Prevendo Ocorrência Cancer

Hudson Santos

2024-04-28

Prevendo Ocorrencia Câncer de mama

Os dados do câncer da mama incluem 569 observações de biópsias de câncer, cada um com 32 características (variáveis). Uma característica é um número de identificação (ID), outro é o diagnóstico de câncer, e 30 são medidas laboratoriais numéricas. O diagnóstico é codificado como “M” para indicar maligno ou “B” para indicar benigno.

Etapa 1 - Coletando Dados

```
#Coletando dados
dados <- read.csv("bc_data.csv", stringsAsFactors = FALSE)
str(dados)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                : int  87139402 8910251 905520 868871 9012568
906539 925291 87880 862989 89827 ...
##  $ diagnosis         : chr  "B" "B" "B" "B" ...
##  $ radius_mean       : num  12.3 10.6 11 11.3 15.2 ...
##  $ texture_mean      : num  12.4 18.9 16.8 13.4 13.2 ...
##  $ perimeter_mean    : num  78.8 69.3 70.9 73 97.7 ...
##  $ area_mean         : num  464 346 373 385 712 ...
##  $ smoothness_mean   : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
##  $ compactness_mean  : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
##  $ concavity_mean    : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
##  $ points_mean       : num  0.037 0.0264 0.0248 0.048 0.0266 ...
##  $ symmetry_mean     : num  0.196 0.192 0.171 0.177 0.172 ...
##  $ dimension_mean    : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
##  $ radius_se         : num  0.236 0.451 0.197 0.338 0.178 ...
##  $ texture_se        : num  0.666 1.197 1.387 1.343 0.412 ...
##  $ perimeter_se      : num  1.67 3.43 1.34 1.85 1.34 ...
##  $ area_se           : num  17.4 27.1 13.5 26.3 17.7 ...
##  $ smoothness_se     : num  0.00805 0.00747 0.00516 0.01127 0.00501 ...
##  $ compactness_se    : num  0.0118 0.03581 0.00936 0.03498 0.01485 ...
##  $ concavity_se      : num  0.0168 0.0335 0.0106 0.0219 0.0155 ...
##  $ points_se         : num  0.01241 0.01365 0.00748 0.01965 0.00915 ...
##  $ symmetry_se       : num  0.0192 0.035 0.0172 0.0158 0.0165 ...
##  $ dimension_se      : num  0.00225 0.00332 0.0022 0.00344 0.00177 ...
##  $ radius_worst      : num  13.5 11.9 12.4 11.9 16.2 ...
##  $ texture_worst     : num  15.6 22.9 26.4 15.8 15.7 ...
##  $ perimeter_worst   : num  87 78.3 79.9 76.5 104.5 ...
##  $ area_worst        : num  549 425 471 434 819 ...
```

```
##  $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
##  $ compactness_worst: num  0.127 0.252 0.148 0.182 0.174 ...
##  $ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
##  $ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
##  $ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
##  $ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
```

Etapa 2 - Explorando os Dados

```
#Para evitar superajuste, sempre exclua a variável ID ao construir
modelos de #aprendizado de máquina, pois o ID pode levar a previsões
incorretas e dificultar a #generalização para novos dados
dados <- dados[-1]
str(dados)
```

```
## 'data.frame':    569 obs. of  31 variables:
##  $ diagnosis        : chr  "B" "B" "B" "B" ...
##  $ radius_mean      : num  12.3 10.6 11 11.3 15.2 ...
##  $ texture_mean     : num  12.4 18.9 16.8 13.4 13.2 ...
##  $ perimeter_mean   : num  78.8 69.3 70.9 73 97.7 ...
##  $ area_mean        : num  464 346 373 385 712 ...
##  $ smoothness_mean  : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
##  $ compactness_mean : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
##  $ concavity_mean   : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
##  $ points_mean      : num  0.037 0.0264 0.0248 0.048 0.0266 ...
##  $ symmetry_mean    : num  0.196 0.192 0.171 0.177 0.172 ...
##  $ dimension_mean   : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
##  $ radius_se        : num  0.236 0.451 0.197 0.338 0.178 ...
##  $ texture_se       : num  0.666 1.197 1.387 1.343 0.412 ...
##  $ perimeter_se     : num  1.67 3.43 1.34 1.85 1.34 ...
##  $ area_se          : num  17.4 27.1 13.5 26.3 17.7 ...
##  $ smoothness_se    : num  0.00805 0.00747 0.00516 0.01127 0.00501 ...
##  $ compactness_se   : num  0.0118 0.03581 0.00936 0.03498 0.01485 ...
##  $ concavity_se     : num  0.0168 0.0335 0.0106 0.0219 0.0155 ...
##  $ points_se        : num  0.01241 0.01365 0.00748 0.01965 0.00915 ...
##  $ symmetry_se      : num  0.0192 0.035 0.0172 0.0158 0.0165 ...
##  $ dimension_se     : num  0.00225 0.00332 0.0022 0.00344 0.00177 ...
##  $ radius_worst     : num  13.5 11.9 12.4 11.9 16.2 ...
##  $ texture_worst    : num  15.6 22.9 26.4 15.8 15.7 ...
##  $ perimeter_worst  : num  87 78.3 79.9 76.5 104.5 ...
##  $ area_worst       : num  549 425 471 434 819 ...
##  $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
##  $ compactness_worst: num  0.127 0.252 0.148 0.182 0.174 ...
##  $ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
##  $ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
##  $ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
##  $ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
```

```
any(is.na(dados))
```

```
## [1] FALSE
```

```r
#Algumas variaveis precisam ser reclassificadas para o tipo fator
table(dados$diagnosis)
```

```
##
##   B   M
## 357 212
```

```r
dados$diagnosis <- factor(dados$diagnosis, levels = c("B", "M"), labels =
c("Benigno", "Maligno"))
str(dados$diagnosis)
```

```
##  Factor w/ 2 levels "Benigno","Maligno": 1 1 1 1 1 1 1 2 1 1 ...
```

```r
#Verificando proporção
round(prop.table(table(dados$diagnosis)) * 100, digits = 1)
```

```
##
## Benigno Maligno
##    62.7    37.3
```

```r
#Em relação às medidas de tendência central, a detecção de um problema de
escala #nos dados indica a necessidade de normalização. No kNN, a
precisão do cálculo de #distância depende da escala dos dados de entrada
summary(dados[c("radius_mean", "area_mean", "smoothness_mean")])
```

```
##   radius_mean       area_mean      smoothness_mean
##  Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
##  1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
##  Median :13.370   Median : 551.1   Median :0.09587
##  Mean   :14.127   Mean   : 654.9   Mean   :0.09636
##  3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
##  Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

```r
# Criando um função de normalização
normalizar <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

# Testando a função de normalização - os resultados devem ser idênticos
normalizar(c(1, 2, 3, 4, 5))
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

```r
normalizar(c(10, 20, 30, 40, 50))
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

```r
# Normalizando os dados
dados_norm <- as.data.frame(lapply(dados[2:31], normalizar))

# Confirmando se a normalização funcionou
summary(dados[c("radius_mean", "area_mean", "smoothness_mean")])
```

```
##   radius_mean       area_mean      smoothness_mean
##  Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
##  1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
##  Median :13.370   Median : 551.1   Median :0.09587
##  Mean   :14.127   Mean   : 654.9   Mean   :0.09636
##  3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
##  Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

```
summary(dados_norm[c("radius_mean", "area_mean", "smoothness_mean")])
```

```
##   radius_mean       area_mean      smoothness_mean
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2233   1st Qu.:0.1174   1st Qu.:0.3046
##  Median :0.3024   Median :0.1729   Median :0.3904
##  Mean   :0.3382   Mean   :0.2169   Mean   :0.3948
##  3rd Qu.:0.4164   3rd Qu.:0.2711   3rd Qu.:0.4755
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

Etapa 3: Treinando o modelo

```
# Carregando o pacote library
#install.packages("class")
library(class)
```

```
## Warning: package 'class' was built under R version 4.2.3
```

```
# Criando dados de treino e dados de teste
dados_treino <- dados_norm[1:469, ]
dados_teste <- dados_norm[470:569, ]

# Criando os labels para os dados de treino e de teste
dados_treino_labels <- dados[1:469, 1]
dados_teste_labels <- dados[470:569, 1]
length(dados_treino_labels)
```

```
## [1] 469
```

```
length(dados_teste_labels)
```

```
## [1] 100
```

```
# Criando o modelo
#A função knn() gera previsões para cada exemplo no conjunto de teste,
#apresentando os resultados em um objeto de fator
modelo <- knn(train = dados_treino,
              test = dados_teste,
              cl = dados_treino_labels,
              k = 21)
```

Etapa 4: Avaliando e Interpretando o Modelo

```
# Carregando o gmodels
#install.packages("gmodels")
library(gmodels)

## Warning: package 'gmodels' was built under R version 4.2.3

# Criando uma tabela cruzada dos dados previstos x dados atuais
# Usaremos amostra com 100 observações: length(dados_teste_labels)
CrossTable(x = dados_teste_labels, y = modelo, prop.chisq = FALSE)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                      | modelo
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## -------------------|-----------|-----------|-----------|
##            Benigno |        61 |         0 |        61 |
##                    |     1.000 |     0.000 |     0.610 |
##                    |     0.968 |     0.000 |           |
##                    |     0.610 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##            Maligno |         2 |        37 |        39 |
##                    |     0.051 |     0.949 |     0.390 |
##                    |     0.032 |     1.000 |           |
##                    |     0.020 |     0.370 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        63 |        37 |       100 |
##                    |     0.630 |     0.370 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

A tabela de confusão apresenta quatro cenários: verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo. Ela compara as previsões do modelo com os dados observados. Por exemplo, 'verdadeiro negativo' significa que o modelo previu corretamente a ausência da doença, enquanto os dados confirmaram isso. A taxa de acerto do modelo é de 98%, indicando que acertou 98 de 100 casos. Os erros do

modelo são classificados como falso positivo (Erro Tipo I) e falso negativo (Erro Tipo II).

Etapa 5: Otimizando a performance do modelo

```
# Usando a função scale() para padronizar o z-score
?scale()
```

```
## starting httpd help server ... done
```

```
dados_z <- as.data.frame(scale(dados[-1]))

# Confirmando transformação realizada com sucesso
summary(dados_z$area_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -1.4532 -0.6666 -0.2949  0.0000  0.3632  5.2459
```

```
# Criando novos datasets de treino e de teste
dados_treino <- dados_z[1:469, ]
dados_teste <- dados_z[470:569, ]

dados_treino_labels <- dados[ 1: 469, 1]
dados_teste_labels <- dados[ 470: 569, 1]

# Reclassificando
modelo_v2 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 21)

# Criando uma tabela cruzada dos dados previstos x dados atuais
CrossTable(x = dados_teste_labels, y = modelo_v2, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                      | modelo_v2
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## -------------------|-----------|-----------|-----------|
##            Benigno |        61 |         0 |        61 |
```

```
##                     |     1.000 |     0.000 |     0.610 |
##                     |     0.924 |     0.000 |           |
##                     |     0.610 |     0.000 |           |
## --------------------|-----------|-----------|-----------|
##             Maligno |         5 |        34 |        39 |
##                     |     0.128 |     0.872 |     0.390 |
##                     |     0.076 |     1.000 |           |
##                     |     0.050 |     0.340 |           |
## --------------------|-----------|-----------|-----------|
##       Column Total |        66 |        34 |       100 |
##                     |     0.660 |     0.340 |           |
## --------------------|-----------|-----------|-----------|
##
##
```

```
# Valores diferentes para k
modelo_v3 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 1)
CrossTable(x = dados_teste_labels, y = modelo_v3, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | modelo_v3
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## --------------------|-----------|-----------|-----------|
##             Benigno |        58 |         3 |        61 |
##                     |     0.951 |     0.049 |     0.610 |
##                     |     0.983 |     0.073 |           |
##                     |     0.580 |     0.030 |           |
## --------------------|-----------|-----------|-----------|
##             Maligno |         1 |        38 |        39 |
##                     |     0.026 |     0.974 |     0.390 |
##                     |     0.017 |     0.927 |           |
##                     |     0.010 |     0.380 |           |
## --------------------|-----------|-----------|-----------|
##       Column Total |        59 |        41 |       100 |
```

```
##                     |     0.590 |     0.410 |           |
## --------------------|-----------|-----------|-----------|
##
##
```

```
modelo_v4 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 5)
CrossTable(x = dados_teste_labels, y = modelo_v4, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | modelo_v4
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## --------------------|-----------|-----------|-----------|
##             Benigno |        61 |         0 |        61 |
##                     |     1.000 |     0.000 |     0.610 |
##                     |     0.968 |     0.000 |           |
##                     |     0.610 |     0.000 |           |
## --------------------|-----------|-----------|-----------|
##             Maligno |         2 |        37 |        39 |
##                     |     0.051 |     0.949 |     0.390 |
##                     |     0.032 |     1.000 |           |
##                     |     0.020 |     0.370 |           |
## --------------------|-----------|-----------|-----------|
##       Column Total |        63 |        37 |       100 |
##                     |     0.630 |     0.370 |           |
## --------------------|-----------|-----------|-----------|
##
##
```

```
modelo_v5 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 11)
CrossTable(x = dados_teste_labels, y = modelo_v5, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | modelo_v5
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## -------------------|-----------|-----------|-----------|
##            Benigno |        61 |         0 |        61 |
##                    |     1.000 |     0.000 |     0.610 |
##                    |     0.953 |     0.000 |           |
##                    |     0.610 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##            Maligno |         3 |        36 |        39 |
##                    |     0.077 |     0.923 |     0.390 |
##                    |     0.047 |     1.000 |           |
##                    |     0.030 |     0.360 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        64 |        36 |       100 |
##                    |     0.640 |     0.360 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```
modelo_v6 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 15)
CrossTable(x = dados_teste_labels, y = modelo_v6, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
```

```
## Total Observations in Table:  100
##
##
##                     | modelo_v6
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## -------------------|-----------|-----------|-----------|
##            Benigno |        61 |         0 |        61 |
##                    |     1.000 |     0.000 |     0.610 |
##                    |     0.953 |     0.000 |           |
##                    |     0.610 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##            Maligno |         3 |        36 |        39 |
##                    |     0.077 |     0.923 |     0.390 |
##                    |     0.047 |     1.000 |           |
##                    |     0.030 |     0.360 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        64 |        36 |       100 |
##                    |     0.640 |     0.360 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```
modelo_v7 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 27)
CrossTable(x = dados_teste_labels, y = modelo_v7, prop.chisq = FALSE)
```

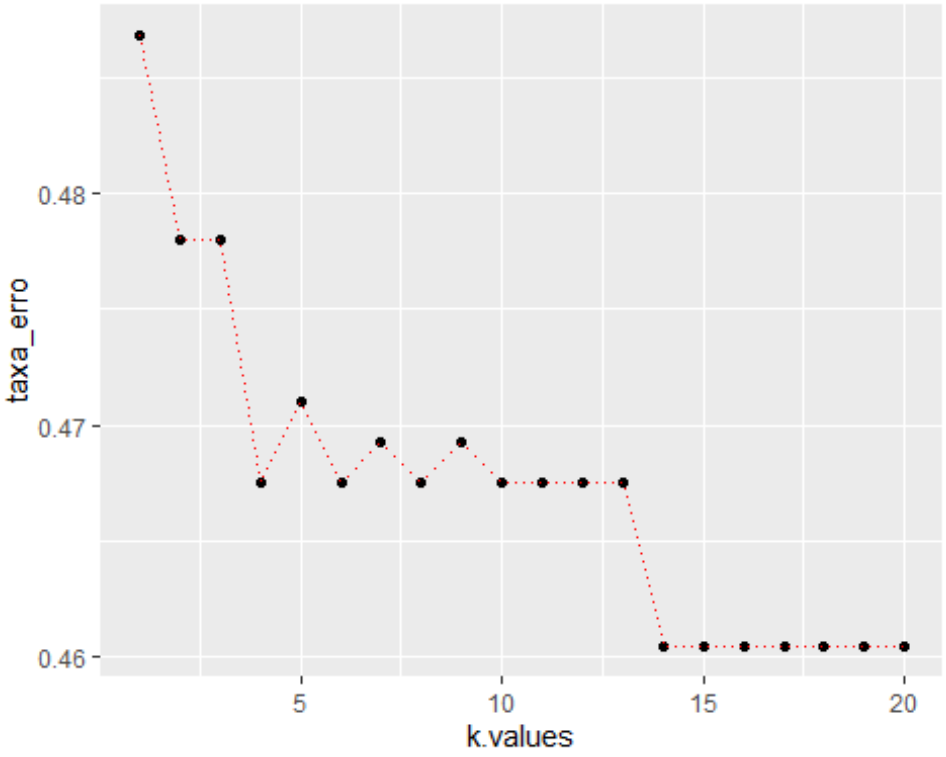
```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | modelo_v7
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## -------------------|-----------|-----------|-----------|
##            Benigno |        61 |         0 |        61 |
##                    |     1.000 |     0.000 |     0.610 |
##                    |     0.938 |     0.000 |           |
##                    |     0.610 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
```

```
##             Maligno |         4 |        35 |        39 |
##                     |     0.103 |     0.897 |     0.390 |
##                     |     0.062 |     1.000 |           |
##                     |     0.040 |     0.350 |           |
## --------------------|-----------|-----------|-----------|
##       Column Total |        65 |        35 |       100 |
##                     |     0.650 |     0.350 |           |
## --------------------|-----------|-----------|-----------|
##
##
```

```
modelo_v2 <- knn(train = dados_treino,
                 test = dados_teste,
                 cl = dados_treino_labels,
                 k = 21)
CrossTable(x = dados_teste_labels, y = modelo_v2, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | modelo_v2
## dados_teste_labels |   Benigno |   Maligno | Row Total |
## --------------------|-----------|-----------|-----------|
##             Benigno |        61 |         0 |        61 |
##                     |     1.000 |     0.000 |     0.610 |
##                     |     0.968 |     0.000 |           |
##                     |     0.610 |     0.000 |           |
## --------------------|-----------|-----------|-----------|
##             Maligno |         2 |        37 |        39 |
##                     |     0.051 |     0.949 |     0.390 |
##                     |     0.032 |     1.000 |           |
##                     |     0.020 |     0.370 |           |
## --------------------|-----------|-----------|-----------|
##       Column Total |        63 |        37 |       100 |
##                     |     0.630 |     0.370 |           |
## --------------------|-----------|-----------|-----------|
##
##
```

```r
## Calculando a taxa de erro
prev = NULL
taxa_erro = NULL

suppressWarnings(
for(i in 1:20){
  set.seed(101)
  prev = knn(train = dados_treino, test = dados_teste, cl =
dados_treino_labels, k = i)
  taxa_erro[i] = mean(dados$diagnosis != prev)
})

# Obtendo os valores de k e das taxas de erro
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.2.3

k.values <- 1:20
df_erro <- data.frame(taxa_erro, k.values)
df_erro

##    taxa_erro k.values
## 1  0.4868190        1
## 2  0.4780316        2
## 3  0.4780316        3
## 4  0.4674868        4
## 5  0.4710018        5
## 6  0.4674868        6
## 7  0.4692443        7
## 8  0.4674868        8
## 9  0.4692443        9
## 10 0.4674868       10
## 11 0.4674868       11
## 12 0.4674868       12
## 13 0.4674868       13
## 14 0.4604569       14
## 15 0.4604569       15
## 16 0.4604569       16
## 17 0.4604569       17
## 18 0.4604569       18
## 19 0.4604569       19
## 20 0.4604569       20

# À medida que aumentamos k, diminuímos a taxa de erro do modelo
ggplot(df_erro, aes(x = k.values, y = taxa_erro)) + geom_point()+
geom_line(lty = "dotted", color = 'red')
```


0.48
0.47
0.46
taxa_erro
5
10
15
20
k.values