

5 rows x 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:

```
Colunas utilizadas para análise vetorial: ['totalAmount', 'preparationTime', 'takeOutTimeInSeconds']

Matriz de pedidos (NumPy array):
[[ 90.91  45.  2131. ]
 [ 99.69  33.  374. ]
 [ 45.97  22.  247. ]
 ...
 [ 66.56  23.  3162. ]
 [ 23.44  19.  2230. ]
 [ 30.13  78.  1861. ]]
```

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)
```

[3] 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.