

# Solução Lista 01

Nome: Raquel Ferreira Campos  
E-mail: raquel.campos@aluno.ufabc.edu.br  
Nome: Rodrigo Kaito Okuyama  
E-mail: kaito.o@aluno.ufabc.edu.br  
(Não é preciso informar os RAs)

24 fevereiro, 2025

## Exercício 01

- a) Problema de classificação: Pode ser usado para detecção de spam no email. Os vetores podem incluir a presença de links, número de caracteres especiais, frequência de palavras ligadas ao spam e outros. Logo, o rótulo vai ser: spam ou não spam.
- b) Problema de regressão: Pode ser usado para a previsão do preço de imóveis. Os vetores podem incluir o tamanho do imóvel, localização, números de quartos, etc. A resposta é o preço estimado do imóvel.
- c) Agrupamento: Pode ser usado para um sistema de organização por fotos, para agrupar imagens parecidas automaticamente, como paisagens ou pessoas. Os vetores podem incluir cores dominantes, formas que estão na imagem, padrões de textura. Assim, vai ter uma saída com grupos como fotos de praia ou fotos de pessoas.

## Exercício 02

A maldição da dimensionalidade ocorre quando trabalhamos com dados em espaços de alta dimensão, e a partir disso pode gerar alguns problemas como o aumento da complexidade computacional e a dificuldade de medir distâncias com precisão, porque os pontos geralmente ficam mais afastados.

## Exercício 03

```
knn <- function(k, x, D) {  
  D$dist <- (x[1] - D$x1)^2 + (x[2] - D$x2)^2  
  
  D_sorted <- D[order(D$dist), ]  
  nearest_neighbors <- head(D_sorted, k)  
  
  most_common <- names(which.max(table(nearest_neighbors$y)))  
  
  return(most_common)  
}  
  
# teste  
set.seed(123)
```

```

D <- data.frame(
  x1 = rnorm(100, 1, 1),
  x2 = rnorm(100, -1, 2),
  y = factor(sample(c("one", "two", "three"), 100, replace = TRUE))
)

x_teste <- c(1, 2)
k_valor <- 10

resultado <- knn(k_valor, x_teste, D)
print(resultado)

```

```
## [1] "one"
```

## Exercício 04

```

library(tidyverse)
library(class)

data("iris")

set.seed(123)
dados <- iris %>%
  select(Petal.Length, Sepal.Length, Species) %>%
  rename(x_1 = Petal.Length, x_2 = Sepal.Length, y = Species)

n <- nrow(dados)
indices <- sample(seq_len(n), size = round(0.8 * n))
dados_treino <- dados[indices, ]
dados_teste <- dados[-indices, ]

treino_matrix <- as.matrix(dados_treino[, c("x_1", "x_2")])
teste_matrix <- as.matrix(dados_teste[, c("x_1", "x_2")])
treino_classes <- as.factor(dados_treino$y)

knn_k1 <- class::knn(train = treino_matrix, test = teste_matrix, cl = treino_classes, k = 1)
knn_k10 <- class::knn(train = treino_matrix, test = teste_matrix, cl = treino_classes, k = 10)

valor_k1 <- mean(knn_k1 == dados_teste$y)
valor_k10 <- mean(knn_k10 == dados_teste$y)

print(paste("k = 1:", round(valor_k1 * 100, 2), "%"))

```

```
## [1] "k = 1: 93.33 %"
```

```
print(paste("k = 10:", round(valor_k10 * 100, 2), "%"))
```

```
## [1] "k = 10: 96.67 %"
```

## Exercício 06

```
calcula_mediana = function(m, d) {  
  return((1 - 0.5^(1/m))^(1/d))  
}
```

```
# Exemplo para m=100 e d=3
```

```
m = 100
```

```
d = 3
```

```
mediana_distancia <- calcula_mediana(m, d)
```

```
print(paste("A mediana da distância do ponto mais próximo à origem :", round(mediana_distancia, 4)))
```

```
## [1] "A mediana da distância do ponto mais próximo à origem : 0.1904"
```