

Aprendizagem Estatística em Altas Dimensões

[MAE0501/MAE5904/IBI5904]

Ícaro Maia Santos de Castro¹
Rayssa de Carvalho Roberto²
Rodrigo Aoyama Nakahara³
Rodrigo Araujo⁴
Vitor Hugo Vieira de Lima⁵

Novembro de 2020

Sumário

Análise descritiva	2
Renomeando a variável	3
Reshape	3
Visualização dos Dados	3
Análise Descritiva	3
Correlação entre cada variável	3
Proporção de diabetes (Benign / Malignant)	7

¹Número USP: 11866921

²Número USP: 10940828

³Número USP: 3510922

⁴Número USP: 9299208

⁵Número USP: 10263886

Análise descritiva

O objetivo geral do problema é prever se a pessoa possui ou não diabetes mellitus, com base em uma série de variáveis preditoras. O desafio foi posto declaradamente como um problema de aprendizagem.

O banco de dados possui todas suas observações constituídas de pessoas do gênero feminino, com idade superior a 21 anos, de ascendência do povo Pima (grupo de nativos norte-americanos). Os dados, que são atualmente abertos e gratuitos para download CC0 1.0, são provenientes do Instituto Nacional de Diabetes e Distúrbios Digestivos e do Rim (NIDDK) dos EUA e mantidos pela UC Irvine Machine Learning Repository (University of California-Irvine).

```
diabetes <- read.csv("diabetes.csv")
head(diabetes) %>% kable(caption="Dados.")
```

Tabela 1: Dados.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33,6	0,63	50	1
1	85	66	29	0	26,6	0,35	31	0
8	183	64	0	0	23,3	0,67	32	1
1	89	66	23	94	28,1	0,17	21	0
0	137	40	35	168	43,1	2,29	33	1
5	116	74	0	0	25,6	0,20	30	0

```
summary(diabetes)
```

```
## Pregnancies      Glucose      BloodPressure      SkinThickness
## Min.   : 0.000    Min.   : 0.0    Min.   : 0.00    Min.   : 0.00
## 1st Qu.: 1.000    1st Qu.: 99.0    1st Qu.: 62.00    1st Qu.: 0.00
## Median : 3.000    Median :117.0    Median : 72.00    Median :23.00
## Mean   : 3.845    Mean   :120.9    Mean   : 69.11    Mean   :20.54
## 3rd Qu.: 6.000    3rd Qu.:140.2    3rd Qu.: 80.00    3rd Qu.:32.00
## Max.   :17.000    Max.   :199.0    Max.   :122.00    Max.   :99.00
## Insulin          BMI          DiabetesPedigreeFunction      Age
## Min.   : 0.0    Min.   : 0.00    Min.   :0.0780    Min.   :21.00
## 1st Qu.: 0.0    1st Qu.:27.30    1st Qu.:0.2437    1st Qu.:24.00
## Median : 30.5    Median :32.00    Median :0.3725    Median :29.00
## Mean   : 79.8    Mean   :31.99    Mean   :0.4719    Mean   :33.24
## 3rd Qu.:127.2    3rd Qu.:36.60    3rd Qu.:0.6262    3rd Qu.:41.00
## Max.   :846.0    Max.   :67.10    Max.   :2.4200    Max.   :81.00
## Outcome
## Min.   :0.000
## 1st Qu.:0.000
## Median :0.000
## Mean   :0.349
## 3rd Qu.:1.000
## Max.   :1.000
```

Renomeando a variável

‘Outcome’ para ‘diabetes’

```
colnames(diabetes)[9] <- "diabetes"
```

Reshape

Diabetes? => 0 : No / 1 : Yes

```
diabetes$diabetes <- as.factor(diabetes$diabetes)
levels(diabetes$diabetes) <- c("No", "Yes")
```

Visualização dos Dados

Estrutura dos Dados

```
str(diabetes)

## 'data.frame': 768 obs. of 9 variables:
## $ Pregnancies : int 6 1 8 1 0 5 3 10 2 8 ...
## $ Glucose : int 148 85 183 89 137 116 78 115 197 125 ...
## $ BloodPressure : int 72 66 64 66 40 74 50 0 70 96 ...
## $ SkinThickness : int 35 29 0 23 35 0 32 0 45 0 ...
## $ Insulin : int 0 0 0 94 168 0 88 0 543 0 ...
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
## $ DiabetesPedigreeFunction: num 0.627 0.351 0.672 0.167 2.288 ...
## $ Age : int 50 31 32 21 33 30 26 29 53 54 ...
## $ diabetes : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 1 2 2 ...
```

Dimensão

```
dim(diabetes)
```

```
## [1] 768 9
```

Análise Descritiva

Correlação entre cada variável

```
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.0.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.0.2
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
##
```

```
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
```

```
##      first, last
```

```
##
```

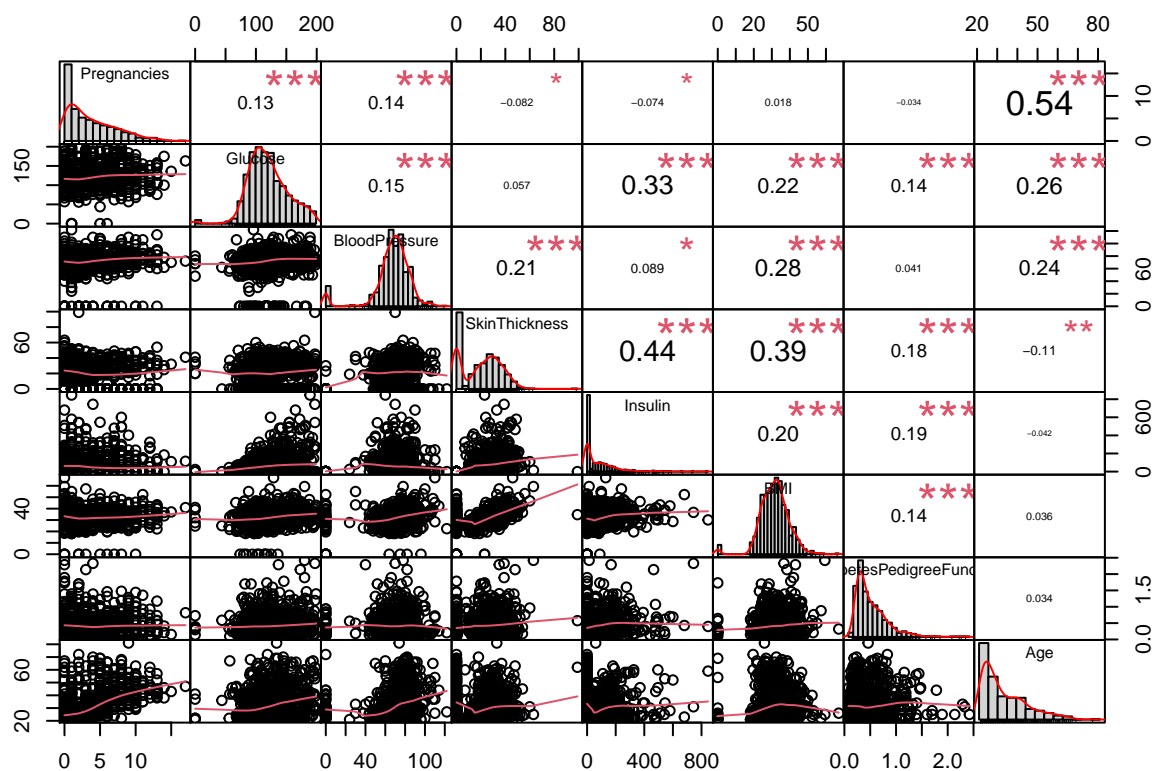
```
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
```

```
##
```

```
##      legend
```

```
chart.Correlation(diabetes[, -9], histogram=TRUE, col="grey10", pch=1, main="Correlação entre as variáveis")
```



```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
```

```
##   method from
```

```
##   +.gg   ggplot2
```

```
ggcorr(diabetes[, -9], name = "corr", label = TRUE) +
```

```
  theme(legend.position="none") +
```

```
  labs(title="Correlação entre as variáveis explicativas") +
```

```
  theme(plot.title=element_text(face='bold', color='black', hjust=0.5, size=12))
```

Correlação entre as variáveis explicativas

						Age
					DiabetesPedigreeFunction	
				BMI	0.1	0
			Insulin	0.2	0.2	0
		SkinThickness	0.4	0.4	0.2	-0.1
	BloodPressure	0.2	0.1	0.3	0	0.2
Glucose	0.2	0.1	0.3	0.2	0.1	0.3
pregnancies	0.1	0.1	-0.1	-0.1	0	0.5

```
library(tidyverse)
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
## The following object is masked from 'package:permute':
##
##   permute
```

```
library(dplyr)

# para reprodução
set.seed(23)

nrows <- NROW(diabetes)

index <- sample(1:nrows, 0.7 * nrows) # shuffle and divide

# train <- diab                        # 768 test data (100%)
train <- diabetes[index,]              # 537 test data (70%)
```

```
test <- diabetes[-index,] # 231 test data (30%)
```

Proporção de diabetes (Benign / Malignant)

train

```
prop.table(table(train$diabetes))
```

```
##  
##           No           Yes  
## 0.6405959 0.3594041
```

test

```
prop.table(table(test$diabetes))
```

```
##  
##           No           Yes  
## 0.6753247 0.3246753
```

RandomForest

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2
```

```
##  
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:vegan':  
##  
##      tolerance
```

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

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

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## randomForest 4.6-14
```

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

```
##
```

```
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:psych':
```

```
##
```

```
## outlier
```

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

```
##
```

```
## combine
```

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

```
##
```

```
## margin
```

```
learn_rf <- randomForest(diabetes~., data=train, ntree=500, proximity=T, importance=T)
```

```
pre_rf <- predict(learn_rf, test[, -9])
```

```
cm_rf <- confusionMatrix(pre_rf, test$diabetes)
```

```
cm_rf
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction No Yes
```

```
##           No 128 26
```

```
##           Yes 28 49
```

```
##
```

```
##           Accuracy : 0.7662
```

```
##           95% CI : (0.7063, 0.8192)
```

```
##           No Information Rate : 0.6753
```

```
##           P-Value [Acc > NIR] : 0.00157
```

```
##
```

```
##           Kappa : 0.4706
```

```
##
```

```
##           McNemar's Test P-Value : 0.89176
```

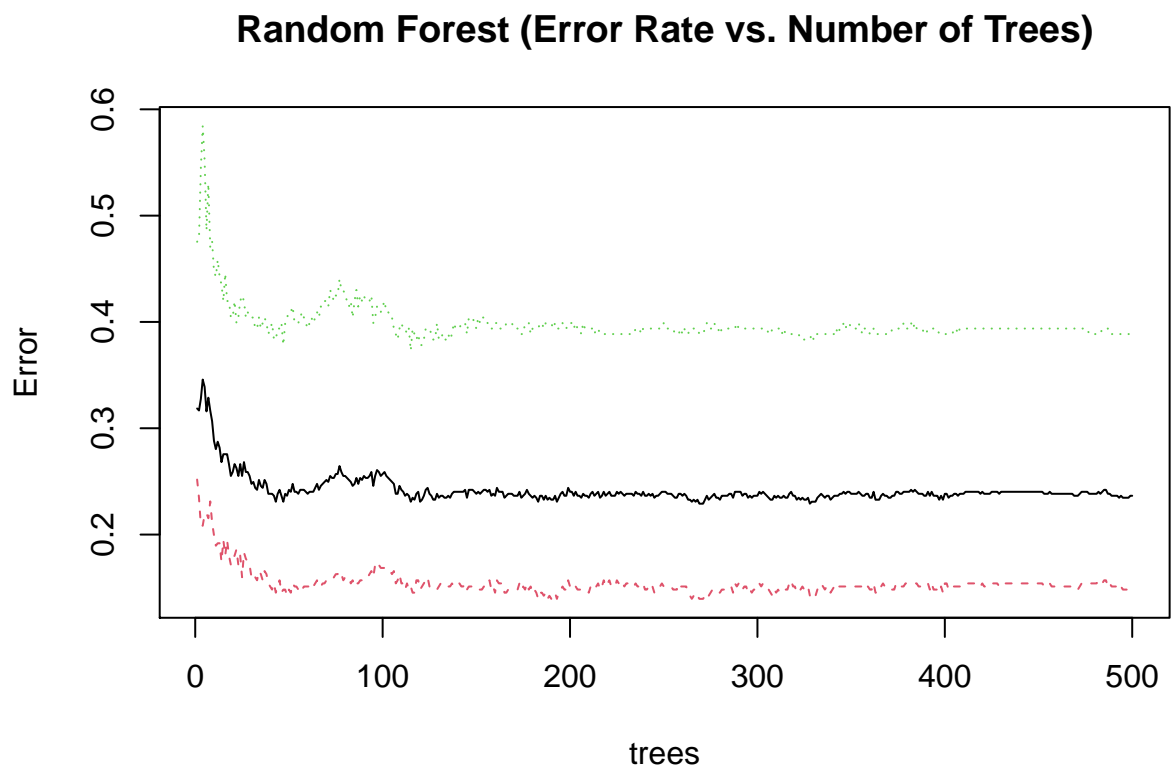
```
##
```

```
##           Sensitivity : 0.8205
```



```
##          Specificity : 0.6533
##          Pos Pred Value : 0.8312
##          Neg Pred Value : 0.6364
##          Prevalence : 0.6753
##          Detection Rate : 0.5541
##          Detection Prevalence : 0.6667
##          Balanced Accuracy : 0.7369
##
##          'Positive' Class : No
```

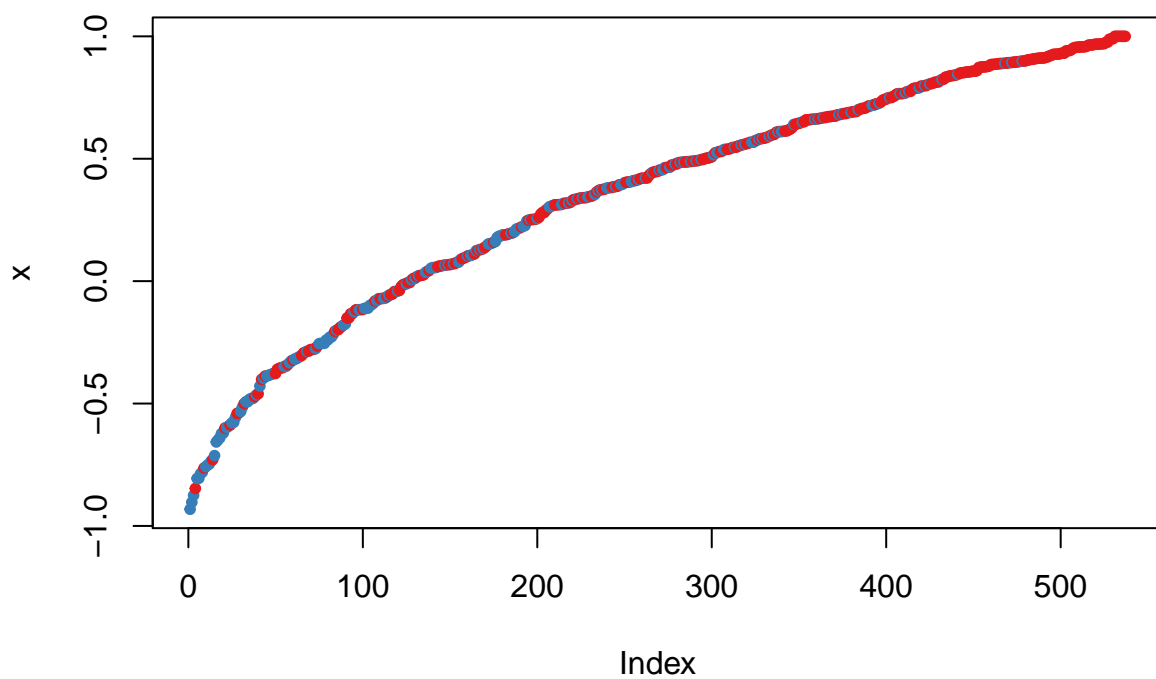
```
plot(learn_rf, main="Random Forest (Error Rate vs. Number of Trees)")
```



Prediction Plot

```
plot(margin(learn_rf, test$diabetes))
```

```
## Warning in RColorBrewer::brewer.pal(nlevs, "Set1"): minimal value for n is 3, returning requested palette
```



Variance Importance Plot

```
varImpPlot(learn_rf)
```

learn_rf

