NumPy Cheat Sheet for Data Science in Python

Datalators Datalators

NumPy is a library for the Python programming language which is used for working with multi-dimensional arrays and matrices. This is very useful in large scientific computing. Because NumPy ndarrays is way faster compared to a regular python list. Arrays are very frequently used in data science too, where speed and resources are very important. That's why NumPy is a very handy tool in data-science.

But remembering all the NumPy commands might be overwhelming for both beginners and professionals. So Datalators makes the complex simple.

It's also a good idea to check the official NumPy documentation from time to time. Even if you can find what you need in the cheat sheet. Reading documentation is a skill every data professional needs. Also, the documentation goes into a lot more detail than we can fit in a single sheet anyway!

Creating Arrays:

np.array([1.3, 2, 3], dtype = float)	Creating a 1d array
np.array([1, 2, 3], [2, 3, 4], [3, 4, 5])	Creating a 3d array
np.arange(2, 8, 2)	Array of evenly spaced values
np.zeros((2, 3))	2×3 array consists of zeros
np.ones((3, 4))	3×4 array consists of ones
np.random.random((3, 3))	Array consists of random values
np.empty((2, 3))	Empty array
np.full((3,2), 5)	Constant array

Import Export:

np.save('saved_array', a)	Save 'a' array as on disk
np.savez('array.npz', a, b)	Save 2 arrays
np.savetxt('array.txt', a, delimiter= " ")	Saving as text file
np.genformatxt('array.csv', a, delimiter= ",")	Saving as CSV file
np.load('array.npz')	Load from disk
np.loadtxt('array.txt')	Load from text file

Inspecting Array:

a.shape	Dimensions of 'a' array
len(a)	Length of array
a.ndim	Array dimensions
a.size	Number of elements
a.dtype	Data type of elements
a == b	Element-wise comparison
a > 1	Element-wise comparison
np.array_equal(a, b)	Array-wise comparison

Data Types:

np.int64	Signed integer types
np.float32	Floating point
np.complex	Complex number
np.bool	Boolean type
np.object	Python object type
np.string_	String type
np.unicode	Unicode type
a.astype(int)	Convert to int type

Array Mathematics:

np.subtract(a, b)	Subtraction
np.add(a, b)	Addition
np.devide(a, b)	Division
np.multiply(a, b)	Multiplication
a+b, a-b, a*b, a/b	Operation – arithmetic sign
np.sin(a), np.cos(a), np.log(a)	Mathematical operation
a.dot(b)	Dot product
np.exp(a), np.sqrt(a)	Exponentiation and Square root

Statistics on NumPy:

a.sum()	Array-wise sum
a.min(), a.max(axis = 0)	Minimum and Maximum value
a.mean(), a.median()	Mean and Median
a.corrcoef()	Correlation coefficient
np.std(a)	Standard deviation
b.cumsum(axis = 1)	Cumulative sum of elements
a.sort(), a.sort(axis = 0)	Sort an array

Indexing and Slicing

b = a.view() / a.copy()	Create a copy of array
a[1, 2]	Subsetting
a[0:2]	Slicing
a[0:2, :-1]	Slicing
a[a<2]	Boolean Indexing

Array Manipulation:

b = np.transpose(a)	Permute array dimensions
a.reshape(3,-2)	Reshape but don’t change data
h.resize((2,4))	Return a new array with shape (2,4)
np.append(a,b)	Append items to an array
np.insert(a, 1, 5)	Insert items in an array
np.delete(a,[1])	Delete items from an array
np.hsplit(a,3)	Split the array horizontally at the 3rd index

I hope this cheat sheet will be useful to you. No matter you are new to python who is learning python for data science or a data professional. Happy Programming. For more visit **blog.datalators.com**