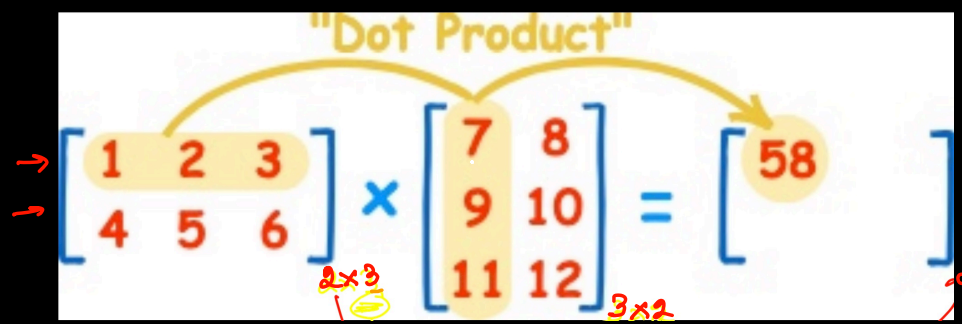
Numpy

4 hours ⟶ 4-5x

Matrix operation

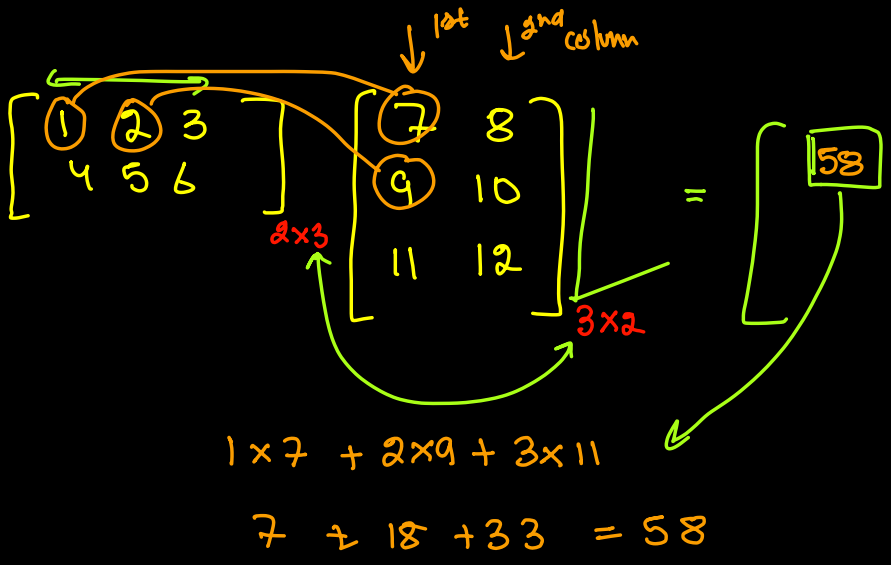
dot product



$$\underset{A}{\begin{bmatrix} \quad \end{bmatrix}_{r_1 c_1}} \quad \underset{B}{\begin{bmatrix} \quad \end{bmatrix}_{r_2 c_2}} = \underset{R}{\begin{bmatrix} \quad \end{bmatrix}_{r_1 c_2}}$$

$c_1 = r_2$

resultant matrix r_1, c_2



Dot Product

$$\begin{bmatrix} a & b \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \end{bmatrix}$$

$1\times2 \quad 2\times1$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

$2\times2 \quad 2\times1 \quad 2\times1$

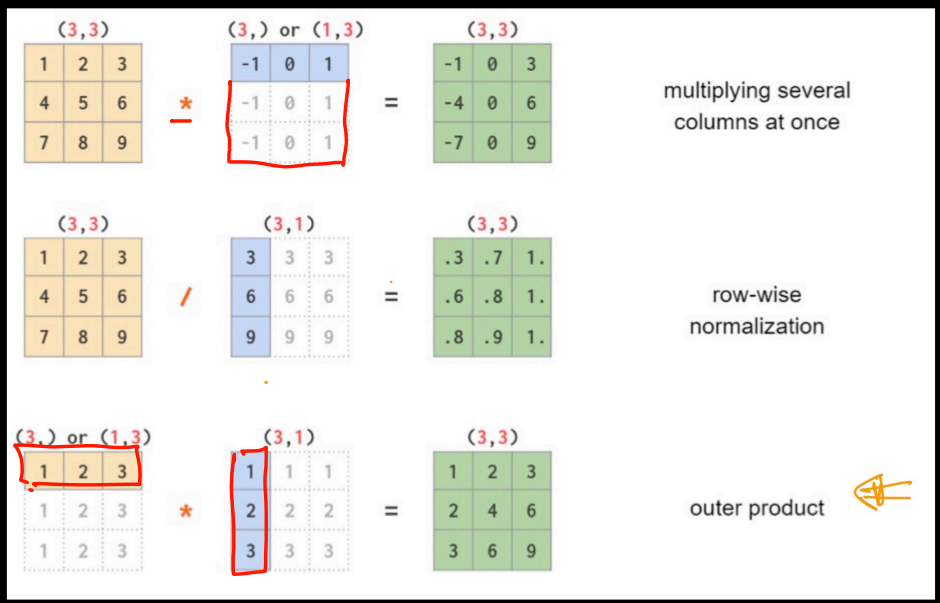
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} aw + by & ax + bz \\ cw + dy & cx + dz \end{bmatrix}$$

$2\times2 \quad 2\times2 \quad 2\times2$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$$

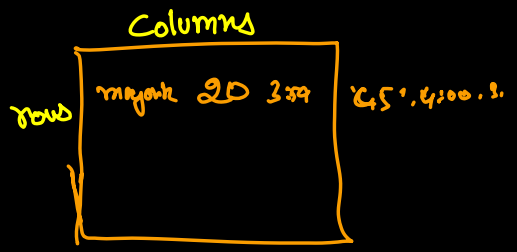
NumPy Array Operation

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} * \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_2 & a_3b_3 \\ a_4b_4 & a_5b_5 & a_6b_6 \\ a_7b_7 & a_8b_8 & a_9b_9 \end{bmatrix}$$


(3,3)
1 2 3
4 5 6
7 8 9
*
(3,) or (1,3)
-1 0 1
-1 0 1
-1 0 1
=
(3,3)
-1 0 3
-4 0 6
-7 0 9
multiplying several columns at once
(3,3)
1 2 3
4 5 6
7 8 9
/
(3,1)
3 3 3
6 6 6
9 9 9
=
(3,3)
.3 .7 1.
.6 .8 1.
.8 .9 1.
row-wise normalization
(3,) or (1,3)
1 2 3
1 2 3
1 2 3
*
(3,1)
1 1 1
2 2 2
3 3 3
=
(3,3)
1 2 3
2 4 6
3 6 9
outer product

Pandas

data manipulation library



Why pandas & not numpy?

homogenity

Column name

CSV ⇒ Comma seperated values

@ seperated values

1 , 2

1 @ 2 · ⌊——⌋ seperated values

Pandas

1. Open source library / free
2. easy to use for data structure & analysis
3. high performance

Types of Data structure in Pandas

Data structure	Dimn	Description
Series	1	1-D labeled homogenous size immutable
Dataframe	2	2-D labeled heterogenous size mutable
Panel	3	3-D labeled mutable

Pandas Dataframe

A pandas df is 2-D labeled & mutable data structure which stores data in form of rows & column.

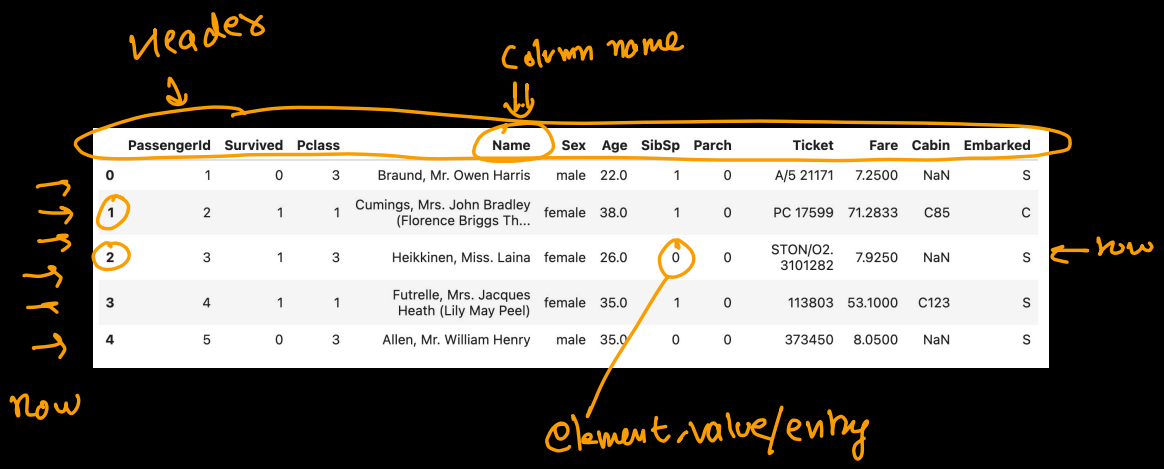
Features

→ Analogous to tabular data

→ can perform arithmetic opⁿ

→ Size mutable

→ Labeled



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Dataframe ⇒ If we provide a data in a (proper format) into a dataframe

Import data to pandas

pd.read_csv
pd.read_json
pd.read_clipboard

read_sql
read_excel

to_json to_csv

Operation on Pandas

Column

Add / Delete

Select

change

Rows

Addition & Deletion

Indexing & slicing

.iloc [____ , ____]

row ⇓ col ⇓