

Solução Lista 01

Nome: Henrique Queiroz Reuter
E-mail: reuter.q@aluno.ufabc.edu.br
Nome: Rodrigo Martins de Souza
E-mail: souza.r@aluno.ufabc.edu.br

24 February, 2025

Exercício 01

- a) Uma aplicação prática de um problema de classificação é fazendo o uso do algoritmo kNN para a classificação de imagens. O algoritmo pega o vetor de características, que é os valores de pixel da imagem e então achar os seus k vizinhos mais próximos e então classifica aquele dado baseado em qual classe é a mais comum dentre esse vizinhos.
- b) Podemos citar como exemplo prático de problema de regressão a resolução mostrada por Saleh, H.(2022) onde foi usando um algoritmo de regressão linear múltipla para treinar um modelo a prever o preço de um imóvel. As características usadas pelo o modelo são: quantidade de quartos, quantidade de banheiros, a área total da casa, a quantidade de andares, área do porão, ano de construção do imóvel, ano de renovação do imóvel e zipcode. Saleh, Haidara, and J. A. Layous. “Machine Learning–Regression.” (2022).
- c) Uma aplicação prática de um problema de agrupamento é fazendo o clustering para a segmentação de clientes. Nesse modelo o histórico de compras, idade, escolaridade, modo mais comum de pagamento de um cliente vem a ser parte do vetor de características, isso com o objetivo de poder agrupar os clientes em grupos semelhantes e com isso facilitando as diferentes campanhas de marketing que uma empresa deve fazer para cada nicho.

Exercício 02

A maldição da dimensionalidade se refere ao aumento das distâncias dos pontos conforme se aumenta a quantidade de dimensões. Ou seja para algoritmos de aprendizado de máquina que dependem desse tipo de cálculo, conforme a complexidade do problema aumenta, levando em consideração mais fatores, fica cada vez mais computacionalmente custoso.

Exercício 03

```
library(tidyverse)

kNN <- function(k,x,D){
  D2 <- D %>% mutate( dist = (x[1] - x1)**2 + (x[2] - x2)**2 ) %>%
  arrange( dist ) %>% head(k) %>% count(y) %>% arrange(n) %>% tail(1)
  r <- D2[1]
  return (toString(pull(r)))
}

D <- tibble( x1 = rnorm(100,1,1),
             x2 = rnorm(100,-1,2),
```

```

y = factor(sample(c("toranja","laranja"),100,replace = T))
x <- c(4,1)
k<-7
print (kNN(k,x,D))

```

```
## [1] "toranja"
```

Exercício 04

```

library(tidyverse)
data("iris")
iris <- as_tibble(iris) %>% # Converte para a dataframe tibble
select(Petal.Length,Sepal.Length,Species) %>% # Seleciona colunas da dataframe
rename( x1 = Petal.Length, x2 = Sepal.Length, y = Species) # Renomeia as colunas

```

```

library(purrr)
vizinhos <- 10

l_iris <- as.list(iris)

compar <- pmap_lgl(l_iris, function(x1,x2,y){
  t <- c(x1, x2)
  rod2 <- toString(t)
  rod<-toString(y)
  #print(str_c(kNN(vizinhos, t, iris),",", " ", rod))
  return((kNN(vizinhos, t, iris))==toString(y))
})
#compar
#sum(c(compar))
vizStr <- toString(vizinhos)
respStr <- toString((sum(c(compar))/150)*100)
print(str_c("a porcentagem certa para vizinhos = ",vizStr, " é ",respStr,"%"))

```

```
## [1] "a porcentagem certa para vizinhos = 10 é 96%"
```

```

vizinhos <- 1

compar2 <- pmap_lgl(l_iris, function(x1,x2,y){
  t <- c(x1, x2)
  rod2 <- toString(t)
  rod<-toString(y)
  #print(str_c(kNN(vizinhos, t, iris),",", " ", rod))
  return((kNN(vizinhos, t, iris))==toString(y))
})

vizStr <- toString(vizinhos)
respStr <- toString((sum(c(compar2))/150)*100)
print(str_c("a porcentagem certa para vizinhos = ",vizStr, " é ",respStr,"%"))

```

```
## [1] "a porcentagem certa para vizinhos = 1 é 99.3333333333333%"
```