Date :

Data Science ?

→ Doing science like studies on Data

→ Involves running experiments, doing analysis, coming to conclusions

→ Involves building models and testing them.

→ Like cleaning a beaker before testing for chemicals, we clean Data.

How to do Data Science ?

→ Involves a lot of steps like data fetching, process cleaning, analyzing, presenting, ML modelling, deployment, scaling etc.

solve problems like :

- ETL (Extract Load Transform) jobs
- A/B Testing
- Data scraping and cleaning
- Doing analyses and presenting results
- Building and maintaining dashboards
- Building and deploying ML models

Numpy : Matrix & Tensor Operations

Pandas : Reading, cleaning & operating on Data

OpenCV : Reading & basic operations on Images

Seaborn & Matplotlib : Plotting of data

Scrapy & beautifulSoup : Fetching data from the Internet

MySQL : Fetching data from Database

Date :

Numpy -

what is numpy?

→ Fundamental of every data science library.

→ Almost every data science library in python is itself written in numpy.

→ Numpy deals with fundamental building block of all data science algos: Vectors and Matrices.

→ If data science was a car, Numpy would be the wheels.

when to use Numpy?

→ Optimized for doing numerical data science work.

→ Numpy is a python library written in C, making it blazingly fast.

→ It has a lot of submodules capable of doing a variety of tasks related to data science.

When NOT to use Numpy?

→ Doesn't work well with Non-numerical data.

→ If a lot of data is in text format, like reviews, you should not directly use Numpy.

→ Numpy always runs on your CPU. If however you are doing deep learning work on a GPU, numpy won't work.

What is a Vector?

→ List of numbers.

e.g: [1, 2, 3, 4, 10, -5]
int

[1, 'A', 2, 'B', 3] → LIST
str int

Date :

What is a matrix?
List of vectors i.e. a list of lists of numbers
e.g - [[1, 2, 3] , [4, 5, 6]]

Jupyter
↳ pip install jupyter

jupyter notebook

new → python

Some practice

```
In [1]: import numpy as np
In [2]: python_list = [1, 2, 3, 4, -5]
In [3]: python_list
Out [3]: [1, 2, 3, 4, -5]
In [4]: np.array                                  → to convert list to vector
Out [4]: array ( [1, 2, 3, 4, -5] )               → 1D Vector
In [5]: numpy_arr = np.array (python_list)
In [6]: numpy_arr.shape
Out [6]: (5,)                                     → tuple
In [7]: len (numpy_arr.shape)
Out [7]: 1
In [8]: python_list_list = [ [1, 2], [3, 4] ]
In [9]: python_list_list
Out [9]: [ [1, 2], [3, 4] ]
In [10]: np.array (python_list_list)
Out [10]: array ( [ [1, 2], [3, 4] ] )            → 2D list
```

Date :

Numpy : Data Types

Numpy arrays have data types.

→ Unlike in Python, numpy arrays have a specific data types

→ All the elements of an array must be same data type.

→ Numpy also has a few unique things to handle non-numeric data, infinities etc.

Data types :

int (Integers) (int 8, int 16, int 32, int 64) → $2^7 = 128$

uint (Unsigned Integers) (uint 8, uint 16, uint 64) → -2^7 to 2^7

bool (Boolean (True or False)) — save memory

float (floating-point numbers) (float 16, float 32, float 128) → accurate

complex (complex-values floating-point numbers)

Numpy : Slicing

→ Used for manipulating arrays

→ Reduce length selection, similar to python list

→ Can use Boolean arrays for selecting and filtering.

arr [:4] (Slice till 4)

(include till index 3)

as 3<4)

arr [:, :4]

↳ select all

Date :

Numpy : Transpose

Swap rows with columns and columns with rows

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3\times3} \quad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}_{3\times3}$$

Can change shape of non-square matrix

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}_{3\times4} = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}^{T}_{4\times3}$$

Numpy : Broadcasting

Operations on arrays of different dimensions

Using broadcasting, we can add this vector, and the vector gets broadcasted / copied so that it can be added.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}_{3\times4} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}_{3\times1} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 12 & 12 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

same rows ✓

↳ Broadcasting

Numpy : Tensors

Like matrix is a 2D array, tensor is a 3D array. Tensor could be also 4D, 5D ... arrays but mostly tensors are 3D arrays.

Date :

Numpy : Tensor Transpose

Transpose is usually done by swapping any 2 axes, or rotating the 3d matrix.

Pandas

→ Pandas is how we read and process data in python.

→ Almost every project uses pandas to read data, process data and make it ready for use in ML.

→ Uses numpy under the hood to make things fast.

→ Similar to using SQL, but more powerful for some use cases.

When do we use Pandas?

→ Reading and processing tabular data.

→ Reading data from files (like csv) locally or from the cloud. Can also be used to read from SQL.

→ Joining multiple data files, removing invalid or wrong data, calculating business relevant numbers and many more things are usually done using pandas.

Pandas is also usually used for feature engg for ML use cases.

When NOT use Pandas?

→ Non-tabular data should not be read using pandas

→ Images, Not suitable for very large data.

Date :

Numpy : Tensor Transpose

Transpose is usually done by swapping any 2 axes, or rotating the 3d matrix.

Pandas

→ Pandas is how we read and process data in python.

→ Almost every project uses pandas to read data, process data and make it ready for use in ML.

→ Uses numpy under the hood to make things fast.

→ Similar to using SQL, but more powerful for some use cases.

When do we use Pandas?

→ Reading and processing tabular data.

→ Reading data from files (like CSV) locally or from the cloud. Can also be used to read from SQL.

→ Joining multiple data files, removing invalid or wrong data, calculating business relevant numbers and many more things are usually done using pandas.

Pandas is also usually used for feature engg for ML use cases.

When NOT use Pandas?

→ Non-tabular data should not be read using Pandas

→ Images, Not suitable for very large data.

Date :

DataFrame & Series

Pandas Tables are called DataFrames and each column for which the DF is made up of is a Series.

Functions like head, tail, info etc allow us to see inside a pandas DataFrame.

Pandas can read data from SQL, CSV, Excel, JSON, Pickel etc.

Data Format	Users
.csv	A Plain tabular only format
.xlsx / .xlx	Familiar Microsoft Excel file
.json	Dictionary-based plain text data fort
Pickle (.pkl / .pck)	entire data frame is stored as Binary pickle file

Pandas : Data types

Pandas supports a few more data types than Numpy.

object (str or mixed)
int 64
float 64
bool

Date :

Read data ⇒

```
import pandas as pd

countries_data = pd.read_csv("d/countries.csv")

countries_data.tail()   # last rows

countries_data["country"].nunique()
```

Pandas Indexing

→ A dataframe index is analogous to python list index.

→ In lists, indices are used because they allow very fast & easy access to the list elements. They are simple integers from 0 onwards.

→ While usually dataframe indices will be simple integers from 0/1 that is not always the case.

→ We can set any column as the index of the dataframe, at which point it will be analogous to keys in a dictionary.

→ Dataframe rows can be accessed using loc and iloc.
↳ (useful when applied some filtering over)

Pandas : Masking

Similar to Numpy we can use Masking to select from a DataFrame.

```
countries_data.query("(country == 'Zimbabwe') & (Date == '23/1/2021')")
```

Date :

Pandas : NULL values

Data sometimes have blank values, similar to points in Yellow.

Pandas often "fill" these blank spaces with a "NaN" or "None".

NaN: stands for "Not a Number". Pandas will often use nan for blank values in numerical columns.

(for string) **None**: Python's equivalent of NULL. It will be used to fill blanks for object type columns.

NaT: Stands for "Not a Time". Pandas often use nat for blank values in datetime

however, not all null values will have NANs / None often an invalid value may be used in place of NaN / None

Eg: Using -1 in an Age column to show that age is unknown.

Pandas : Kinds of join / merge

There are 5 main types of joins / merge

INNER: The default, where only rows which are present on both dataframes are kept.

Date :

LEFT: Where rows which are present on both dataframes are kept, along with leftover rows from the left dataframe.

RIGHT: Where rows which are present on both dataframes are kept, along with leftover rows from the right dataframe.

OUTER/FULL: Where rows which are present on both dataframes are kept, along with leftover rows from both left & right dataframes.

ANTI-JOIN: Where rows which are present on both dataframes are removed, keeping only leftover rows from both dataframes.

There are also anti-left and anti-right join which only keep left over rows from left data frame and right dataframe respectively.

Data Scraping -
Process of getting useful, semi-structured data from the Internet is called Web Scraping.

Basics of HTML, CSS and JSON needed for Web Scraping.

Types of web scrapping -
- Request/API scraping
- Static HTML Scraping
- Dynamic JS scraping

Date :

Request / API scraping -
We directly load data from Internet via an "API"
Considered to be the easiest type of scraping.

Static HTML scraping -
We load a simple "static" website. We extract HTML data available on the website. Also considered to be easy.

Dynamic JS scraping -
We load a complex "dynamic" website. We interact with it like a person.

1] API Scraping
API (Application Programming Interface) is a way to get an external program to do something.

For scraping, we send a "request" to the API program to get data for us.

JSON is a way to store data in text

→ We will be using requests python library for easy API scraping.

2] HTML scraping -
Tools - Scrapy & BeautifulSoup.

→ Scrapy will allow us to create a simple way to scrape and crawl our desired website, and even save our data as CSV. It will allow us to easily get all images for the books.

Date : _____

How to use Scrapy?

1) Initialise a project
2) Initialise a spider with a start URL.
3) Write the scraper
4) Run the scraper

Spider in web scraping →

In web scraping, a spider is a function that allows us to find all pages you want to scrape.

Data Visualization

Intro to DV

Creating charts and graphs for easy data exploration and understanding.

We can use Excel, Power BI, Tableau etc.

Advantage of doing data visualization in python are larger amount of usable data, ability to integrate visualizations into an API or website

How to do Data Visualization in Python?

Main library for data viz in Python is Matplotlib

Scatter Plot –

→ The most basic type of plot.
→ Shows points in a 2D grid
Useful to see
→ Patterns and relationships
→ See clusters and patterns.

Parts of Plot

- Figure
- Axes (1 pair of X axis & Y axis are called as Axes).
- X axis & Y axis
- Legend.

Bar Plot

Plots for number vs category

→ Bar plot is useful when visualising relationships between a category and income.

eg Gender vs Income

Histograms

Histograms are almost vertical bar plots where the height of the bar denotes the count of the category.

Histograms of numerical data can be made using Binning.

$$binSize = \frac{max(X) - min(X)}{bins}$$