Regressão

Labdaps

Pacotes

```
library(tidyverse)
library(caret)
library(hydroGOF)
library(lattice)
library(psych)
library(reshape)
library(reshape2)
library(e1071)
library(corrplot)
```

Importar banco de dados

```
banco <- read.table ("dados_expecvida2.csv", header = T ,sep=';', dec='.',colClass=c(rep("numeric",24)))
str(banco)
```

```
## 'data.frame':    5565 obs. of  24 variables:
##  $ cod_municipio        : num  1100015 1100023 1100031 1100049 1100056 ...
##  $ PopResid             : num  24392 90353 6313 78574 17029 ...
##  $ ExpecVida            : num  70.8 73.4 70.4 74.3 72.9 ...
##  $ idoso                : num  8.79 6.67 10.09 8.1 9.97 ...
##  $ mulher               : num  48.1 49.6 48.3 50.2 49.8 ...
##  $ nascimentopercapta   : num  16.7 18.1 15.1 15.7 14.1 ...
##  $ casado               : num  38.6 30.1 43.9 42.2 41.7 ...
##  $ evangelico           : num  26.9 33.7 28.6 39 29.9 ...
##  $ deficiente           : num  23.9 21.6 23.8 25.3 25 ...
##  $ densidadepop         : num  3.45 20.41 4.8 20.72 6.12 ...
##  $ geladeira            : num  90.7 94.6 93.7 97.2 97.3 ...
##  $ moradorfavela        : num  0.09 0.1 0 0.03 0 0.13 0 0.15 0.03 0 ...
##  $ pavimentacao         : num  0.157 0.635 0.241 0.326 0.137 ...
##  $ brancos              : num  42.4 37.1 46.7 43.3 45.9 ...
##  $ nvelsuperiorconcludo : num  5.5 8.16 4.93 8.74 7.01 7.91 4.62 5.03 5.74 7.23 ...
##  $ conclusoensinomdio   : num  13.8 23.4 15.5 19.4 17 ...
##  $ ResideMenos10anos    : num  6.51 9.19 7.05 7.94 12.47 ...
##  $ resideoutranacionalid: num  0 0.12 0.05 0.05 0.11 0.16 0.06 1.58 0.03 2.27 ...
##  $ taxadedesemprego     : num  2.99 3.23 1.58 3.95 3.51 ...
##  $ taxatrabinfantil     : num  11.3 10.2 13 14.1 12.9 ...
##  $ horastrabalho        : num  32.9 32.8 21.7 25.6 36.6 ...
##  $ Gini                 : num  0.589 0.55 0.517 0.589 0.515 ...
##  $ CobBFamilia          : num  83.5 94.9 71.4 57.7 70 ...
##  $ traboutromun         : num  2.3 3.93 3.48 2.57 5.44 3.03 1.36 0.39 2.86 4.63 ...
```

Seleção de municípios com mais de 10.000 habitantes

```
#filtrar mun > 10.000 habitantes
banco_filtrado <- banco %>%
                  filter(PopResid > 10000) %>%
                  select(-PopResid)
summary(banco_filtrado)
```

```
##  cod_municipio       ExpecVida         idoso            mulher
##  Min.   :1100015   Min.   :65.55   Min.   : 2.627   Min.   :38.71
##  1st Qu.:2501881   1st Qu.:70.85   1st Qu.: 9.107   1st Qu.:49.13
##  Median :3109808   Median :73.26   Median :11.133   Median :50.04
##  Mean   :3116082   Mean   :72.98   Mean   :11.049   Mean   :49.88
##  3rd Qu.:3548527   3rd Qu.:75.22   3rd Qu.:13.093   3rd Qu.:50.82
##  Max.   :5300108   Max.   :78.64   Max.   :20.404   Max.   :54.24
##  nascimentopercapta     casado        evangelico      deficiente
##  Min.   : 5.97      Min.   : 6.55   Min.   : 2.03   Min.   :10.19
##  1st Qu.:12.73      1st Qu.:29.04   1st Qu.:11.04   1st Qu.:21.32
##  Median :14.54      Median :35.12   Median :17.26   Median :24.07
##  Mean   :14.98      Mean   :34.26   Mean   :18.30   Mean   :24.19
##  3rd Qu.:16.78      3rd Qu.:40.61   3rd Qu.:24.45   3rd Qu.:26.88
##  Max.   :35.17      Max.   :53.42   Max.   :77.26   Max.   :38.79
##   densidadepop        geladeira      moradorfavela     pavimentacao
##  Min.   :    0.20   Min.   : 27.98   Min.   :0.0000   Min.   :0.01085
##  1st Qu.:   16.11   1st Qu.: 80.47   1st Qu.:0.0000   1st Qu.:0.29253
##  Median :   36.06   Median : 91.57   Median :0.0300   Median :0.45825
##  Mean   :  177.01   Mean   : 87.49   Mean   :0.1675   Mean   :0.48698
##  3rd Qu.:   87.29   3rd Qu.: 97.56   3rd Qu.:0.1700   3rd Qu.:0.68435
##  Max.   :13024.60   Max.   :100.00   Max.   :7.4000   Max.   :0.99351
##     brancos       nvelsuperiorconcludo conclusoensinomdio
##  Min.   : 4.296   Min.   : 0.410       Min.   : 1.86
##  1st Qu.:24.495   1st Qu.: 3.110       1st Qu.:12.49
##  Median :37.854   Median : 5.070       Median :16.64
##  Mean   :43.604   Mean   : 6.027       Mean   :17.58
##  3rd Qu.:62.615   3rd Qu.: 7.982       3rd Qu.:21.93
##  Max.   :96.108   Max.   :33.840       Max.   :47.47
##  ResideMenos10anos  resideoutranacionalid taxadedesemprego
##  Min.   : 0.06705   Min.   :0.00000       Min.   : 0.3477
##  1st Qu.: 2.86014   1st Qu.:0.00000       1st Qu.: 2.9364
##  Median : 4.43156   Median :0.02000       Median : 3.9163
##  Mean   : 5.63131   Mean   :0.08684       Mean   : 4.2003
##  3rd Qu.: 6.89445   3rd Qu.:0.09000       3rd Qu.: 5.1607
##  Max.   :45.74883   Max.   :5.73000       Max.   :16.9871
##  taxatrabinfantil horastrabalho        Gini         CobBFamilia
##  Min.   : 0.84    Min.   : 3.45   Min.   :0.3051   Min.   :  0.00
##  1st Qu.: 7.33    1st Qu.:22.23   1st Qu.:0.4796   1st Qu.: 60.54
##  Median :10.59    Median :28.00   Median :0.5182   Median : 77.04
##  Mean   :11.87    Mean   :28.09   Mean   :0.5196   Mean   : 72.49
##  3rd Qu.:15.04    3rd Qu.:33.91   3rd Qu.:0.5588   3rd Qu.: 86.66
##  Max.   :63.22    Max.   :63.60   Max.   :0.8082   Max.   :100.00
##   traboutromun
##  Min.   : 0.00
##  1st Qu.: 3.99
##  Median : 6.98
##  Mean   :10.66
##  3rd Qu.:13.09
##  Max.   :65.04
```

Separar conjunto de dados em treinamento e teste

```
#Divisão aleatória do conjunto de dados original
set.seed(1)
train_obs <- sample(nrow(banco_filtrado), 0.7*nrow(banco_filtrado))

#*********************************************************************
#Preditores conjunto de treinamento
X <- banco_filtrado[train_obs, ]
X <- X[,-which(names(X) %in% (c("ExpecVida")))]

#Resposta do conjunto de treinamento
Y <- banco_filtrado$ExpecVida[train_obs]

#Banco treino
data_train <- cbind(Y,X)

#*********************************************************************
#Preditores conjunto de teste
X_holdout <- banco_filtrado[-train_obs, ]
X_holdout <- X_holdout[,-which(names(X_holdout) %in% (c("ExpecVida")))]

#Resposta do conjunto de teste
Y_holdout <- banco_filtrado$ExpecVida[-train_obs]

#Banco teste
data_test <- cbind(Y_holdout,X_holdout)
```

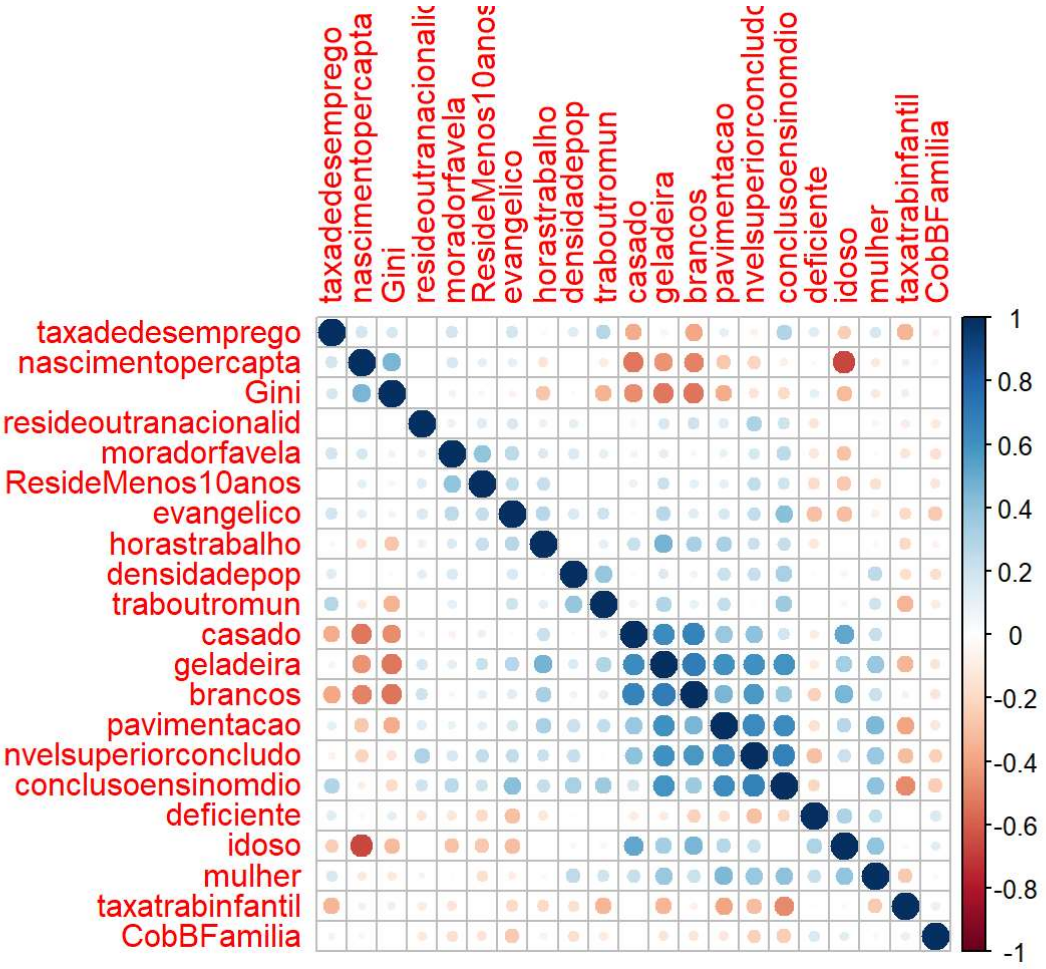
Pré-processamento dos dados de treinamento

- Variaveis quantitativas

```
#Padronizar variáveis quantitativas
nums <- sapply(data_train, is.numeric)
quantis<-data_train[,nums]
quantis_filter <- select(quantis, -c(cod_municipio))
scale_variables <- preProcess(quantis_filter, method = c("center", "scale"))
quantis_train_scale<-predict(scale_variables,quantis_filter)
```

```
#Investigar presença de colinearidade entre os preditores
quantis_predictors<-select(quantis_train_scale,-Y)
correlacao<-cor(quantis_predictors)

#visualizar correlação
corrplot::corrplot(correlacao, method="circle", order="hclust")
```




```
#Analisar correlações altas
corrAlta<-correlacao
corrAlta[lower.tri(corrAlta,diag=TRUE)]=NA
corrAlta[abs(corrAlta) < 0.7] <- NA
corrAlta <- na.omit(melt(corrAlta))
corrAlta[order(-abs(corrAlta$value)),]
```

```
## [1] X1    X2    value
## <0 rows> (or 0-length row.names)
```

```
#Avaliar assimetria
skewvalues= apply(quantis_train_scale,2,skewness)
skewvalues[order(-abs(skewvalues))]
```

```
## resideoutranacionalid          densidadepop         moradorfavela
##            11.1857994            10.4697730             4.9112948
##     ResideMenos10anos          traboutromun  nvelsuperiorconcludo
##             2.7895941             2.2743308             1.6089746
##      taxatrabinfantil             geladeira      taxadedesemprego
##             1.5378183            -1.2526187             1.1349204
##                mulher    nascimentopercapta           CobBFamilia
##            -1.0268658             0.9750582            -0.9091830
##            evangelico    conclusoensinomdio                casado
##             0.7838433             0.5627441            -0.5020415
##               brancos                     Y                 Gini
##             0.4797972            -0.2767508             0.2414956
##          pavimentacao         horastrabalho            deficiente
##             0.2084901             0.1504551             0.1423415
##                idoso
##            -0.1332964
```

```
#Filtrar preditores com variância proxima de zero
nearZeroVar(quantis_filter)
```

```
## integer(0)
```

```
#Banco final
data_train_final<-quantis_train_scale
summary(data_train_final)
```

```
##        Y                idoso              mulher
##  Min.   :-2.71290   Min.   :-2.96449   Min.   :-7.8020
##  1st Qu.:-0.75708   1st Qu.:-0.66753   1st Qu.:-0.5443
##  Median : 0.09542   Median : 0.03048   Median : 0.1154
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.80249   3rd Qu.: 0.71868   3rd Qu.: 0.6755
##  Max.   : 2.04579   Max.   : 2.82285   Max.   : 2.8106
##  nascimentopercapta     casado           evangelico        deficiente
##  Min.   :-2.7333    Min.   :-3.3956   Min.   :-1.7521   Min.   :-3.42335
##  1st Qu.:-0.6810    1st Qu.:-0.6318   1st Qu.:-0.7779   1st Qu.:-0.70082
##  Median :-0.1384    Median : 0.1010   Median :-0.1201   Median :-0.02372
##  Mean   : 0.0000    Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.5535    3rd Qu.: 0.7767   3rd Qu.: 0.6388   3rd Qu.: 0.65277
##  Max.   : 5.9486    Max.   : 2.3476   Max.   : 6.4387   Max.   : 3.57972
##   densidadepop       geladeira       moradorfavela       pavimentacao
##  Min.   :-0.2219   Min.   :-4.6451   Min.   :-0.44769   Min.   :-1.9420
##  1st Qu.:-0.2022   1st Qu.:-0.5637   1st Qu.:-0.44769   1st Qu.:-0.8066
##  Median :-0.1769   Median : 0.3370   Median :-0.36582   Median :-0.1067
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.1127   3rd Qu.: 0.8194   3rd Qu.: 0.01625   3rd Qu.: 0.8101
##  Max.   :16.1609   Max.   : 1.0190   Max.   :10.44139   Max.   : 2.0758
##     brancos         nvelsuperiorconcludo conclusoensinomdio
##  Min.   :-1.7167   Min.   :-1.4485      Min.   :-1.9956
##  1st Qu.:-0.8414   1st Qu.:-0.7483      1st Qu.:-0.7730
##  Median :-0.2544   Median :-0.2509      Median :-0.1343
##  Mean   : 0.0000   Mean   : 0.0000      Mean   : 0.0000
##  3rd Qu.: 0.8494   3rd Qu.: 0.4975      3rd Qu.: 0.6701
##  Max.   : 2.2892   Max.   : 7.1342      Max.   : 4.5832
##  ResideMenos10anos resideoutranacionalid taxadedesemprego
##  Min.   :-1.2405   Min.   :-0.33964      Min.   :-2.0861
##  1st Qu.:-0.6114   1st Qu.:-0.33964      1st Qu.:-0.6787
##  Median :-0.2561   Median :-0.26060      Median :-0.1509
##  Mean   : 0.0000   Mean   : 0.00000      Mean   : 0.0000
##  3rd Qu.: 0.2635   3rd Qu.: 0.01602      3rd Qu.: 0.5213
##  Max.   : 8.1277   Max.   :22.30399      Max.   : 6.9222
##  taxatrabinfantil  horastrabalho            Gini          CobBFamilia
##  Min.   :-1.7300   Min.   :-2.642516   Min.   :-2.9325   Min.   :-3.6671
##  1st Qu.:-0.7197   1st Qu.:-0.629552   1st Qu.:-0.6593   1st Qu.:-0.5943
##  Median :-0.2097   Median :-0.001537   Median :-0.0282   Median : 0.2322
##  Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4919   3rd Qu.: 0.625943   3rd Qu.: 0.6471   3rd Qu.: 0.7108
##  Max.   : 8.2893   Max.   : 3.792765   Max.   : 4.7206   Max.   : 1.3841
##   traboutromun
##  Min.   :-0.9984
##  1st Qu.:-0.6207
##  Median :-0.3383
##  Mean   : 0.0000
##  3rd Qu.: 0.2003
##  Max.   : 5.1478
```

Aplicar alterações do banco treino no banco teste

```
#Padronizar dados de teste com base nos parâmetros estimados nos dados de treinamento
nums_t <- sapply(data_test, is.numeric)
quanti_t<-data_test[,nums]
quantis_filter_test <- select(quanti_t, -c(cod_municipio))
names(quantis_filter_test)[1]<-"Y"
quantis_test_scale<-predict(scale_variables, quantis_filter_test)
```

```
#Banco final
data_test_final<-quantis_test_scale
names(data_test_final)[1]<-"Y_holdout"
summary(data_test_final)
```

```
##    Y_holdout            idoso               mulher        
##  Min.   :-2.59657   Min.   :-2.971608   Min.   :-7.64099  
##  1st Qu.:-0.85250   1st Qu.:-0.733067   1st Qu.:-0.45159  
##  Median : 0.08633   Median :-0.006639   Median : 0.16444  
##  Mean   :-0.04118   Mean   :-0.023788   Mean   : 0.05329  
##  3rd Qu.: 0.79613   3rd Qu.: 0.693465   3rd Qu.: 0.67722  
##  Max.   : 2.03852   Max.   : 3.285784   Max.   : 3.06961  
##  nascimentopercapta      casado           evangelico       
##  Min.   :-2.381663   Min.   :-3.19584   Min.   :-1.72164  
##  1st Qu.:-0.696203   1st Qu.:-0.65416   1st Qu.:-0.72650  
##  Median :-0.132363   Median : 0.11322   Median :-0.04766  
##  Mean   :-0.004633   Mean   :-0.00203   Mean   : 0.06328  
##  3rd Qu.: 0.511052   3rd Qu.: 0.78256   3rd Qu.: 0.77001  
##  Max.   : 6.118388   Max.   : 2.18834   Max.   : 2.90890  
##    deficiente         densidadepop         geladeira        
##  Min.   :-3.44299   Min.   :-0.221873   Min.   :-4.80693  
##  1st Qu.:-0.73213   1st Qu.:-0.200845   1st Qu.:-0.54792  
##  Median :-0.06792   Median :-0.176509   Median : 0.34069  
##  Mean   :-0.01891   Mean   : 0.001701   Mean   : 0.02281  
##  3rd Qu.: 0.66628   3rd Qu.:-0.111937   3rd Qu.: 0.82483  
##  Max.   : 3.07880   Max.   :12.874344   Max.   : 1.01088  
##  moradorfavela       pavimentacao        brancos          
##  Min.   :-0.44769   Min.   :-1.9229   Min.   :-1.708880  
##  1st Qu.:-0.44769   1st Qu.:-0.7464   1st Qu.:-0.820956  
##  Median :-0.36582   Median :-0.1247   Median :-0.244340  
##  Mean   : 0.03118   Mean   : 0.0157   Mean   :-0.005431  
##  3rd Qu.: 0.01625   3rd Qu.: 0.8179   3rd Qu.: 0.774795  
##  Max.   :19.74760   Max.   : 2.0189   Max.   : 2.211518  
##  nvelsuperiorconcludo conclusoensinomdio ResideMenos10anos
##  Min.   :-1.40746     Min.   :-2.39814   Min.   :-1.17600  
##  1st Qu.:-0.76561     1st Qu.:-0.76683   1st Qu.:-0.60911  
##  Median :-0.25213     Median :-0.14041   Median :-0.25743  
##  Mean   :-0.02143     Mean   : 0.02488   Mean   : 0.04705  
##  3rd Qu.: 0.47829     3rd Qu.: 0.69610   3rd Qu.: 0.32617  
##  Max.   : 6.62847     Max.   : 3.48879   Max.   : 9.05992  
##  resideoutranacionalid taxadedesemprego   taxatrabinfantil  
##  Min.   :-0.33964      Min.   :-1.91274   Min.   :-1.79469  
##  1st Qu.:-0.33964      1st Qu.:-0.69428   1st Qu.:-0.78638  
##  Median :-0.26060      Median :-0.16188   Median :-0.23231  
##  Mean   : 0.01179      Mean   :-0.00134   Mean   :-0.03718  
##  3rd Qu.: 0.01602      3rd Qu.: 0.50707   3rd Qu.: 0.51777  
##  Max.   :13.25444      Max.   : 5.08336   Max.   : 3.87612  
##  horastrabalho           Gini           CobBFamilia       
##  Min.   :-2.56441   Min.   :-3.52060   Min.   :-3.66707  
##  1st Qu.:-0.63704   1st Qu.:-0.66418   1st Qu.:-0.66817  
##  Median :-0.03149   Median :-0.03065   Median : 0.19710  
##  Mean   :-0.02004   Mean   :-0.02397   Mean   :-0.01837  
##  3rd Qu.: 0.58876   3rd Qu.: 0.60574   3rd Qu.: 0.69514  
##  Max.   : 3.40761   Max.   : 3.54939   Max.   : 1.38413  
##   traboutromun      
##  Min.   :-0.99085  
##  1st Qu.:-0.62326  
##  Median :-0.34023  
##  Mean   : 0.03044  
##  3rd Qu.: 0.29905  
##  Max.   : 5.06461
```

Treinamento de modelos preditivos

Técnica de reamostragem (para evitar sobreajuste)

```
control <- trainControl(method="cv",
                        number=10,
                        savePredictions=TRUE)
```

Escolha de hiperparâmetros

Modelos lineares

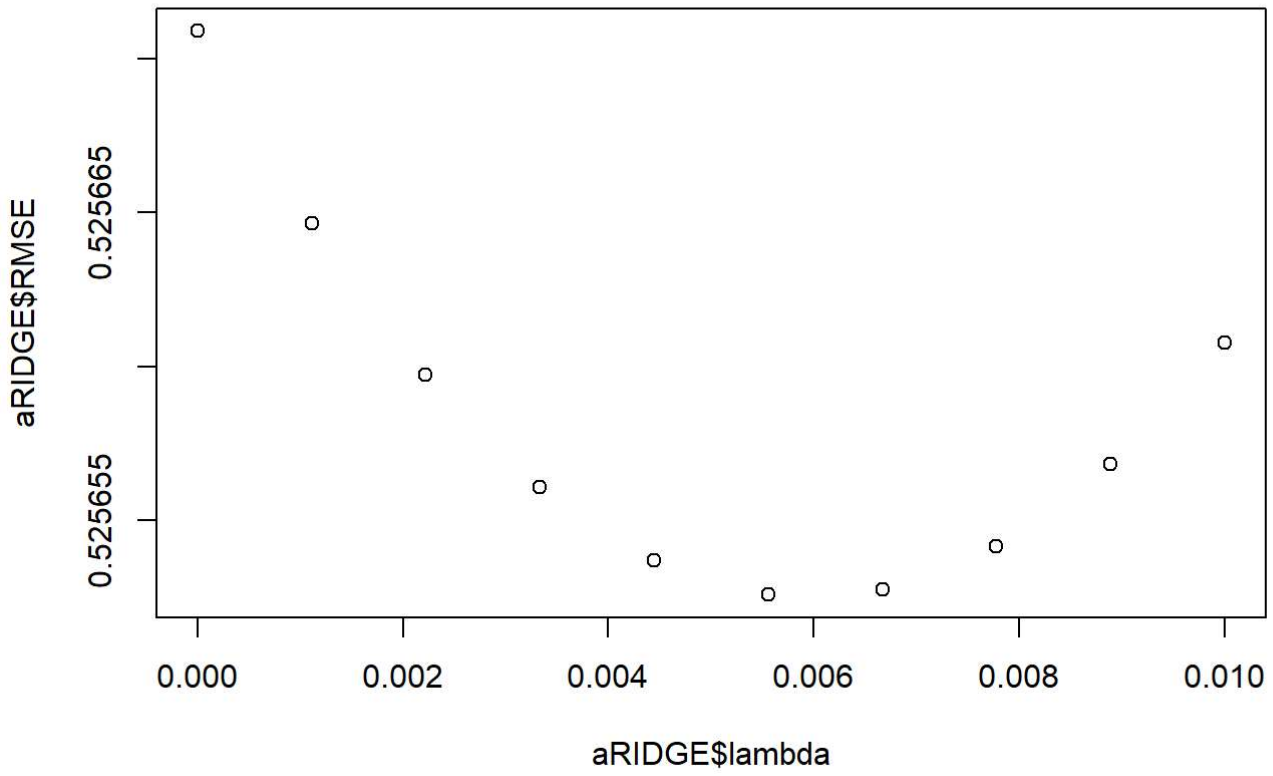
- Regressão Rigde

```
ridgeGrid <- data.frame(.lambda = seq(0, .01, length = 10))

set.seed(2712)
ridgeRegFit <- train(Y~., data = data_train_final,
                     method = "ridge",
                     tuneGrid = ridgeGrid,
                     trControl = control)
ridgeRegFit
```

```
## Ridge Regression
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   lambda       RMSE       Rsquared   MAE
##   0.000000000  0.5256709  0.7248088  0.4176031
##   0.001111111  0.5256647  0.7248233  0.4176112
##   0.002222222  0.5256597  0.7248374  0.4176200
##   0.003333333  0.5256561  0.7248510  0.4176298
##   0.004444444  0.5256537  0.7248642  0.4176406
##   0.005555556  0.5256526  0.7248768  0.4176519
##   0.006666667  0.5256528  0.7248891  0.4176632
##   0.007777778  0.5256542  0.7249009  0.4176754
##   0.008888889  0.5256568  0.7249122  0.4176886
##   0.010000000  0.5256608  0.7249231  0.4177028
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was lambda = 0.005555556.
```

```
aRIDGE<-ridgeRegFit$results
plot(aRIDGE$lambda,aRIDGE$RMSE)
```




```
ridgeRegFit$bestTune
```

```
##        lambda
## 6 0.005555556
```

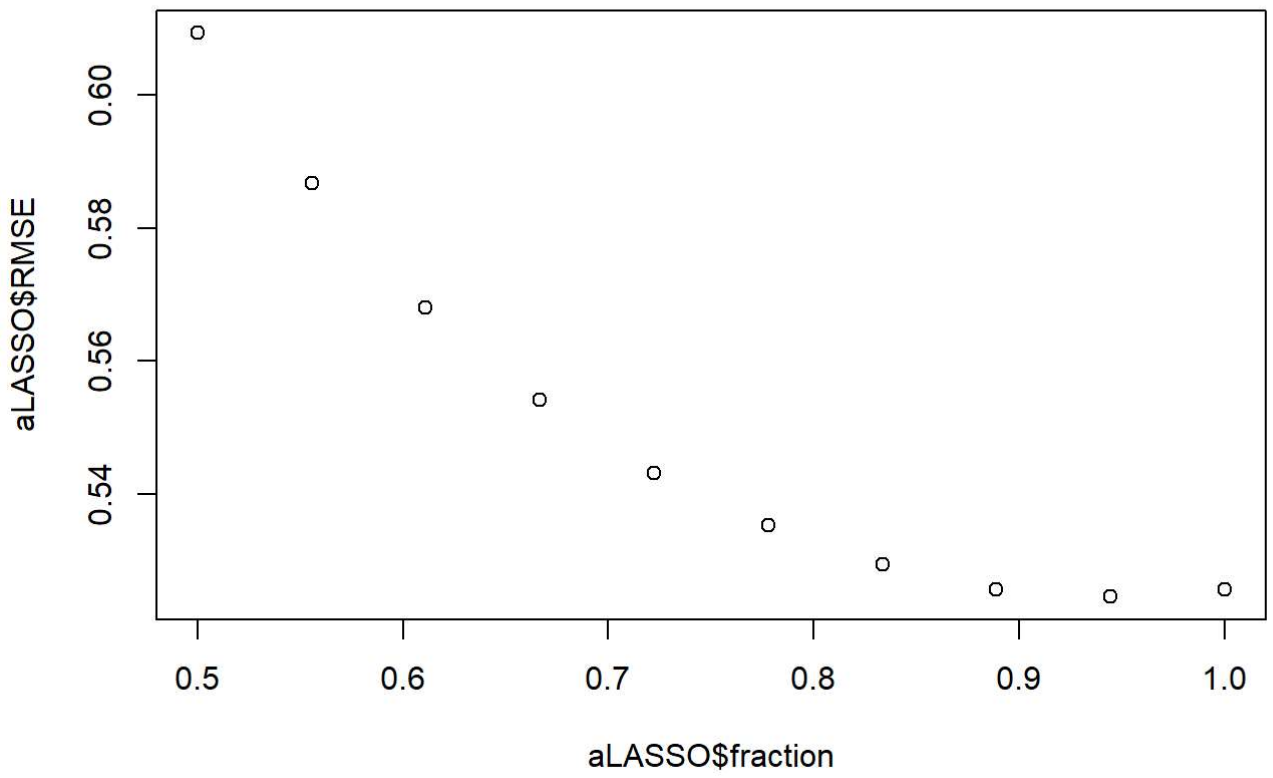
- Lasso

```
LassoGrid <- expand.grid(.fraction = seq(.5, 1, length = 10))

set.seed(2712)
LassoFit <- train(Y~., data = data_train_final,
                  method = "lasso",
                  tuneGrid = LassoGrid,
                  trControl = control)
LassoFit
```

```
## The lasso
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   fraction   RMSE       Rsquared   MAE
##   0.5000000  0.6092877  0.6796978  0.4750162
##   0.5555556  0.5866830  0.6923249  0.4572066
##   0.6111111  0.5680750  0.7009238  0.4430106
##   0.6666667  0.5542436  0.7073555  0.4334395
##   0.7222222  0.5431663  0.7140191  0.4262714
##   0.7777778  0.5353579  0.7190018  0.4215561
##   0.8333333  0.5294576  0.7230758  0.4180236
##   0.8888889  0.5257595  0.7254318  0.4164641
##   0.9444444  0.5246044  0.7259434  0.4163921
##   1.0000000  0.5256709  0.7248088  0.4176031
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was fraction = 0.9444444.
```

```
aLASSO<-LassoFit$results
plot(aLASSO$fraction,aLASSO$RMSE)
```



```
LassoFit$bestTune
```

```
##    fraction
## 9 0.9444444
```

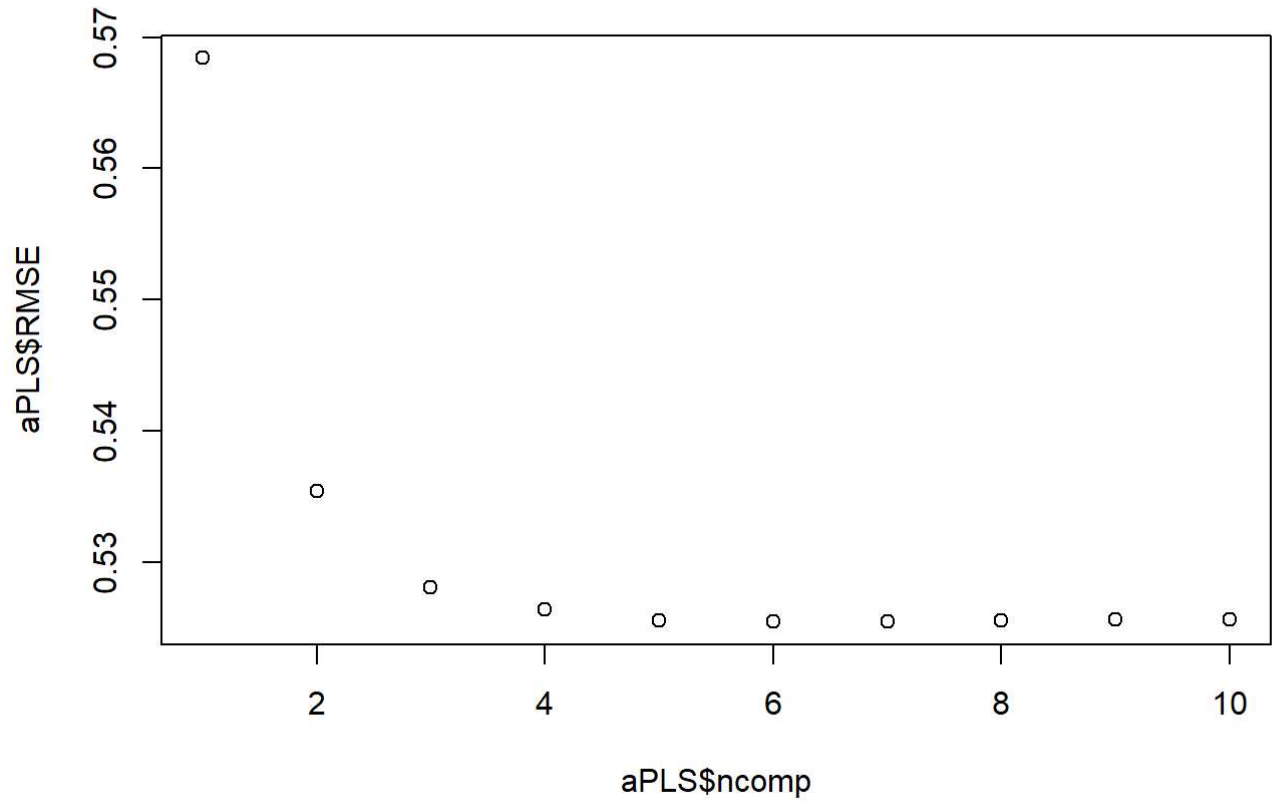
- Mínimos quadrados parciais

```
plsGrid <- expand.grid(.ncomp = c(1:10))

set.seed(2712)
plsFit<-train(Y~., data = data_train_final,
              method="pls",
              tuneGrid = plsGrid,
              trControl=control)
plsFit
```

```
## Partial Least Squares
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   MAE
##    1     0.5684041  0.6797384  0.4473528
##    2     0.5354225  0.7146769  0.4220530
##    3     0.5281280  0.7221157  0.4180227
##    4     0.5263982  0.7240647  0.4181676
##    5     0.5256291  0.7248772  0.4173392
##    6     0.5255038  0.7249951  0.4175216
##    7     0.5255366  0.7249387  0.4174615
##    8     0.5256259  0.7248497  0.4175843
##    9     0.5256581  0.7248195  0.4175974
##   10     0.5256599  0.7248189  0.4175908
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was ncomp = 6.
```

```
aPLS<-plsFit$results
plot(aPLS$ncomp,aPLS$RMSE)
```




```
plsFit$bestTune
```

```
##   ncomp
## 6     6
```

Modelos não lineares

- MARS

```
marsGrid <- expand.grid(.degree = 1, .nprune = 2:11)

set.seed(2712)
marsFit <- train(Y~., data = data_train_final,
                 method = "earth",
                 tuneGrid = marsGrid,
                 trControl = control)
```

```
## Warning: package 'earth' was built under R version 3.3.3
```

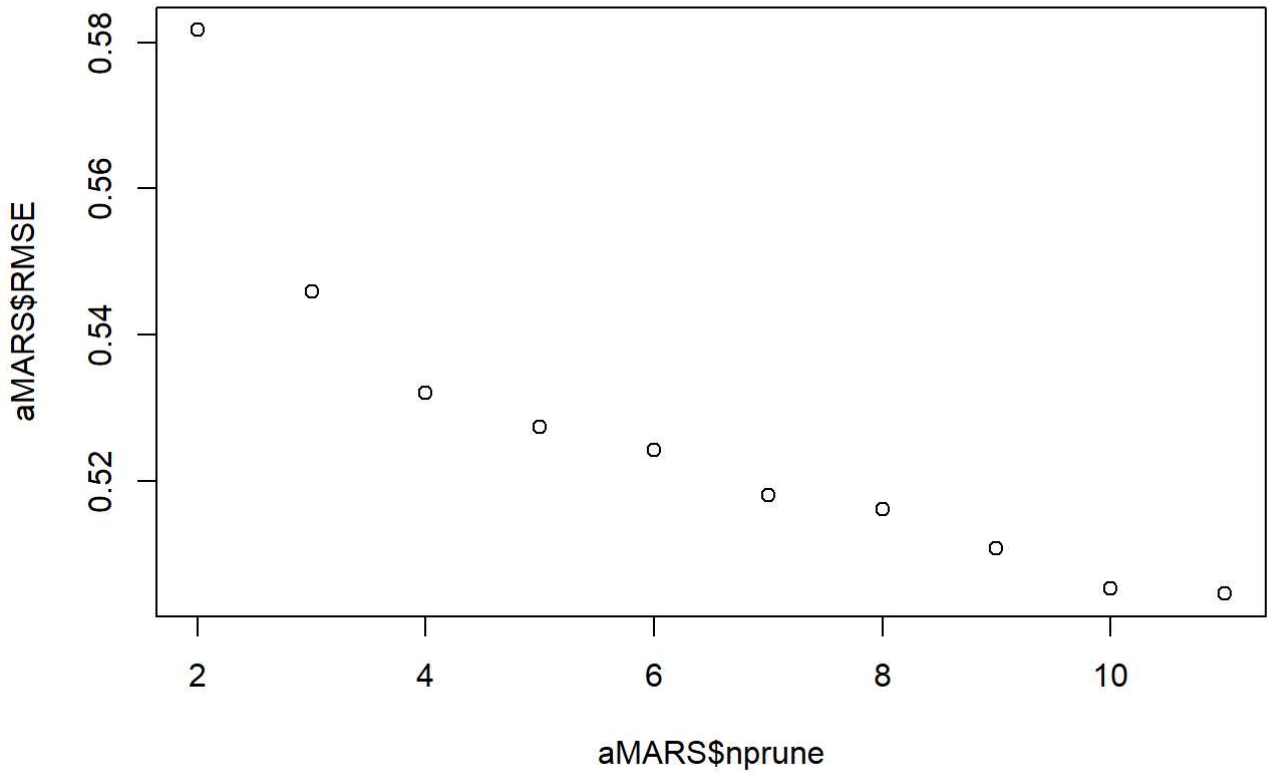
```
## Warning: package 'plotmo' was built under R version 3.3.3
```

```
## Warning: package 'TeachingDemos' was built under R version 3.3.3
```

```
marsFit
```

```
## Multivariate Adaptive Regression Spline
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   nprune  RMSE       Rsquared   MAE
##    2      0.5816575  0.6624262  0.4581123
##    3      0.5459351  0.7031751  0.4309867
##    4      0.5320774  0.7180737  0.4204459
##    5      0.5274626  0.7229984  0.4159400
##    6      0.5243463  0.7261550  0.4144851
##    7      0.5181911  0.7326646  0.4099236
##    8      0.5162397  0.7349799  0.4078467
##    9      0.5108267  0.7404544  0.4039156
##   10      0.5053622  0.7458053  0.3997280
##   11      0.5046823  0.7464102  0.3985500
##
## Tuning parameter 'degree' was held constant at a value of 1
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were nprune = 11 and degree = 1.
```

```
aMARS<-marsFit$results
plot(aMARS$nprune,aMARS$RMSE)
```



```
marsFit$bestTune
```

```
##    nprune degree
## 10     11      1
```

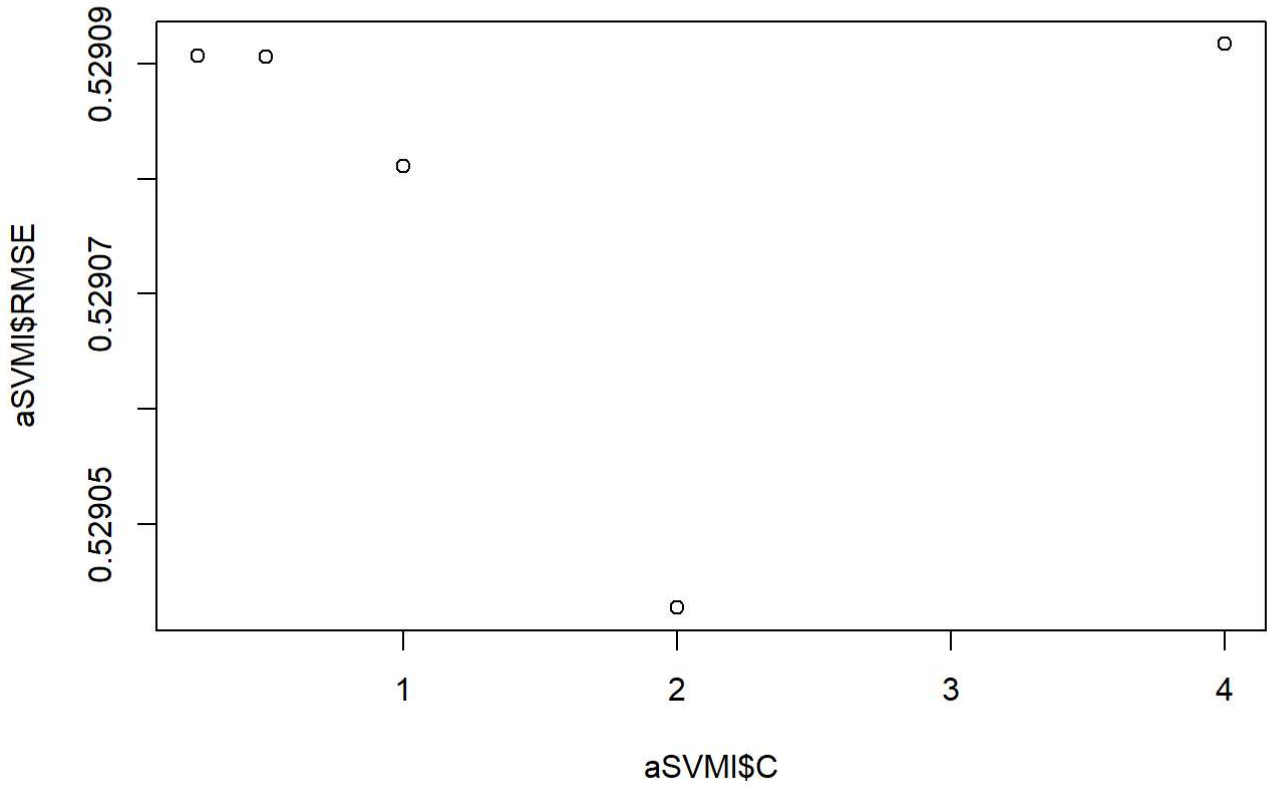
- SVM linear

```
svmlGrid <- expand.grid(.C= c(.25,.50,1,2,4))

set.seed(2712)
svmLFit <- train(Y~., data = data_train_final,
                 method = "svmLinear",
                 tuneGrid=svmlGrid,
                 trControl = control)
svmLFit
```

```
## Support Vector Machines with Linear Kernel
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   C     RMSE       Rsquared   MAE
##   0.25  0.5290907  0.7225051  0.4171292
##   0.50  0.5290906  0.7224992  0.4171416
##   1.00  0.5290811  0.7225051  0.4171405
##   2.00  0.5290427  0.7225362  0.4171135
##   4.00  0.5290917  0.7224818  0.4171767
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was C = 2.
```

```
aSVMl<-svmLFit$results
plot(aSVMl$C,aSVMl$RMSE)
```




```
svmLFit$bestTune
```

```
##   C
## 4 2
```

Modelos de arvore

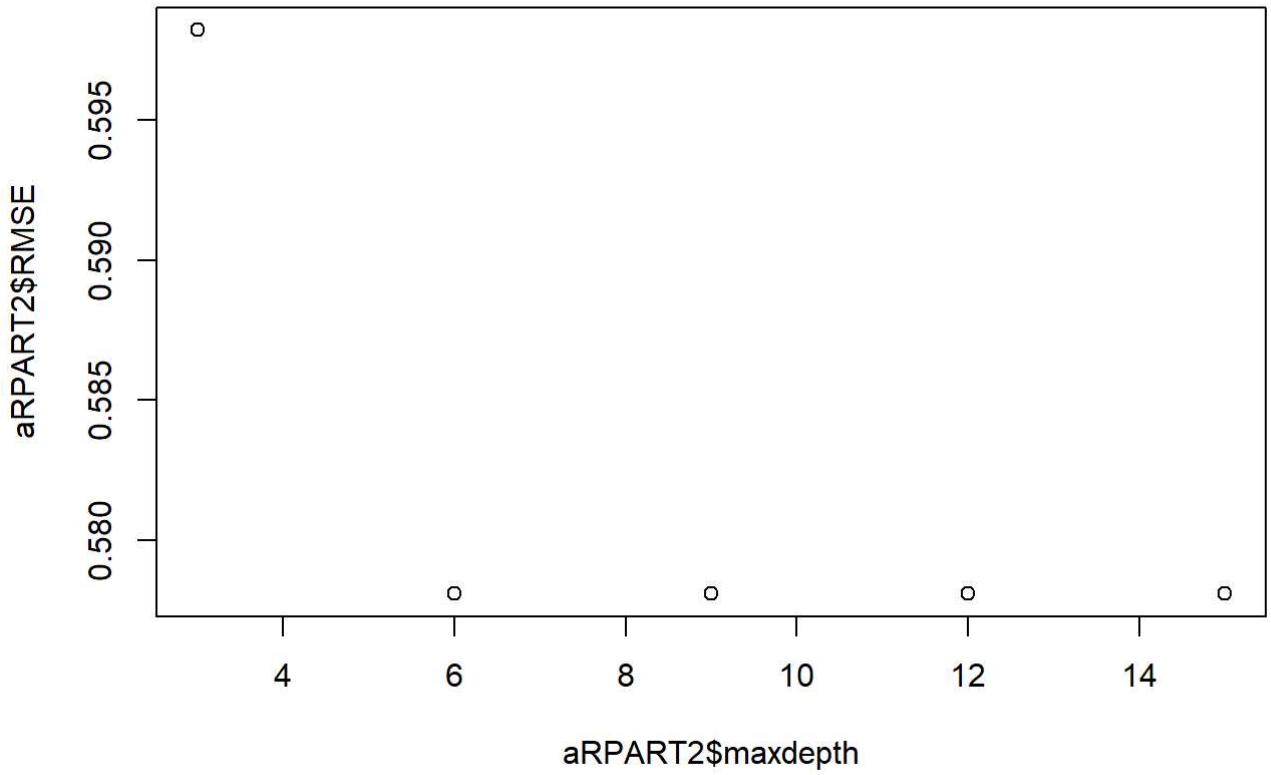
- Árvore de regressão (método .maxdepth)

```
rpart2Grid <- expand.grid(.maxdepth=c(3,6,9,12,15))

set.seed(2712)
rpart2Fit <- train(Y~., data = data_train_final,
                   method = "rpart2",
                   #tuneLength = 10,
                   tuneGrid = rpart2Grid,
                   trControl = control)
rpart2Fit
```

```
## CART
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   maxdepth  RMSE       Rsquared   MAE
##    3        0.5982108  0.6430965  0.4714300
##    6        0.5780765  0.6668855  0.4532768
##    9        0.5780765  0.6668855  0.4532768
##   12        0.5780765  0.6668855  0.4532768
##   15        0.5780765  0.6668855  0.4532768
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was maxdepth = 6.
```

```
aRPART2<-rpart2Fit$results
plot(aRPART2$maxdepth,aRPART2$RMSE)
```




```
rpart2Fit$bestTune
```

```
##   maxdepth
## 2        6
```

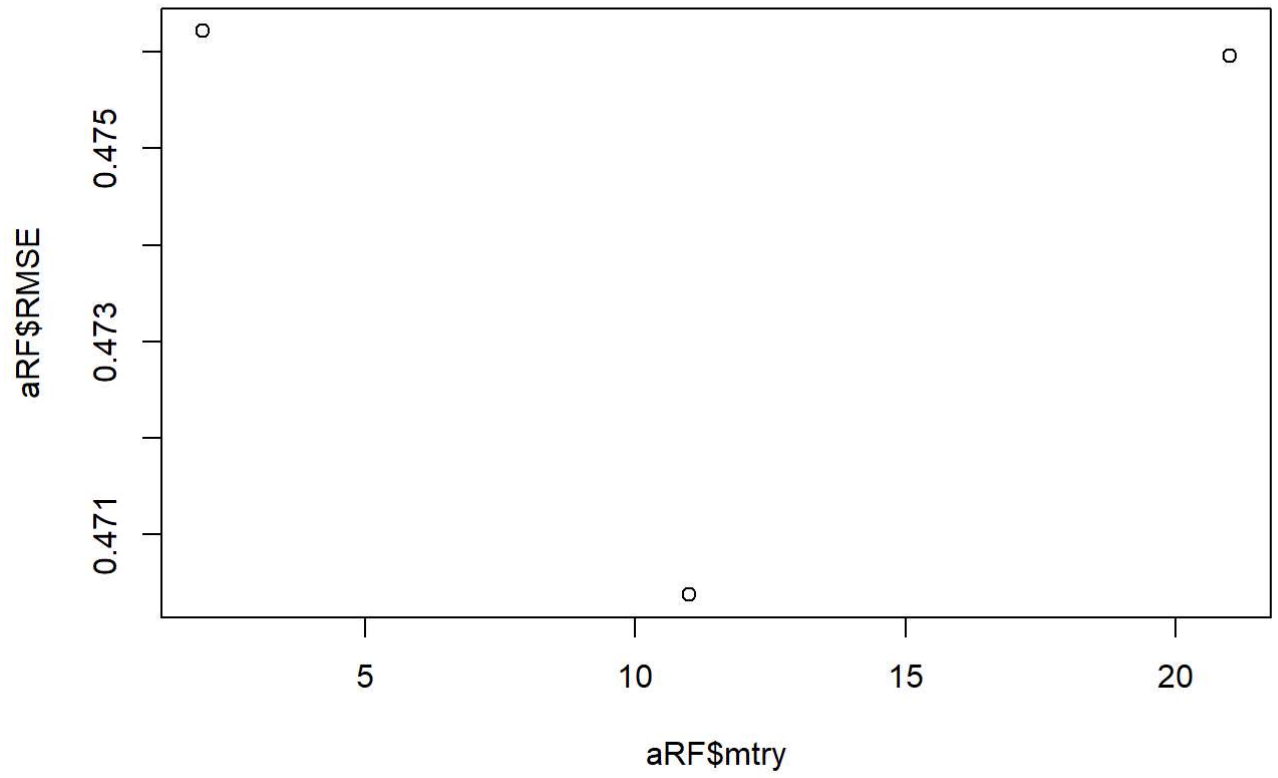
- Random forest

```
rfGrid <- expand.grid(.mtry= c(2,11,21))

set.seed(2712)
rfFit <- train(Y~., data = data_train_final,
               method="rf",
               tuneGrid =rfGrid,
               trControl=control)
rfFit
```

```
## Random Forest
##
## 2136 samples
##   21 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1922, 1923, 1923, 1923, 1923, 1922, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE       Rsquared   MAE
##    2    0.4762171  0.7793081  0.3747332
##   11    0.4703762  0.7802051  0.3702594
##   21    0.4759606  0.7744608  0.3751903
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was mtry = 11.
```

```
aRF<-rfFit$results
plot(aRF$mtry,aRF$RMSE)
```




```
rfFit$bestTune
```

```
##   mtry
## 2   11
```

Ajuste de modelos para o conjunto de treinamento após escolha de hiperparâmetros e avaliação da performance preditiva

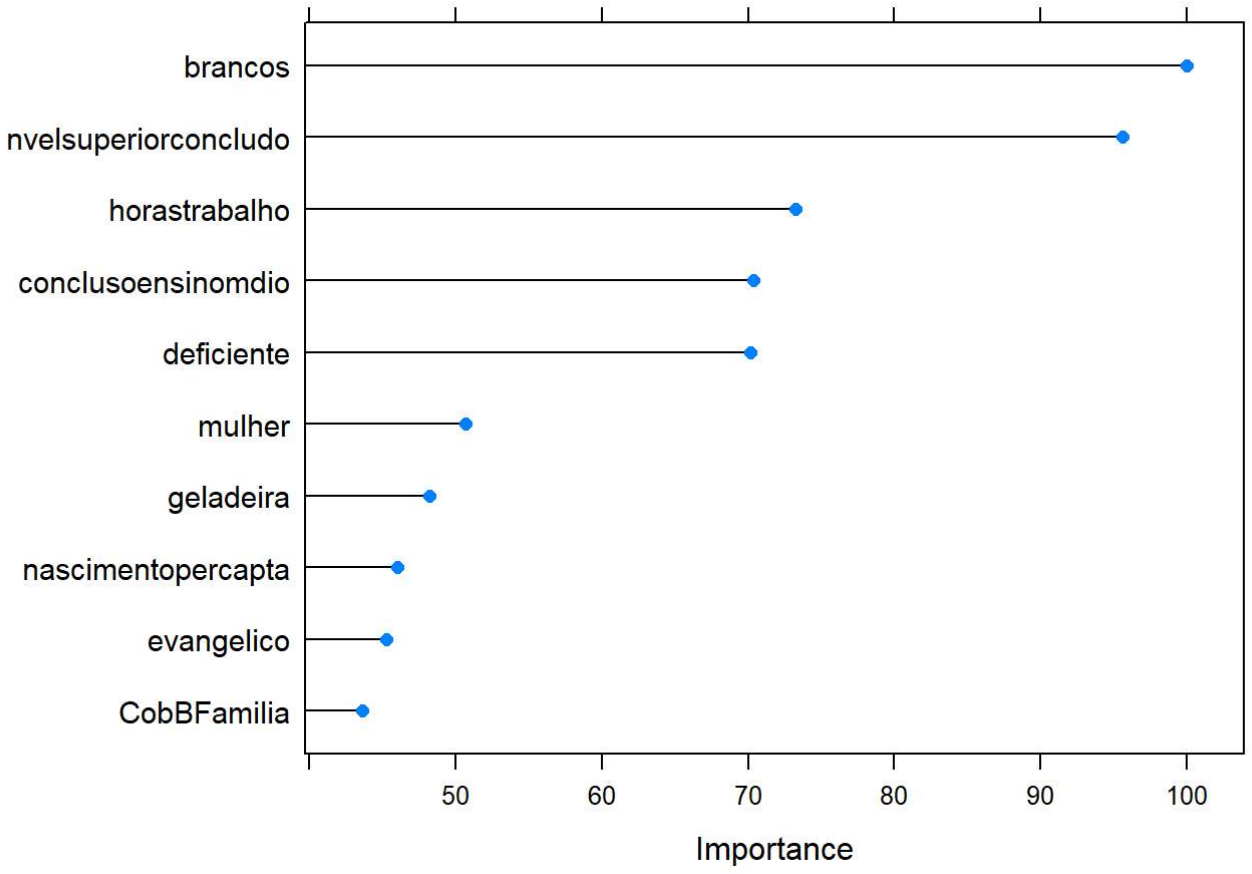
```
control2 <- trainControl(method="none",savePredictions=TRUE)
```

Modelos lineares e modelos relacionados

- Regressão Linear

```
set.seed(2712)
lmFit<-train(Y~., data = data_train_final,
             method="lm",
             trControl=control2)

varimp_lm<-varImp(lmFit)
plot(varimp_lm, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoLM<-predict(lmFit,select(data_test_final,-Y_holdout))
rmseLM<-rmse(data_test_final$Y_holdout,predicaoLM)
rmseLM
```

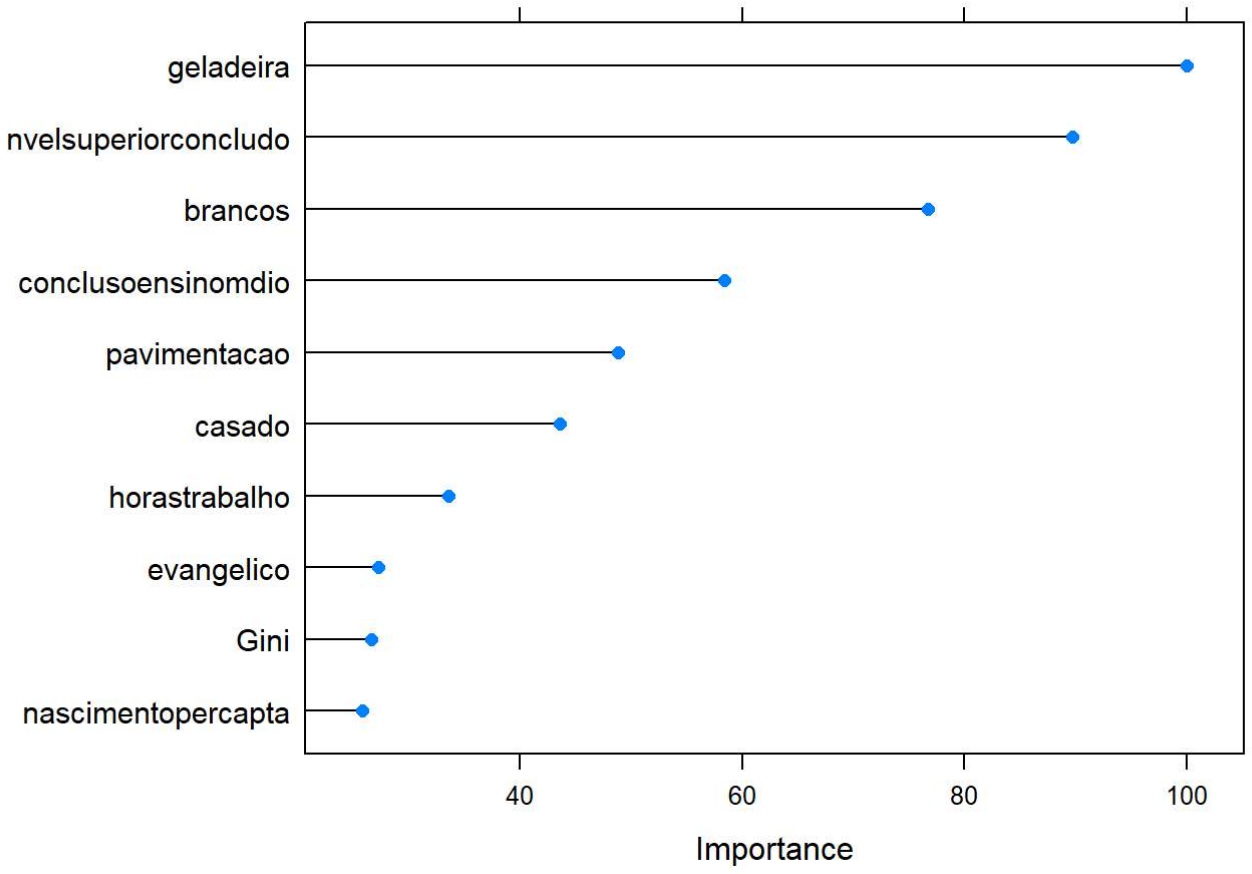
```
## [1] 0.5495419
```

- Regressão Rigde

```
ridgeGrid2 <- data.frame(.lambda = .0006)

set.seed(2712)
ridgeRegFit2 <- train(Y~.,data= data_train_final,
                      method = "ridge",
                      tuneGrid = ridgeGrid2,
                      trControl = control2)

varimp_ridge<-varImp(ridgeRegFit2)
plot(varimp_ridge, top=10,scales=list(y=list(cex=.95)))
```



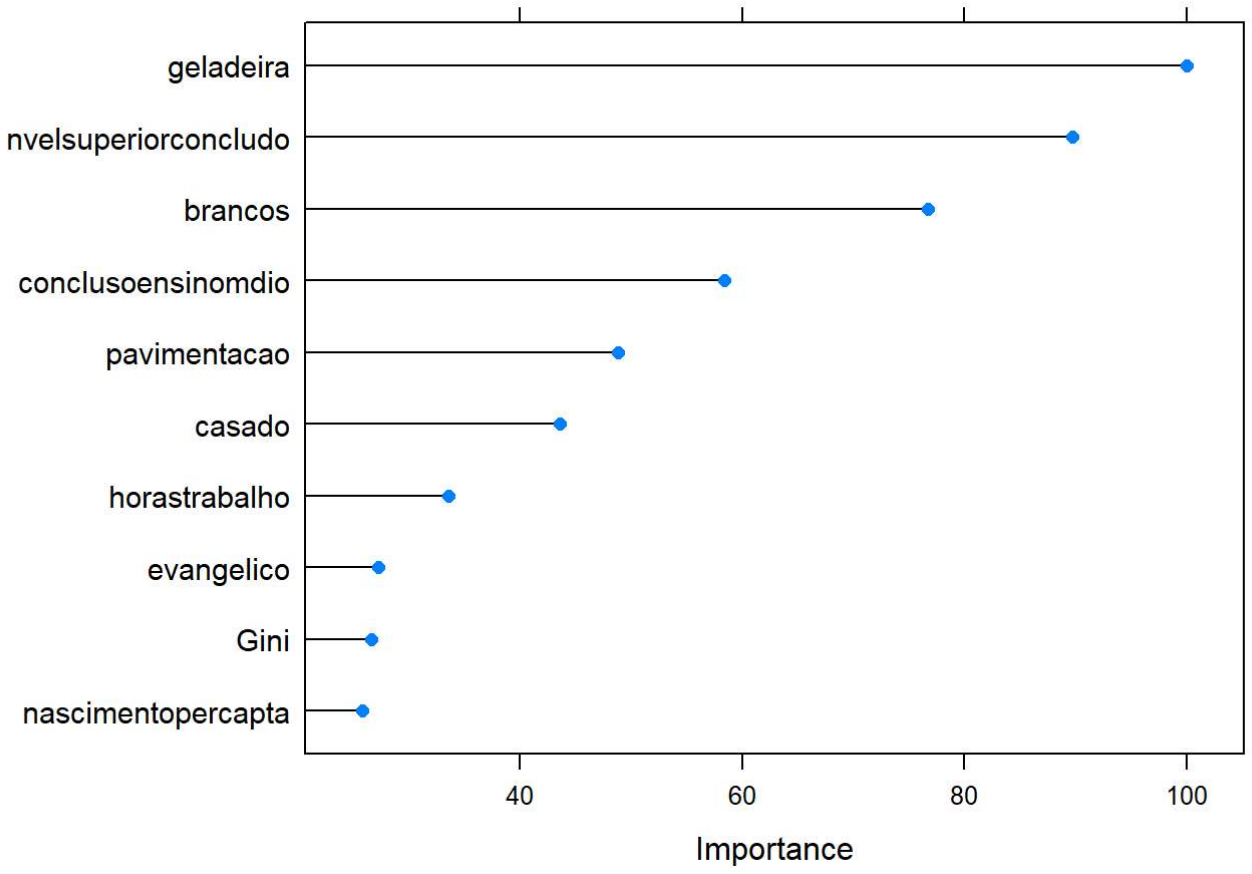

```
predicaoRIDGE<-predict(ridgeRegFit2,select(data_test_final,-Y_holdout))
rmseRIDGE<-rmse(data_test_final$Y_holdout,predicaoRIDGE)
rmseRIDGE
```

```
## [1] 0.5495251
```

- Lasso

```
LassoGrid2 <- expand.grid(.fraction = 0.94)
set.seed(2712)
LassoFit2 <- train(Y~., data = data_train_final,
                   method = "lasso",
                   tuneGrid = LassoGrid2,
                   trControl = control2)

varimp_lasso<-varImp(LassoFit2)
plot(varimp_lasso, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoLASSO<-predict(LassoFit2,select(data_test_final,-Y_holdout))
rmseLASSO<-rmse(data_test_final$Y_holdout,predicaoLASSO)
rmseLASSO
```

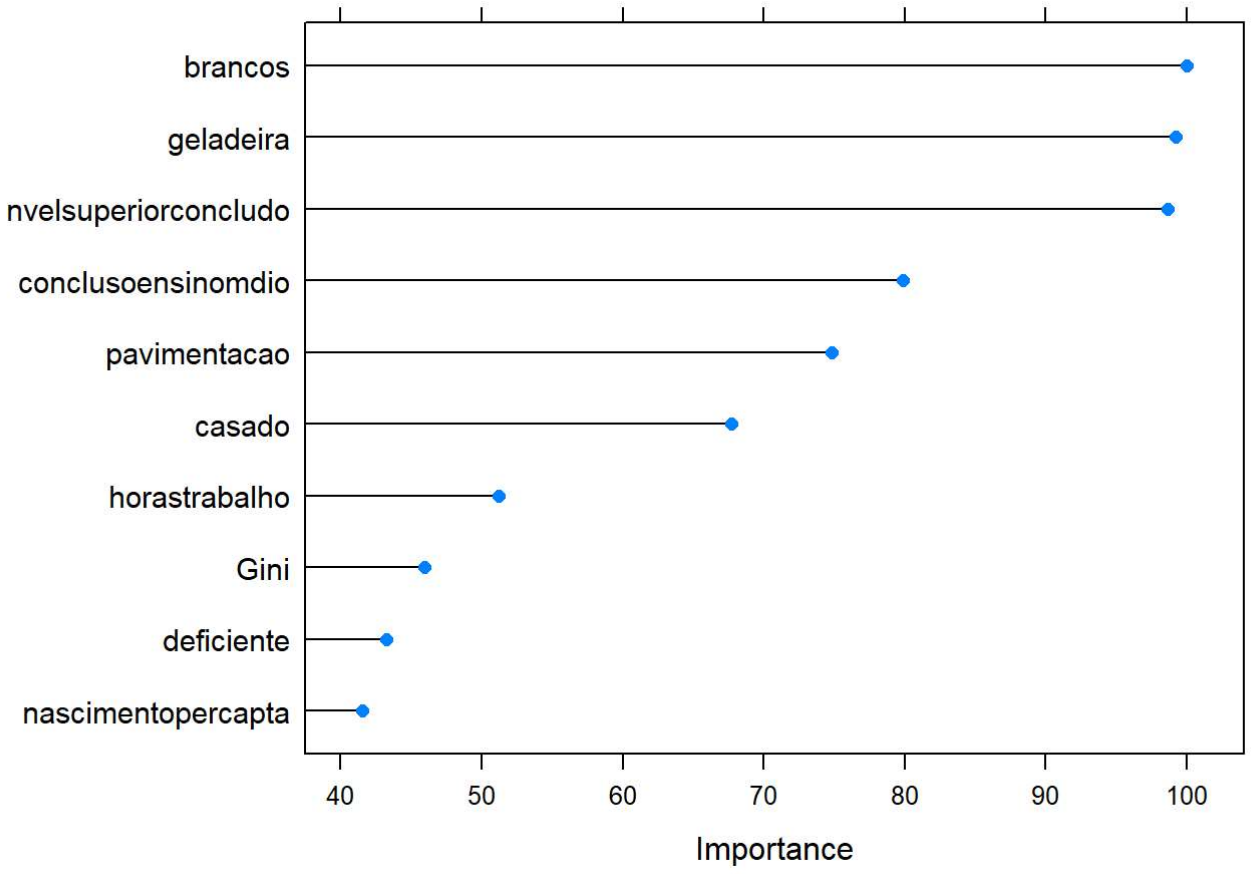
```
## [1] 0.5502414
```

- Mínimos quadrados parciais

```
plsGrid2 <- expand.grid(.ncomp = 6)

set.seed(2712)
plsFit2<-train(Y~., data = data_train_final,
               method="pls",
               tuneGrid = plsGrid2,
               trControl=control2)

varimp_pls<-varImp(plsFit2)
plot(varimp_pls, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoPLS<-predict(plsFit2,select(data_test_final,-Y_holdout))
rmsePLS<-rmse(data_test_final$Y_holdout,predicaoPLS)
rmsePLS
```

```
## [1] 0.5486196
```

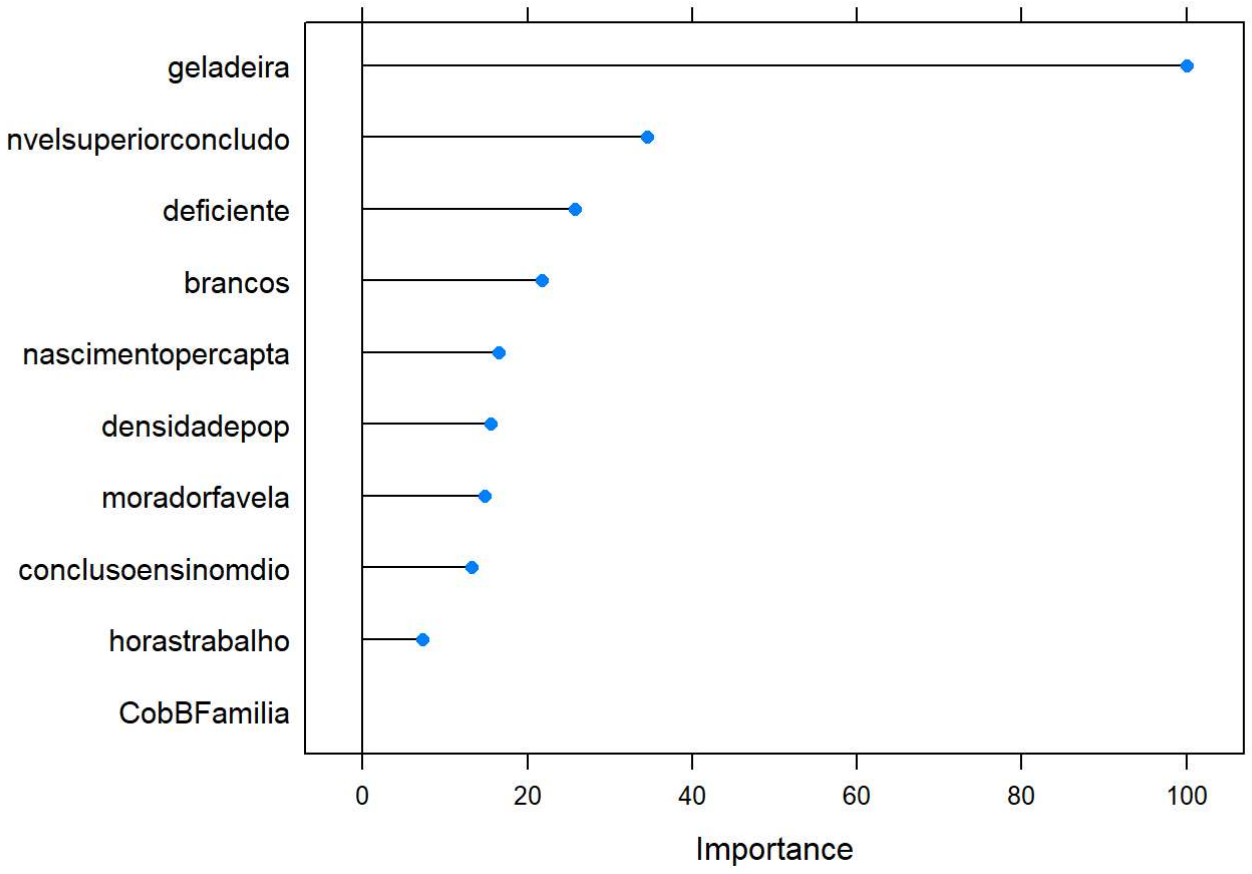
Modelos não lineares

- MARS

```
marsGrid2 <- expand.grid(.degree = 1, .nprune = 10)

set.seed(2712)
marsFit2 <- train(Y~., data = data_train_final,
                  method = "earth",
                  tuneGrid = marsGrid2,
                  trControl = control2)

varimp_mars<-varImp(marsFit2)
plot(varimp_mars, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoMARS<-as.vector(predict(marsFit2,select(data_test_final,-Y_holdout)))
rmseMARS<-rmse(data_test_final$Y_holdout,predicaoMARS)
rmseMARS
```

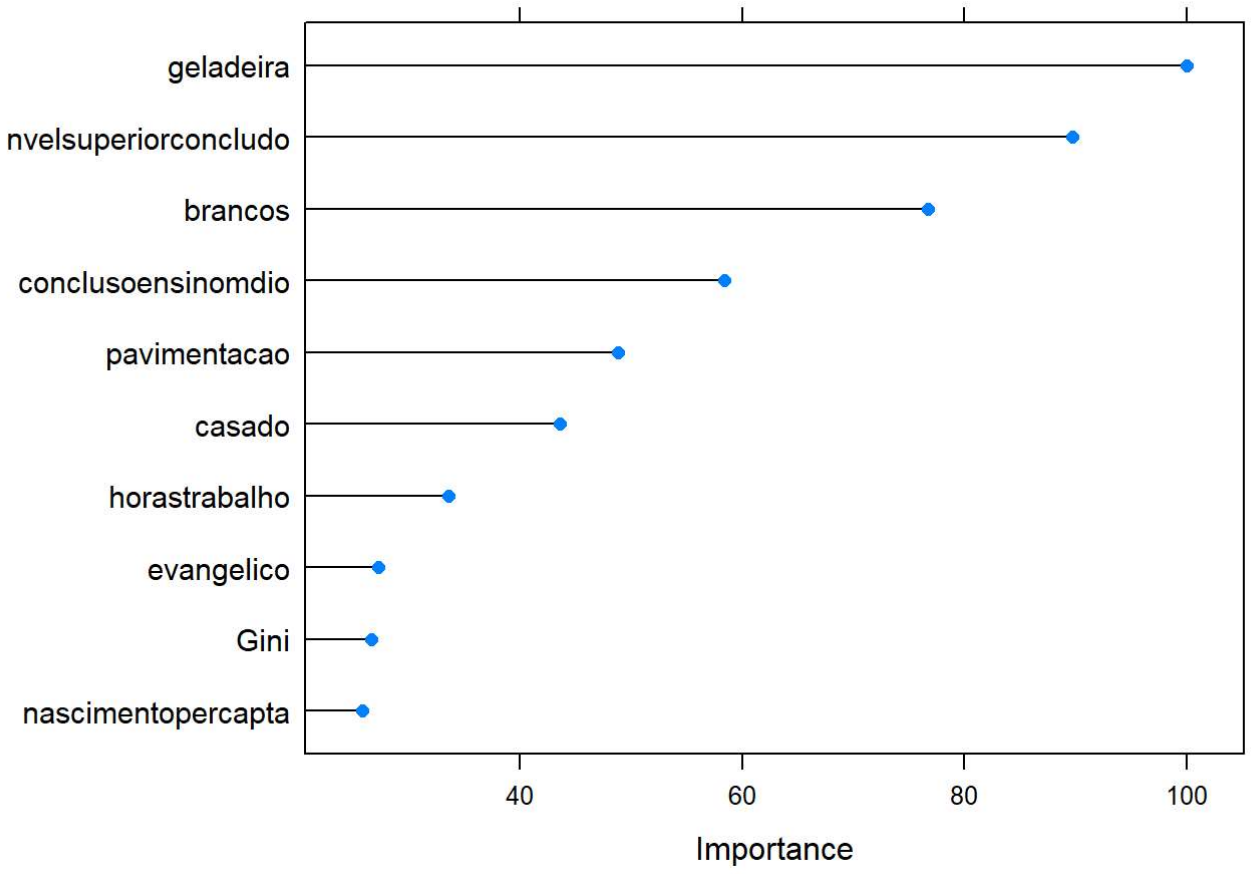
```
## [1] 0.5165598
```

- SVM linear

```
svmlGrid2 <- expand.grid(.C= 2)

set.seed(2712)
svmLFit2 <- train(Y~., data = data_train_final,
                  method = "svmLinear",
                  tuneGrid=svmlGrid2,
                  trControl = control2)

varimp_svm<-varImp(svmLFit2)
plot(varimp_svm, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoSVM1<-predict(svmLFit2,select(data_test_final,-Y_holdout))
rmseSVM1<-rmse(data_test_final$Y_holdout,predicaoSVM1)
rmseSVM1
```

```
## [1] 0.5574941
```

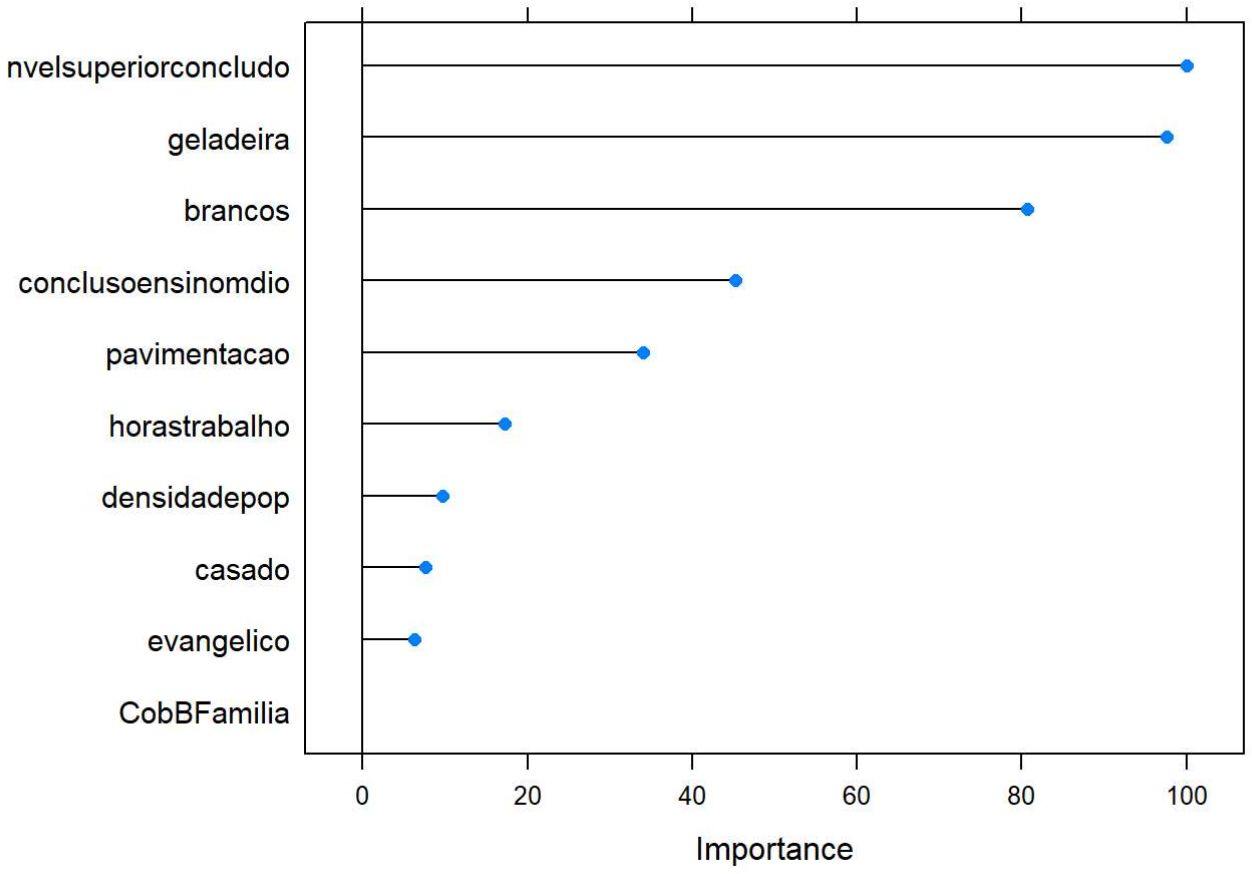
Modelos de árvore

- Árvore de regressão (método .maxdepth)

```
rpart2Grid2 <- expand.grid(.maxdepth=6)

set.seed(2712)
rpart2Fit2 <- train(Y~., data = data_train_final,
                    method = "rpart2",
                    tuneGrid = rpart2Grid2,
                    trControl = control2)

varimp_rpart<-varImp(rpart2Fit2)
plot(varimp_rpart, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoRPART2<-predict(rpart2Fit2,select(data_test_final,-Y_holdout))
rmseRPART2<-rmse(data_test_final$Y_holdout,predicaoRPART2)
rmseRPART2
```

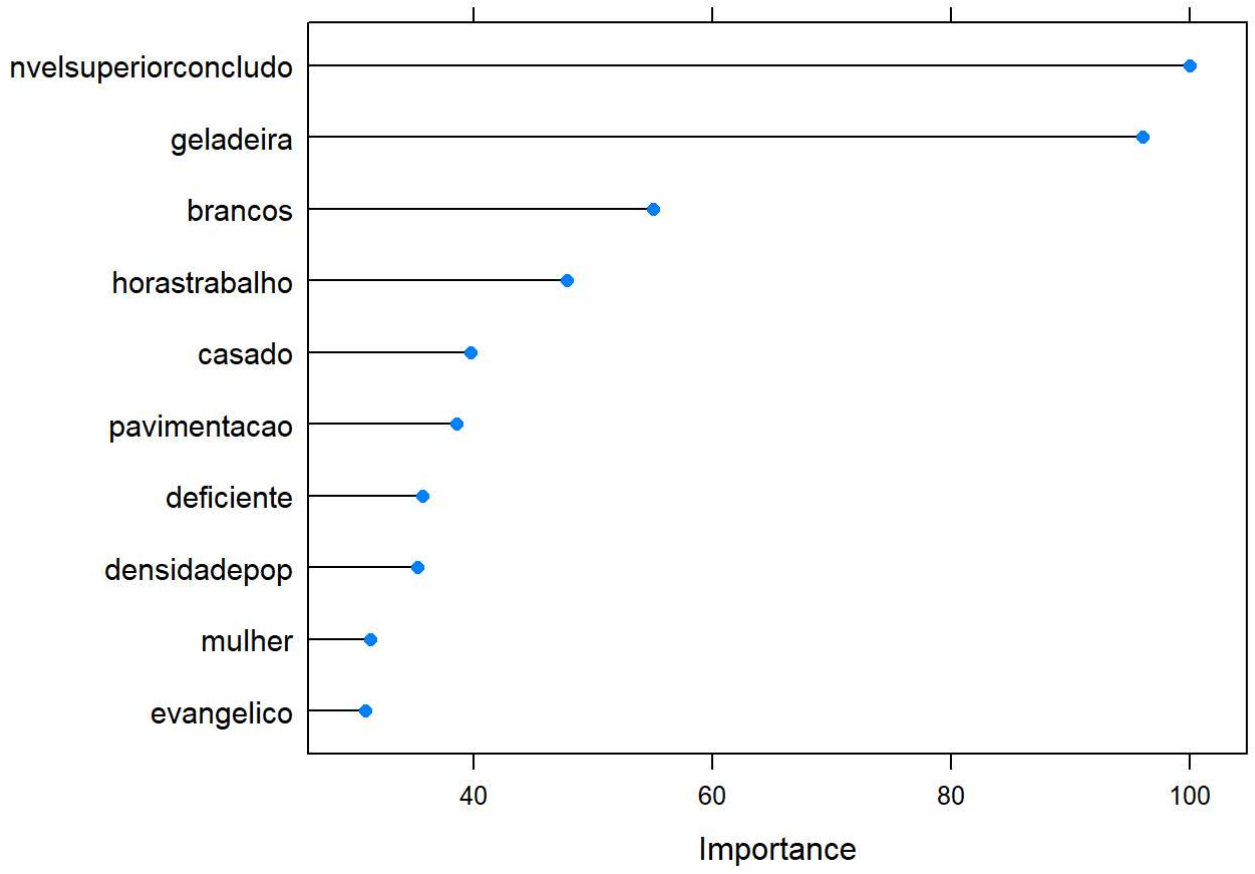
```
## [1] 0.5885154
```

- Random forest

```
rfGrid2 <- expand.grid(.mtry= 11)

set.seed(2712)
rfFit2 <- train(Y~., data = data_train_final,
                method="rf",
                tuneGrid =rfGrid2,
                trControl=control2,
                importance=T)

varimp_rf<-caret::varImp(rfFit2)
plot(varimp_rf, top=10,scales=list(y=list(cex=.95)))
```




```
predicaoRF<-predict(rfFit2,select(data_test_final,-Y_holdout))
rmseRF<-rmse(data_test_final$Y_holdout,predicaoRF)
rmseRF
```

```
## [1] 0.4900818
```

Performance (dados de teste) dos algoritmos ajustados

```
RMSE_test<-as.data.frame(rbind(rmseLM,rmseRIDGE,rmseLASSO,rmsePLS,rmseMARS,rmseSVMl,rmseRPART
2,rmseRF),row.names = F)
Algorithm<-c("LM","RIDGE","LASSO","PLS","MARS","SVMl","RPART","RF")
RMSE_test<-cbind(Algorithm, RMSE_test)
names(RMSE_test)[2]<-"performance"
RMSE_test[with(RMSE_test, order(performance)), ]
```

```
##   Algorithm performance
## 8        RF   0.4900818
## 5      MARS   0.5165598
## 4       PLS   0.5486196
## 2     RIDGE   0.5495251
## 1        LM   0.5495419
## 3     LASSO   0.5502414
## 6      SVMl   0.5574941
## 7     RPART   0.5885154
```