Solução Lista 01

Nome: Leandro dos Anjos
E-mail: leandro.anjos@aluno.ufabc.edu.br

23 fevereiro, 2025

Exercício 01

Problemas de classificação: O aprendizado por classificação é utilizado para problemas em que os valores os quais queremos determinar, são valores discretos, em que aplicamos um rótulo ou classe a um conjunto de pontos (x, y), determinamos a f:x->y que melhor "Prevê" a classe de futuros x. Conforme dito anteriormente, o vetor características desse tipo de aprendizado é dado por um ponto x e uma classe y, que o define. Sendo que uma mesma classe y, pode ter várias caracterísitas, e cada uma vai ser uma dimensão do nosso problema.

Problema de regressão: Esse aprendizado é útil para problemas em que o valor de x é contínuo e não mais discreto, logo para um determinado x vamos estimar um valor de y contínuo que o represente da melhor forma possível. Um exemplo é dado o peso e idade de uma pessoa (fatores X), tentar estimar a altura dela (fator Y). O vetor característica desse problema é um conjunto de pontos x com valores contínuos y (Também, podendo ter mais de uma característica, como no exemplo acima)

Problema de Agrupamento: Esse tipo de aprendizado é utilizado para problemas em que não sabemos os rótulos dos pontos em uma determinada base de dados, e queremos agrupar eles da melhor forma possível. Basicamente o vetor características desse tipo de aprendizado é um conjunto de características x e não nos é dado a característica y.

Exercício 02

A maldição da dimensionalidade é um conceito que afirma que conforme a dimensão de um problema aumenta, a distância média entre os pontos da base de dados tende a aumentar também, tendo em vista que agora os pontos tem uma possibilidade maior de posições. Esse conceito, por exemplo, se aplica a técnica de aproximação de distribuição chamada Knn. Essa técnica estuda os K pontos mais próximos de um determinado X. Ele é bom para problemas com poucas dimensões, mas conforme as dimensões vão aumentando, os vizinhos mais próximos vão ficando cada vez mais distantes e a utilização desse método deve ser questionada para tal problema. Geralmente ele funciona bem quando o problemas tem muitos pontos para um espaço considerávelmente menor.

Exercício 03

```
library(tidyverse)

define_classe <- function(k,x,D){
  D2 <- D %>%
  mutate( dist = (x[1] - x_1)^2 + (x[2] - x_2)^2 ) %>%
  arrange( dist ) %>% head(k) %>% count(y)
```

```
  ultimo_elemento <- D2 %>%
    arrange(desc(n)) %>%
    slice(1) %>%
    pull(y)  # Pega o valor da coluna 'y' da primeira linha

  return(ultimo_elemento)
}

D <- tibble( x_1 = rnorm(100,1,1),
  x_2 = rnorm(100,-1,2),
  y = factor(sample(c("one","two","three"),100,replace = T)))
head(D)
```

```
## # A tibble: 6 x 3
##      x_1    x_2 y
##    <dbl>  <dbl> <fct>
## 1  3.06  -3.63  one
## 2 -0.614 -3.24  two
## 3  0.422  1.48  two
## 4 -0.709 -3.41  one
## 5  1.05  -2.25  two
## 6 -0.289  0.443 one
```

```
x = c(1,2)
k = 10

print("A classe mais provável é: ")
```

```
## [1] "A classe mais provável é: "
```

```
print(define_classe(k,x,D))
```

```
## [1] two
## Levels: one three two
```

Exercício 04

```
library(tidyverse)

#Copiando a função do exercício anterior para usar aqui
define_especie <- function(k,x,D){
  D2 <- D %>%
  mutate( dist = (x[1] - x_1)^2 + (x[2] - x_2)^2 ) %>%
  arrange( dist ) %>% head(k) %>% count(y)

  ultimo_elemento <- D2 %>%
    arrange(desc(n)) %>%
    slice(1) %>%
    pull(y)  # Pega o valor da coluna 'y' da primeira linha
```

```r
  return(ultimo_elemento)
}

#Pegando o banco de dado iris
data("iris") # Carrega o banco no ambiente global
iris <- as_tibble(iris) %>% # Converte para a dataframe tibble
select(Petal.Length,Sepal.Length,Species) %>% # Seleciona colunas da dataframe
rename( x_1 = Petal.Length, x_2 = Sepal.Length, y = Species) # Renomeia as colunas

#Para verificar se o algorítimo classifica bem os pontos, eu vou dividir os dados em 90%
#para teste e 10% para testarmos (Pois já devemos saber o resultado correto, então não
#podemos escolher um ponto novo e aleatório)
set.seed(42)
indices <- sample(1:nrow(iris), size = 0.9 * nrow(iris))
treino <- iris[indices, ]
teste <- iris[-indices, ]

list_test <- as.data.frame(teste)

#Aqui eu vou analisar k = 1 até k = 15
for(k in 1:15){
  acertos = 0
  total = nrow(list_test)

  for(i in 1:total) {
    ponto_test <- list_test[i,]  # Pega a linha i de list_test
    result <- define_especie(k, c(ponto_test$x_1, ponto_test$x_2), treino)
    #print(paste("Resultado esperado: ", ponto_test[[3]]))
    #print(paste("Resultado do metodo: ", result))
    if (ponto_test$y == result){
      acertos <- acertos + 1
    }
  }
  print(paste("Percentual de acertos com k =", k, ": ", ... = (acertos/total)*100, "%"))
  }
```

```
## [1] "Percentual de acertos com k = 1 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 2 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 3 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 4 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 5 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 6 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 7 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 8 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 9 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 10 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 11 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 12 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 13 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 14 :  93.3333333333333 %"
## [1] "Percentual de acertos com k = 15 :  93.3333333333333 %"
```