Day 05 - Python Numpy Arrays

This includes:

- Numpy
- Array
- shape
- Single and Multidimensional Arrays
- EDA over arrays
- reshape()
- arange()
- ones()
- zeros()
- zero_likes()
- random methods and submethods

Numpy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python

What is an array

An array is a data structure that stores values of same data type. In Python, this is the main difference between arrays and lists. While python lists can contain values corresponding to different data types, arrays in python can only contain values corresponding to same data type.

```
# import libraries
import numpy as np

lst=[1,2,3,4]
arr=np.array(lst) # it creates a numpy array from a list, 1D array.

type(arr)
```

```
numpy.ndarray
```

```
arr.shape # it returns the shape of the array
```

```
(4,)
```

```
# 3D array
lst1=[1,2,3,4]
lst2=[4,5,6,3]
lst3=[7,8,9,2]
```

```
arr1=np.array([lst1,lst2,lst3])

arr1.shape

(3, 4)

arr1

array([[1, 2, 3, 4],
       [4, 5, 6, 3],
       [7, 8, 9, 2]])

# Indexing in single dimension array
arr[2]

3

arr

array([1, 2, 3, 4])

arr[2]=5

arr

array([1, 2, 5, 4])

arr[-1]

4

# it excludes the last element which's index in written
arr[:-1]

array([1, 2, 5])

arr[1:3] # there is 3 means (3-1)

array([2, 5])

arr[::-1] # it reverses the array

array([4, 5, 2, 1])

arr[::-2] # it reverses the array with jump of 2 elements

array([4, 2])

arr1

array([[1, 2, 3, 4],
       [4, 5, 6, 3],
       [7, 8, 9, 2]])
```

```
# indexing of multi dimensional array
arr1[:,1] # it returns the 1st column

array([2, 5, 8])

arr1[:,2]  # it returns the 2nd column

array([3, 6, 9])

arr1[1,:] # it returns the 1st row

array([4, 5, 6, 3])

arr1[1:,1:3] # it returns the 1st row and 2nd column

array([[5, 6],
       [8, 9]])

arr1

array([[1, 2, 3, 4],
       [4, 5, 6, 3],
       [7, 8, 9, 2]])

arr1[1:,2:]

array([[6, 3],
       [9, 2]])
```

EDA

```
arr[3]=-5

arr

array([ 1,  2,  5, -5])

arr<2 # it returns the boolean array

array([ True, False, False,  True])

arr1

array([[1, 2, 3, 4],
       [4, 5, 6, 3],
       [7, 8, 9, 2]])

arr1.reshape(4,3) # it reshapes the array

array([[1, 2, 3],
       [4, 4, 5],
       [6, 3, 7],
       [8, 9, 2]])
```

```
# mechanism to create an array
np.arange(1,10,1) # it creates an array from 1 to 10 with jump of 1
```

```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
np.arange(1,10,2) # it creates an array from 0 to 10 with jump of 2
```

```
array([1, 3, 5, 7, 9])
```

```
np.arange(1,20,2).reshape(5,2) # it creates an array from 1 to 20 with
jump of 2 and reshapes it to 5x2
```

```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11],
       [13, 15],
       [17, 19]])
```

```
np.arange(1,20,2).reshape(1,10) # it creates an array from 1 to 20
with jump of 2 and reshapes it to 5x2
```

```
array([[ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19]])
```

```
arr
```

```
array([ 1,  2,  5, -5])
```

```
arr*arr
```

```
array([ 1,  4, 25, 25])
```

```
arr[3]=7
```

```
arr
```

```
array([1, 2, 5, 7])
```

```
arr*arr
```

```
array([ 1,  4, 25, 49])
```

```
arr1 * arr1
```

```
array([[ 1,  4,  9, 16],
       [16, 25, 36,  9],
       [49, 64, 81,  4]])
```

```
np.ones((3,3))
# means it creates an array of 3x3 with all elements 1
```

```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

```
np.zeros((3,3)) # it creates an array of 3x3 with all elements 0
```

```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

```
np.zeros_like(arr1) # it creates an array of same shape as arr1 with
all elements 0
```

```
array([[0, 0, 0, 0],
       [0, 0, 0, 0],
       [0, 0, 0, 0]])
```

```
np.random.randint(10,50)
# it returns a random number between 10 and 50
# it takes input as a range
```

```
16
```

```
np.random.randint(10,50,2)
# it returns 2 random numbers between 10 and 50
```

```
array([23, 24])
```

```
np.random.randint(10,50,4).reshape(2,2)
# it returns 4 random numbers between 10 and 50 and reshapes it to 2x2
```

```
array([[31, 31],
       [27, 15]])
```

```
np.random.randn(3,3)
# it returns a 3x3 array with random numbers
```

```
array([[-1.82128206,  0.62891643, -0.74837251],
       [ 0.50806661, -0.17429957,  0.29242122],
       [-1.12601566,  0.67112277,  1.09086865]])
```

```
np.random.random_sample((3,3))
# it returns a 3x3 array with random numbers
# it only initiliazes numbers between 0 and 1
```

```
array([[0.29028366, 0.92508677, 0.27347589],
       [0.85258057, 0.11138806, 0.0268907 ],
       [0.20355696, 0.7489771 , 0.89897142]])
```