Jason Schlottman

HWRS 401

10/04/2021

**Cheat Sheet 3\_Numpy**

Numpy Array-

A numpy array is a sequence of values which are all of the same type and are composed of indexed by a tuple of non-negative integers. These arrays are useful for performing complex operations on large sets of data by calling on the defined parameters in a efficient small amount of code.

Numpy Array Format:

The standard numpy array is composed as shown in the following random example:

Numpy.array(object, dtype="None", boolean=True/(False), order= "n", sub-bool="False/(True), min =(0))

Where we can break down the example components:

Object: returns any object which represents array interface/nested sequence

Dtype: a Boolean expression which confirms (or denies) the true data type of every object within the array which are all the same.

Boolean: a

Order: return memory type of the array

Sub-bool: Another Boolean expression which determines whether the return is a base-class array or if the sub-classes pass through.

Min: returns integer minimum value of dimensions of the array based on the data.

Numpy Array vs. Lists

These are both methods of performing various operations on multiple data objects in a simpler, reduced amount of code which still outlines the returns given from multiple lines of code with separate operations. Both list and arrays are very similar but may prove more beneficial in certain circumstances such as if the series needs to be sizeable or set in stone and whether the objects may be the same or can be composed of different types.

Lists are very similar in written format when designing the initial code , however they are built-in python functions which can contain elements of several different types unlike the numpy array which is composed of values that are all of the same type. Lists are sizable while numpy arrays aren't. \

Both rely on using contiguous blocks of data which results in very quick indexing. Both are highly efficient and rely on data called in from another file source.

Numpy key methods & Attributes:

Operations can be used for many purposes related to arrays such as returning the dimensions of the array in a tuple or resizing or reshaping the array

An example of a numpy attribute used to analyze the elements within the array:

Ex) numpy.itemsize = which returns the length of each element of the array in bytes.

An example of a key numpy method may be:

Ex) np.dtype() where the parameters are included, which returns the data type of each object called in the method definition in parentheses.

\*\*There are thousands more useful examples of returns for key numpy array attributes and methods which can be observed with great examples here on numpy.org

<https://numpy.org/doc/stable/user/basics.rec.html>