Lecture 3 - Numpy + Case Study

Monday, 27 October 2025 9:01 PM

Numpy

Broadcasting

Vectorization

np v split & concat.

Matrix Multiplication

A

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

2×3

B

$\begin{bmatrix} 4 & 5 \\ 7 & 8 \\ 9 & 10 \end{bmatrix}$

3×2

np.matmul
np.dot
a @ b

$2 \times 3 \quad \cancel{2 \times 3}$

Broadcasting

One message share to all

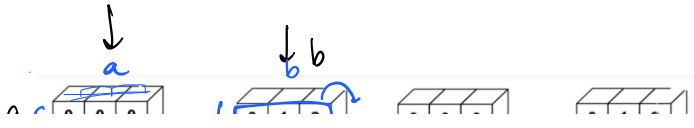
$[1, 2, 3] + 2$

$[1, 2, 3] + [4, 5]$ ✗

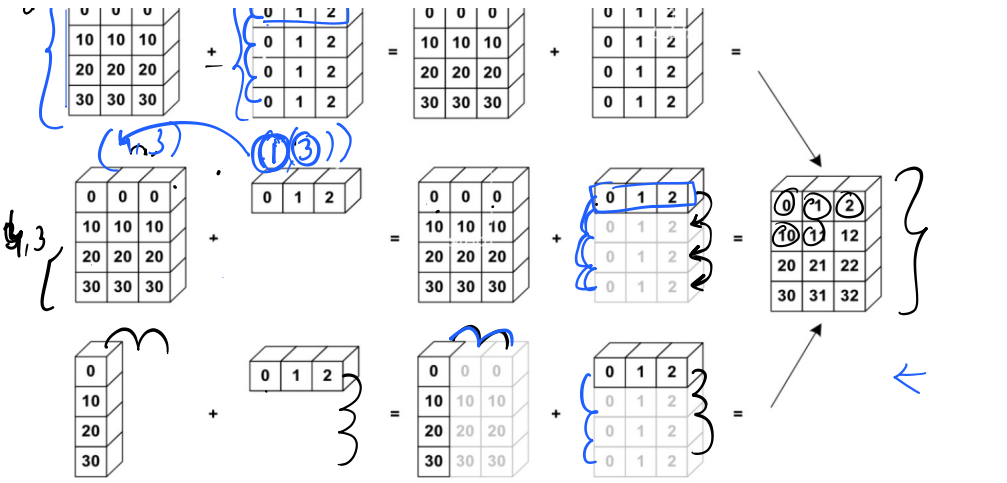
$1+4 \quad 1+5$

② Rules

① Dimensions of array needs to be same



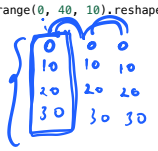
element wise add



Broadcasting — element wise Mapping

① → Dimension should exactly match, add 1 in the Starting

```
a = np.arange(0, 40, 10).reshape(4, 1)
```



repeats

np.tile (a, (1,3))

a

$$\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix}$$

b

[0 1 2]

4,3 — 2 — (4,3)

3, — 1 — (1,3)

2 } Add 1

7, 3, 4, 1 — (7, 3, 4, 1)

(2, 4) — (1, 1, 2, 4)

(4, 3) == (1, 3) / (4, 3) ✓

① Dimension should Match, If dimension are not matching add 1 in starting.

② Every element = every element only broad cast where element is 1

a = (4, 1) (3,)

(4, 1) → 4 3

(1, 3) → 4 3

(5) 10 ✓ (2,) 10. (1, 3) (1, 2)

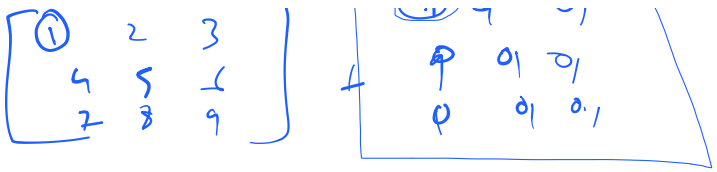
(8, 1, 6, 1) (7, 1, 5)

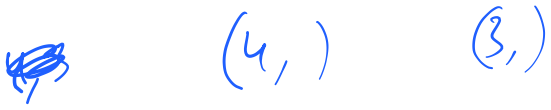
8, 1, 6, 1 → 8 7 6 5

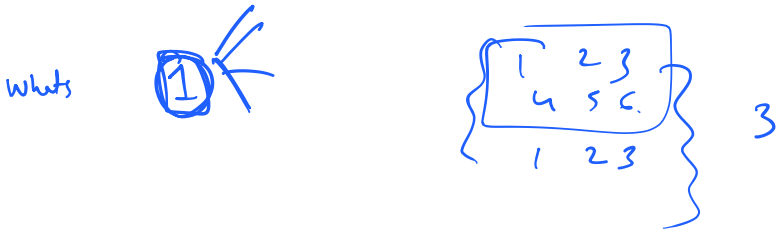
1, 7, 1, 5 → 8 7 6 5

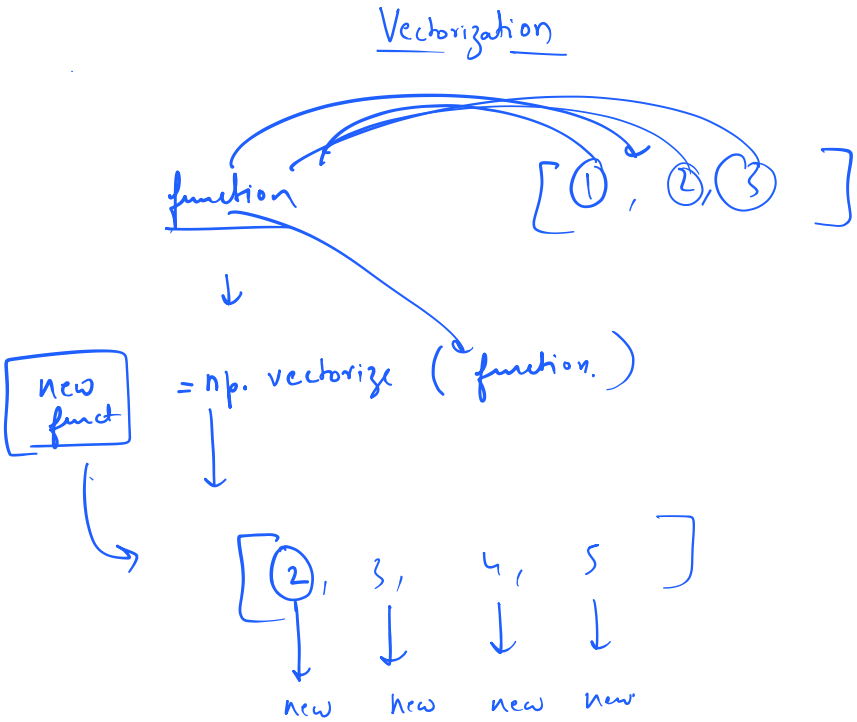
cm [20 30 40]
sas [50 60 10]

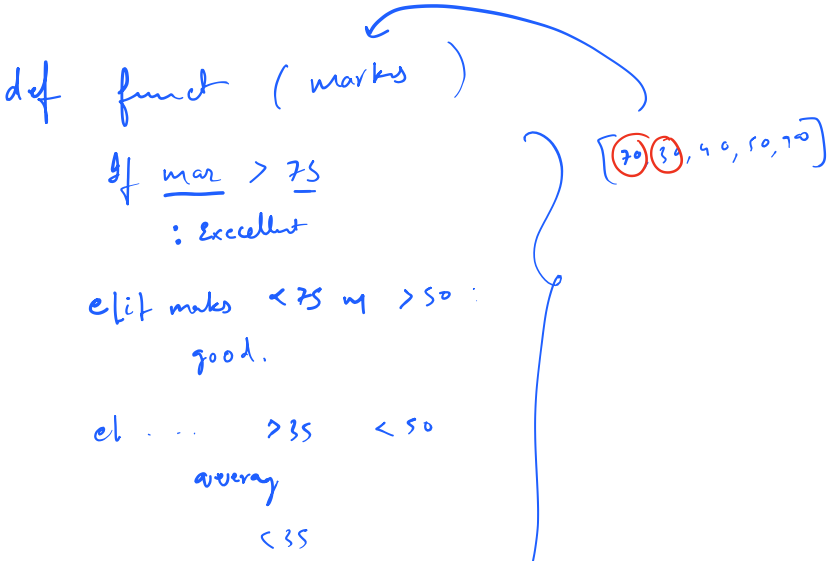
[20] [2]






whats
3


Vectorization
function
new funct
= np. vectorize (function)
new
new
new
new


def funct (marks)
if mar > 75
: excellent
elif marks < 75 & > 50 :
good.
el ... > 35 < 50
averag
< 35

hour.

def funct (marks)

if mar > 75
: Excellent

elif marks < 75 & > 50 :
good.

el ... > 35 < 50
average

< 35

hour.

[70, 50, 40, 20, 90]

np. vectorize → parallel.

Chatgpt — solving — Thinking — Interviews

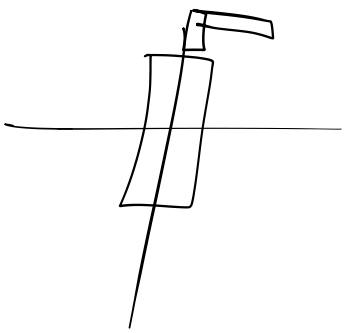
100 — 100 = .

- bill
- arr
- sum
- mul
- arr

array

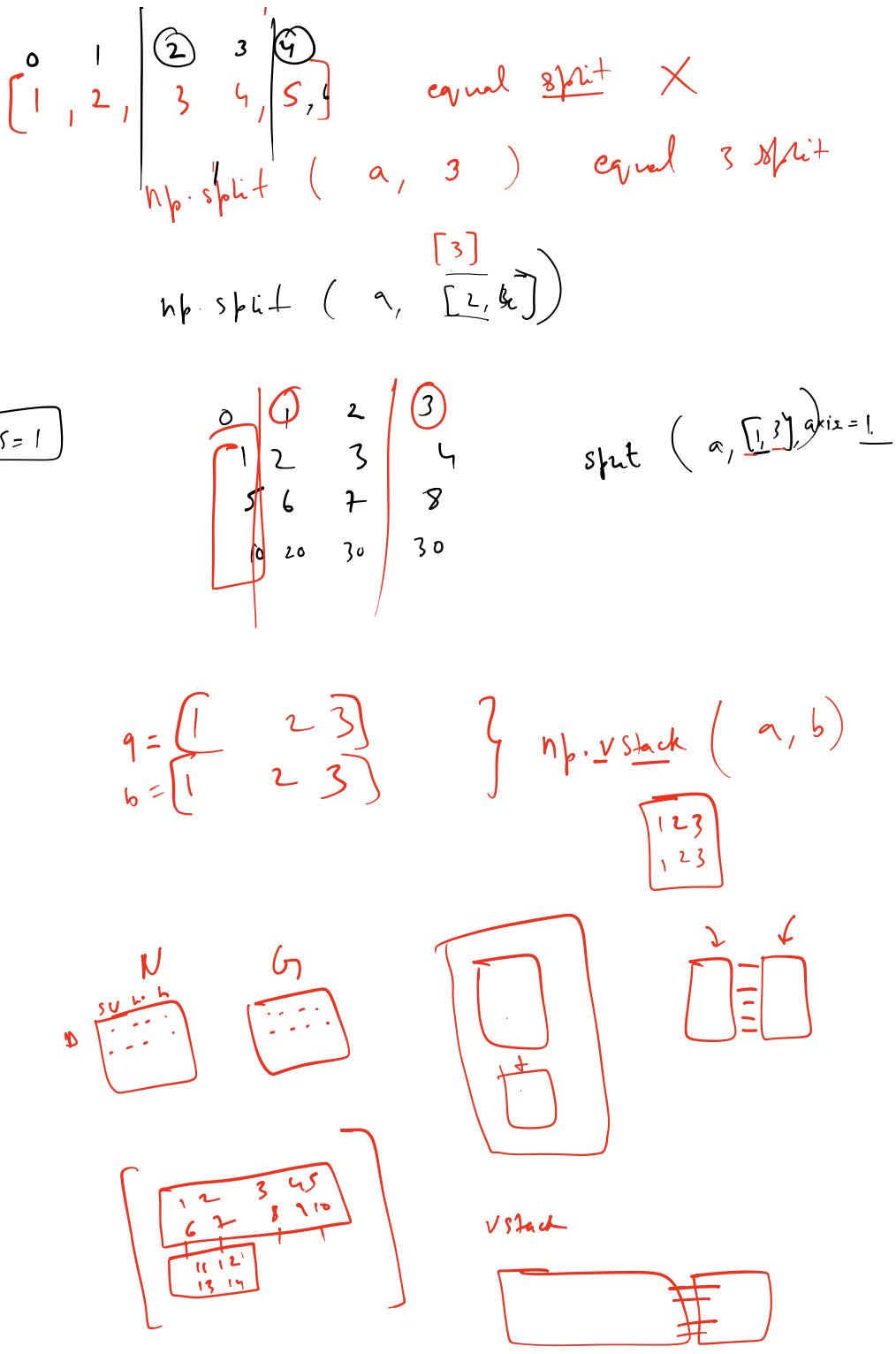
1	2	3
4	5	6
7	8	9

1	2	3
4	5	6



1	2	3
4	5	6
7	8	9

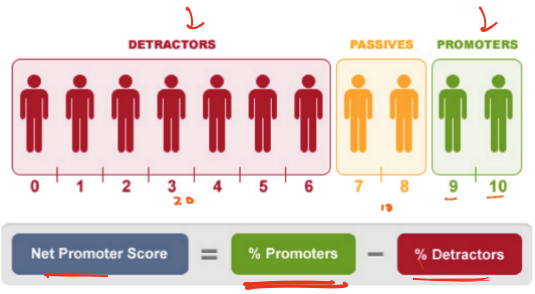
axis = 1


0 1 2 3 4
1 2 3 4 5 6
np.split (a, 3) equal 3 split
np.split (a, [2, 4])
[3]
axis=1
split (a, [1, 3], axis=1
a = [1 2 3]
b = [1 2 3]
np.vstack (a, b)
1 2 3
1 2 3
N
G
vstack
epam
N P S
Net promotor Score.
1 2 . . . 9 10

(6)

1 - 6
detractors

7 - 8
Netural



NPS [-100 100]

70

35
70 x 100
200
2

fit.txt

#date	step_count	mood	calories_burned	hours_of_sleep	active
06-10-2017	5464	Neutral	181	5	Inactive
07-10-2017	6041	Sad	197	8	Inactive
08-10-2017	25	Sad	0	5	Inactive
09-10-2017	5461	Sad	174	4	Inactive
10-10-2017	6915	Neutral	223	5	Active
11-10-2017	4545	Sad	149	6	Inactive
12-10-2017	4340	Sad	140	6	Inactive
13-10-2017	1230	Sad	38	7	Inactive
14-10-2017	61	Sad	1	5	Inactive
15-10-2017	1258	Sad	40	6	Inactive
16-10-2017	3148	Sad	101	8	Inactive
17-10-2017	4687	Sad	152	5	Inactive
18-10-2017	4732	Happy	150	6	Active
19-10-2017	3519	Sad	113	7	Inactive
20-10-2017	1580	Sad	49	5	Inactive
21-10-2017	2822	Sad	86	6	Inactive
22-10-2017	181	Sad	6	8	Inactive
23-10-2017	3158	Neutral	99	5	Inactive
24-10-2017	4383	Neutral	143	4	Inactive
25-10-2017	3881	Neutral	125	5	Inactive
26-10-2017	4037	Neutral	129	6	Inactive

.astype(int)

* added case study

9545126535

1pm

① → Broadcasting

[1, 2, 3] [4, 5, 6]

1+4
2+5
3+6

[1 2 3.] + 5 X

Jio k → [42, 41, 35, 22] → Kelvin.

+273,

↓

Rules

B → 1 ← all 7, 7000 - - -

① Hi I AKASH

② welcom to c

① Dimension should Match ✓ ×

R,C

(3,3)

1 2 3
4 5 6
7 8 9

(3,3)
3,3

[1 2 3] 1

(3,1)

(1, 3)
3 3

3 × 1

kg

R_1	70
R_2	82
R_3	91

1 × 2

kg cm

1	1000	10000

(3,1)
3 2

(1,2)
3 2

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

5 - scalar × ( )

Vectorization → def f1 (v) [ ]

age

< 18
> 18

return

> 18