When you use np. array(existing data) then you will remove all commas

Exit1= np.array(existing data) ->

[[ 1 2 3 4 5]

[ 6 7 8 9 10]

[11 12 13 14 15]]

NumPy is the fundamental package for scientific computing with Python. It's used for working with arrays and contains functions for working in the domain of linear algebra, Fourier transform, and matrices.

***arange(start, stop, step)*** will return evenly spaced values within a given interval. The default step size is 1 inch

**np.arange(0, 15).reshape(3, 5) ->** it will start from 0 to 14 (15 not including) and it has 3 rows and 5 columns

np**.**zeros[**ones](**15) -> all zeros 15 zeros or 15 ones, so it will give us **FLOAT NUMBERS**

**np.ones(15, dtype=int).reshape(3, 5)** The main difference is we use dtypy=int

np.eye(number) -> gives us matrix with **number \*number** and diagonal is 1’s

np**.**random**.**randint(1,100) -> you can now use it without importing the **“Random”**

**np.random.rand(5) -**(random number between 0 and 1 all number are positive 5 numbers)

Data.max() and Data.min we use it in np.arange(0,5).reshape(1, 5)

0/data -> 0 and data/0 -> Error

If you want to multiple 2 datas you have to use then **np.matmul(Data1,Data2)**

**np.sqrt(NUMBER) -> You can use square root without Importing the “IMPORT MATH”**

**np.log(NUMBER) -LOG**

To modify the index of the data we can just replace it with

A[0:10] **=** 500 -> now al numbers between 0 to 10(not including the 10 because it always start with 0) are 500

Print (Data[0, 2])

Print ([3, 3])

Print ([4, 2]) it will give us from 0 to 4(columns) and from 0 to 2 columns. Basically to say it will give us the **5th column and 3rd row**

**Data [Data % 2 == 0 ] ->** it gives all numbers which divisible to 2