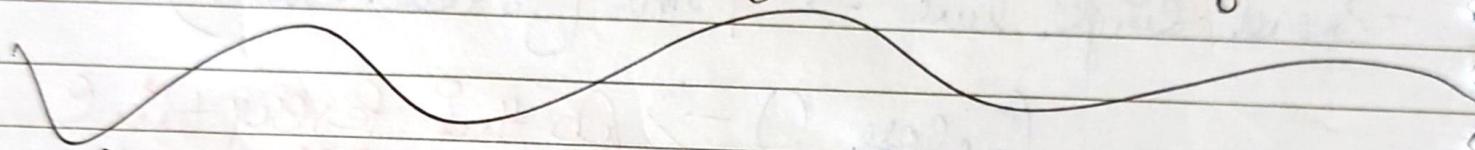


25/01/23 Day -> 1 Numpy -> 1

Date 25/01/23

Data Analysis & Visualization (DAN)

- ↳ DAY 1 → Python Libraries ≈ 1 month
- DAY 2 → Probability & stats
- DAY 3 → Hypothesis Testing



Python List → heterogeneous

$a = [1, 'Hello']$

Problems :- i) Take Diff amount memory

ii) So, they cannot store together

Numpy → $\begin{bmatrix} 4 & 1 & 4 & 1 & 4 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ d_1 & d_2 & d_3 & d_4 & d_5 \end{bmatrix}$ List → $\boxed{1, 2}$

↳ Refining

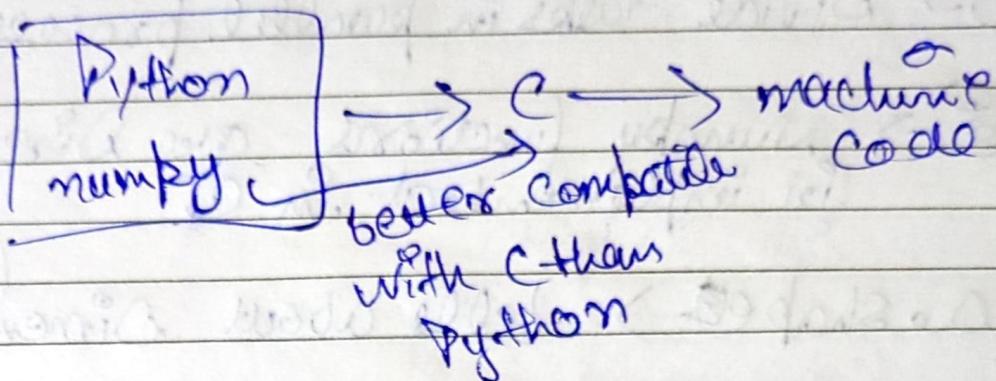
Numpy → homogeneous

↳ Sequence of Same Data Type

25/9/23

Date: 1M1

numpy ← Directly connected to C++



Importing Numpy →

`import numpy as np`

`b = np.array(a)`

n-dimension

`type(b) → numpy.ndarray`

Numpy support element wise operations

Broadcasting → $b^{* * 2}$

`b = a**2` (1, 2, 3) ↓

[1, 4, 9]

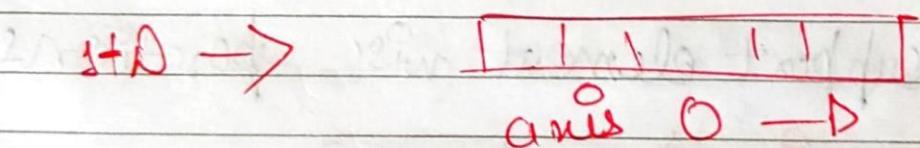
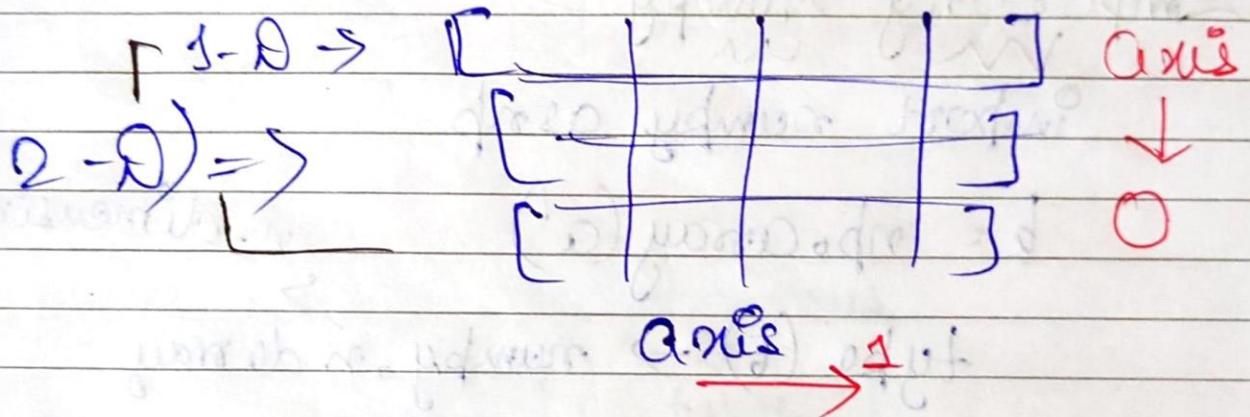
Numpy ← faster list

`l = np.array(range(10000))`, % time it 2**3
↓
Microseconds

23/9/23

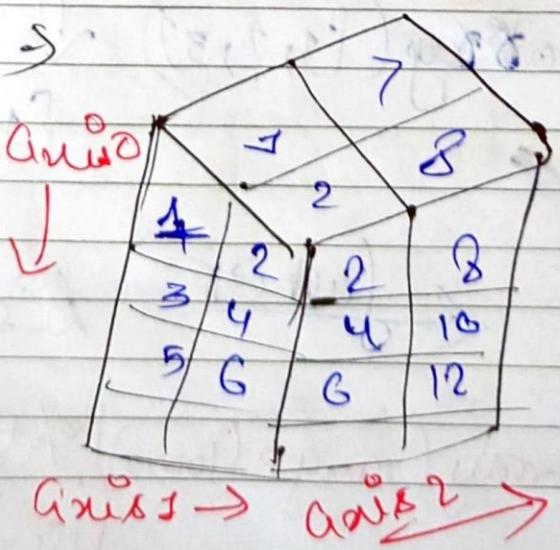
Date 1/12

- (1) \rightarrow It is Homogeneous
- (2) \rightarrow Divide tasks in parallel processes
- (3) \rightarrow NumPy functions are directly implemented in C
- a. shape \rightarrow tells about Dimension $\begin{pmatrix} \text{row} \\ \text{col} \end{pmatrix}$
- b. dimensions \rightarrow tells about Dimension $\begin{pmatrix} 1/2/3 \end{pmatrix}$



3-D arrays

shape $(3, 2, 2)$
axis 0 \rightarrow
axis 1 \rightarrow
axis 2 \rightarrow



26/9/23

Date 17/3

Similar to range

$\hookrightarrow \text{orange} \rightarrow \text{np.array}(3, 2, 5)$

$\hookrightarrow \text{start}$

$\hookrightarrow \text{end}$

$\text{np.array}(1, 5, 2)$

$\hookrightarrow \text{Exclusive}$

\downarrow Inclusive

* Orange works with float, range don't

$\Rightarrow b = \text{np.array}(1, 5, 0.5)$

$\Rightarrow [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5]$

All Integer automatically converted to float

* All slicing & index works just like list.

Auto Typecasting / Intrinsic / Implicit

$b = \text{np.array}([6, 11, 12, 13])$

String Data Type

$b \in \{11, 12, 13\}$

This will get convert to string because numbers can be converted into string.

Explicit Typecasting

$| d = \text{np.array}([3, 2], \text{dtype} = \text{"float"})$

26/9/23

Date 174
String $\xrightarrow{\text{To}}$ float $\rightarrow X$
float $\xrightarrow{\text{To}}$ string $\rightarrow \checkmark$

Ranking Indexing ↗

$m_1 = np.array([range(5)])$

$m_1 \rightarrow \text{array}([1, 2, 3, 4])$

$m_1 > 6 \rightarrow \text{array}([False, False, False, False])$

$m_1 < 6 \rightarrow \text{array}([True, True, True, True])$

$a = [0, 1, 2, 3, 4, 5]$

$a[0, 2, 4] \rightarrow$ I want these index elements

→ Cannot do that in Python

But in NumPy

$a[a \% 2 \neq 0] \rightarrow [1, 3, 5]$

$a[a \% 2 \neq 0] \rightarrow [1, 3, 5]$

Remove false items

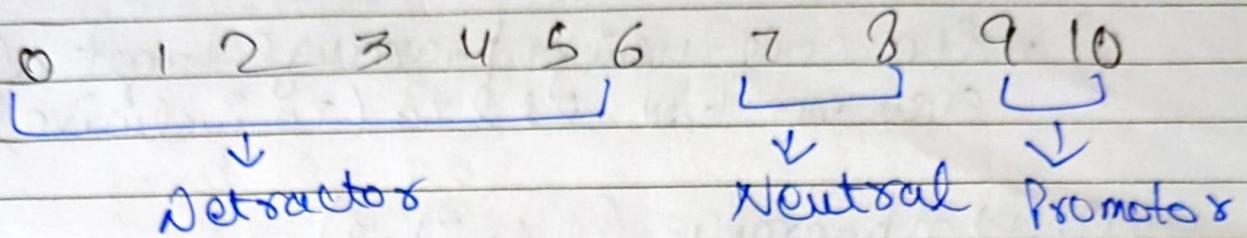
False \rightarrow Remove it

True \rightarrow Don't Remove it

Date 14/5

26/1/23

NPS → Net Promoter Score



$$\boxed{NPS = \% \text{ promoters} - \% \text{ detractors}}$$

H/W → Calculate NPS for this given dataset

Score
→ `df = pd.read_csv('survey.csv', dtype = int)`

Promoter → score [score > 9]
num of promoter = len(score[score > 9])

Also no of detractor = len(score[score ≤ 6])

% of Promoter = $\frac{\text{num_promoter}}{\text{len(score)}} \times 100$
Total
↳ Multiply by 100

% Detractor = $\frac{\text{num_detractor}}{\text{len(score)}} \times 100$

NPS = (Promoter% - Detractor%)

27/9/23 Assignment

Date: 176

Q) Even & odd.

a) Write code using np.arange to get all even no b/w 23 270 (20 inclusive)

option 1) `np.arange(21, 71, 2)`

b) odd no b/w 20 271 (-1 inclusive)

option 2) `np.arange(21, 72, 2)`

Q) Cast a type

`arr = np.array([10, 20, 30, 40, 50])`
↳ change to float

option 3) `arr.astype('float64')`

Q) Slice array

`a = np.arange[10]`

option 4)

`print(a[2:5])` → 2, 3, 4

Q) What about output

$x = np.array([-5, 9, 20, 25, -3, 5, 16, 10, 8])$
 $x[(x > -5) \& (x \leq 15)] = -1$

$\rightarrow [-5, 9, 20, 25, 31 - 5, 16, -10, 8]$

option 2) Given code will change the sign of all
the numbers in range [-5, 15] in

Date 4/17

27/02/23

⑤ → Comparison Returns!

$a = \text{np.array}([100, 200, 300, 400])$
 $b = \text{np.array}([300, 200, 100, 400])$

print($a == b$)

option 0 → [False, True, False, True]

2 Additional Problems

① → The dawn of the planet of numpy

$\text{np.round}(\text{np.arange}(\text{start}, \text{end}, \text{step}), 2)$

② → Index of Pokédex

first Element → arr[0]
last Element → arr[-1]

③ → np.array([1, 2, 3, 4])

print(arr[2] + arr[-2]) → 3 + 3 = 6 [option 0]

④ → Create the sequence

using arange ⇒ end = start + (step * length)
seq = np.arange(start, end, step)

→ Can also use linspace → But Not taught yet

⑤ → Shape of you

shape = arr.shape

dim = arr.ndim