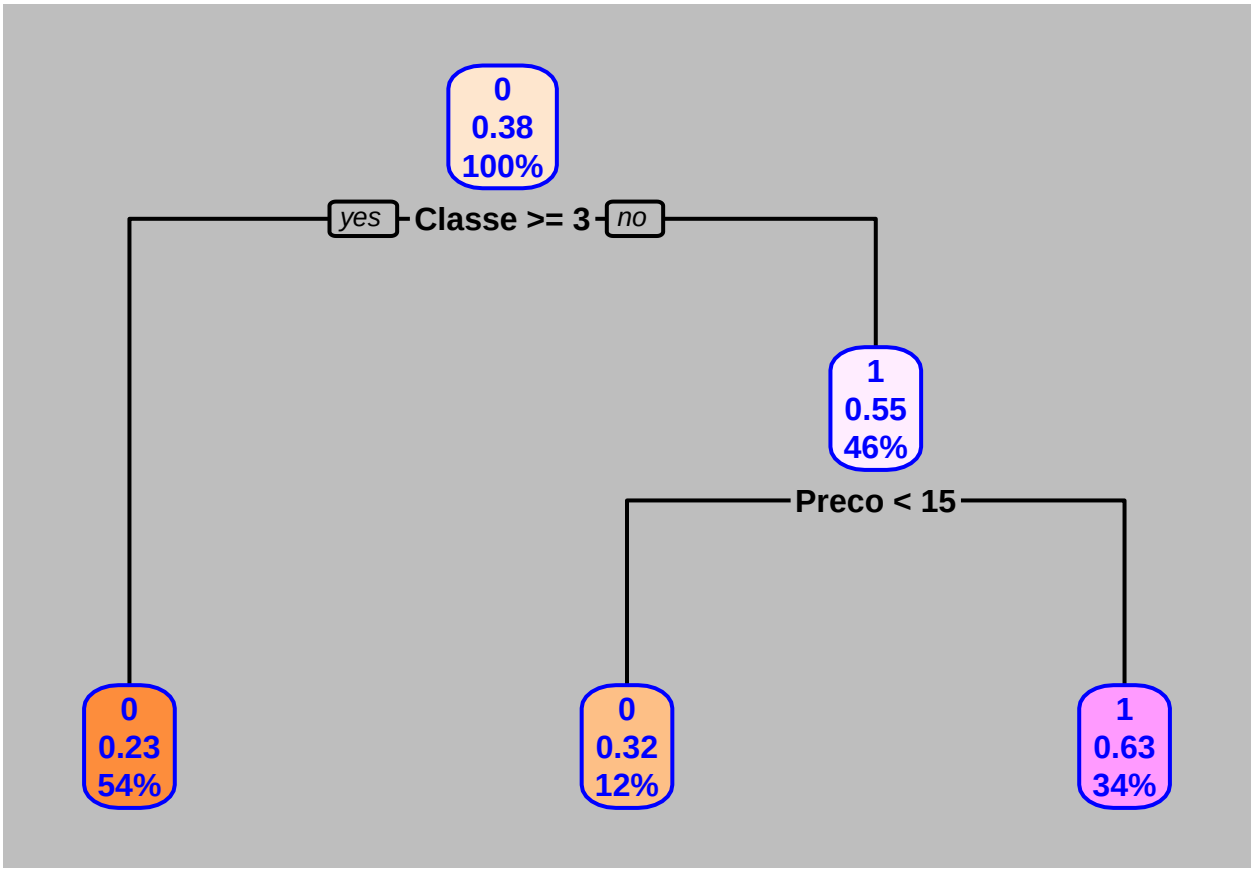
```
##                   Real Negativo Real Positivo
## Previsto Negativo            78            29
## Previsto Positivo            29            42
```

4.2 Árvore de Decisão

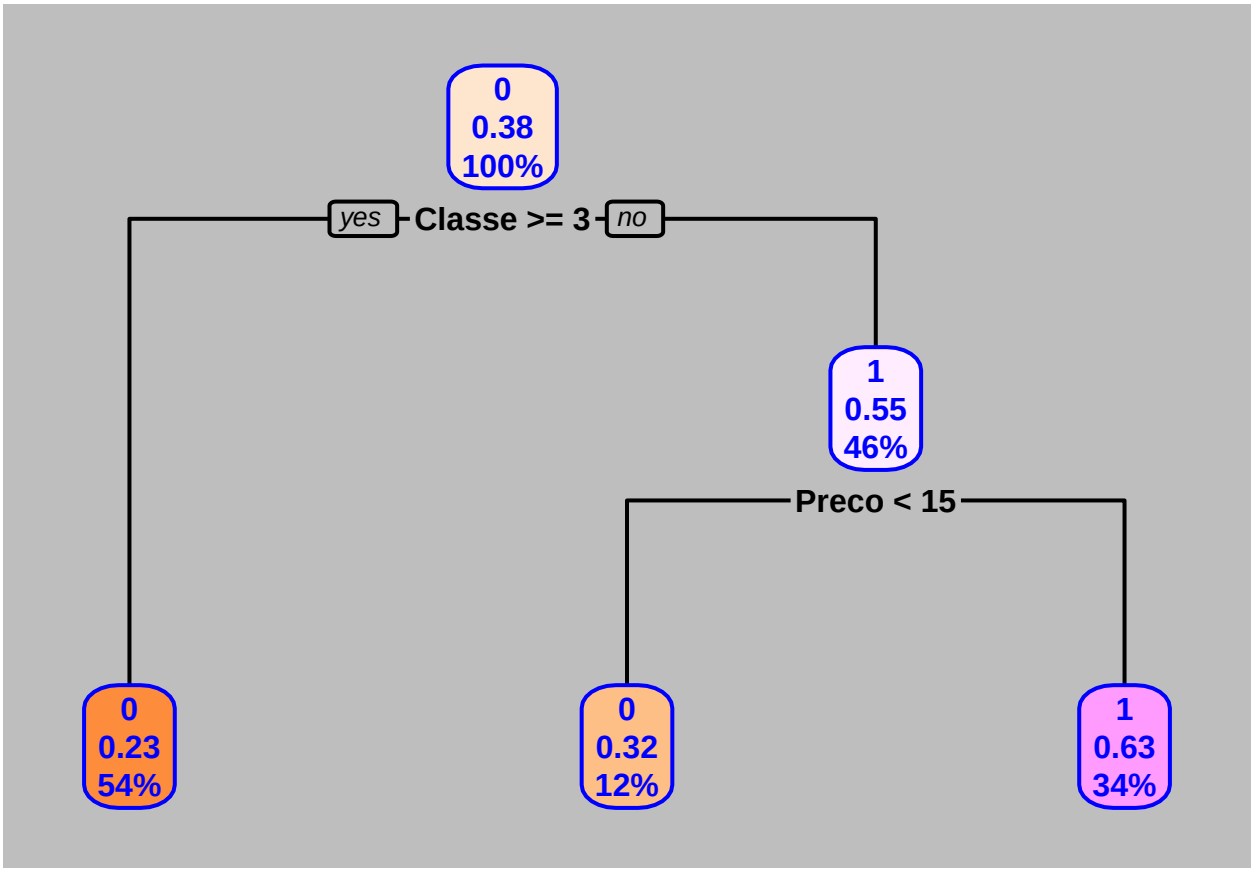
A segunda técnica a ser utilizada será **Árvore de Decisão**.

Carregando as bibliotecas necessárias para utilizar a técnica. Será necessário a biblioteca **rpart** que possui a função para gerar a árvore de decisão e a **rpart.plot** que possui a função para plotar a árvore em modo gráfico.

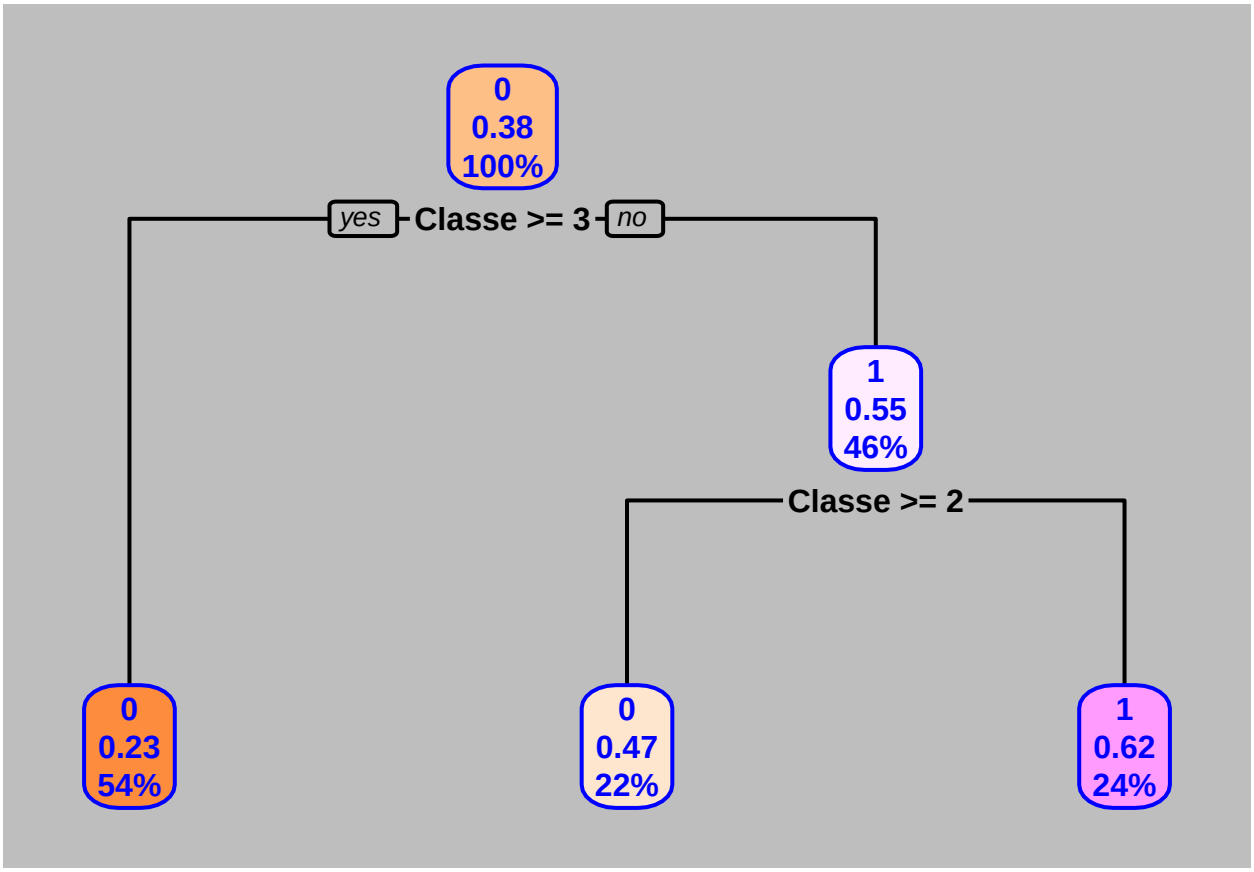
Primeiro ajuste foi considerando somente as variáveis *Classe*, *Preço* e *Embarque*, pois são as variáveis que possuem alguma relação com critérios sócio-econômicos dos passageiros.



No segundo ajuste retirou-se a variável *Embarque*.



No terceiro ajuste foi mantido somente a variável *Classe* no modelo.



Como a classificação para ambos os ajustes, primeiro e segundo, é basicamente a mesma, optou-se pelo segundo pois ele é mais parcimonioso.

```
## Call:
## rpart(formula = Sobreviveu ~ Classe + Preco + 1, data = treino)
##   n= 711
##
##          CP nsplit rel error    xerror       xstd
## 1 0.1152416      0 1.0000000 1.0000000 0.04807289
## 2 0.0100000      2 0.7695167 0.7843866 0.04528341
##
## Variable importance
## Classe  Preco
##     53     47
##
## Node number 1: 711 observations,    complexity param=0.1152416
##   predicted class=0  expected loss=0.3783404  P(node) =1
##     class counts:   442   269
##    probabilities: 0.622 0.378
##   left son=2 (384 obs) right son=3 (327 obs)
##   Primary splits:
##       Classe < 2.5      to the right, improve=34.6098, (0 missing)
##       Preco  < 10.825   to the left,  improve=28.6193, (0 missing)
##   Surrogate splits:
##       Preco < 10.48125 to the left,  agree=0.799, adj=0.563, (0 split)
##
```

```
## Node number 2: 384 observations
##   predicted class=0  expected loss=0.234375  P(node) =0.5400844
##     class counts:   294    90
##    probabilities: 0.766 0.234
##
## Node number 3: 327 observations,    complexity param=0.1152416
##   predicted class=1  expected loss=0.4525994  P(node) =0.4599156
##     class counts:   148   179
##    probabilities: 0.453 0.547
##   left son=6 (87 obs) right son=7 (240 obs)
##   Primary splits:
##       Preco  < 15.4     to the left,  improve=12.061900, (0 missing)
##       Classe < 1.5      to the right, improve= 3.443034, (0 missing)
##   Surrogate splits:
##       Classe < 1.5      to the right, agree=0.768, adj=0.126, (0 split)
##
## Node number 6: 87 observations
##   predicted class=0  expected loss=0.3218391  P(node) =0.1223629
##     class counts:    59    28
##    probabilities: 0.678 0.322
##
## Node number 7: 240 observations
##   predicted class=1  expected loss=0.3708333  P(node) =0.3375527
##     class counts:    89   151
##    probabilities: 0.371 0.629
```

Validando o ajuste escolhido.

A **Matriz de Confusão** da validação mostra que o modelo é capaz de prever somente cerca de 50% das pessoas que sobreviveram segundo a métrica **Sensibilidade (RECALL)**, considerando o limite de corte com probabilidade de 50%.

```
## [1] "Limite de corte:  50 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            87            36
## Previsto Positivo            20            35
```

```
## [1] "RECALL:  49.2958 %"
```

A **Precisão** e **F1 Score** confirmam a baixa capacidade de previsão do modelo.

```
## [1] "Precisão:  63.6364 %"
```

```
## [1] "F1 Score:  55.5556 %"
```

O passo seguinte é obter um **Limite de Corte** otimizado que represente a melhor escolha e que pontencialize a métrica **F1 Score**.

```
## Warning in optimize(f1_score, c(0, 1), tol = 1e-04, maximum = TRUE): NA/Inf
## substituido pelo máximo valor positivo
```

```
## [1] "Limite de Corte Otimizado:  0.382019207359601"
```

```
## [1] "Valor Máximo alcançado pela métrica F1 Score:  0.555555555555555"
```

Utilizando o novo **Limite de Corte** otimizado para obter os valores da **Precisão**, **Recall** e **F1 Score**, tem-se as medidas de qualidade para o ajuste utilizando a técnica de **Árvore de Decisão**.

```
## [1] "Precisão:  63.6364 %"
```

```
## [1] "RECALL:  49.2958 %"
```

```
## [1] "F1 Score:  55.5556 %"
```

```
## [1] "Matriz de Confusão:"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            87            36
## Previsto Positivo            20            35
```

Nota-se que não houve uma evolução nos valores das métricas com o novo limite de corte otimizado.

Estabelecendo um limite de corte de **32,0%**, obtem-se um valor para **F1 Score** melhor, pois aparentemente a função *optimize* não convergiu para um máximo global.

```
## [1] "Precisão:  59.1549 %"
```

```
## [1] "RECALL:  59.1549 %"
```

```
## [1] "F1 Score:  59.1549 %"
```

```
## [1] "Matriz de Confusão:"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            78            29
## Previsto Positivo            29            42
```

4.3 Análise de Discriminante

A terceira técnica a ser utilizada será **Análise de Discriminante**.

Carregando as bibliotecas necessárias para utilizar a técnica. Será necessário a biblioteca **MASS** que contém a função capaz de gerar a função e as bilbiotecas **heplots** que contém o teste *Box-M*, para verificar a semelhança da matriz de variância-covariância entre os grupos e **rrcov** que contém o teste de *Lambda de Wilks*.

Primeiro ajuste foi considerando somente as variáveis *Classe*, *Preço* e *Embarque*, pois são as variáveis que possuem alguma relação com critérios sócio-econômicos dos passageiros.

```
## Call:
## lda(Sobreviveu ~ Classe + Preco + Embarque + 1, data = treino)
##
## Prior probabilities of groups:
##         0         1
## 0.6216596 0.3783404
##
## Group means:
##     Classe    Preco EmbarqueC  EmbarqueQ
## 0 2.515837 22.41375 0.1380090 0.08597285
## 1 1.940520 46.52127 0.2862454 0.08921933
##
## Coefficients of linear discriminants:
##                     LD1
## Classe    -1.050515276
## Preco      0.003652901
## EmbarqueC  0.742018022
## EmbarqueQ  0.963830856
```

Analisando se o modelo atende a um dos pressupostos da técnica de **Análise de Discriminante** que é homogeneidade da matriz de variância-covariância, aplicamos o teste de *Box-M* que estabelece como hipótese nula a homogeneidade dessa matriz.

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  treino_exp[, c(2, 6, 9)]
## Chi-Sq (approx.) = 235.11, df = 6, p-value < 2.2e-16
```

```
## [1] "p-Valor: 6.2149183486046e-48"
```

Verificou-se que o p-Valor é menor que 5%, portanto rejeita-se a hipótese nula que estabelece que as matrizes de variância e covariância são homogêneas. Assim não é possível utilizar o ajuste acima, pois ele não atende ao pressuposto da homogeneidade da matriz de variância-covariância da técnica.

No segundo ajuste retirou-se a variável *Embarque*.

```
## Call:
## lda(Sobreviveu ~ Classe + Preco + 1, data = treino)
##
## Prior probabilities of groups:
##         0         1
## 0.6216596 0.3783404
##
## Group means:
##     Classe    Preco
## 0 2.515837 22.41375
## 1 1.940520 46.52127
##
## Coefficients of linear discriminants:
##                 LD1
## Classe -1.083075604
## Preco   0.005316472
```

Analisando se o modelo atende a um dos pressupostos da técnica de **Análise de Discriminante** que é homogeneidade da matriz de variância-covariância, aplicamos o teste de *Box-M* que estabelece como hipótese nula a homogeneidade dessa matriz.

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  treino_exp[, c(2, 6)]
## Chi-Sq (approx.) = 233.25, df = 3, p-value < 2.2e-16
```

```
## [1] "p-Valor: 2.73607957610163e-50"
```

Verificou-se que o p-Valor é menor que 5%, portanto rejeita-se a hipótese nula que estabelece que as matrizes de variância e covariância são homogêneas. Assim não é possível utilizar o ajuste acima, pois ele não atende ao pressuposto da homogeneidade da matriz de variância-covariância da técnica.

No terceiro ajuste foi mantido somente a variável *Classe* no modelo.

```
## Call:
## lda(Sobreviveu ~ Classe + 1, data = treino)
##
## Prior probabilities of groups:
##         0         1
## 0.6216596 0.3783404
##
## Group means:
##     Classe
## 0 2.515837
## 1 1.940520
##
## Coefficients of linear discriminants:
##             LD1
## Classe 1.273108
```

Como não há como analisar se o modelo atende a um dos pressupostos da técnica de **Análise de Discriminante** que é homogeneidade da matriz de variância-covariância, pois ele possui somente uma variável dependente, foi descartado esse ajuste.
Como o primeiro e segundo ajustes não atenderam a um dos pressupostos da técnica, a **Análise de Discriminante** não será utilizado nessa predição.

4.4 k-NN (k-Nearest Neightbors)

Quarto técnica a ser testado será uma **k-NN (k-Nearest Neighbors)**.

Carregando as bibliotecas necessárias para utilizar a técnica. Será necessário a biblioteca **class** que contém a função responsável por implementar a técnica **k-NN**.

Primeiro ajuste foi considerando somente as variáveis *Classe*, *Preço* e *Embarque*, pois são as variáveis que possuem alguma relação com critérios sócio-econômicos dos passageiros.

```
##   [1] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
##  [36] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1
## [106] 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0
```

```
## [141] 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1
## [176] 0 0 0
## attr(,"prob")
##   [1] 0.5333333 0.6333333 0.5666667 0.9444444 0.5333333 0.5333333 0.8378378
##   [8] 0.8139535 0.8703704 0.7096774 0.7741935 0.6129032 0.5666667 0.8857143
##  [15] 0.5333333 0.8333333 0.8857143 0.9354839 0.8139535 0.8727273 0.8857143
##  [22] 0.6666667 0.6285714 0.9354839 0.9375000 0.6333333 0.5666667 0.8727273
##  [29] 0.6000000 0.6857143 0.5312500 0.6666667 0.9444444 0.5312500 0.6000000
##  [36] 0.8627451 0.8333333 0.5666667 0.7000000 0.7000000 0.6984127 0.6562500
##  [43] 0.6451613 0.7666667 1.0000000 0.6333333 0.5000000 0.6000000 0.9444444
##  [50] 0.6000000 0.8571429 0.6000000 0.7096774 0.8139535 0.5312500 0.7096774
##  [57] 0.7666667 0.5666667 0.6000000 0.6764706 0.6333333 0.7666667 0.7666667
##  [64] 0.7096774 0.6285714 0.6285714 0.8333333 0.8333333 0.5161290 0.5116279
##  [71] 0.5312500 0.6666667 0.9375000 0.9354839 0.5666667 0.9393939 0.8857143
##  [78] 0.9354839 0.6333333 0.6129032 0.6285714 0.7666667 0.5333333 0.8387097
##  [85] 0.5161290 0.5483871 0.8139535 0.6000000 0.8857143 0.7096774 0.8333333
##  [92] 0.6333333 0.5483871 0.8703704 0.5636364 0.6285714 0.9375000 0.5636364
##  [99] 0.6333333 0.8387097 0.6333333 1.0000000 0.8857143 0.8333333 0.7096774
## [106] 0.5312500 0.6000000 0.8333333 0.6333333 0.8648649 0.7096774 0.8048780
## [113] 0.5428571 0.5161290 0.6666667 0.8378378 0.7741935 0.6129032 0.8139535
## [120] 0.7000000 0.7333333 0.6285714 0.8593750 0.5937500 0.8378378 0.5483871
## [127] 0.6451613 0.5937500 1.0000000 0.7812500 0.8139535 0.6666667 0.7666667
## [134] 0.5416667 0.8857143 0.5666667 0.7096774 0.7096774 0.8857143 0.6000000
## [141] 0.8064516 0.5660377 0.5636364 0.8333333 0.6285714 0.7000000 0.7741935
## [148] 0.9375000 0.8139535 0.7096774 0.6000000 0.5714286 0.7741935 0.6333333
## [155] 0.7333333 0.7666667 0.6000000 0.7714286 0.6000000 0.7096774 0.9375000
## [162] 0.6000000 0.5312500 0.9375000 0.5416667 0.6333333 0.9393939 1.0000000
## [169] 0.6666667 0.8717949 0.7333333 0.8000000 0.7741935 0.6562500 0.5588235
## [176] 0.6363636 0.8139535 0.8139535
## Levels: 0 1
```

Analisando o primeiro ajuste utilizando a matriz de confusão e a métrica F1 Score como medida de qualidade do ajuste, verificamos que o nível de predição não chega a 50%.

```
## [1] "F1 Score: 49.557522 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            93            43
## Previsto Positivo            14            28
```

No segundo ajuste retirou-se a variável *Embarque*.

```
##   [1] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
##  [36] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1
## [106] 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0
## [141] 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1
## [176] 0 0 0
## attr(,"prob")
##   [1] 0.5333333 0.6333333 0.5483871 0.8636364 0.8857143 0.5000000 0.8235294
##   [8] 0.9032258 0.9032258 0.7096774 0.6521739 0.6129032 0.5666667 0.8857143
##  [15] 0.5312500 0.8387097 0.8857143 0.8235294 0.9032258 0.8644068 0.8857143
##  [22] 0.7666667 0.6285714 0.8181818 0.8378378 0.6333333 0.5666667 0.8644068
```

```
##  [29] 0.6000000 0.6060606 0.5161290 0.6666667 0.8636364 0.5312500 0.6000000
##  [36] 0.8545455 0.8387097 0.5333333 0.7000000 0.7000000 0.7297297 0.6333333
##  [43] 0.6451613 0.7500000 0.9459459 0.6333333 0.5000000 0.6000000 0.8636364
##  [50] 0.6000000 0.8333333 0.6000000 0.7096774 0.9032258 0.5312500 0.7096774
##  [57] 0.7666667 0.5666667 0.6000000 0.6764706 0.6333333 0.7500000 0.7666667
##  [64] 0.7096774 0.6285714 0.6285714 0.8000000 0.8387097 0.9062500 0.5769231
##  [71] 0.5312500 0.7666667 0.8285714 0.8235294 0.5666667 0.8461538 0.8857143
##  [78] 0.8235294 0.6333333 0.6129032 0.6285714 0.7666667 0.5333333 0.8437500
##  [85] 0.5161290 0.5483871 0.9032258 0.6250000 0.8857143 0.7096774 0.8387097
##  [92] 0.6333333 0.5312500 0.9032258 0.5636364 0.6285714 0.8285714 0.5636364
##  [99] 0.6333333 0.8437500 0.6333333 0.9459459 0.8857143 0.8387097 0.7096774
## [106] 0.5312500 0.6000000 0.8387097 0.6000000 0.7407407 0.7096774 0.8181818
## [113] 0.5428571 0.5161290 0.6046512 0.8235294 0.6590909 0.6129032 0.9032258
## [120] 0.7000000 0.7666667 0.6285714 0.8524590 0.5937500 0.8235294 0.5333333
## [127] 0.6451613 0.5937500 0.9459459 0.6744186 0.9032258 0.8421053 0.7666667
## [134] 0.5510204 0.8857143 0.5666667 0.7096774 0.7096774 0.8857143 0.6250000
## [141] 0.8064516 0.5660377 0.5636364 0.8000000 0.6285714 0.7000000 0.6590909
## [148] 0.8285714 0.9032258 0.7096774 0.6000000 0.5416667 0.6521739 0.6666667
## [155] 0.7666667 0.7666667 0.6000000 0.6000000 0.6000000 0.7096774 0.8285714
## [162] 0.6000000 0.5312500 0.8378378 0.5306122 0.6333333 0.8461538 0.9459459
## [169] 0.7666667 0.8571429 0.7333333 0.8461538 0.6590909 0.6333333 0.5588235
## [176] 0.6363636 0.9032258 0.9032258
## Levels: 0 1
```

Analisando o segundo ajuste utilizando a matriz de confusão e a métrica F1 Score como medida de qualidade do ajuste, verificamos que o nível de predição também não chega a 50%.

```
## [1] "F1 Score: 48.648649 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            94            44
## Previsto Positivo            13            27
```

No terceiro ajuste foi mantido somente a variável *Classe* no modelo.

```
##   [1] 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
##  [36] 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0
##  [71] 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1
## [106] 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 1 0 0
## [141] 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
## [176] 0 0 0
## attr(,"prob")
##   [1] 0.6162791 0.6162791 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250
##   [8] 0.7656250 0.7656250 0.5290323 0.7656250 0.7656250 0.5290323 0.7656250
##  [15] 0.5290323 0.7656250 0.7656250 0.7656250 0.7656250 0.7656250 0.7656250
##  [22] 0.7656250 0.5290323 0.7656250 0.7656250 0.5290323 0.7656250 0.7656250
##  [29] 0.7656250 0.7656250 0.6162791 0.7656250 0.7656250 0.5290323 0.7656250
##  [36] 0.7656250 0.7656250 0.7656250 0.6162791 0.5290323 0.5290323 0.7656250
##  [43] 0.7656250 0.6162791 0.7656250 0.5290323 0.6162791 0.7656250 0.7656250
##  [50] 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250 0.5290323 0.6162791
##  [57] 0.6162791 0.7656250 0.7656250 0.6162791 0.7656250 0.6162791 0.6162791
##  [64] 0.6162791 0.5290323 0.5290323 0.7656250 0.7656250 0.7656250 0.7656250
##  [71] 0.5290323 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250 0.7656250
```

```
##  [78] 0.7656250 0.6162791 0.7656250 0.5290323 0.6162791 0.7656250 0.7656250
##  [85] 0.5290323 0.7656250 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250
##  [92] 0.5290323 0.5290323 0.7656250 0.6162791 0.5290323 0.7656250 0.6162791
##  [99] 0.7656250 0.7656250 0.6162791 0.5290323 0.7656250 0.7656250 0.6162791
## [106] 0.6162791 0.7656250 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250
## [113] 0.6162791 0.6162791 0.5290323 0.7656250 0.7656250 0.6162791 0.7656250
## [120] 0.5290323 0.7656250 0.5290323 0.7656250 0.6162791 0.7656250 0.7656250
## [127] 0.5290323 0.6162791 0.6162791 0.7656250 0.7656250 0.7656250 0.6162791
## [134] 0.6162791 0.7656250 0.6162791 0.5290323 0.6162791 0.7656250 0.7656250
## [141] 0.6162791 0.6162791 0.6162791 0.7656250 0.5290323 0.5290323 0.7656250
## [148] 0.7656250 0.7656250 0.6162791 0.7656250 0.7656250 0.7656250 0.5290323
## [155] 0.7656250 0.6162791 0.7656250 0.7656250 0.7656250 0.6162791 0.7656250
## [162] 0.7656250 0.5290323 0.7656250 0.7656250 0.5290323 0.7656250 0.6162791
## [169] 0.7656250 0.7656250 0.7656250 0.7656250 0.7656250 0.7656250 0.6162791
## [176] 0.7656250 0.7656250 0.7656250
## Levels: 0 1
```

Analisando o terceiro ajuste utilizando a matriz de confusão e a métrica F1 Score como medida de qualidade do ajuste, verificamos que o nível de predição também não chega a 50%.

```
## [1] "F1 Score: 49.557522 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            93            43
## Previsto Positivo            14            28
```

Utilizando o primeiro ajuste e um número de vizinhos (k) ligeiramente maior, igual a 50, tentou-se verificar se haveria uma melhora na predição.

```
## [1] "k = 50"
```

```
## [1] "Precisão:  68.1818 %"
```

```
## [1] "RECALL:  42.2535 %"
```

```
## [1] "F1 Score: 52.173913 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            93            41
## Previsto Positivo            14            30
```

Realmente produzimos uma melhora na capacidade de predição segundo a métrica F1 Score. Na tentativa de aprimorar a capacidade de predição, aumentamos o número de vizinhos, sendo que constatamos que a qualidade de predição segundo a métrica F1 Score não evoluiu. Portanto, optou-se por utilizar 50 vizinhos próximos com o primeiro ajuste.

4.5 Random Forest

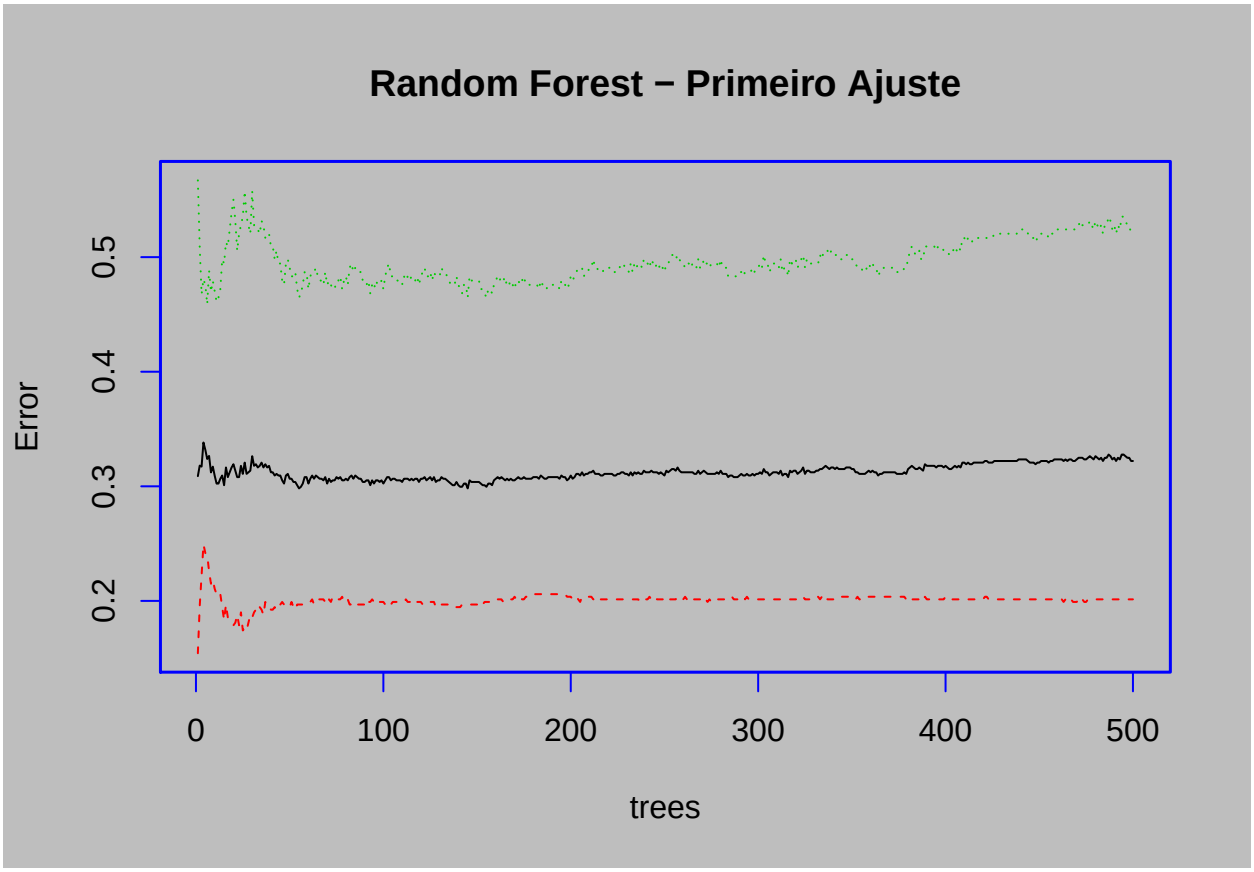
Quinto técnica a ser testado será uma **Random Forest**, que na verdade não chega a ser uma técnica e sim uma evolução na técnica de **Árvore de Decisão**, utilizando várias árvores acopladas para se chegar em uma predição melhor.

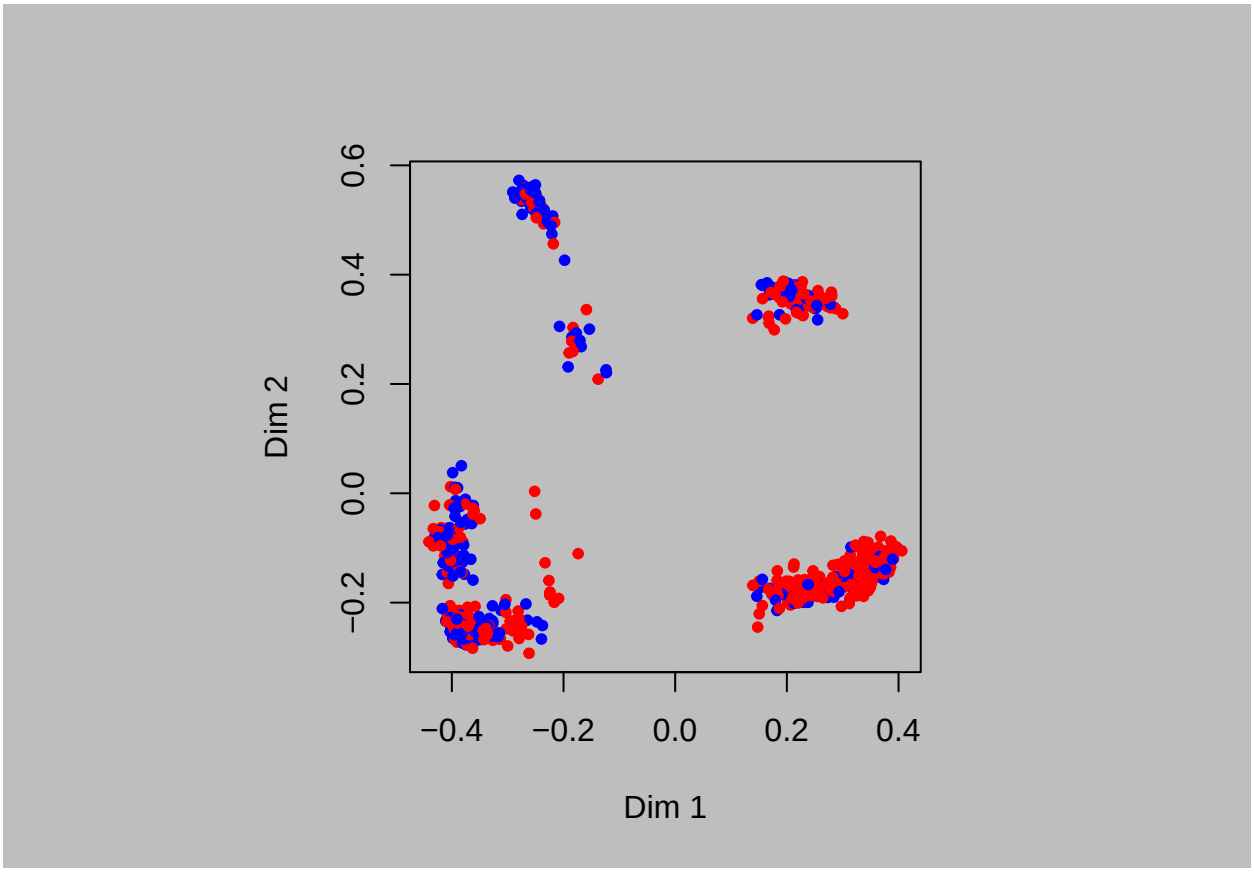
Carregando as bibliotecas necessárias para utilizar a técnica. Será utilizado duas bibliotecas, a **randomForest** que contém a função para gerar as árvores de decisão e a **caret** que contém as métricas que serão utilizadas para avaliar a qualidade do modelo.

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:ModelMetrics':
##
##     confusionMatrix, precision, recall, sensitivity, specificity

## The following object is masked from 'package:purrr':
##
##     lift
```

Primeiro ajuste foi considerando somente as variáveis *Classe*, *Preço* e *Embarque*, pois são as variáveis que possuem alguma relação com critérios sócio-econômicos dos passageiros.

```
##
## Call:
##  randomForest(formula = Sobreviveu ~ Classe + Preco + Embarque +      1, data = treino_exp, proximity
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 32.21%
## Confusion matrix:
##     0   1 class.error
## 0 353  89   0.2013575
## 1 140 129   0.5204461
```


Random Forest − Primeiro Ajuste
Error
0.2
0.3
0.4
0.5
0
100
200
300
400
500
trees



Avaliando o primeiro ajuste, obtivemos um **F1 Score** em torno de 50%.

```
##     0   1 class.error
## 0 353  89   0.2013575
## 1 140 129   0.5204461
```

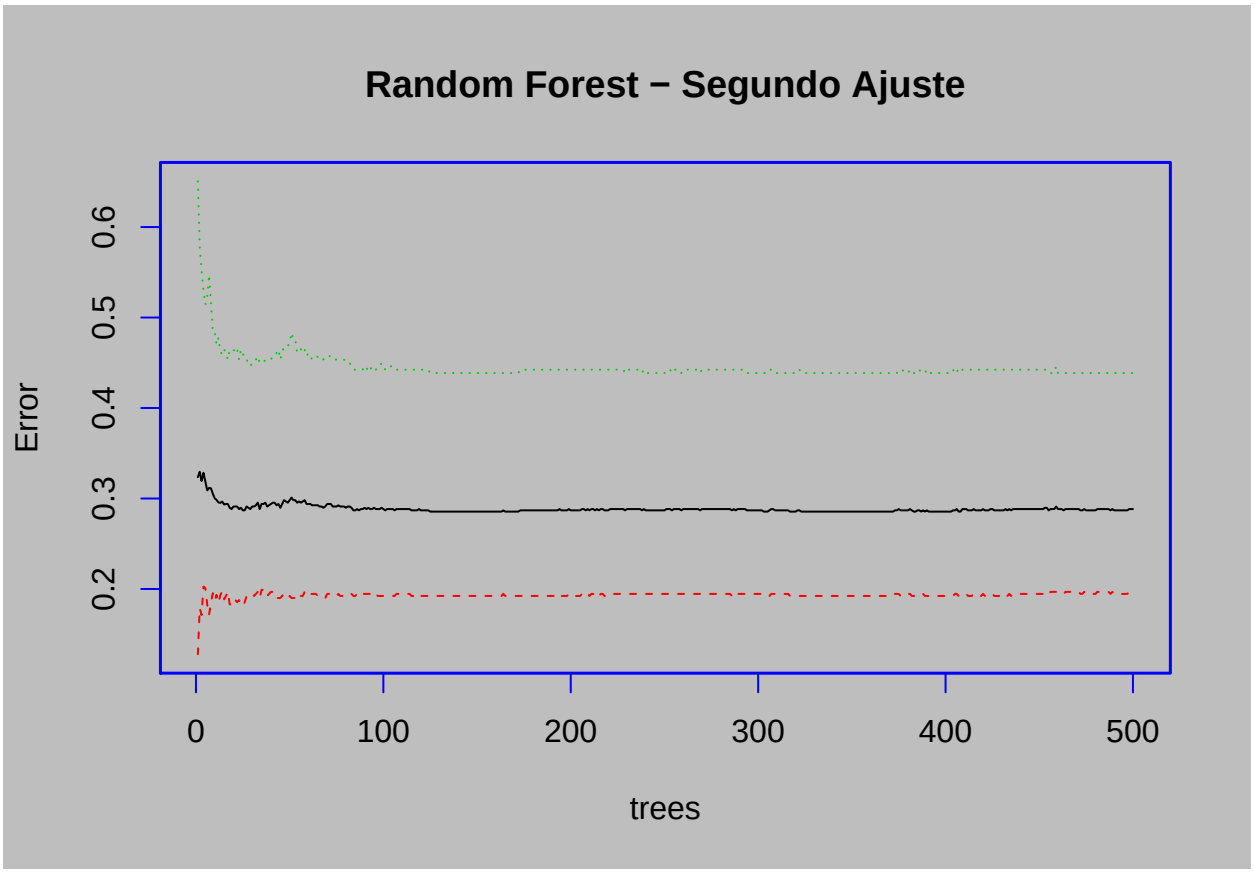
```
## [1] "Precisão: 0.479553903345725"
```

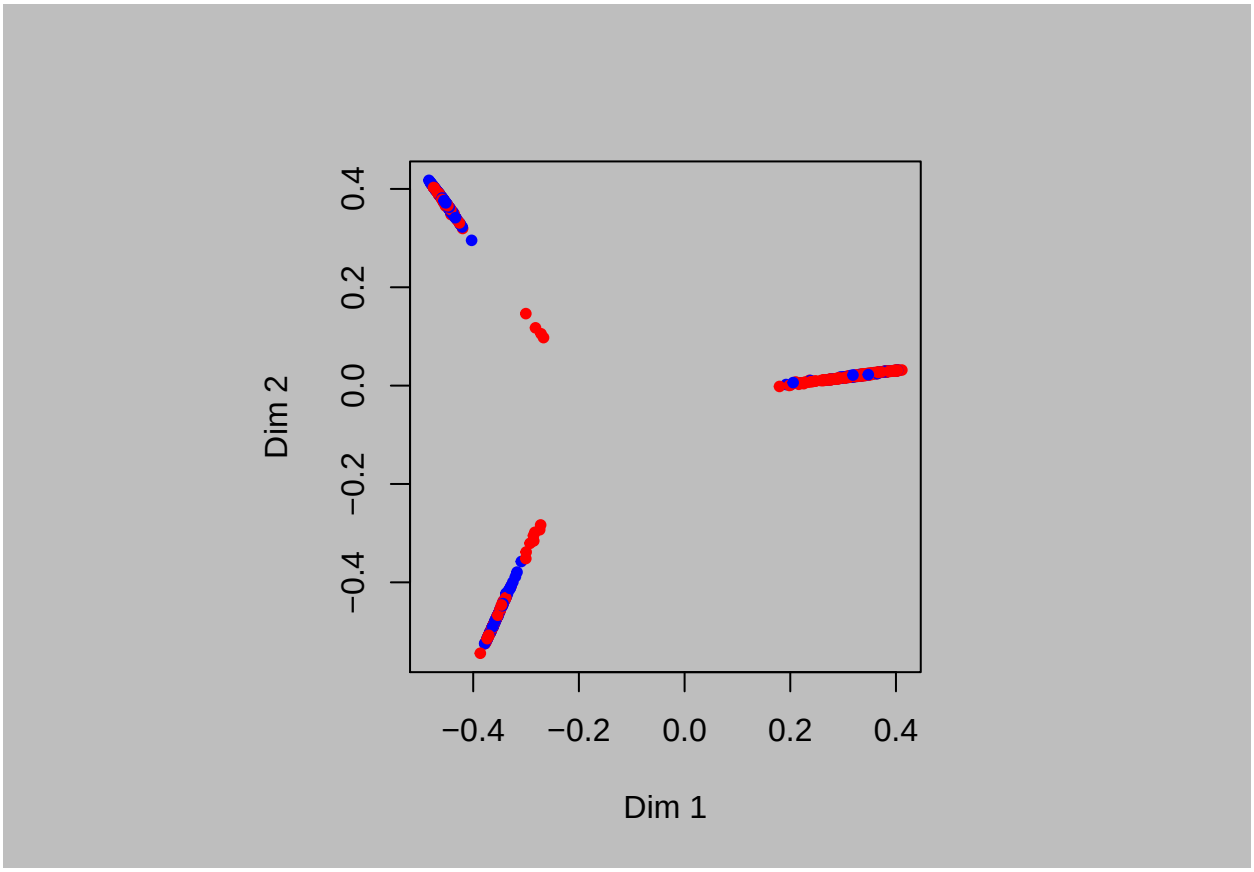
```
## [1] "Sensibilidade (RECALL): 0.591743119266055"
```

```
## [1] "F1 Score: 0.529774127310062"
```

No segundo ajuste retirou-se a variável *Embarque*.

```
##
## Call:
##  randomForest(formula = Sobreviveu ~ Classe + Preco + 1, data = treino_exp,      proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 28.83%
## Confusion matrix:
##     0   1 class.error
## 0 355  87   0.1968326
## 1 118 151   0.4386617
```


Random Forest − Segundo Ajuste
Error
0.2
0.3
0.4
0.5
0.6
0
100
200
300
400
500
trees



Avaliando o segundo ajuste, o **F1 Score** se aproxima de 60%.

```
##     0   1 class.error
## 0 355  87   0.1968326
## 1 118 151   0.4386617
```

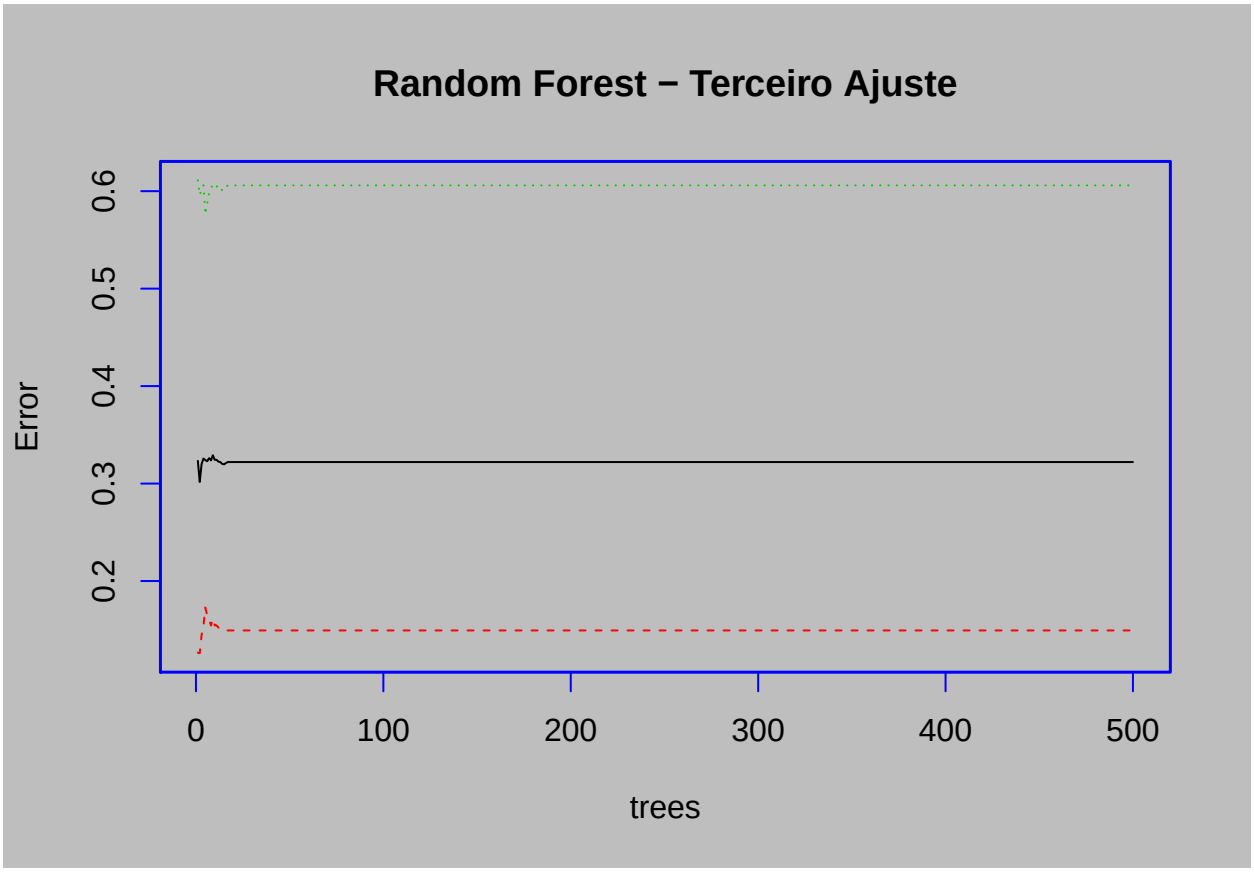
```
## [1] "Precisão: 0.561338289962825"
```

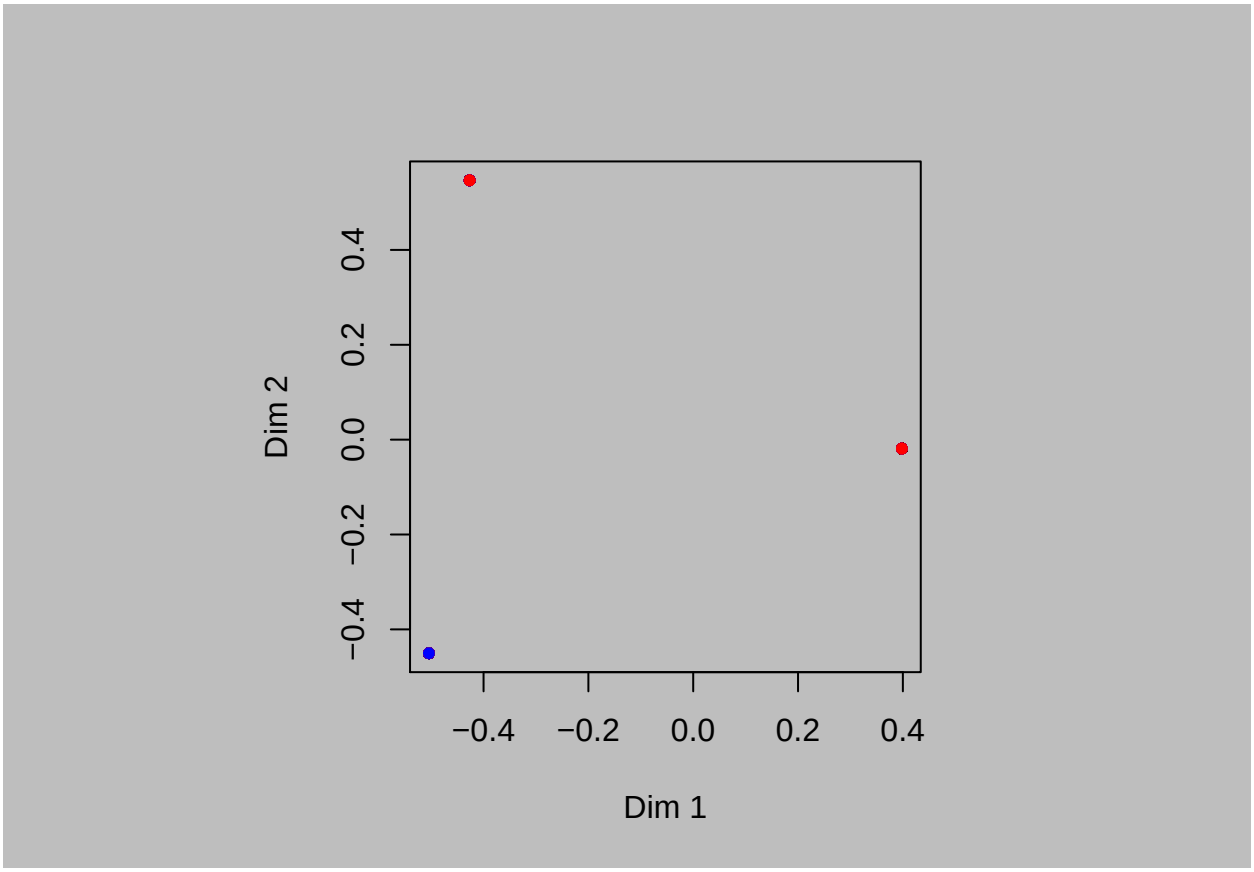
```
## [1] "Sensibilidade (RECALL): 0.634453781512605"
```

```
## [1] "F1 Score: 0.595660749506903"
```

No terceiro ajuste foi mantido somente a variável *Classe* no modelo.

```
##
## Call:
##  randomForest(formula = Sobreviveu ~ Classe + 1, data = treino_exp,      proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 32.21%
## Confusion matrix:
##     0   1 class.error
## 0 376  66   0.1493213
## 1 163 106   0.6059480
```


Random Forest − Terceiro Ajuste
Error
0.2
0.3
0.4
0.5
0.6
0
100
200
300
400
500
trees



Avaliando o terceiro ajuste, verificamos que o **F1 Score** retorna para um valor em torno de 50%.

```
##     0   1 class.error
## 0 376  66   0.1493213
## 1 163 106   0.6059480
```

```
## [1] "Precisão: 0.394052044609665"
```

```
## [1] "Sensibilidade (RECALL): 0.616279069767442"
```

```
## [1] "F1 Score: 0.480725623582766"
```

De acordo com uma avaliação na **F1 Score** e **Recall** dos três ajustes, optou-se pelo Segundo Ajuste pois produziu valores de métrica melhores.

Validando o ajuste escolhido.

A **Matriz de Confusão** da validação mostra que o modelo é capaz de prever somente cerca de 50% das pessoas que sobreviveram segundo a métrica **Sensibilidade (RECALL)**.

```
## [1] "Precisão para RandomForest: 0.648148148148148"
```

```
## [1] "Sensibilidade (RECALL) para RandomForest: 0.492957746478873"
```

```
## [1] "F1 Score: 0.56"
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 88 36
##          1 19 35
##
##                Accuracy : 0.691
##                  95% CI : (0.6175, 0.758)
##     No Information Rate : 0.6011
##     P-Value [Acc > NIR] : 0.008131
##
##                   Kappa : 0.3286
##
##  Mcnemar's Test P-Value : 0.030971
##
##             Sensitivity : 0.4930
##             Specificity : 0.8224
##          Pos Pred Value : 0.6481
##          Neg Pred Value : 0.7097
##               Precision : 0.6481
##                  Recall : 0.4930
##                      F1 : 0.5600
##              Prevalence : 0.3989
##          Detection Rate : 0.1966
##    Detection Prevalence : 0.3034
##       Balanced Accuracy : 0.6577
##
##        'Positive' Class : 1
##
```

5. Conclusão

Avaliando as quatro técnicas avaliadas, pois a **Análise de Discriminante** poderá ser utilizada nesse caso, verificamos que a **Regressão Logística** e a **Árvore de Decisão** simples obtiveram uma perfomance melhor, considerando a métrica escolhida para avaliação que foi a **Sensibilidade (Recall)**.

Para a **Regressão Logística** obtivemos as seguintes métricas na validação do modelo.

```
## [1] "Precisão: 59.15493 %"
```

```
## [1] "Sensibilidade (RECALL): 59.15493 %"
```

```
## [1] "F1 Score: 59.15493 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            78            29
## Previsto Positivo            29            42
```

Para a **Árvore de Decisão** obtivemos as seguintes métricas na valiação do modelo.

```
## [1] "Precisão: 59.15493 %"
```

```
## [1] "Sensibilidade (RECALL): 59.15493 %"
```

```
## [1] "F1 Score: 59.15493 %"
```

```
##                   Real Negativo Real Positivo
## Previsto Negativo            78            29
## Previsto Positivo            29            42
```

Com base nos resultados obtidos, qualquer um dos dois modelos poderia ser empregado para gerar a predição do modelo. Nesse caso, optaríamos pelo modelo de **Regressão Logística** por uma escolha pessoal.