

Lista 01 - Aprendizado de Máquinas

25 fevereiro, 2025

Exercício 01

Faça uma pesquisa sobre aplicações práticas de aprendizado de máquinas e descreva aplicações de cada um dos problemas abaixo, incluindo a descrição dos vetores de características, rótulos ou respostas quando apropriado para cada um dos problemas:

- Problema de classificação. Um exemplo real poderia ser o filtro de spam dos emails. No sentido que ele recebe como entrada o conteúdo do email, palavras suspeitas, arquivos anexados, etc. E o modelo aprende a prever se o email é suspeito de spam ou não.
- Problema de regressão. Um exemplo real poderia ser a previsão de um preço de imóvel, com entrada sendo a área da casa, localização, quantos quartos, suítes, área construída, etc. E a saída sendo o preço estimado do imóvel
- Problema de Agrupamento. Um exemplo real poderia ser um banco identificando perfis de clientes em certos tipos de produtos do banco. Usando como entrada: frequência e tipo de compras, idade, histórico de transação, salário; e saída como: Perfis de cliente como “poupadores”, “gastadores”, “confiáveis”, “não confiáveis”, etc ## Exercício 02

Descreva com suas próprias palavras o que é a “maldição da dimensionalidade”.

Em casos onde há muitas variáveis começam a aparecer problemas como aumento de custo computacional com processamentos e memórias muito requisitadas e dependendo das dimensões a complexidade aumenta exponencialmente. Outro problema comum é o dados estarem muito espalhados, a variar das dimensões também, fazendo com que atrapalhe a busca de padrões.

Exercício 03

Implemente o método dos vizinhos mais próximos para um problema de classificação. Sua função deve receber como argumento:

- Um número k representando o número de vizinhos para usar no kNN;
- Um vetor $x = (x_1, x_2)$ com duas componentes;
- Uma *dataframe* D com colunas x_1 , x_2 e y , em que x_1 e x_2 são valores numéricos representando duas componentes do vetor de características x (i.e., $x = (x_1, x_2)$) e y é um fator representando os rótulos dos pontos. Abaixo tem-se um exemplo dessa *dataframe* com duas classes:

```
D <- tibble( x_1 = rnorm(100,1,1),
             x_2 = rnorm(100,-1,2),
             y   = factor(sample(c("one", "two", "three"), 100, replace = T)))
head(D)
```

x_1	x_2	y
-0.1825480	-2.2289259	two
2.4495117	-2.7319493	three
0.4729542	-4.1085046	two
0.9793536	-1.9734251	two
0.3723913	0.9209118	two
1.9859052	-3.5777720	two

A função deve ter a assinatura `function(k,x,D)` e deve retornar a classe mais provável associada ao ponto x .

dica: Você pode fazer o kNN usando uma sequencia de comandos encadeados pelo operador pipe `%>%`. Por exemplo, teste a seguinte sequencia de comandos com a *dataframe* D anterior:

```
x = c(1,2)
k = 10
D2 <- D %>%
  mutate(dist = (x[1] - x_1)^2 + (x[2] - x_2)^2) %>%
  arrange(dist) %>% head(k) %>% count(y)

library(dplyr)

knn_classifier <- function(k, x, D) {

  D <- D %>%
    mutate(dist = (x[1] - x_1)^2 + (x[2] - x_2)^2) %>%
    arrange(dist) %>%
    head(k) %>%
    count(y)

  return(D$y[which.max(D$n)])
}

D <- tibble(
  x_1 = rnorm(100, 1, 1),
  x_2 = rnorm(100, -1, 2),
  y = factor(sample(c("A", "B", "C"), 100, replace = TRUE))
)

x <- c(1, 2)
k <- 10
resultado <- knn_classifier(k, x, D)

print(resultado)
```

```
## [1] A
## Levels: A B C
```

Exercício 04

Usando o banco de dados *iris* e sua implementação do kNN do exercício anterior, calcule quantos pontos são classificados corretamente de acordo com o rótulo *Species* usando as colunas *Petal.length* e *Sepal.length* com $k = 10$ e com $k = 1$.

dica 1: Você pode carregar o banco Iris no R da seguinte forma:

```
library(tidyverse)
data("iris") # Carrega o banco no ambiente global
iris <- as_tibble(iris) %>% # Converte para a dataframe tibble
  select(Petal.Length, Sepal.Length, Species) %>% # Seleciona colunas da dataframe
  rename( x_1 = Petal.Length, x_2 = Sepal.Length, y = Species) # Renomeia as colunas

head(iris)
```

x_1	x_2	y
1.4	5.1	setosa
1.4	4.9	setosa
1.3	4.7	setosa
1.5	4.6	setosa
1.4	5.0	setosa
1.7	5.4	setosa

dica 2: As funções map da biblioteca purrr do pacote tidyverse são muito úteis! Por exemplo, a função pmap_lgl aplica uma função à argumentos fornecidos por uma lista e retorna os resultados da função como um vetor de booleanos (neste caso, a função deve retornar valores booleanos). Rode o exemplo abaixo:

```
l_iris <- as.list(iris) # converte a dataframe em uma lista
                        # que possui elementos com nomes x_1, x_2 e y

# Aplica a função usando os valores dados pelos valores na lista,
# concatena os resultados da função em um vetor booleano.
v_bool <- pmap_lgl(l_iris, function(x_1,x_2,y){
  print(str_c(x_1," ", "x_2"," ", "y"))
  return( y == "setosa" )
})
```

A função sum pode também ser útil. Lembre-se de usar o comando ? para ajuda.

```
library(tidyverse)
library(purrr)

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

test_knn <- function(k) {
  acertos <- pmap_lgl(as.list(iris), function(x_1, x_2, y) {
    predicted_class <- knn_classifier(k, c(x_1, x_2), iris)
    return(predicted_class == y)
  })

  return(mean(acertos))
}
```

```

accuracy_k10 <- test_knn(10)
accuracy_k1 <- test_knn(1)

print(paste("Acurácia com k = 10:", round(accuracy_k10 * 100, 2), "%"))

## [1] "Acurácia com k = 10: 95.33 %"

print(paste("Acurácia com k = 1:", round(accuracy_k1 * 100, 2), "%"))

## [1] "Acurácia com k = 1: 99.33 %"

```

Exercício 5 (opcional)

Em aula vimos como calcular a função de regressão $f : \mathcal{X} \rightarrow \mathcal{Y}$ ótima que minimiza o risco esperado:

$$\mathcal{R}(f) = \mathbb{E}_{XY}[\ell(Y, f(X))]$$

quando a função de perda é dada por $\ell_2(y, y') := (y - y')^2$. Essa função de perda é geralmente usada pela simplicidade de soluções. Mas existem outras funções de perda, como a função de perda do erro absoluto, que é dada por: $\ell_1(y, y') := |y - y'|$. Mostre que a função f ótima, que minimiza o risco esperado com essa função de perda, é dada por $f(x) := \text{Mediana}(Y|X = x)$. Suponha que a distribuição de Y é contínua.

dica 1: A mediana de uma variável aleatória contínua tomando valor em \mathbb{R} é definida como sendo o valor real m tal que $P(Y > m) = P(Y < m) = 1/2$.

dica 2: A derivada de $\ell_1(y, y')$ em relação a y' quando $y' \neq y$ existe e é limitada. Nestes casos, sabe-se que

$$\frac{\partial}{\partial z} \mathbb{E}[\ell_1(Y, z)|X = x] = \mathbb{E}\left[\frac{\partial}{\partial z} \ell_1(Y, z) \Big| X = x\right].$$

Exercício 6 (opcional)

Considere que m pontos são espalhados uniformemente em uma hipersfera de raio unitário e dimensão d . Mostre que a mediana da distância do ponto mais próximo à origem é dada por: $(1 - 0.5^{1/m})^{1/d}$.