

# Solução Lista 01

Nome: Bruna Alves Maziero  
E-mail: bruna.maziero@aluno.ufabc.edu.br  
Nome: Pedro Cardoso Alves Barbuti  
E-mail: pedro.barbuti@aluno.ufabc.edu.br  
(Não é preciso informar os RAs)

25 February, 2025

## Exercício 01

### a) Problema de Classificação

Aplicação: Prever se uma biópsia de tecido mamário é benigna ou maligna.

Vetor de características: Espessura do grupo, Uniformidade do tamanho da célula, Uniformidade da forma da célula, Adesão marginal, Tamanho da célula epitelial única, Núcleos nus, Cromatina suave, Nucléolos normais, Mitoses.

Rótulo: Benigno (0) ou Maligno (1).

### b) Problema de Regressão

Aplicação: Prever o preço do vinho Bordeaux com base nas condições climáticas. Vetor de características: Temperatura média do verão, precipitação da colheita, precipitação do inverno, idade do vinho. Resposta: Preço do vinho.

### c) Problema de Agrupamento

Aplicação: Agrupar legisladores da Califórnia com base em seu alinhamento com diferentes organizações.

Vetor de características: Pontuações de concordância com várias organizações (por exemplo, California Medical Association, Sierra Club California).

Resposta: Grupos de legisladores com padrões de alinhamento semelhantes (por exemplo, “conservadores convictos”, “conservadores fiscais e ambientalmente conscientes”, “moderados fiscais e ambientalmente conscientes”, “liberais ambientalistas”).

## Exercício 02

A maldição da dimensionalidade descreve o fenômeno em que o volume do espaço de dados aumenta tão rapidamente com o aumento do número de dimensões que os dados disponíveis se tornam esparsos, dificultando a análise e modelagem. Isso pode levar a problemas como aumento da complexidade computacional, overfitting e perda de precisão em algoritmos de aprendizado de máquina.

## Exercício 03

```

library(dplyr)
library(tibble)

knn_function <- function(k, x, D) {
  D <- D %>% mutate(dist = (x[1] - x_1)^2 + (x[2] - x_2)^2)
  D_sorted <- D %>% arrange(dist) %>% head(k)
  class_counts <- D_sorted %>% count(y)
  return(class_counts$y[which.max(class_counts$n)])
}

D <- tibble(
  x_1 = rnorm(100, mean = 1, sd = 1),
  x_2 = rnorm(100, mean = -1, sd = 2),
  y = factor(sample(c("one", "two", "three"), 100, replace = TRUE))
)

```

## Exercício 04

```

library(dplyr)
library(tibble)
library(datasets)
library(purrr)

knn_function <- function(k, x, D) {
  D <- D %>% mutate(dist = (x[1] - x_1)^2 + (x[2] - x_2)^2)
  D_sorted <- D %>% arrange(dist) %>% head(k)
  class_counts <- table(D_sorted$y)
  return(names(which.max(class_counts)))
}

iris_df <- as_tibble(iris) %>%
  select(Petal.Length, Sepal.Length, Species) %>%
  rename(x_1 = Petal.Length, x_2 = Sepal.Length, y = Species)

evaluate_function <- function(k, D) {
  correct <- sum(pmap_lgl(as.list(D), function(x_1, x_2, y) {
    x_test <- c(x_1, x_2)
    D_train <- D %>% filter(!(x_1 == x_test[1] & x_2 == x_test[2]))
    predicted_label <- knn_function(k, x_test, D_train)
    return(predicted_label == y)
  })))
  return(correct)
}

correct_k10 <- evaluate_function(10, iris_df)
correct_k1 <- evaluate_function(1, iris_df)

print(correct_k10)

```

```
## [1] 141
```

```
print(correct_k1)
```

```
## [1] 137
```