

Solução Lista 01

Nome: Pedro Henrique Machado Silva Campos

E-mail: campos.pedro@aluno.ufabc.edu.br

Nome: Yasmin Nicole Rezende

E-mail: yasmin.rezende@aluno.ufabc.edu.br

(Não é preciso informar os RAs)

25 fevereiro, 2025

Exercício 01

- a) Classificação - Detecção de Fraude em Transações Financeiras Descrição: Identificar se uma transação financeira é legítima ou se foi golpe. Vetores de Características: Valor da transação, localização geográfica, horário da transação e histórico de transações do usuário. Rótulos: “Legítima” ou “Golpe”.
- b) Regressão - Previsão de Preços Imobiliários Descrição: Estimar o preço de venda de imóveis com base em suas características. Vetores de Características: Tamanho do imóvel, número de quartos, localização, idade do imóvel e proximidade de serviços públicos. Rótulos: O preço de venda do imóvel.
- c) Agrupamento - Segmentação de Clientes em Marketing Descrição: Agrupar clientes com características semelhantes para direcionar campanhas de marketing específicas no setor de compras. Vetores de Características: Idade, gênero, histórico de compras, frequência de compras, valor gasto e preferências de produtos. Rótulos: Não há rótulos pré-definidos. Os dados são ajustados conforme a semelhança de características.

Exercício 02

Ao aumentar muito a dimensão, esses dados se tornam muito esparsos e medir as distâncias entre eles se torna difícil. Há uma noção de proximidade entre os pontos quando a dimensão é baixa, ou seja, reduzir faz com que ganhe um desempenho.

Exercício 03

```
library(dplyr)
library(tibble)
library(ggplot2)

#função que retorna a classe mais comum do ponto x
knn <- function(k, x, D) {
  #copia a tabela D para fazer alterações que ficarão salvas na D2
  D2 <- D %>%

  #calcula a distância de cada ponto da tabela D para o ponto x
  mutate( dist = (x[1] - x_1)^2 + (x[2] - x_2)^2 ) %>%
```

```

#ordena essa distância por ordem crescente
arrange( dist ) %>%

#seleciona os k (10) primeiros da tabela
head(k) %>%

#agrupa (conta) as classes y e ordena do maior para o menor
count(y, sort = TRUE)

#retorna o primeiro valor de y da tabela D2, contendo a classe mais comum
return(D2$y[1])
}

#criação da tabela com 100 dados aleatórios
set.seed(42)
D <- tibble(
  x_1 = rnorm(100, 1, 1),
  x_2 = rnorm(100, -1, 2),
  y = factor(sample(c("one", "two", "three"), 100, replace = TRUE))
)

#define a observação nova a ser rotulada (com k vizinhos)
x = c(1, 2)
k = 10

#chamada da função que retorna e imprime a classe rotulada
classe = knn(k, x, D)
classe

```

```

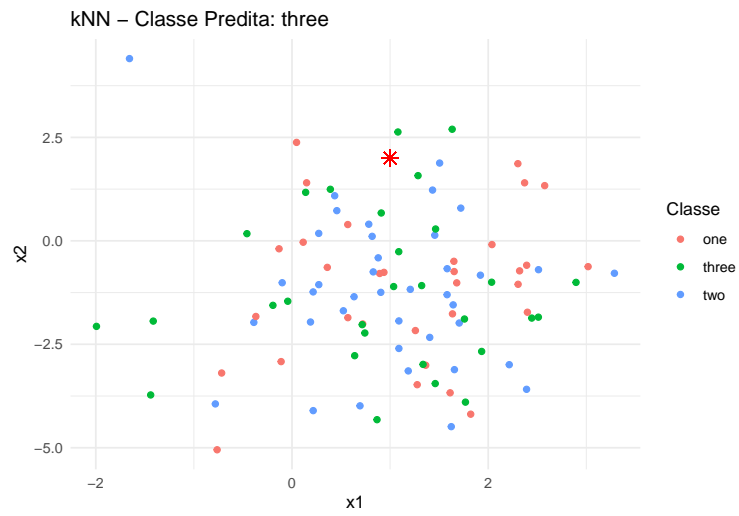
## [1] three
## Levels: one three two

```

```

#grafico para verificação
ggplot(D, aes(x = x_1, y = x_2, color = y)) +
  geom_point() +
  geom_point(aes(x = x[1], y = x[2]), color = "red", size = 3, shape = 8) +
  labs(title = paste("kNN - Classe Predita:", classe),
       x = "x1", y = "x2", color = "Classe") +
  theme_minimal()

```



Exercício 04