

Numpy Library Common Functions

1. Introduction

NumPy (commonly imported as `np`) is a powerful Python library used for numerical and scientific computing. It provides support for arrays, matrices, and many mathematical functions that make working with large datasets or performing complex mathematical operations easier and faster.

2. Array Creation and Initialization

- `np.array()`: Create an n-dimensional array from a list or existing data
- `np.zeros()`: Create an array filled with zeros
- `np.ones()`: Create an array filled with ones
- `np.full()`: Create an array filled with a specific value
- `np.arange()`: Create an array with a range of values
- `np.linspace()`: Create an array with evenly spaced values between two numbers
- `np.eye()` / `np.identity()`: Create an identity matrix
- `np.random.rand()`: Generate an array with random values between 0 and 1

3. Reshaping and Manipulating Arrays

- `np.reshape()`: Reshape the array without changing its data.
- `np.transpose()`: Transpose the array (swap rows and columns).
- `np.flatten()`: Convert a multi-dimensional array into a 1D array.
- `np.concatenate()`: Join two or more arrays along an axis.
- `np.hstack()` / `np.vstack()`: Stack arrays horizontally or vertically.
- `np.split()`: Split an array into multiple sub-arrays.

4. Mathematical Operations

- `np.add()`, `np.subtract()`, `np.multiply()`, `np.divide()`: Basic element-wise arithmetic operations.
- `np.dot()`: Perform a dot product of two arrays (important for linear algebra).
- `np.sum()`: Sum of array elements along a given axis.
- `np.mean()`: Calculate the mean of the array.
- `np.median()`: Compute the median of the array.
- `np.std()`: Calculate the standard deviation.
- `np.var()`: Compute the variance.
- `np.cumsum()`: Cumulative sum of array elements.
- `np.prod()`: Product of array elements.

5. Examples

<https://github.com/Bido5/Data-Science/blob/main/Numpy%20Arrays.ipynb>