Machine Learning - Exercício

Walter Humberto Subiza Pina

22 de novembro de 2016

Contents

Objetivo do exercício

Com a base de dados sobre crédito bancário, dividir o conjunto de dados em treino e teste e criar uma arvore capaz de detectar se o cliente tem capacidade de pagar um empréstimo. Prof: Tiago Mendes Dantas - FGV # Execução ## Carregando dados e arrumando

```
setwd("H:/FGV/05 Machine Learning - Tiago")
load("H:/FGV/05 Machine Learning - Tiago/credito.RData")
# conhecendo os dados...
str(credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ Creditability                    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Account.Balance                  : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 1 1 1 4 2 ...
##  $ Duration.of.Credit..month.       : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ Payment.Status.of.Previous.Credit: Factor w/ 5 levels "0","1","2","3",..: 5 5 3 5 5 5 5 5 5 3 ...
##  $ Purpose                          : Factor w/ 10 levels "0","1","2","3",..: 3 1 9 1 1 1 1 1 4 4 ...
##  $ Credit.Amount                    : int  1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
##  $ Value.Savings.Stocks             : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 3 ...
##  $ Length.of.current.employment     : Factor w/ 5 levels "1","2","3","4",..: 2 3 4 3 3 2 4 2 1 1 ...
##  $ Instalment.per.cent              : Factor w/ 4 levels "1","2","3","4": 4 2 2 3 4 1 1 2 4 1 ...
##  $ Sex...Marital.Status             : Factor w/ 4 levels "1","2","3","4": 2 3 2 3 3 3 3 3 2 2 ...
##  $ Guarantors                       : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Duration.in.Current.address      : Factor w/ 4 levels "1","2","3","4": 4 2 4 2 4 3 4 4 4 4 ...
##  $ Most.valuable.available.asset    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 3 4 ...
##  $ Age..years.                      : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ Concurrent.Credits               : Factor w/ 3 levels "1","2","3": 3 3 3 3 1 3 3 3 3 3 ...
##  $ Type.of.apartment                : Factor w/ 3 levels "1","2","3": 1 1 1 1 2 1 2 2 2 1 ...
##  $ No.of.Credits.at.this.Bank       : Factor w/ 3 levels "1","2","3": 1 2 1 2 2 2 2 1 2 1 ...
##  $ Occupation                       : Factor w/ 4 levels "1","2","3","4": 3 3 2 2 2 2 2 2 1 1 ...
##  $ No.of.dependents                 : Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 2 1 1 ...
##  $ Telephone                        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Foreign.Worker                   : int  1 1 1 2 2 2 2 2 1 1 ...
```

```r
summary(credit)
```

```
##  Creditability Account.Balance Duration.of.Credit..month.
##  0:300         1:274           Min.   : 4.0
##  1:700         2:269           1st Qu.:12.0
##                3: 63           Median :18.0
##                4:394           Mean   :20.9
##                                3rd Qu.:24.0
##                                Max.   :72.0
##
##  Payment.Status.of.Previous.Credit    Purpose    Credit.Amount
##  0: 40                             3      :280   Min.   :  250
##  1: 49                             0      :234   1st Qu.: 1366
##  2:530                             2      :181   Median : 2320
##  3: 88                             1      :103   Mean   : 3271
##  4:293                             9      : 97   3rd Qu.: 3972
##                                    6      : 50   Max.   :18424
##                                    (Other): 55
##  Value.Savings.Stocks Length.of.current.employment Instalment.per.cent
##  1:603                1: 62                        1:136
##  2:103                2:172                        2:231
##  3: 63                3:339                        3:157
##  4: 48                4:174                        4:476
##  5:183                5:253
##
##
##  Sex...Marital.Status Guarantors Duration.in.Current.address
##  1: 50                1:907      1:130
##  2:310                2: 41      2:308
##  3:548                3: 52      3:149
##  4: 92                           4:413
```

```
##
##
##
##  Most.valuable.available.asset  Age..years.    Concurrent.Credits
##  1:282                          Min.   :19.00   1:139
##  2:232                          1st Qu.:27.00   2: 47
##  3:332                          Median :33.00   3:814
##  4:154                          Mean   :35.54
##                                 3rd Qu.:42.00
##                                 Max.   :75.00
##
##  Type.of.apartment No.of.Credits.at.this.Bank Occupation No.of.dependents
##  1:179             1:633                      1: 22      1:845
##  2:714             2:333                      2:200      2:155
##  3:107             3: 34                      3:630
##                                               4:148
##
##
##
##  Telephone Foreign.Worker
##  1:596     Min.   :1.000
##  2:404     1st Qu.:1.000
##            Median :1.000
##            Mean   :1.037
##            3rd Qu.:1.000
##            Max.   :2.000
##
```

```r
# arrumando algumas variaveis...
# padronizando Credit.Amount
credit$Credit.Amount <- scale(credit$Credit.Amount)
# Foreign.Worker como fator
credit$Foreign.Worker <- as.factor(credit$Foreign.Worker)

# definicao do tamanho do conjunto teste
set.seed(1234)
teste.ind <- sample(1:nrow(credit), size = 600)

# separacao de dados treino e teste
cred.treino<-credit[teste.ind,]
cred.teste <- credit[-teste.ind,]
```

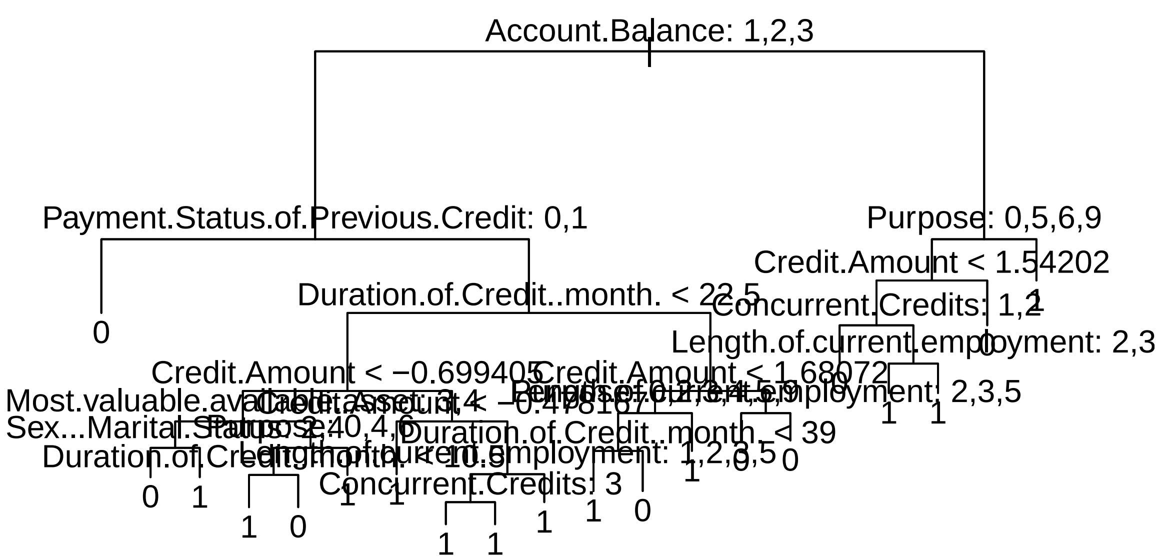
Primeiro método: Árvores de Classificacao

```r
library(tree)
set.seed(100)

# arvore com todas as variaveis consideradas
arvore <- tree(Creditability~.,cred.treino)

summary(arvore)
```

```
##
## Classification tree:
## tree(formula = Creditability ~ ., data = cred.treino)
## Variables actually used in tree construction:
## [1] "Account.Balance"                  "Payment.Status.of.Previous.Credit"
## [3] "Duration.of.Credit..month."       "Credit.Amount"
## [5] "Most.valuable.available.asset"    "Sex...Marital.Status"
## [7] "Purpose"                          "Length.of.current.employment"
## [9] "Concurrent.Credits"
## Number of terminal nodes:  20
## Residual mean deviance:  0.7665 = 444.6 / 580
## Misclassification error rate: 0.1817 = 109 / 600
```

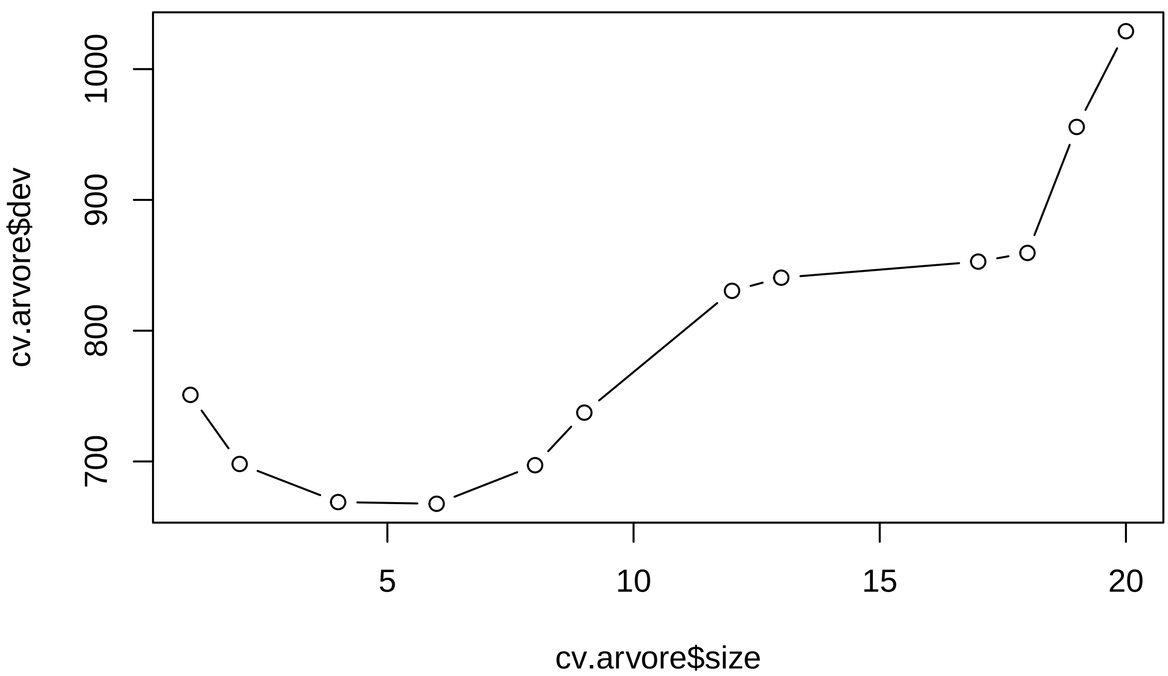
- As variáveis encontradas mais importantes são:
    1. Account.Balance
    2. Payment.Status.of.Previous.Credit
    3. Duration.of.Credit..month.
    4. Credit.Amount
    5. Most.valuable.available.asset
    6. Sex. . . Marital.Status
    7. Purpose
    8. Length.of.current.employment
    9. Concurrent.Credits

Visualizando a arvore com 20 nós



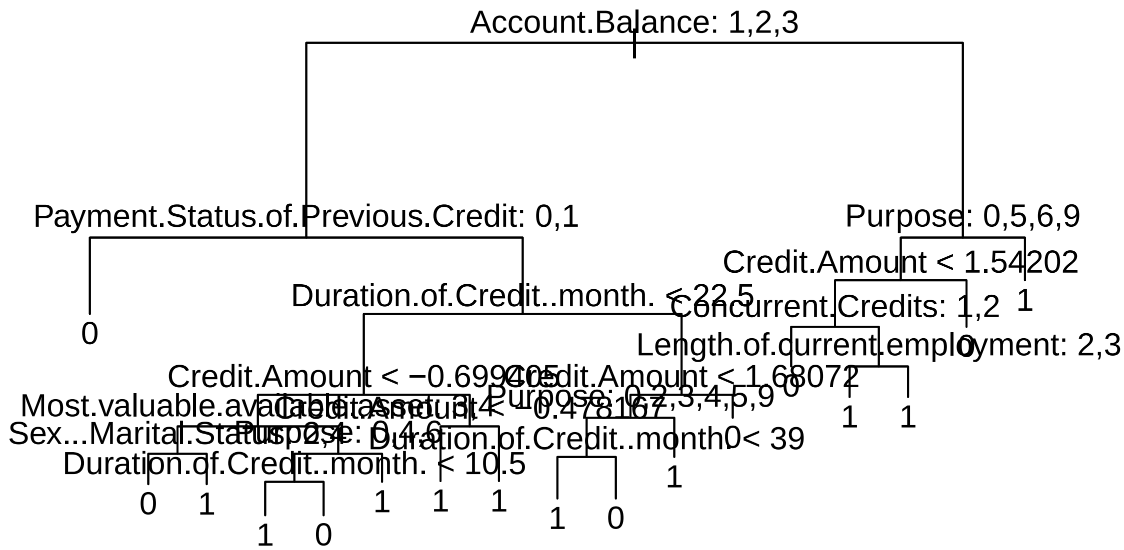
Realizando validacao cruzada para identificar o melhor tamanho de arvore

```
cv.arvore<-cv.tree(arvore)
plot(cv.arvore$size,cv.arvore$dev,type="b")
```



Poda da arvore com 14 nós terminais

```
arvore.poda <- prune.tree(arvore, best=14)
plot(arvore.poda)
text(arvore.poda, pretty = 0)
```



Modelo usado para predecir os novos dados

```r
pred.arvore <- predict(arvore.poda, newdata = cred.teste)
# matriz de confusao
(mat.conf <- table(round(pred.arvore[,2],0), cred.teste$Creditability))
```

```
##
##       0   1
##   0  36  37
##   1  79 248
```

```r
# taxa de acertos
acertos.tree <- (round((mat.conf[1,1] + mat.conf[2,2]) / sum(mat.conf),2))*100
paste0("Taxa de acertos Tree= ", acertos.tree," %")
```

```
## [1] "Taxa de acertos Tree= 71 %"
```

Segundo método: Random Forest

```r
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```r
# Bagging
set.seed(100)
# 20 é o total de variaveis, entao e bagging...
(ajuste.bagging<-randomForest(Creditability~., data=cred.treino, mtry=20))
```

Call: randomForest(formula = Creditability ~ ., data = cred.treino, mtry = 20) Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 20

```
    OOB estimate of  error rate: 23%
```

Confusion matrix: 0 1 class.error 0 93 92 0.4972973 1 46 369 0.1108434

vamos testar o metodo no conjunto de teste

```r
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```r
pred.bagging<-predict(ajuste.bagging,newdata=cred.teste)
conf1 <- (confusionMatrix(pred.bagging, cred.teste$Creditability))
#calculando o mse
acertos.bag <- postResample(pred.bagging, cred.teste$Creditability)
paste0("Taxa de acertos modelo Bagging= ", acertos.bag[1]*100," %")
```

```
## [1] "Taxa de acertos modelo Bagging= 74.75 %"
```

Modelo 2 com algumas variaveis selecionadas (8)

```
 library(knitr)
set.seed(100)
modelo.rf2 <- randomForest(Creditability ~ Account.Balance + Duration.of.Credit..month. +
                             Payment.Status.of.Previous.Credit + Purpose  +
                             Guarantors  + Duration.in.Current.address +
                             Most.valuable.available.asset + Sex...Marital.Status,
                           data = cred.treino, importance = TRUE, ntree = 300, nodesize = 1)

#vamos testar o novo metodo no conjunto de teste
pred.rf2 <-predict(modelo.rf2,newdata=cred.teste)
conf2 <- (confusionMatrix(pred.rf2, cred.teste$Creditability))

#calculando o mse
acertos.tree.8 <- postResample(pred.rf2, cred.teste$Creditability)
paste0("Taxa de acertos modelo Com 8 variaveis = ", acertos.tree.8[1]*100," %")
```

[1] "Taxa de acertos modelo Com 8 variaveis = 74 %"

Agora vamos fazer um modelo com as variaveis que o metodo da arvore (tree) achou mais importantes na predição:

```
library(knitr)
set.seed(100)
modelo.rf3 <- randomForest(Creditability ~ Account.Balance + Duration.of.Credit..month. +
                             Payment.Status.of.Previous.Credit + Purpose  + Most.valuable.available.asse
                             Sex...Marital.Status + Credit.Amount +
                             Length.of.current.employment + Concurrent.Credits,
                             data = cred.treino, importance = TRUE, ntree = 300, nodesize = 1)

#vamos testar o novo metodo no conjunto de teste
pred.rf3 <-predict(modelo.rf3,newdata=cred.teste)
conf3 <- (confusionMatrix(pred.rf3, cred.teste$Creditability))

#calculando o mse
acertos.tree.imp <- postResample(pred.rf3, cred.teste$Creditability)
paste0("Taxa de acertos modelo com variaveis + importantes= ", acertos.tree.imp[1]*100," %")
```

[1] "Taxa de acertos modelo com variaveis + importantes= 74.25 %"

Terceiro Método: regressão logistica

glm ajusta um modelo linear generalizado, no nosso caso como family=binomial, estamos ajustando um modelo de regressao logistica

```
library(ISLR)
ajuste.glm<-glm(Creditability ~.,
                data=cred.treino,family=binomial)
```

```r
summary(ajuste.glm)$coefficients
```

```
##                                        Estimate    Std. Error     z value
## (Intercept)                        -0.620401296   1.46847780 -0.42247918
## Account.Balance2                    0.390528961   0.29702733  1.31479133
## Account.Balance3                    0.422395845   0.47560355  0.88812594
## Account.Balance4                    1.696814322   0.32261512  5.25956234
## Duration.of.Credit..month.         -0.030514959   0.01283462 -2.37754998
## Payment.Status.of.Previous.Credit1 -0.060875619   0.84087734 -0.07239536
## Payment.Status.of.Previous.Credit2  1.314486525   0.70550747  1.86317874
## Payment.Status.of.Previous.Credit3  2.376481007   0.76612544  3.10194763
## Payment.Status.of.Previous.Credit4  2.435106880   0.73131200  3.32977835
## Purpose1                            1.991780456   0.51423958  3.87325387
## Purpose2                            0.905477830   0.36575604  2.47563329
## Purpose3                            1.118813944   0.33972615  3.29328175
## Purpose4                            0.413202815   0.87825517  0.47048150
## Purpose5                            1.003490955   0.80755414  1.24262995
## Purpose6                            0.242932293   0.50246490  0.48348112
## Purpose8                           14.966154343 495.66287467  0.03019422
## Purpose9                            0.579630687   0.44460721  1.30369160
## Purpose10                           2.584371969   1.24005691  2.08407530
## Credit.Amount                      -0.596012759   0.17835838 -3.34165837
## Value.Savings.Stocks2               0.148410051   0.37088838  0.40014748
## Value.Savings.Stocks3              -0.277390610   0.47106318 -0.58886073
## Value.Savings.Stocks4               1.367584023   0.67574708  2.02381048
## Value.Savings.Stocks5               1.007180534   0.35596277  2.82945470
## Length.of.current.employment2      -0.237636782   0.61776812 -0.38466987
## Length.of.current.employment3      -0.134970611   0.59730621 -0.22596552
## Length.of.current.employment4       0.827290419   0.64699655  1.27866279
## Length.of.current.employment5      -0.110470699   0.60597031 -0.18230382
## Instalment.per.cent2               -0.590523915   0.42133877 -1.40154184
## Instalment.per.cent3               -1.105982398   0.46622783 -2.37219299
## Instalment.per.cent4               -1.471097503   0.42061734 -3.49747236
## Sex...Marital.Status2               0.557708981   0.52583770  1.06061049
## Sex...Marital.Status3               1.299100265   0.51589999  2.51812421
## Sex...Marital.Status4               0.758269255   0.62473924  1.21373719
## Guarantors2                         0.008244206   0.59074070  0.01395571
## Guarantors3                         1.074538907   0.58754647  1.82885773
## Duration.in.Current.address2       -0.858043204   0.41081716 -2.08862550
## Duration.in.Current.address3       -0.979709393   0.44845011 -2.18465636
## Duration.in.Current.address4       -0.490882903   0.42070100 -1.16682133
## Most.valuable.available.asset2     -0.062605807   0.34791479 -0.17994580
## Most.valuable.available.asset3     -0.253550097   0.31995131 -0.79246462
## Most.valuable.available.asset4     -0.788996840   0.59221655 -1.33227758
## Age..years.                         0.014380078   0.01254287  1.14647391
## Concurrent.Credits2                 0.171858281   0.61103761  0.28125647
## Concurrent.Credits3                 0.449340253   0.32972054  1.36279121
## Type.of.apartment2                  0.302723510   0.32005430  0.94585047
## Type.of.apartment3                  0.830087062   0.67572987  1.22843032
## No.of.Credits.at.this.Bank2        -0.835809522   0.35228363 -2.37254716
## No.of.Credits.at.this.Bank3        -0.789459525   0.74210976 -1.06380426
## Occupation2                        -1.273204735   1.00470125 -1.26724708
## Occupation3                        -1.118504588   0.98592311 -1.13447446
```

```
## Occupation4                          -1.088037337   1.01334806 -1.07370545
## No.of.dependents2                    -0.296394972   0.32973137 -0.89889830
## Telephone2                            0.228882956   0.28630881  0.79942688
## Foreign.Worker2                       0.805263035   0.84706583  0.95064988
##                                          Pr(>|z|)
## (Intercept)                          6.726753e-01
## Account.Balance2                     1.885800e-01
## Account.Balance3                     3.744730e-01
## Account.Balance4                     1.443987e-07
## Duration.of.Credit..month.           1.742808e-02
## Payment.Status.of.Previous.Credit1 9.422873e-01
## Payment.Status.of.Previous.Credit2 6.243712e-02
## Payment.Status.of.Previous.Credit3 1.922520e-03
## Payment.Status.of.Previous.Credit4 8.691514e-04
## Purpose1                             1.073919e-04
## Purpose2                             1.330001e-02
## Purpose3                             9.902519e-04
## Purpose4                             6.380110e-01
## Purpose5                             2.140042e-01
## Purpose6                             6.287542e-01
## Purpose8                             9.759122e-01
## Purpose9                             1.923388e-01
## Purpose10                            3.715332e-02
## Credit.Amount                        8.327949e-04
## Value.Savings.Stocks2                6.890479e-01
## Value.Savings.Stocks3                5.559547e-01
## Value.Savings.Stocks4                4.298966e-02
## Value.Savings.Stocks5                4.662740e-03
## Length.of.current.employment2        7.004820e-01
## Length.of.current.employment3        8.212282e-01
## Length.of.current.employment4        2.010158e-01
## Length.of.current.employment5        8.553443e-01
## Instalment.per.cent2                 1.610521e-01
## Instalment.per.cent3                 1.768285e-02
## Instalment.per.cent4                 4.696894e-04
## Sex...Marital.Status2                2.888670e-01
## Sex...Marital.Status3                1.179817e-02
## Sex...Marital.Status4                2.248481e-01
## Guarantors2                          9.888653e-01
## Guarantors3                          6.742092e-02
## Duration.in.Current.address2         3.674145e-02
## Duration.in.Current.address3         2.891405e-02
## Duration.in.Current.address4         2.432825e-01
## Most.valuable.available.asset2       8.571951e-01
## Most.valuable.available.asset3       4.280898e-01
## Most.valuable.available.asset4       1.827690e-01
## Age..years.                          2.515991e-01
## Concurrent.Credits2                  7.785137e-01
## Concurrent.Credits3                  1.729483e-01
## Type.of.apartment2                   3.442249e-01
## Type.of.apartment3                   2.192855e-01
## No.of.Credits.at.this.Bank2          1.766591e-02
## No.of.Credits.at.this.Bank3          2.874174e-01
## Occupation2                          2.050670e-01
```

```
## Occupation3                        2.565956e-01
## Occupation4                        2.829547e-01
## No.of.dependents2                  3.687068e-01
## Telephone2                         4.240429e-01
## Foreign.Worker2                    3.417821e-01
```

```r
# funcao predict calcula a probabilidade de Creditability = 1, nao dar calote
probs.glm<-predict(ajuste.glm, type="response")
probs.glm <- ifelse(probs.glm >0.5,"1","0")

#calculo da matriz de confusao
mat_conf<-table(probs.glm, cred.treino$Creditability)
#calculo da taxa de acerto do modelo
acertos.glm <- round(((mat_conf[1,1]+ mat_conf[2,2])/sum(mat_conf)*100),2)
paste0("Taxa de acertos GLM treino = ", acertos.glm, " %")
```

```
## [1] "Taxa de acertos GLM treino = 80.33 %"
```

```r
# vamos ver no conjunto de teste
probs.glm.teste <- predict(ajuste.glm, newdata = cred.teste, type="response")
pred.glm <- ifelse(probs.glm.teste >0.5,"1","0")

# matriz de confusao e taxa de acertos...
mat_conf<-table(pred.glm, cred.teste$Creditability)

acertos.glm.teste <- ((mat_conf[1,1]+mat_conf[2,2])/sum(mat_conf))*100
paste0("Taxa de acertos GLM teste = ", acertos.glm.teste," %")
```

```
## [1] "Taxa de acertos GLM teste = 77.5 %"
```

Quarto método Boosting

Agora vamos utilizar o metodo de boosting utilizaremos o pacote gbm como e um problema de regressao devemos colocar como argumento distribution="gaussian", para determinar o numero de arvores a ser considerado utilizamos o argumento n.trees, e o argumento interaction.depth determina o numero maximo de nos das arvores

```r
library(gbm)
```

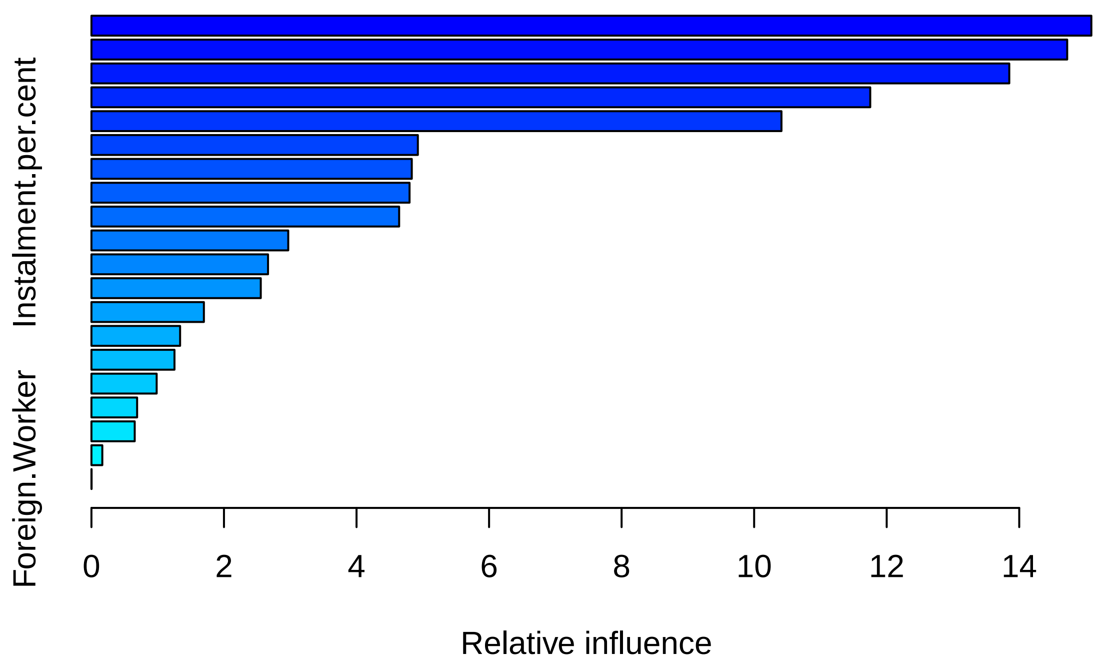
```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.1
```

```r
set.seed(100)
ajuste.boosting<-gbm(Creditability~.,data=cred.treino,
                     distribution = "gaussian",
                     n.trees=5000,
                     interaction.depth = 4)
summary(ajuste.boosting)
```




```
##                                                                   var
## Credit.Amount                                           Credit.Amount
## Account.Balance                                       Account.Balance
## Purpose                                                       Purpose
## Duration.of.Credit..month.                 Duration.of.Credit..month.
## Payment.Status.of.Previous.Credit Payment.Status.of.Previous.Credit
## Age..years.                                               Age..years.
## Value.Savings.Stocks                             Value.Savings.Stocks
## Instalment.per.cent                               Instalment.per.cent
## Length.of.current.employment             Length.of.current.employment
## Sex...Marital.Status                             Sex...Marital.Status
## Duration.in.Current.address               Duration.in.Current.address
```

```
## Most.valuable.available.asset         Most.valuable.available.asset
## Concurrent.Credits                                Concurrent.Credits
## Guarantors                                                Guarantors
## Occupation                                                Occupation
## Type.of.apartment                                  Type.of.apartment
## No.of.Credits.at.this.Bank                No.of.Credits.at.this.Bank
## Telephone                                                  Telephone
## No.of.dependents                                    No.of.dependents
## Foreign.Worker                                        Foreign.Worker
##                                       rel.inf
## Credit.Amount                     15.088531892
## Account.Balance                   14.724127500
## Purpose                           13.849870052
## Duration.of.Credit..month.        11.751999002
## Payment.Status.of.Previous.Credit 10.412150241
## Age..years.                        4.925832722
## Value.Savings.Stocks               4.833819632
## Instalment.per.cent                4.800090207
## Length.of.current.employment       4.643274301
## Sex...Marital.Status               2.969461613
## Duration.in.Current.address        2.664487802
## Most.valuable.available.asset      2.555053018
## Concurrent.Credits                 1.696929735
## Guarantors                         1.337943803
## Occupation                         1.253018628
## Type.of.apartment                  0.984356049
## No.of.Credits.at.this.Bank         0.689283104
## Telephone                          0.652895141
## No.of.dependents                   0.164333722
## Foreign.Worker                     0.002541836
```

```
#vamos testar o metodo no conjunto de teste
pred.boosting<-predict(ajuste.boosting, newdata=cred.teste, n.trees = 5000)
summary(pred.boosting)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.042   1.559   1.744   1.701   1.875   2.061
```

```
real <- cred.teste$Creditability
predito <- (round((pred.boosting),0)-1)

#calculando o mse

mat_conf<-table(predito,real)
acertos.boosting <- ((mat_conf[1,1]+mat_conf[2,2])/sum(mat_conf))*100
paste0("Taxa de acertos Boosting = ", acertos.boosting," %")
```

```
## [1] "Taxa de acertos Boosting = 76.5 %"
```

Resultados finais

```
## [1] "Taxa de acertos Tree                      = 71 %"
```

```
## [1] "Taxa de acertos modelo Bagging            = 74.75 %"

## [1] "Taxa de acertos modelo Com 8 variaveis    = 74 %"

## [1] "Taxa de acertos modelo com variaveis imp. = 74.25 %"

## [1] "Taxa de acertos GLM                       = 77.5 %"

## [1] "Taxa de acertos Boosting                  = 76.5 %"
```

Fim do exercício

```
## [1] "Tue Nov 22 14:35:15 2016"
```