Entrega 1 | Álgebra Linear, Vetores e Geometria Analítica

Tema: Representação de Dados de Pedidos em Matrizes e Operações Lineares

Clique aqui para baixar o arquivo detalhado

Introdução

Neste relatório utilizamos dados reais de pedidos (arquivo `Order_semicolon.csv`) provenientes do kit de dados fornecido pela Cannoli. Utilizaremos apenas o arquivo de pedidos pois ele já nos traz colunas numéricas contínuas, ideais para representar diretamente como vetores e matrizes, aplicando operações de álgebra linear.

Cada pedido possui atributos como:

- totalAmount: valor total do pedido (R\$)
- **preparationTime**: tempo de preparo (minutos)
- takeOutTimeInSeconds: tempo de retirada (segundos)

Cada linha será representada como um **vetor**, e o conjunto dos pedidos formará uma **matriz**. Com isso, aplicamos operações de Álgebra Linear (transposição, soma de vetores e multiplicação de matrizes).

1. Importação e visualização dos dados

```
import pandas as pd
import numpy as np

# Ler os dados de pedidos

df_orders = pd.read_csv("Order_semicolon.csv", sep=";")

print("Primeiras linhas do arquivo de pedidos:")

display(df_orders.head())
```

Output:

Primeiras linhas do arquivo de pedidos:											
	id	companyld	containerId	createdAt	customer	displayld	engineld	engineName	engineType	extrainfo	
		I0VP1DZXGU	AKVXT2FH	08/01/2025 15:47	525	O8CC98	E3R037	DirectOrder	POS	Adipisci maiores nam eius vero nesciunt sed.	
	2	04MEUULZ6W	KKMY5TG5	03/11/2024 22:30	694	HJ99VA	EJVOBV	DirectOrder	APP	NaN	
2		w2ZZS7CCZW	187JLGTV	09/09/2024 05:49	491	XJJDH2	UDX1ZF	KDSPro	POS	NaN	
	4	SQ5MBL8L24	W3ZXOQHO	22/05/2025 06:22	846	J8QDM3	30FLOC	KDSPro	WEB	Autem mollitia voluptas quasi.	
4		0RCNJ6BRJD	NA0C0FOS	22/02/2025 01:23	77	H4LV73	QI3URN	KDSPro	POS	NaN	
5 rows × 23 columns											

2. Seleção de colunas para formar a matriz

```
# Selecionar colunas de interesse
colunas_interesse = ["totalAmount", "preparationTime",
"takeOutTimeInSeconds"]
colunas_existentes = [col for col in colunas_interesse if col in
df_orders.columns]
print("\nColunas utilizadas para análise vetorial:", colunas_existentes)
# Criar matriz de dados com numpy
matriz_pedidos = df_orders[colunas_existentes].dropna().to_numpy()
print("\nMatriz de pedidos (NumPy array):")
print(matriz_pedidos)
```

Output:

3. Representação em Vetores e Matrizes + Transposição

Cada linha representa um **pedido** (vetor) e cada coluna representa uma **característica** (atributo):

```
# Transposição da matriz
matriz_transposta = matriz_pedidos.T
print("\nMatriz-Transposta:")
print(matriz_transposta)

Python

Matriz Transposta:
[[ 90.91 99.69 45.97 ... 66.56 23.44 30.13]
[ 45. 33. 22. ... 23. 19. 78. ]
[2131. 374. 247. ... 3162. 2230. 1861. ]]
```

4. Soma de Matrizes e Vetores

```
# Soma de vetores (Pedido 1 + Pedido 2)
if matriz_pedidos.shape[0] >= 2:
    soma_vetores = matriz_pedidos[0] + matriz_pedidos[1]
    print("\nSoma do Pedido 1 e Pedido 2:")
    print(soma_vetores)
```

Output:

```
Soma do Pedido 1 e Pedido 2:
[ 190.6 78. 2505. ]
```

5. Multiplicação de matrizes

Definimos uma matriz de pesos e o resultado representa dois "modelos" diferentes avaliando os pedidos:

```
# Multiplicação de matrizes (pesos x dados)
pesos = np.array([
      [0.5, 0.3, 0.2],
      [0.2, 0.5, 0.3]
])

resultado = np.dot(pesos, matriz_pedidos.T)
print("\nMultiplicação de matrizes (pesos x pedidos.T):")
print(resultado)
```

Output:

```
Multiplicação de matrizes (pesos x pedidos.T):
[[485.155 134.545 78.985 ... 672.58 463.42 410.665]
[679.982 148.638 94.294 ... 973.412 683.188 603.326]]
```

Conclusão

Neste relatório, representamos dados de pedidos em forma de **matriz** e aplicamos operações fundamentais da Álgebra Linear:

- **Transposição** → troca linhas por colunas.
- **Soma de vetores** → combina características de diferentes pedidos.
- Multiplicação de matrizes→ aplica pesos diferentes às variáveis.

Essas operações são a base de técnicas em **Inteligência Artificial** e **Aprendizado de Máquina**, mostrando como dados do mundo real podem ser manipulados matematicamente para gerar conhecimento.