Explorando o Poder do NumPy na Ciência de Dados

O pacote Numpy é um dos pilares da ciência de dados e da análise de dados em Python. Ele fornece suporte para arrays multidimensionais e uma ampla coleção de funções matemáticas e estatísticas, tornando as operações de manipulação de dados mais eficientes e menos propensas a erros. NumPy é a base sobre a qual muitos outros pacotes de ciência de dados, como Pandas, SciPy e Scikit-learn, são construídos. Sua capacidade de realizar operações em grandes volumes de dados com rapidez e precisão faz dele uma ferramenta indispensável para cientistas de dados e analistas.

Abaixo, apresento uma tabela com 15 das principais funções do pacote numpy utilizadas em ciência de dados e análise de dados. Estas funções foram selecionadas por sua utilidade e frequência de uso em tarefas comuns, como manipulação de arrays, cálculos estatísticos e operações matemáticas.

Os 15 principais métodos/funções Numpy utilizados em Análise e Ciência de Dados

Comando	Descrição	Exemplo de Utilização	Operação em Python
`np.add`	Soma elemento a elemento de dois arrays.	`np.add([1, 2], [3, 4])`	`[a + b for a, b in zip([1, 2], [3, 4])]`
`np.argmax`	Retorna o índice do maior valor ao longo de um eixo.	`np.argmax([1, 3, 2])`	`max(range(len([1, 3, 2])), key=[1, 3, 2].__getitem__)`
`np.array`	Cria um array Numpy a partir de uma lista.	`np.array([1, 2, 3])`	`list([1, 2, 3])`
`np.concatenate`	Junta uma sequência de arrays ao longo de um eixo existente.	`np.concatenate([[1, 2], [3, 4]])`	`[1, 2] + [3, 4]`
`np.dot`	Produto escalar de dois arrays.	`np.dot([1, 2], [3, 4])`	`sum(a * b for a, b in zip([1, 2], [3, 4]))`
`np.mean`	Calcula a média aritmética ao longo de um eixo.	`np.mean([1, 2, 3])`	`sum([1, 2, 3]) / len([1, 2, 3])`
`np.median`	Calcula a mediana ao longo de um eixo.	`np.median([1, 2, 3])`	`sorted([1, 2, 3])[len([1, 2, 3]) // 2]`
`np.reshape`	Redimensiona um array sem alterar os dados.	`np.reshape([1, 2, 3, 4], (2, 2))`	`[[1, 2], [3, 4]]`
`np.std`	Calcula o desvio padrão ao longo de um eixo.	`np.std([1, 2, 3])`	`(sum((x - sum([1, 2, 3])/len([1, 2, 3]))**2 for x in [1, 2, 3])/len([1, 2, 3]))**0.5`
`np.sum`	Soma dos elementos ao longo de um eixo.	`np.sum([1, 2, 3])`	`sum([1, 2, 3])`
`np.transpose`	Transpõe as dimensões de um array.	`np.transpose([[1, 2], [3, 4]])`	`list(zip(*[[1, 2], [3, 4]]))`
`np.var`	Calcula a variância ao longo de um eixo.	`np.var([1, 2, 3])`	`sum((x - sum([1, 2, 3])/len([1, 2, 3]))**2 for x in [1, 2, 3])/len([1, 2, 3])`
`np.vstack`	Junta arrays verticalmente (em filas).	`np.vstack(([1, 2], [3, 4]))`	`[([1, 2]), ([3, 4])]`
`np.where`	Retorna os índices que satisfazem uma condição.	`np.where([1, 2, 3] > 2)`	`[i for i, x in enumerate([1, 2, 3]) if x > 2]`
`np.zeros`	Cria um array preenchido com zeros.	`np.zeros((2, 2))`	`Manualmente --> [[0, 0], [0, 0]]`

Espero que você tenha achado esta seleção de comandos e comparação com a linguagem Python útil. Seja você um iniciante no mundo da Ciência de Dados ou um profissional experiente, entender como utilizar essas funções do NumPy pode melhorar significativamente a eficiência e a clareza do seu código. Se você tiver alguma dúvida ou sugestão, não hesite em entrar em contato. Quero garantir que meus conteúdos sejam sempre relevantes e úteis para você.

Fique atento à próxima edição, onde continuaremos a explorar ferramentas, técnicas e dicas valiosas para aprimorar suas habilidades em Ciência de Dados. Até lá, continue explorando e aprendendo!

Saudações,

Prof. Dr. Dilermando Piva Jr
Coordenador de Ciência de Dados para Negócios / Fatec Votorantim
E-mail: f301.cdn@fatec.sp.gov.br