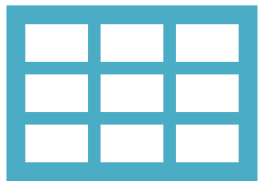


Aula 3 - NumPy Avançado

Indexação e álgebra linear

Objetivos



Manipular arrays



Usar slicing



Operações matriciais

Introdução à Indexação



Indexação básica permite acessar elementos individuais ou fatias de arrays.



Indexação avançada possibilita seleção por arrays de índices, máscaras booleanas ou combinações complexas.



Facilita manipulação de subconjuntos de dados sem laços explícitos.



Permite operações mais eficientes e código mais legível para análise de dados.

Exemplos de Indexação Avançada

Indexação por lista ou array de índices: `arr[[0, 2, 4]]`



Máscaras booleanas para filtrar dados: `arr[arr > 10]`



Indexação combinada e multidimensional para acesso seletivo.



Útil para manipular dados específicos sem criar cópias desnecessárias.



Slicing em NumPy



Definição: Técnica para extrair sub-arrays de arrays maiores utilizando índices de início, fim e passo.



Sintaxe básica: **arr[início:fim:passo]**



Permite selecionar fatias contíguas ou não contíguas de dados de forma simples e eficiente.



Suporta slicing multidimensional:
arr[início1:fim1, início2:fim2]



Muito usado para manipulação e análise de subconjuntos de dados em ciência de dados.

Exemplos de Indexação Avançada

Indexação por lista ou array de índices:
arr[[0, 2, 4]]

Máscaras booleanas para filtrar dados:
arr[arr > 10]

Indexação combinada e
multidimensional para acesso seletivo.

Útil para manipular dados específicos
sem criar cópias desnecessárias.

Álgebra Linear com NumPy - Conceitos Básicos



NumPy suporta operações essenciais de álgebra linear para análise científica.



Principais operações: multiplicação matricial, transposição, inversão de matrizes, cálculo de determinantes e autovalores.



Essas operações são fundamentais para modelagem, simulações e análise de dados em engenharia agrícola.

Operações de Álgebra Linear com Exemplos



Multiplicação matricial: **`np.dot(A, B)`** ou **`A @ B`**



Transposição: **`A.T`**



Inversão: **`np.linalg.inv(A)`**

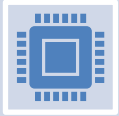


Cálculo de determinante: **`np.linalg.det(A)`**

Aplicação



Distribuição de calor em uma câmara de secagem



Análise e manipulação de grandes conjuntos de dados de sensores e ensaios.



Modelagem de sistemas agrícolas, simulações de comportamento do solo e plantas.



Suporte para modelagem estatística, otimização e aprendizado de máquina.

Exercício

Modele uma matriz de temperaturas e calcule média.

Solução

--- Análise da Matriz de Temperaturas (com NumPy) ---

Matriz de Temperaturas Coletadas (°C):

[25.5, 26.1, 25.8, 26.0]

[27.3, 27.5, 28.0, 27.8]

[24.9, 25.0, 25.3, 25.1]

Temperatura Média Geral: 26.19°C

```
1 import numpy as np
2
3 # 1. Modela a matriz de temperaturas
4 matriz_de_temperaturas = [
5     [25.5, 26.1, 25.8, 26.0],
6     [27.3, 27.5, 28.0, 27.8],
7     [24.9, 25.0, 25.3, 25.1],
8 ]
9 # 2. Converte a lista Python para um array NumPy
10 array_de_temperaturas = np.array(matriz_de_temperaturas)
11
12 # 3. Calcula a média usando a função np.mean()
13 media_geral = np.mean(array_de_temperaturas)
14
15 # 4. Imprime os resultados
16 print("--- Análise da Matriz de Temperaturas (com NumPy) ---")
17 print("\nMatriz de Temperaturas Coletadas (°C):")
18 # Imprime a matriz de forma mais legível
19 for linha in matriz_de_temperaturas:
20     print(linha)
21
22 print(f"\nTemperatura Média Geral: {media_geral:.2f}°C")
```