Abalone_Regression

2023-04-24

Neste trabalho analiso os dados de https://archive.ics.uci.edu/ml/datasets/abalone.

O abalone é um molusco gastrópode pertencente à família Haliotidae e é encontrado sob a forma de diversas espécies em águas costeiras de quase todo o mundo. Por causa de seu uso como jóia e alimento,há duas espécies de abalone que se encontram em risco de extinção.

Neste projeto, irei prever a idade do abalone baseada em fatores físicos.

A idade do abalone é determinada cortando a casca através do cone, manchando-a e contando o número de anéis através de um microscópio. Outras medidas, mais fáceis de obter, são usadas para prever a idade.

Nome	Tipo de Dado	Unidade de Medida	Descrição
Sex (Sexo)	nominal	–	M, F e I (infantil)
Length (Comprimento)	contínuo	mm	Medição mais longa da concha
Diameter (Diâmetro)	contínuo	mm	Perpendicular ao comprimento
Height (Altura)	contínuo	mm	Com carne na concha
Whole weight (Peso total)	contínuo	gramas	Abalone inteiro
Shucked weight (Peso da carne)	contínuo	gramas	Peso da carne
Viscera weight (Peso das vísceras)	contínuo	gramas	Peso do intestino (após sangria)
Shell weight (Peso da concha)	contínuo	gramas	Depois de seco
Rings (Anéis)	inteiro	–	+1,5 dá a idade em anos

Bibliotecas

```
#install.packages("tidyverse")
#install.packages("ggplot2")
#install.packages("GGally")
#install.packages("ggcorrplot")
#install.packages("DataExplorer")
#install.packages("caret")
#install.packages("corrplot")
#install.packages("doParallel")
#install.packages("caret")
#install.packages("rpart.plot")
#install.packages("rpart")
#install.packages("VIM")
#install.packages("rattle")
#install.packages("RColorBrewer")
```

Chamada das Bibliotecas

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggcorrplot)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(readr)
library(DataExplorer)
library(doParallel)
```

```
## Carregando pacotes exigidos: foreach
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
##
## Carregando pacotes exigidos: iterators
## Carregando pacotes exigidos: parallel
```

```
library(caret)
```

```
## Carregando pacotes exigidos: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(rattle)
```

```
## Carregando pacotes exigidos: bitops
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart.plot)
library(RColorBrewer)
library(VIM)
```

```
## Carregando pacotes exigidos: colorspace
## Carregando pacotes exigidos: grid
## VIM is ready to use.
##
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
##
## The following object is masked from 'package:rattle':
##
##     wine
##
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(ModelMetrics)
```

```
##
## Attaching package: 'ModelMetrics'
##
## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, recall, sensitivity, specificity
##
## The following object is masked from 'package:base':
##
##     kappa
```

Limpando o ambiente de execução

```
rm(list = ls())
```

Setando o Local de trabalho

```
setwd("C:/Users/karin/OneDrive/Desktop/Mestrado/Mineração")
```

Visualização Geral do DataFrame

```r
options(scipen = 999) #visualização dos dados sem a notação científica

abalone <- read_csv("abalone.csv", show_col_types = FALSE)
abalone <- as_tibble(abalone)
abalone
```

```
## # A tibble: 4,177 x 9
##    Sex   Length Diameter Height `Whole weight` `Shucked weight` `Viscera weight`
##    <chr>  <dbl>    <dbl>  <dbl>          <dbl>            <dbl>            <dbl>
##  1 M      0.455    0.365  0.095          0.514           0.224            0.101
##  2 M      0.35     0.265  0.09           0.226           0.0995           0.0485
##  3 F      0.53     0.42   0.135          0.677           0.256            0.142
##  4 M      0.44     0.365  0.125          0.516           0.216            0.114
##  5 I      0.33     0.255  0.08           0.205           0.0895           0.0395
##  6 I      0.425    0.3    0.095          0.352           0.141            0.0775
##  7 F      0.53     0.415  0.15           0.778           0.237            0.142
##  8 F      0.545    0.425  0.125          0.768           0.294            0.150
##  9 M      0.475    0.37   0.125          0.509           0.216            0.112
## 10 F      0.55     0.44   0.15           0.894           0.314            0.151
## # i 4,167 more rows
## # i 2 more variables: `Shell weight` <dbl>, Rings <dbl>
```

O dataset possui 9 atributos e 4177 instâncias

```r
#Atributos
ncol(abalone)
```

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

```r
#Instâncias
nrow(abalone)
```

```
## [1] 4177
```

Dos 9 atributos, 8 são do tipo num e 1 do tipo chr.

```r
str(abalone)
```

```
## tibble [4,177 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Sex           : chr [1:4177] "M" "M" "F" "M" ...
##  $ Length        : num [1:4177] 0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ Diameter      : num [1:4177] 0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ Height        : num [1:4177] 0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ Whole weight  : num [1:4177] 0.514 0.226 0.677 0.516 0.205 ...
##  $ Shucked weight: num [1:4177] 0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ Viscera weight: num [1:4177] 0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ Shell weight  : num [1:4177] 0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ Rings         : num [1:4177] 15 7 9 10 7 8 20 16 9 19 ...
```

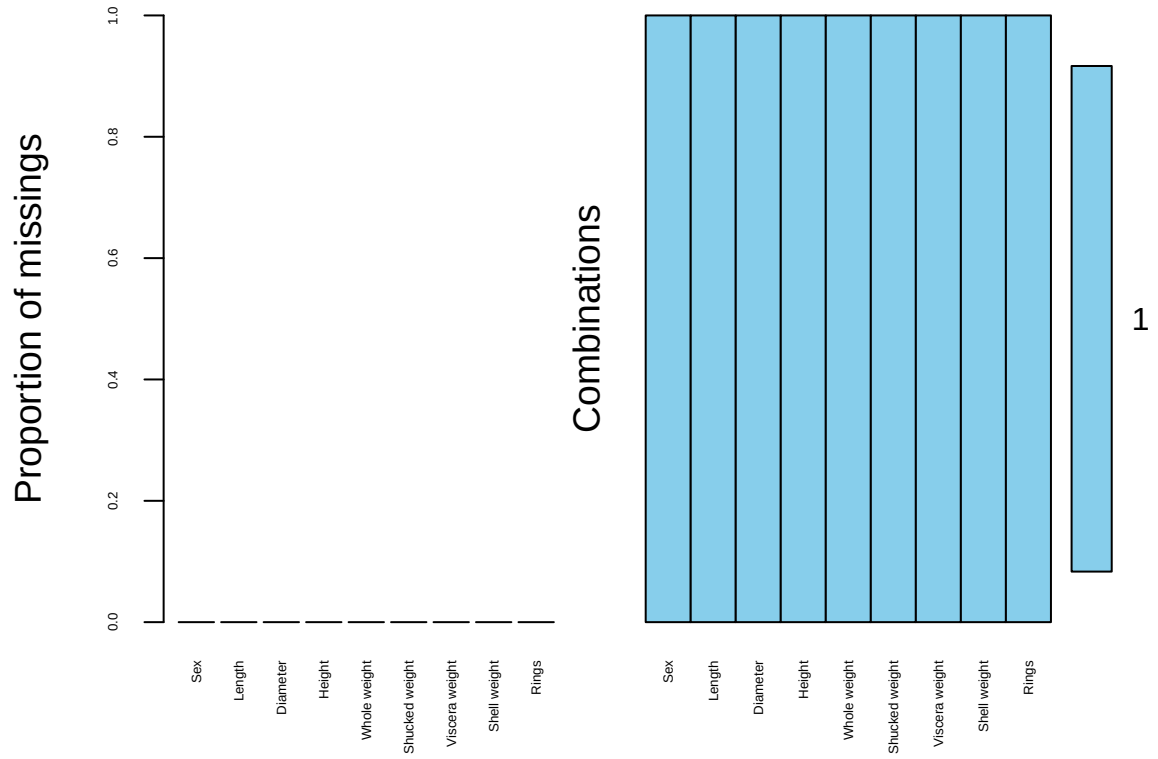
Aqui estão presentes o máximo, mínimo, média e mediana dos atributos númericos.*

```
summary(abalone)
```

```
##      Sex                Length         Diameter          Height
##  Length:4177        Min.   :0.075   Min.   :0.0550   Min.   :0.0000
##  Class :character   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150
##  Mode  :character   Median :0.545   Median :0.4250   Median :0.1400
##                     Mean   :0.524   Mean   :0.4079   Mean   :0.1395
##                     3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650
##                     Max.   :0.815   Max.   :0.6500   Max.   :1.1300
##   Whole weight    Shucked weight   Viscera weight    Shell weight
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.0005   Min.   :0.0015
##  1st Qu.:0.4415   1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300
##  Median :0.7995   Median :0.3360   Median :0.1710   Median :0.2340
##  Mean   :0.8287   Mean   :0.3594   Mean   :0.1806   Mean   :0.2388
##  3rd Qu.:1.1530   3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290
##  Max.   :2.8255   Max.   :1.4880   Max.   :0.7600   Max.   :1.0050
##      Rings
##  Min.   : 1.000
##  1st Qu.: 8.000
##  Median : 9.000
##  Mean   : 9.934
##  3rd Qu.:11.000
##  Max.   :29.000
```

Verificação de dados Missing

```
ppData <- abalone
missPlotData <- aggr(ppData, numbers = TRUE, sortvars = TRUE, labels = names(ppData), cex.axis = 0.4, g
```



Criação a coluna Age

```
abalone$Age <- abalone$Rings
```

Cópia dos dados de Rings para Age e soma de 1.5 para ter a idade

```
abalone$Age <- as.numeric(abalone$Age) #convertendo de inteiro para float
abalone$Age <- abalone$Age + 1.5
abalone
```

```
## # A tibble: 4,177 x 10
##    Sex   Length Diameter Height `Whole weight` `Shucked weight` `Viscera weight`
##    <chr>  <dbl>    <dbl>  <dbl>          <dbl>            <dbl>            <dbl>
##  1 M      0.455    0.365  0.095          0.514           0.224            0.101
##  2 M      0.35     0.265  0.09           0.226           0.0995           0.0485
##  3 F      0.53     0.42   0.135          0.677           0.256            0.142
##  4 M      0.44     0.365  0.125          0.516           0.216            0.114
##  5 I      0.33     0.255  0.08           0.205           0.0895           0.0395
##  6 I      0.425    0.3    0.095          0.352           0.141            0.0775
##  7 F      0.53     0.415  0.15           0.778           0.237            0.142
##  8 F      0.545    0.425  0.125          0.768           0.294            0.150
##  9 M      0.475    0.37   0.125          0.509           0.216            0.112
## 10 F      0.55     0.44   0.15           0.894           0.314            0.151
## # i 4,167 more rows
## # i 3 more variables: `Shell weight` <dbl>, Rings <dbl>, Age <dbl>
```

Retirada da coluna Rings

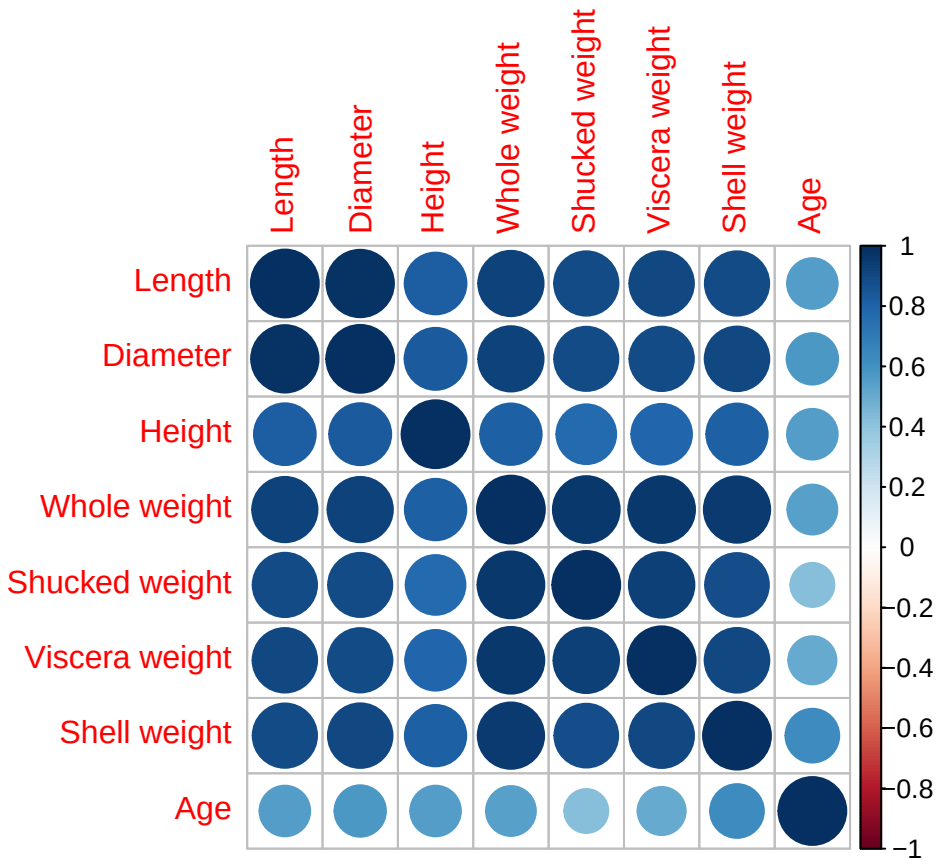
```
myvars <- names(abalone) %in% c("Rings")
abalone <- abalone[!myvars]
str(abalone)
```

```
## tibble [4,177 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Sex           : chr [1:4177] "M" "M" "F" "M" ...
##  $ Length        : num [1:4177] 0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ Diameter      : num [1:4177] 0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ Height        : num [1:4177] 0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ Whole weight  : num [1:4177] 0.514 0.226 0.677 0.516 0.205 ...
##  $ Shucked weight: num [1:4177] 0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ Viscera weight: num [1:4177] 0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ Shell weight  : num [1:4177] 0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ Age           : num [1:4177] 16.5 8.5 10.5 11.5 8.5 9.5 21.5 17.5 10.5 20.5 ...
```

Seleção de Características

Verificação da correlação entre os atributos

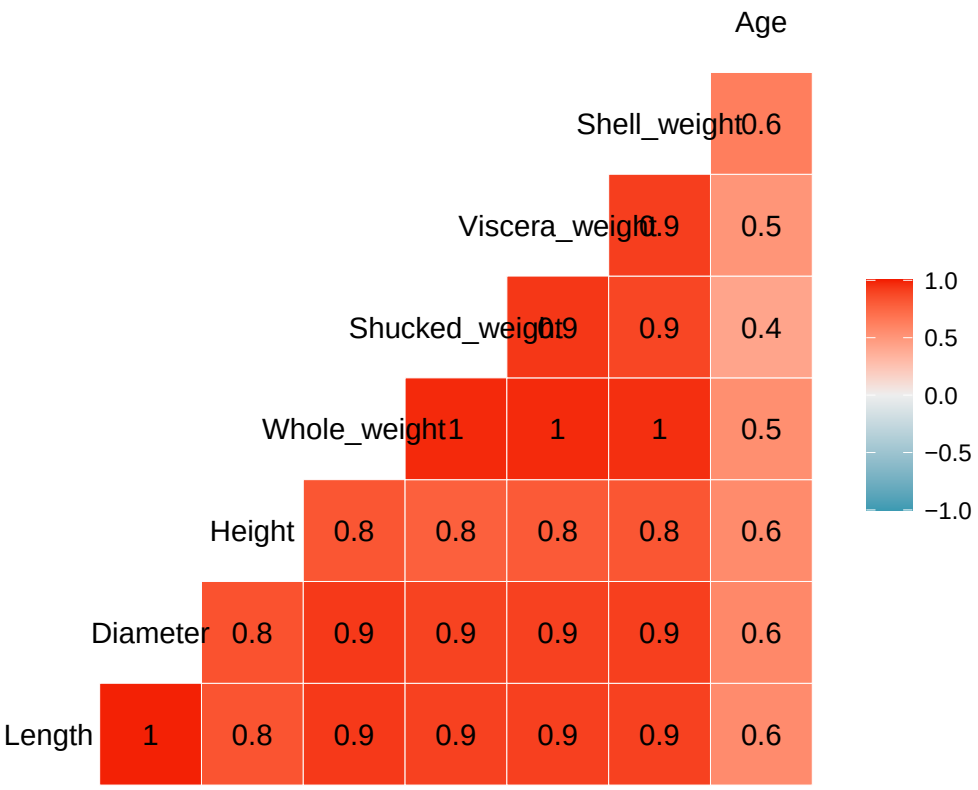
```
numericCol <- unlist(lapply(abalone, is.numeric))
numericData <- abalone[,numericCol]

correlationMatrix <- cor(numericData)
corrplot(correlationMatrix, method = "circle")
```



Como são altamente relacionados, plotei novo gráfico para saber o ponto de corte para a seleção

```
ggcorr(abalone, label=T)
```

```
## Warning in ggcorr(abalone, label = T): data in column(s) 'Sex' are not numeric
## and were ignored
```



Definido o ponto de corte, fiz a seleção

```
p <- 0.9

highlyCorrelated <- findCorrelation(correlationMatrix, cutoff = p, names = TRUE)

myvars <- names(abalone) %in% c(highlyCorrelated)
abalone_regression <- abalone[!myvars]

str(abalone_regression)
```

```
## tibble [4,177 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Sex           : chr [1:4177] "M" "M" "F" "M" ...
##  $ Height        : num [1:4177] 0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ Shucked weight: num [1:4177] 0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ Age           : num [1:4177] 16.5 8.5 10.5 11.5 8.5 9.5 21.5 17.5 10.5 20.5 ...
```

Separação entre treino e teste

```r
set.seed(123)
partition <- createDataPartition(abalone_regression$Age, p=0.75, list = FALSE)

train.set <- abalone_regression[partition,]
test.set <- abalone_regression[-partition,]

#train.set
```

Modelo de Regressão Linear

```r
tc <- trainControl(method = "repeatedcv", number = 10, repeats = 5)

regressao_linear <- train(Age ~. , data = train.set, method = "lm", trControl = tc)

regressao_linear
```

```
## Linear Regression
##
## 3134 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 2820, 2821, 2822, 2821, 2821, 2820, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.469378  0.4154475  1.776433
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Predições - Regressão Linear

```r
predictionsL <- predict(regressao_linear, test.set)

RMSEL <- rmse(abalone_regression$Age,predictionsL)

RMSEL
```

```
## [1] 10.24926
```

Support Vector Machines com Núcleo Linear

```r
train_control <- trainControl(method="repeatedcv", number=10, repeats=3)

svm1 <- train(Age ~., data = train.set, method = "svmLinear", trControl = train_control,  preProcess = c

svm1
```

```
## Support Vector Machines with Linear Kernel
##
```

```
## 3134 samples
##    3 predictor
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 2821, 2821, 2821, 2821, 2820, 2821, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.529422  0.4139736  1.724406
##
## Tuning parameter 'C' was held constant at a value of 1
```

Predições - Support Vector Machines com Núcleo Linear

```
predictionsSVM <- predict(svm1, test.set)

RMSEL <- rmse(abalone_regression$Age,predictionsSVM)

RMSEL
```

```
## [1] 10.24021
```

Fiz duas execuções de maneiras diferentes utilizando modelo linear (SVM com kernel linear e a própria regressão linear). Como esperado, ambos obtiveram RMSE por volta de 2.5,o que significa um desvio padrão de 2.5 dos valores reais. Para o caso do abalone, onde os valores são baixos, considero um resultado mediano para ruim.