Numpy Tutorials:

1. Why do we use numpy:
2. It is more efficient, takes relatively lesser space to store data, fast, easy to learn.

Import numpy using : import numpy as np

NUmpy Array:

* myArray=np.array([list],datatype)
* for example: myArray=np.array([1,2,3,4],int8)   
  It stores 8 bit integer

we can use Numpy array by indexing as well, and as we create numpy array we can also create a 2d array.

myArray.shape – gives the shape (dimension of the array)

myArray.dtype-gives the datatype of elements stored in the array

np.zeros(x,y) – creates a float type array of dimenstion x \* y and fill all the values by 0.

np.arange(x) – creates an array from 0 to x-1 and returns it

example zero=np.zeros(2,5) gives zero=[[0.,0.,0.,0.,0.]

[0.,0.,0.,0.,0.]]

np.arange(15) – gives array([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14])

np.linspace(x,y,sep): gives sep numbers from x to y equally spaced:

example: lnspc=np.linspace(1,5,4)

np.empty((x,y)) creates an empty array with random values

np.empty\_like(arr) gives an empty array of size of arr

np.identity(x) it gives x\*x identity matrix

arr.reshape(x,y) it can reshape a 1d matrix into 2d matrix and vice versa only when number of elements are equal in both of the arrays

arr.ravel() it undo the reshaped

arr.shape gives the shape (dimension) of the array

numpy axis

axis 0 :- axis 0 represents the rows: axis 0 means elements of rows

axis 1:- axis 1 represents the columns: axis 1 means elements of columns

arr.sum(axis = 0) sum of all rows individually

arr.sum(axis=1) sum of all columns individually

arr.T transposes the matrix

arr.flat gives the pointer of the array

arr.ndim gives the number of dimensions of the array

arr.size gives the number of elements in the array

ar.nbytes gives the storage how much it is occupying

ar.argmax() returns the maximum element index

ar.argmin() returns the minimum element index

ar.argsort() returs the indices in which elements need to arrage to be sorted

Numpy array also supports slicing of the arrays.

ar.astype(float) changes the datatype of ar array from int to float

mathematical operation in numpy arrays:

concatenate function: to concatenate two arrays.

For horizontal and vertical concatenation: we use axis values 1 or 0 but also we can use:

np.hstack([arr1,arr2])

np.vstack([arr1,arr2]) for horizontal and vertical stacking respectively

np.array\_split(arr,n) it splits array into n partss