Modelo de Regressão Logística - Diabetes

Marcos de Morais

01/03/2022

DATASET UCI - DIABETES

O objetivo desse script é encontrar um modelo preditivo que consiga identificar quando uma pessoa terá ou não diabetes.

O dicionário de dados apresenta variáveis categóricas e a idade como variável numérica. A variável Class é o target do modelo preditivo.

Dicionário dos dados:

Age 1.20-65

Sex 1. Male, 2.Female

Polyuria 1.Yes, 2.No.

Polydipsia 1.Yes, 2.No.

sudden weight loss 1.Yes, 2.No.

weakness 1.Yes, 2.No.

Polyphagia 1.Yes, 2.No.

Genital thrush 1.Yes, 2.No.

visual blurring 1.Yes, 2.No.

Itching 1.Yes, 2.No.

Irritability 1.Yes, 2.No.

delayed healing 1.Yes, 2.No.

partial paresis 1.Yes, 2.No.

muscle stiness 1.Yes, 2.No.

Alopecia 1.Yes, 2.No.

Obesity 1.Yes, 2.No. Class 1.Positive, 2.Negative.

Link do dataset: https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset.

Diabetes é uma doença causada pela produção insuficiente ou má absorção de insulina, hormônio que regula a glicose no sangue e garante energia para o organismo. O diabetes pode causar o aumento da glicemia e as altas taxas podem levar a complicações no coração, nas artérias, nos olhos, nos rins e nos nervos. Em casos mais graves, o diabetes pode levar à morte.

De acordo com a Sociedade Brasileira de Diabetes, existem atualmente, no Brasil, mais de 13 milhões de pessoas vivendo com a doença, o que representa 6,9% da população nacional.

A melhor forma de prevenir é praticando atividades físicas regularmente, mantendo uma alimentação saudável e evitando consumo de álcool, tabaco e outras drogas.

Fonte: https://www.saude.pr.gov.br/Pagina/Diabetes-diabetes-mellitus

1 - Definindo Diretório de Trabalho

```
setwd('c:/FCD/R/UCI/Diabetes')
getwd()
```

```
## [1] "c:/FCD/R/UCI/Diabetes"
```

2 - Carga do Dataset e Pacotes

Além do dataset da UCI, é necessário a carga dos pacotes caret (Machine Learning), Amelia (Identificar valores NA), ROCR para a curva ROC e por último o RMarkdown para gerar o PDF.

```
DB <- read.csv('diabetes_data_upload.csv')

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

```
## Carregando pacotes exigidos: lattice
```

```
## Carregando pacotes exigidos: ggplot2
```

```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 4.1.1
```

```
## Carregando pacotes exigidos: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 4.1.1
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(rmarkdown)
```

```
## Warning: package 'rmarkdown' was built under R version 4.1.1
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.1.2
```

3 - Análise Exploratória dos Dados

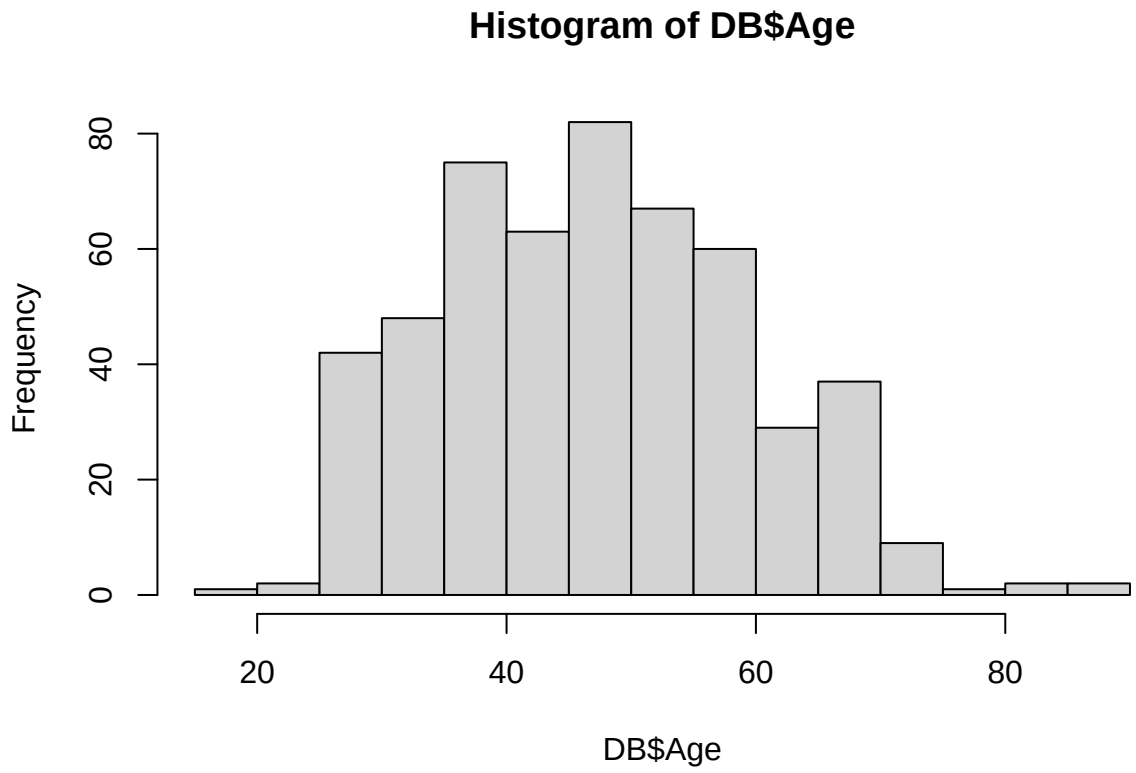
Nesse passo é feito uma análise exploratória dos dados, como analisar valores missing no dataset e a forma da distribuição, por exemplo.

```
View(DB)
str(DB) #Será necessário transformar os dados em factor.
```

```
## 'data.frame':    520 obs. of  17 variables:
##  $ Age              : int  40 58 41 45 60 55 57 66 67 70 ...
##  $ Gender           : chr  "Male" "Male" "Male" "Male" ...
```

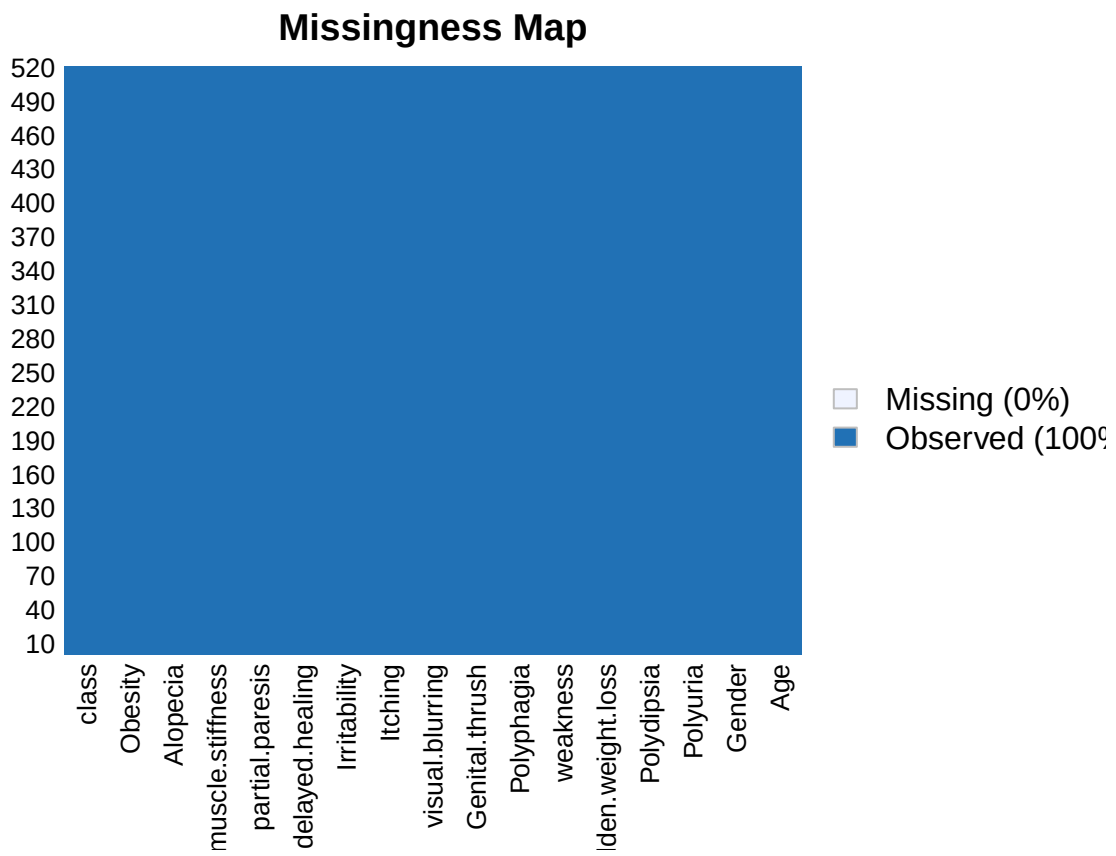
```
##  $ Polyuria          : chr  "No" "No" "Yes" "No" ...
##  $ Polydipsia        : chr  "Yes" "No" "No" "No" ...
##  $ sudden.weight.loss: chr  "No" "No" "No" "Yes" ...
##  $ weakness          : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Polyphagia        : chr  "No" "No" "Yes" "Yes" ...
##  $ Genital.thrush    : chr  "No" "No" "No" "Yes" ...
##  $ visual.blurring   : chr  "No" "Yes" "No" "No" ...
##  $ Itching           : chr  "Yes" "No" "Yes" "Yes" ...
##  $ Irritability      : chr  "No" "No" "No" "No" ...
##  $ delayed.healing   : chr  "Yes" "No" "Yes" "Yes" ...
##  $ partial.paresis   : chr  "No" "Yes" "No" "No" ...
##  $ muscle.stiffness  : chr  "Yes" "No" "Yes" "No" ...
##  $ Alopecia          : chr  "Yes" "Yes" "Yes" "No" ...
##  $ Obesity           : chr  "Yes" "No" "No" "No" ...
##  $ class             : chr  "Positive" "Positive" "Positive" "Positive" ...
```

```
hist(DB$Age)
```




```
# Os dados da idade estão bem distribuídos no histograma, podemos avançar na análise.

missmap(DB) # Não há dados missing.
```




```
round(prop.table(table(DB$class))*100,2)
```

```
##
## Negative Positive
##    38.46    61.54
```

```
# O DataSet veio com um distribução muito maior de casos positivos.
# Isso implica em um modelo mais tendecioso para os casos positivos.
# Em função de ser um dataset de aprendizado, o modelo será treinado e
# testado com essas condições
# Em um caso prático, teríamos que buscar mais registros negativos para
# balancear o dataset.
```

3 - Pré-Processamento dos Dados

Nessa etapa é comum tirar os valores missing, uso de joins com outros datasets, formatar/criar variáveis e etc.

```
#É uma boa prática preservar o dataset original
DB2 <- DB

#Função para transformar as variaveis em tipo factor
for (i in 2:17) {
  DB2[,i] <- as.factor(DB2[,i])
}
```

4 - Dataset de treino e teste

Nesa etapa é feita a divisão do dataset em treino e teste para uso no modelo preditivo.

```
BD_treino <- DB2[1:312,]
BD_teste <- DB2[313:520,]
```

5 - Treinamento Modelo Regressão Logística

Com os dados pré-processados e separados em treino e teste, é feito o treinamento do modelo preditivo. Para isso será usada a função glm do pacote caret.

```
modelo1 <- glm(formula = 'class ~ .', data = BD_treino, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#O summary mostra que a variável Polyuria e Polydipsia são muito representativas
#no modelo de regressão logistica.

#Polyuria é a urina constante e Polydipsia é sintoma de sede e/ou boca seca.
#Ambos os sintomas são característicos da diabetes.
summary(modelo1)
```

```
##
## Call:
## glm(formula = "class ~ .", family = "binomial", data = BD_treino)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.66868  -0.00098   0.00000   0.00290   2.96703
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              32.9712  2345.4995   0.014  0.98878
## Age                      -0.2290     0.0873  -2.623  0.00871 **
## GenderMale              -30.1646  2345.4954  -0.013  0.98974
## PolyuriaYes               6.9997     1.8198   3.846  0.00012 ***
## PolydipsiaYes             8.0669     2.0205   3.993 6.54e-05 ***
## sudden.weight.lossYes    -0.3462     1.0274  -0.337  0.73618
## weaknessYes               0.9961     0.8916   1.117  0.26388
## PolyphagiaYes             2.7485     1.2186   2.255  0.02411 *
## Genital.thrushYes         4.0277     1.4029   2.871  0.00409 **
## visual.blurringYes       -0.5565     1.0966  -0.507  0.61184
## ItchingYes               -3.9075     1.4002  -2.791  0.00526 **
## IrritabilityYes           5.2491     1.7248   3.043  0.00234 **
## delayed.healingYes       -0.4094     0.9828  -0.417  0.67696
## partial.paresisYes        4.2094     1.6744   2.514  0.01194 *
## muscle.stiffnessYes      -2.6807     1.1572  -2.317  0.02053 *
## AlopeciaYes               4.8143     1.7073   2.820  0.00480 **
## ObesityYes                0.3897     1.3377   0.291  0.77082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 353.08  on 311  degrees of freedom
## Residual deviance:  55.32  on 295  degrees of freedom
## AIC: 89.32
##
## Number of Fisher Scoring iterations: 21
```

```
#Dataset de treino sem a variável target
BD_teste1 <- BD_teste[,1:16]
```

6 - Previsão do Modelo de Regressão

Após todas as etapas, o modelo treinado é exposto a dados nunca antes vistos. Nesse caso é o dataset de treino e para isso removemos a variável target para que possamos comparar o resultado obtido x observado na Confusion Matrix.

```
resultado_modelo <- predict(modelo1, newdata = BD_teste1, type = 'response')

#Variável resposta com valores convertidos em binário
resultado <- ifelse(resultado_modelo > 0.5, 1, 0)

#Variável target com valores convertidos em binário
BD_teste$Testemodelo <- ifelse(BD_teste$class == "Positive", 1, 0)

#A Confusion Matrix mostra uma acurácia de 84% em relação ao que foi previsto no
# dataset de treino e o observado originalmente no mesmo dataset.

#É uma acurária excelente para a v1 do modelo.
confusionMatrix(table(data = resultado, reference = BD_teste$Testemodelo))
```

```
## Confusion Matrix and Statistics
##
##     reference
## data  0  1
##    0 91  3
##    1 30 84
##
##                Accuracy : 0.8413
##                  95% CI : (0.7845, 0.8882)
##     No Information Rate : 0.5817
##     P-Value [Acc > NIR] : 7.184e-16
##
##                   Kappa : 0.6876
##
##  Mcnemar's Test P-Value : 6.011e-06
##
##             Sensitivity : 0.7521
##             Specificity : 0.9655
##          Pos Pred Value : 0.9681
##          Neg Pred Value : 0.7368
##              Prevalence : 0.5817
##          Detection Rate : 0.4375
##    Detection Prevalence : 0.4519
##       Balanced Accuracy : 0.8588
##
##        'Positive' Class : 0
```

```
##
resultado_final <- prediction(resultado_modelo, BD_teste$Testemodelo)
```

7 - Apresentação do Resultado

Uma das formas de apresentar o resultado é usar a curva ROC. A curva vermelha representa 50% de precisão do modelo. Quanto mais longe e próximo de 1 no eixo Y, melhor para o modelo.

```
# Função para Plot ROC
plot.roc.curve <- function(predictions, title.text){
  perf <- performance(predictions, "tpr", "fpr")
  plot(perf,col = "black",lty = 1, lwd = 2,
       main = title.text, cex.main = 0.6, cex.lab = 0.8,xaxs = "i", yaxs = "i")
  abline(0,1, col = "red")
  auc <- performance(predictions,"auc")
  auc <- unlist(slot(auc, "y.values"))
  auc <- round(auc,2)
  legend(0.4,0.4,legend = c(paste0("AUC: ",auc)), cex = 0.6, bty = "n", box.col = "white")

}

# Plot Curva ROC
plot.roc.curve(resultado_final, title.text = "Curva ROC")
```