

Solução Lista 01

Nome: Gabriel Jorge Menezes
E-mail: g.jorge@aluno.ufabc.edu.br

25 fevereiro, 2025

Exercício 01

1. Problemas de classificação são aqueles em que estamos interessados em classificar e designar categorias pre definidas aos nossos dados. Dentre suas aplicações temos classificação de imagens, vídeos, filtragem de spam, e até mesmo classificação de texto escrito a mão
2. Problemas de regressão são aqueles nos quais tenta prever um valor contínuo, por exemplo previsão do tempo, ou preço de algum produto dependendo de condições do mercado e etc. . .
3. Problemas de agrupamento é similar a problemas de classificação, mas nestes não temos categorias bem definidas, mas sim estamos mais interessados em achar padrões em comuns que nos permitem agrupar os dados. Como por exemplo imagine o cenário hipotético onde temos dados de pacientes e queremos achar quais responderam bem a um certo tratamento e quais não.

Exercício 02

Maldição da dimensionalidade se refere ao problema enfrentado ao lidar com dados de alta dimensões, isto se torna um problema pois encontrar padrões pode se tornar difícil, aumento do custo computacional resultado em um pior modelo. Além disso treinar em altas dimensões pode gerar o overfitting dos dados, sendo assim não generalizando para novos dados.

Exercício 03

```
knn_classifier <- function(k, x, D) {  
  # Calcula a distância euclidiana entre x e todos os pontos em D  
  distancias <- sqrt((D$x_1 - x[1])^2 + (D$x_2 - x[2])^2)  
  
  # Obtém os índices dos k vizinhos mais próximos  
  indices <- order(distancias)[1:k]  
  
  # Obtém os rótulos dos k vizinhos mais próximos  
  rotulos <- D$y[indices]  
  
  # Retorna a classe mais frequente entre os vizinhos  
  return(names(sort(table(rotulos), decreasing = TRUE))[1])  
}
```

Exercício 04

```
data("iris")  
  
iris <- as_tibble(iris) %>%
```

```

select(Petal.Length, Sepal.Length, Species) %>%
  rename( x_1 = Petal.Length, x_2 = Sepal.Length, y = Species)

l_iris <- as.list(iris)

k1 <- pmap_lgl(l_iris, function(x_1, x_2, y) {
  return( knn_classifier(1, c(x_1, x_2), iris) == y)
})

k100 <- pmap_lgl(l_iris, function(x_1, x_2, y) {
  return( knn_classifier(100, c(x_1, x_2), iris) == y)
})

print(str_c("k = 1 | esperado: ", length(k1), " | obtido: ", sum(k1)))

## [1] "k = 1 | esperado: 150 | obtido: 149"

print(str_c("k = 100 | esperado: ", length(k100), " | obtido: ", sum(k100)))

## [1] "k = 100 | esperado: 150 | obtido: 98"

```