Solução Lista 01

Nome: Caio Stoduto Ervilha
E-mail: caio.stoduto@aluno.ufabc.edu.br
Nome: Nicolas Gomes Greco
E-mail: nicolas.greco@aluno.ufabc.edu.br

25 February, 2025

Exercício 01

a) Uma aplicação clássica na área de classificação é a de 'análise de crédito', onde o agente do empréstimo utiliza os dados do cliente para assim calcular a viabilidade de concessão de crédito. Um exemplo prático seria um dataset que possui inúmeros vetores, que representam os clientes, e cada componente desse vetor seria uma característica que ajude o algoritmo a tomar a decisão entre conceder ou não o crédito àquele usuário, como: histórico de pagamento, renda e outras informações relevantes de sua vida financeira e cada um desses vetores é rotulado como sendo de clientes inadimplentes ou não. Finalmente o algoritmo é devidamente treinado e dado novos vetores, pode-se estimar se o novo cliente pagará suas dívidas devidamente ou não.

b) Um problema de regressão é o cálculo de preços de imóveis numa determinada região, usa-se essa estimativa para dar uma previsão do preço de um imóvel com base nos imóveis de característica mais parecida. O método começa com vetores que são os imóveis do dataset de treinamento, cada vetor possui componentes que são as caractéristicas de cada imóvel, por exemplo, quantidade de quartos, metros quadrados, quantidade de banheiros entre outros. Cada vetor desse também possui seu valor funcional (análogo ao rótulo para problemas de classificação), nesse contexto, é o preço do imóvel. Finalmente, o modelo treinado consegue calcular o valor funcional de um novo vetor, dado os vetores presentes em seu dataset, o que retorna finalmente seu preço estimado.

c) As técnicas de agrupamento são normalmente utilizadas para resolver problemas em que há a necessidade de associações entre dados que não são facilmente vistas sem uma análise estatística completa, dessa forma a máquina fornece insights valiosos a respeito de um conjunto de dados não rotulado. Como por exemplo, no uso de técnicas de agrupamento por exmpresas de vendas online, que usam suas grandes quantidades de dados para melhorar seu marketing, descobrindo associações interessantes, como a relação de um certo perfil de consumidor com um produto específico.

Exercício 02

A Maldição da Dimensionalidade é um fenômeno observado na área de Aprendizado de Máquina, no qual o aumento do número de dimensões (ou variáveis de entrada) em um conjunto de dados ocasiona o aumento exponencial da distância entre os pontos do espaço, causando sua dispersão. Esse fenômeno é prejudicial ao aprendizado de máquina, pois pode diminuir sua performance, aumentar o tempo de treinamento, dificultar a identificação de padrões e até causar o *overfitting* — quando um modelo se ajusta excessivamente aos dados de treinamento e não generaliza bem para novos dados.

Uma abordagem comum para mitigar esse problema é a aplicação de métodos de redução de dimensionalidade, que utilizam diferentes técnicas para diminuir o número de variáveis mantendo a maior quantidade possível de informação relevante.

Exercício 03

```
library(tibble)

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)
```

```
## # A tibble: 6 x 3
##      x_1    x_2 y
##    <dbl>  <dbl> <fct>
## 1 -0.887 -6.12  two
## 2  0.135 -2.45  two
## 3  1.40   0.856 three
## 4  1.57   3.91  two
## 5  1.46  -1.67  one
## 6  1.15  -1.54  two
```

```
import pandas as pd
import math

df = pd.DataFrame(r.D)

def knn(k: int, x: tuple, D: pd.DataFrame) -> str:
  filteredDf = D.drop(['y'], axis=1, inplace=False)
  distancias_rank = []

  for i in range(len(D)):
    distancia = 0
    for index, j in enumerate(filteredDf.iloc[i]):
      distancia+=(x[index]-j)**2
    distanciaFinal = math.sqrt(distancia)
    distancias_rank.append((i, distanciaFinal))

  distancias_rank.sort(key=lambda e: e[1])

  vizinhos = []

  for i in range(k):
    vizinhos.append(distancias_rank[i][0])

  contagem = {classe: 0 for classe in D['y'].unique()}

  for index in vizinhos:
    rotulo = D.iloc[index]['y']

    contagem[rotulo]+=1

  def rotulo_resultante(contagem):
    return max(contagem, key=contagem.get)
```

```
  return rotulo_resultante(contagem)

knn(10, (2,0), df)
```

```
## 'one'
```

Exercício 04

```
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)
```

```
## # A tibble: 6 x 3
##     x_1   x_2 y
##   <dbl> <dbl> <fct>
## 1   1.4   5.1 setosa
## 2   1.4   4.9 setosa
## 3   1.3   4.7 setosa
## 4   1.5   4.6 setosa
## 5   1.4   5   setosa
## 6   1.7   5.4 setosa
```

```
import pandas as pd

dfIris = pd.DataFrame(r.iris)

def validador(k: int, df: pd.DataFrame, knn: callable) -> None:
  pontos = 0
  linhas = len(df)

  for i in range(linhas):
    vector = (df.iloc[i,0],df.iloc[i,1])
    filteredDf = df.drop(i, inplace=False)
    rotulo = knn(k, vector, filteredDf)

    if rotulo == df.iloc[i, 2]:
      pontos+=1

  print(f'A funcao acertou {pontos} do total de {linhas} rótulos, com k = {k}')


validador(10, dfIris, knn)
```

```
## A funcao acertou 141 do total de 150 rótulos, com k = 10
```

```
validador(1, dfIris, knn)
```

```
## A funcao acertou 136 do total de 150 rótulos, com k = 1
```

Exercício 05 (Opcional)

Inicialmente, é necessário definir a função de risco esperado a ser considerda e sua função de perda.

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

Em seguida, fixa-se a função de risco esperado em um valor arbitrário de x.

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

Com isso, é possível derivar a função de risco esperado em função de x.

$$\frac{\partial}{\partial x}\mathbb{E}[\ell(Y, f(X))|X = x]$$

Substitui-se função de perda $\ell(Y, f(X))$ por $|y - y'|$, sabendo que a derivada de $\ell(y, y') := |y - y'|$ existe e é limitada quando $y \neq y'$.

$$\mathbb{E}[\frac{\partial}{\partial x}|y - f(x)||X = x]$$

Analisando a derivada, é possível dividir em dois casos:

$$\begin{cases} y < f(x) : \frac{\partial}{\partial x}|y - f(x)| = +1 \\ y > f(x) : \frac{\partial}{\partial x}|y - f(x)| = -1 \end{cases}$$

Substituindo os casos com suas respectivas probabilidades, pode-se descobrir o valor esperado.

$$\mathbb{E}[\frac{\partial}{\partial x}|y - f(x)||X = x] = (+1) \cdot P(Y < f(x)|X = x) + (-1) \cdot P(Y > f(x)|X = x)$$
$$\mathbb{E}[\frac{\partial}{\partial x}|y - f(x)||X = x] = P(Y < f(x)|X = x) - P(Y > f(x)|X = x)$$

Com isso, encontra-se o ponto crítico igualando a derivada a 0 para descobrir o ponto mínimo.

$$P(Y < f(x)|X = x) - P(Y > f(x)|X = x) = 0$$
$$P(Y < f(x)|X = x) = P(Y > f(x)|X = x)$$

Finalmente, como a probabilidade de Y ser maior e menor que $f(x)$ são iguais, e como isso cobre todo o universo de casos por ser uma variável aleatória contínua, é possível deduzir que essas probabilidades são iguais a $\frac{1}{2}$.

$$P(Y < f(x)|X = x) = P(Y > f(x)|X = x) = \frac{1}{2}$$

Finalmente, pela definição de mediana de uma variável aleatória contínua tomando valor em $\mathbb{R}$ como o valor real m tal que $P(Y > m) = P(Y < m) = \frac{1}{2}$, conclui-se que $f(x)$ é a mediana, já que metade da probabilidade de Y se encontra acima de $f(x)$ e a outra metade abaixo de $f(x)$.

Exercício 06 (Opcional)

Sabe-se que o volume de uma hiperesfera é proporcional ao raio elevado a dimensão do espaço.

$$V_R \propto R^d$$

Considerando a distribuição uniforme em uma hiperesfera, é possível escrever a probabilidade de um ponto estar a uma distância $\leq$ r proporcionalmnete ao volume da hiperesfera.

$$P(distância \leq r) = \frac{V_r}{V_1} = \frac{r^d}{1^d} = r^d$$

Logo, define a probabilidade de que todos os m pontos estejam à uma distância maior que r da origem.

$$P(distância > r) = (1 - r^d)$$
$$P(distância > r)^m = (1 - r^d)^m$$

Enfim, define a probabilidade de ao menos um ponto estar a uma distância $\leq$ r.

$$P(\exists\ p\ \epsilon\ V_R) = 1 - (1 - r^d)^m$$

Como já discutido no exercício anterior, no âmbito da distância absoluta podemos considerar que a distribuição probabilistica partindo da mediana é dada por $\frac{1}{2}$ para a distribuição de pontos com distância menor e maior que a distância de r. Com isso, podemos igualar $P(\exists\ p\ \epsilon\ V_R)$ com $\frac{1}{2}$.

$$1 - (1 - r^d)^m = 0.5$$
$$(1 - r^d)^m = 0.5$$
$$1 - r^d = 0.5^{\frac{1}{m}}$$
$$r^d = 1 - 0.5^{\frac{1}{m}}$$
$$r = (1 - 0.5^{\frac{1}{m}})^{\frac{1}{d}}$$

Notas adicionais

- Todos os códigos desta lista foram realizados em Python 3.12.2;