Abalone_Classification_Neural

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 concha 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

Limpando o ambiente de execução

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

Setando o Local de trabalho

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

Bibliotecas

```
#install.packages("tidyverse")
#install.packages("ggplot2")
#install.packages("GGally")
#install.packages("ggcorrplot")
#install.packages("DataExplorer")
#install.packages("caret")
#install.packages("VIM")
#install.packages("rattle")
```

```
#install.packages("RColorBrewer")
#install.packages("neuralnet")
#install.packages("sampling")
#install.packages("knitr")
```

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(readr)
library(DataExplorer)
library(caret)
```

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

```
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)
```

```
## Carregando pacotes exigidos: rpart
```

```
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(neuralnet)
```

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

```
library(sampling)
```

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

```
library(knitr)
```

Visualização Geral do DataFrame

```
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

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

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

```
#Instâncias
nrow(abalone)
```

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

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

```
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.*

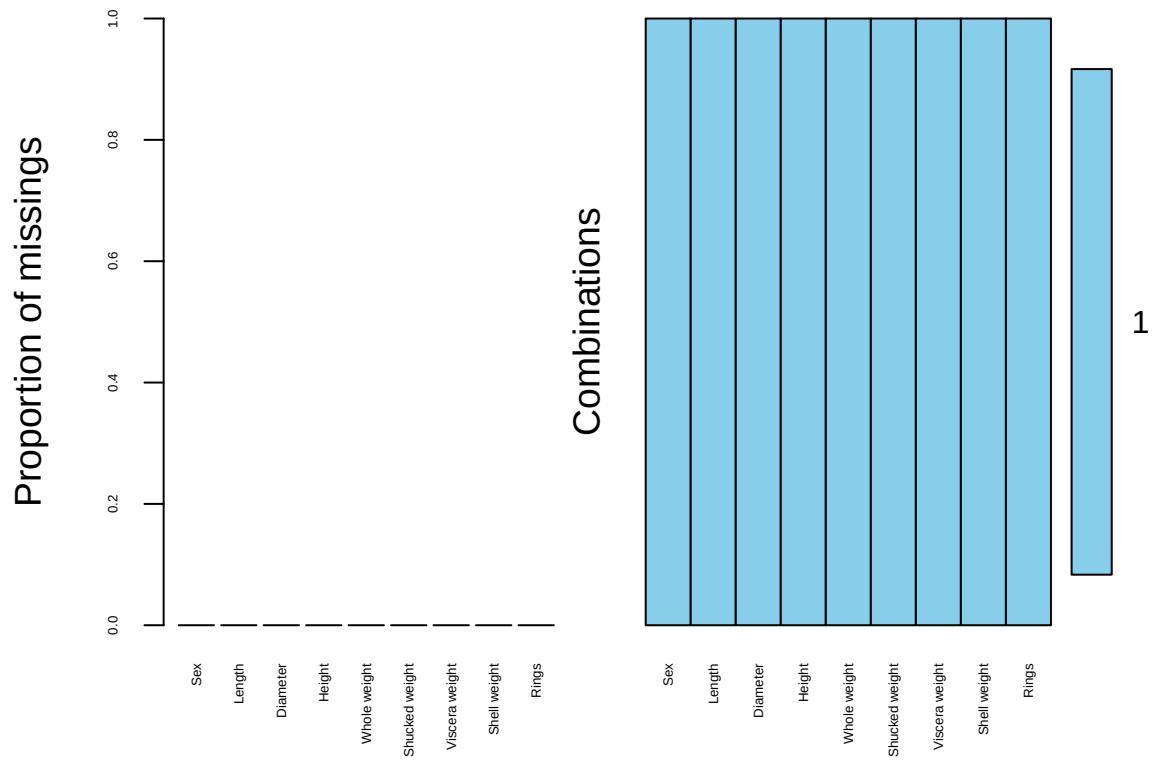
```
summary(abalone)
```

```
##      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
```

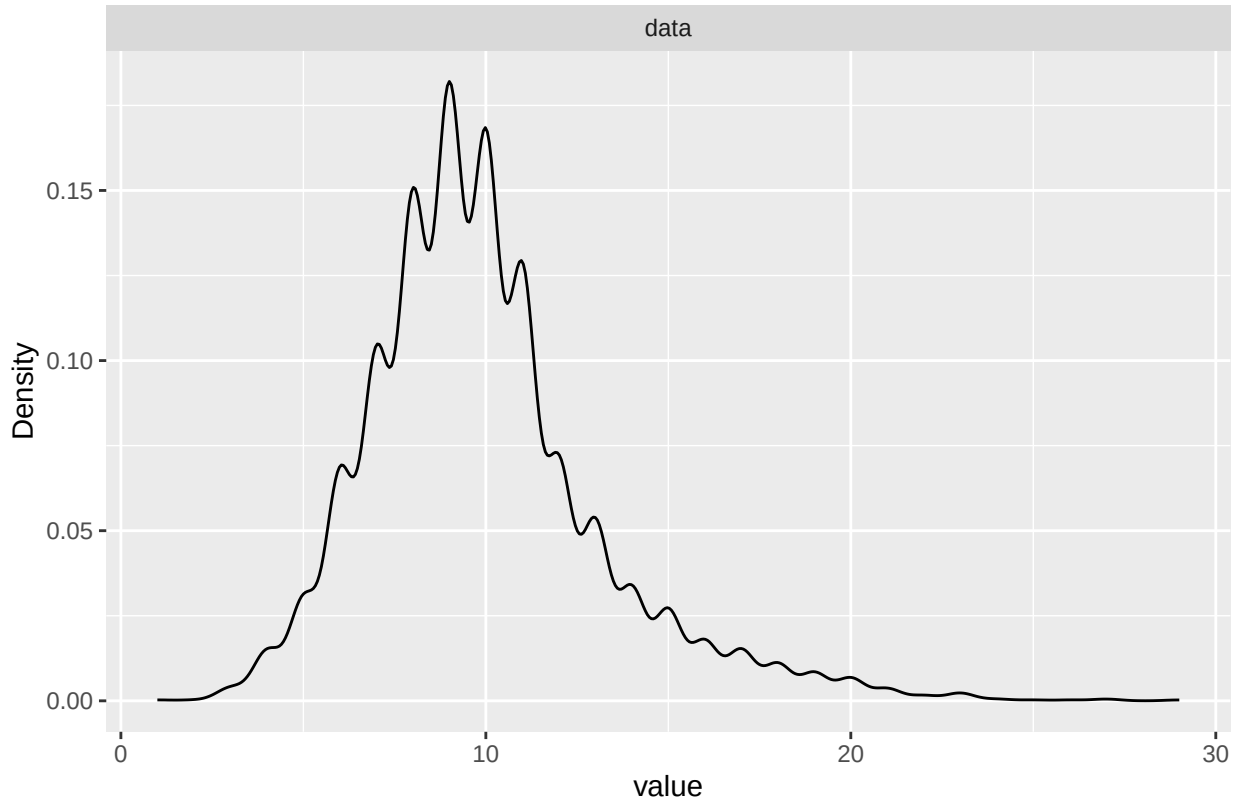
Verificação de dados Missing

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



Categorização da variável Rings em Old, Adult e Young

```
plot_density(abalone$Rings)
```




```
abalone_class <- abalone %>%
  mutate(Age=case_when(
    Rings %in% 1:5 ~ "young",
    Rings %in% 6:13 ~ "adult",
    Rings %in% 14:30 ~ "old"
  ))

#converte AGE em factor
abalone_class$Age <- as.factor(abalone_class$Age)
str(abalone_class)
```

```
## tibble [4,177 x 10] (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 ...
##  $ Age           : Factor w/ 3 levels "adult","old",..: 2 1 1 1 1 1 2 2 1 2 ...
```

Retirada da coluna Rings

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

```
## 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           : Factor w/ 3 levels "adult","old",..: 2 1 1 1 1 1 2 2 1 2 ...
```

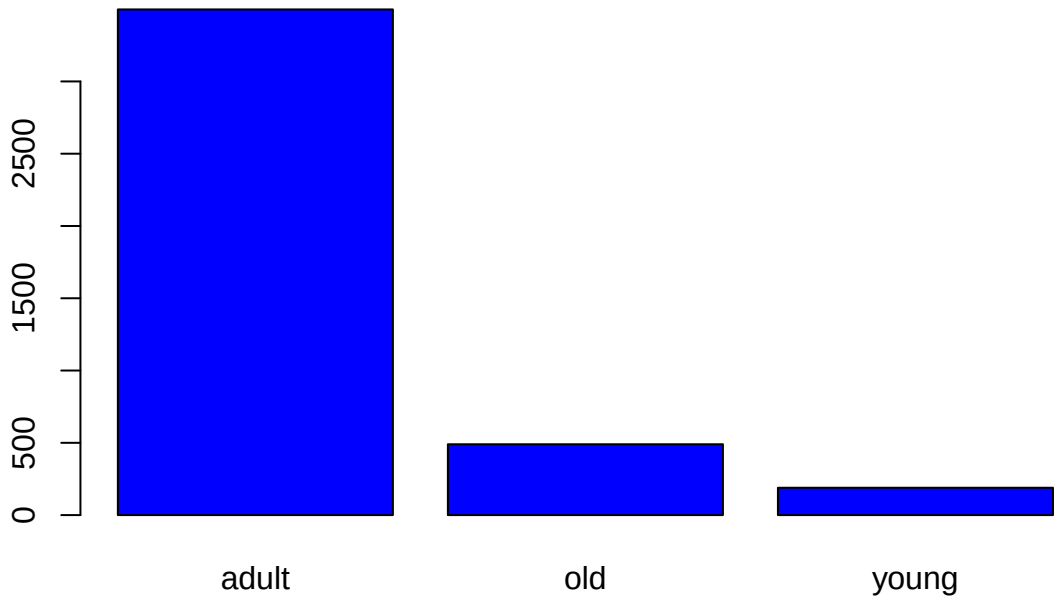
Renomeei os atributos para evitar algum problema com o espaço presente nos nomes

```
abalone_class <- rename(abalone_class, WholeWeight = `Whole weight`, ShuckedWeight = `Shucked weight`, V
abalone_class
```

```
## # A tibble: 4,177 x 9
##    Sex   Length Diameter Height WholeWeight ShuckedWeight VisceraWeight
##    <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: ShellWeight <dbl>, Age <fct>
```

Verificando o balanceamento das classes

```
 barplot(table(abalone_class$Age), col = "blue")
```



Verificação da quantidade de instâncias em cada classe

```
contagem <- table(abalone_class$Age)
contagem
```

```
##
## adult   old young
##  3498   490   189
```

Amostragem

```
id2 <- strata(abalone_class, stratanames="Age", size=c(189,189,189), method="srswor")
abalone_class_amos <- abalone_class %>% slice(id2$ID_unit)
summary(abalone_class_amos)
```

```
##      Sex                Length          Diameter          Height
##  Length:567         Min.   :0.0750   Min.   :0.0550   Min.   :0.0100
##  Class :character   1st Qu.:0.2975   1st Qu.:0.2175   1st Qu.:0.0750
##  Mode  :character   Median :0.5000   Median :0.4000   Median :0.1350
##                     Mean   :0.4615   Mean   :0.3584   Mean   :0.1245
##                     3rd Qu.:0.6000   3rd Qu.:0.4700   3rd Qu.:0.1650
##                     Max.   :0.8150   Max.   :0.6500   Max.   :0.2500
##   WholeWeight     ShuckedWeight    VisceraWeight      ShellWeight
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.00050   Min.   :0.0015
##  1st Qu.:0.1242   1st Qu.:0.0560   1st Qu.:0.02675   1st Qu.:0.0375
```

```
##  Median :0.6605   Median :0.2660   Median :0.14400   Median :0.2000
##  Mean   :0.7045   Mean   :0.2891   Mean   :0.15245   Mean   :0.2134
##  3rd Qu.:1.0962   3rd Qu.:0.4600   3rd Qu.:0.24225   3rd Qu.:0.3275
##  Max.   :2.8255   Max.   :1.1465   Max.   :0.52350   Max.   :0.8970
##     Age
##  adult:189
##  old  :189
##  young:189
##
##
##
```

Separação entre treino e teste

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

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

#test.set
train.set
```

```
## # A tibble: 426 x 9
##    Sex   Length Diameter Height WholeWeight ShuckedWeight VisceraWeight
##    <chr>  <dbl>    <dbl>  <dbl>       <dbl>         <dbl>         <dbl>
##  1 F      0.545    0.425  0.125       0.768         0.294         0.150
##  2 F      0.55     0.44   0.15        0.894         0.314         0.151
##  3 F      0.525    0.38   0.14        0.606         0.194         0.148
##  4 F      0.68     0.56   0.165       1.64          0.606         0.280
##  5 M      0.665    0.525  0.165       1.34          0.552         0.358
##  6 F      0.52     0.425  0.165       0.988         0.396         0.225
##  7 M      0.595    0.475  0.16        1.32          0.408         0.234
##  8 M      0.665    0.535  0.195       1.61          0.576         0.388
##  9 M      0.55     0.435  0.145       0.843         0.328         0.192
## 10 M      0.53     0.435  0.16        0.883         0.316         0.164
## # i 416 more rows
## # i 2 more variables: ShellWeight <dbl>, Age <fct>
```

Modelo e plot da Rede Neural

```
  Abalone_Neural_Net <- neuralnet((Age == "young") +
                (Age == "adult") +
                (Age == "old") ~
                 Length+
                 Diameter+
                 Height+
                 WholeWeight+
                 ShuckedWeight+
                 VisceraWeight+
                 ShellWeight,
               train.set,
               hidden = c(3,3),
```

```
                threshold = 0.7,
                stepmax = 1e+05,
                learningrate=0.01,
                algorithm = "backprop",
                linear.output = FALSE)

pred2 <- predict(Abalone_Neural_Net, test.set)
```

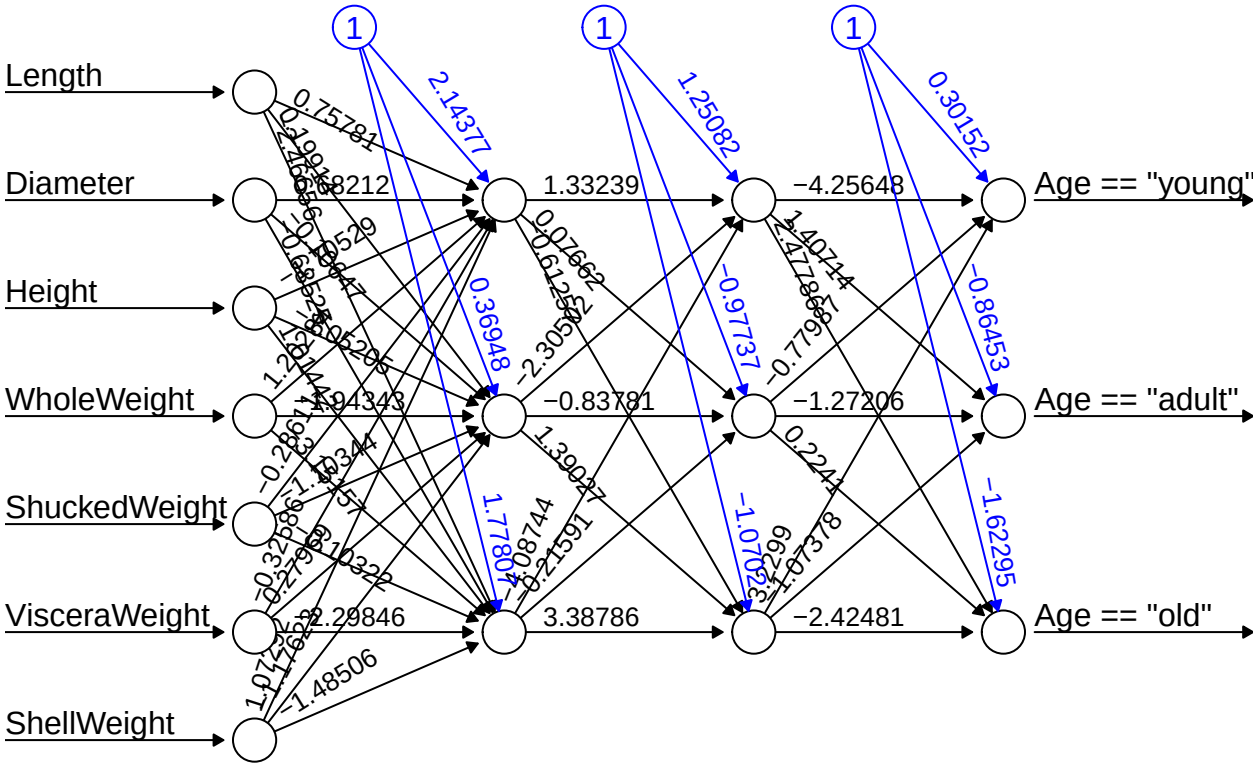
Predições e Matriz de Confusão

```
a<-apply(pred2, 1, which.max)

a[a==1]<-"young"
a[a==2]<-"adult"
a[a==3]<-"old"

a<-factor(a,levels = c("adult","old","young"))
result2<-table(test.set$Age,a)

cm <- confusionMatrix(result2, mode = "everything")
cm
```

```
## Confusion Matrix and Statistics
##
##        a
##         adult old young
##   adult     6  37     4
##   old       1  45     1
##   young     1   0    46
##
## Overall Statistics
##
##                Accuracy : 0.6879
##                  95% CI : (0.6045, 0.7633)
##     No Information Rate : 0.5816
##     P-Value [Acc > NIR] : 0.006067
##
##                   Kappa : 0.5319
##
##  Mcnemar's Test P-Value : 0.00000004819
##
## Statistics by Class:
##
##                      Class: adult Class: old Class: young
## Sensitivity               0.75000     0.5488       0.9020
## Specificity               0.69173     0.9661       0.9889
## Pos Pred Value            0.12766     0.9574       0.9787
## Neg Pred Value            0.97872     0.6064       0.9468
## Precision                 0.12766     0.9574       0.9787
## Recall                    0.75000     0.5488       0.9020
## F1                        0.21818     0.6977       0.9388
## Prevalence                0.05674     0.5816       0.3617
## Detection Rate            0.04255     0.3191       0.3262
```

```
## Detection Prevalence      0.33333     0.3333       0.3333
## Balanced Accuracy         0.72086     0.7574       0.9454
```

```
plot(Abalone_Neural_Net,rep = "best")
```



Error: 80.204208 Steps: 171

Utilizando todo o dataset, o modelo prediz que todas as instâncias são da classe "adult". Comportamento este esperado tendo em vista o desbalanceamento. Com a amostragem, embora a acurácia tenha dimunído, o modelo acertou mais de cada classe. Assim optei por utilizar a amostragem

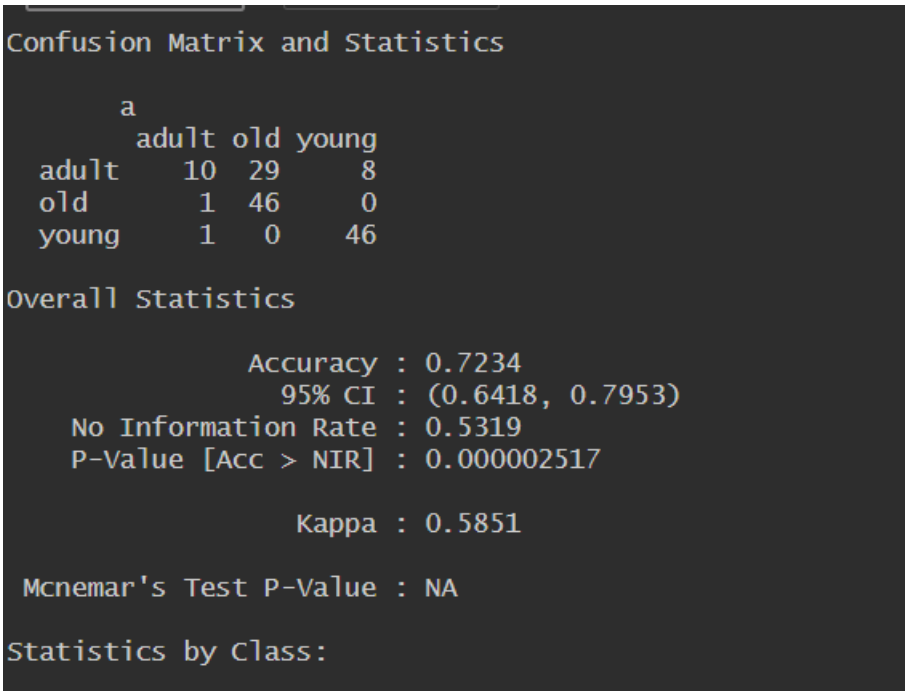


Figure 1: Teste da Rede Neural com a Amostragem

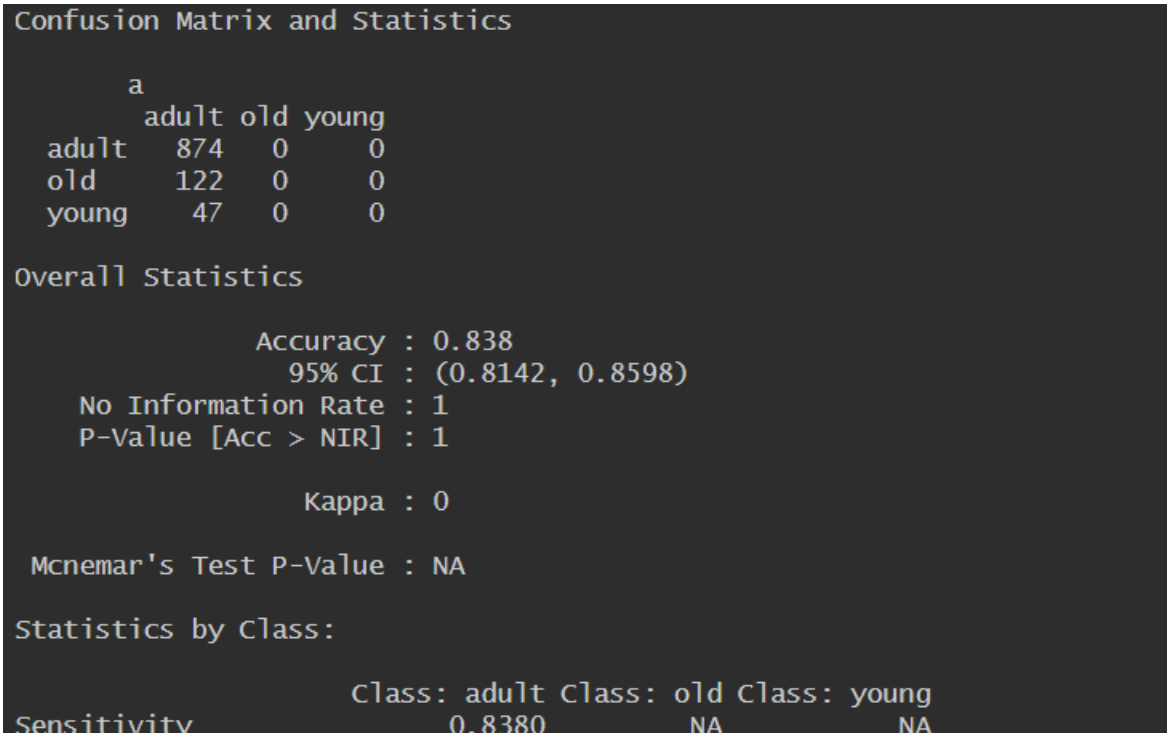


Figure 2: Teste da Rede Neural Com Todos os Dados