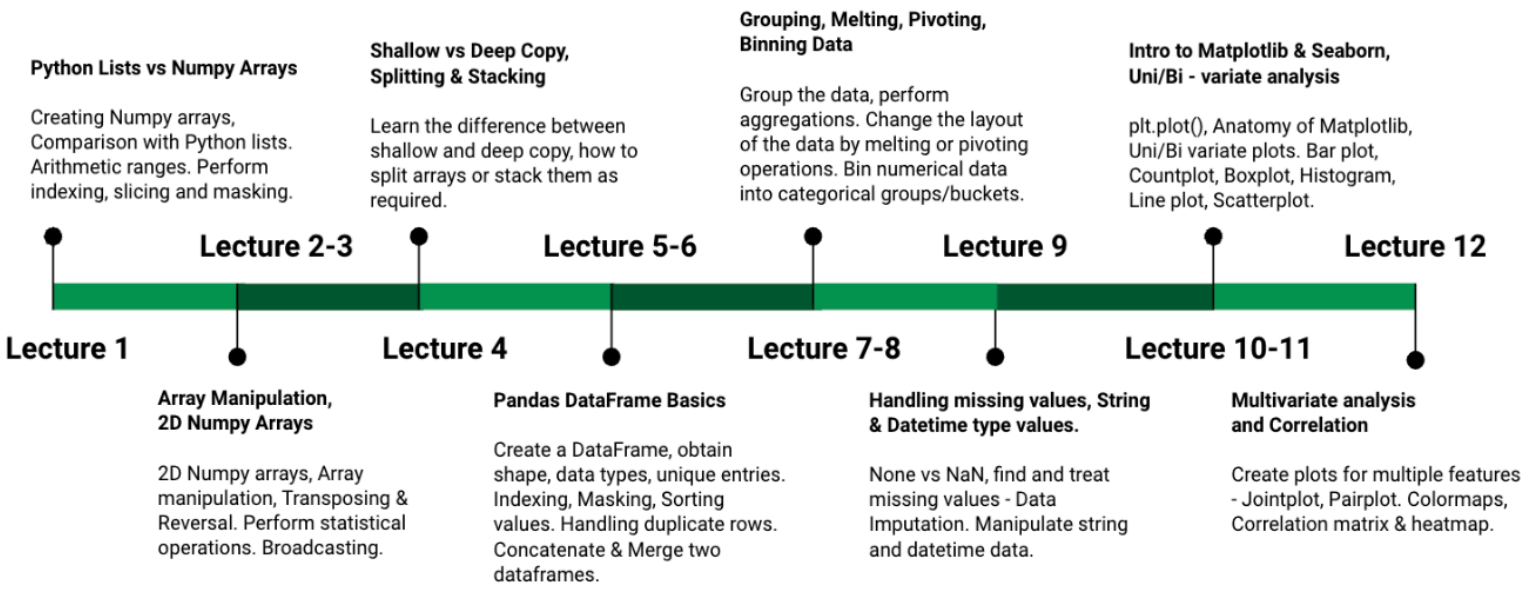
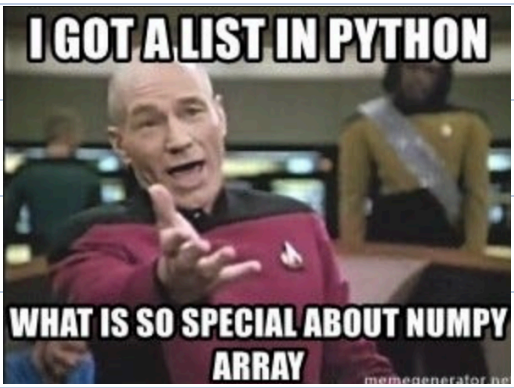
## Module Importance

- Libraries like numpy, pandas, matplotlib & seaborn form the foundation of Python-based data analysis and visualization.
- They enable Data Analysts to efficiently manipulate, analyze, and visualize data, leading to informed decision-making.
- **Pandas:** Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrame and Series, which are efficient for handling labelled data.
- **NumPy:** NumPy is the fundamental package for scientific computing in Python. It provides support for arrays, matrices, and mathematical functions, enabling efficient operations.
- **Matplotlib:** Matplotlib is a versatile plotting library for creating static, interactive, and publication-quality visualizations.
- **Seaborn:** Seaborn is a statistical data visualization library built on top of Matplotlib. It provides high-level functions for creating informative and attractive visuals.

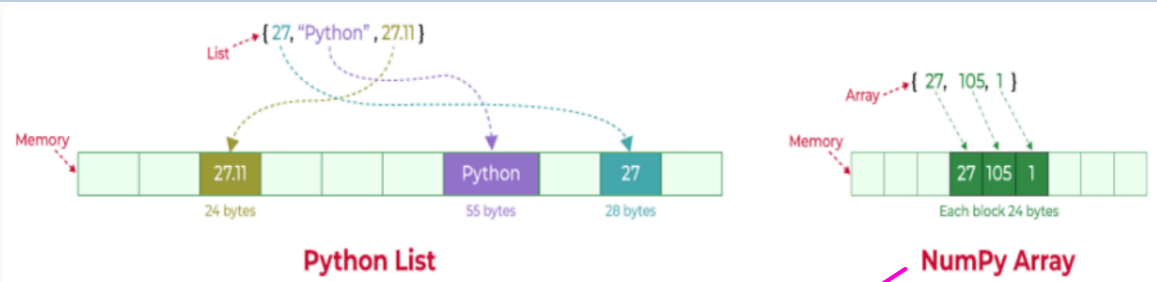


## Agenda

- Python Lists vs Numpy Array
  - Importing Numpy
  - Why use Numpy?
- Dimension & Shape
- Type Conversion in Numpy Arrays
- Indexing & Slicing
- Case Study - NPS




# Python Lists vs Numpy Arrays




Heterogenous data

int, str, boolean

Homogenous data → single data type.

⇒ Continuous memory allocation

Very fast

All code is written in C.

All properties / functions of list can be used in numpy array.

Indexing & slicing is also same as list.

## Importing Numpy

Recall how we used to import a module/library in Python.

- In order to use Python Lists, we do not need to import anything extra.
- However to use Numpy Arrays, we need to import it into our environment, as it is a Library.

Generally, we do so while using the alias `np`.

```
1 import numpy as np
```

**Note:**

- In this terminal, we will already have numpy installed as we are working on Google Colab.
- However, when working on an evironment that does not have it installed, you'll have to install it the first time working.
- This can be done with the command: `!pip install numpy`

**What is the major reason behind numpy's faster computation?**

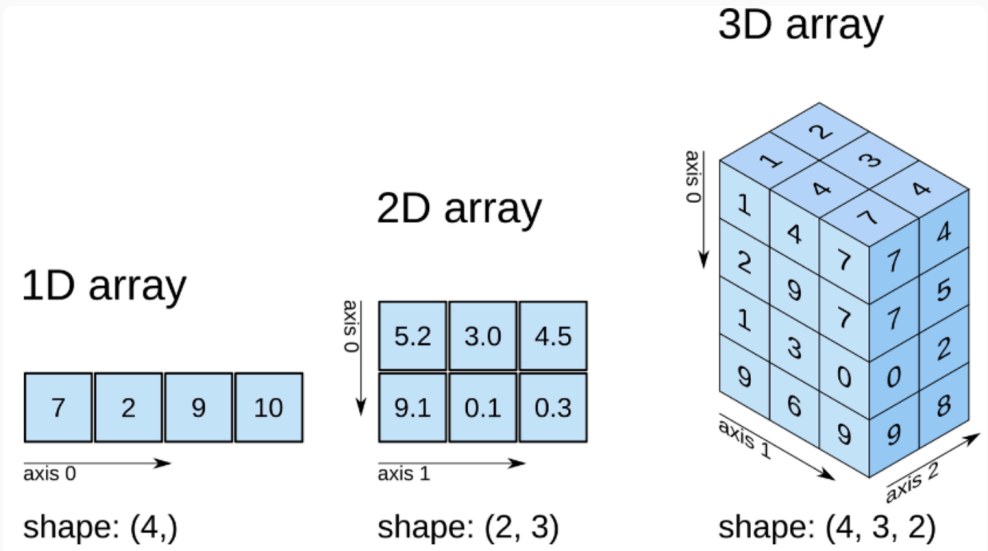
- Numpy array is densely packed in memory due to it's **homogenous** type.
- Numpy functions are implemented in **C programming launguage**.
- Numpy is able to divide a task into multiple subtasks and process them **parallelly.**

# Dimensions & Shape

a.ndim a.shape

`ndim` specifies the number of dimensions of the array i.e. 1D (1), 2D(2), 3D (3) and so on.

`shape` returns the exact shape in all dimensions, that is (3,3) which implies 3 in axis 0 and 3 in axis 1.




## `np.arange()`

Let's create some sequences in Numpy.

We can pass **starting** point, **ending** point (not included in the array) and **step-size**.

**Syntax:**

- `arange(start, end, step)`

`np.arange()` behaves in the same way as `range()` function.

**But then why not call it np.range?**

- In `np.arange()`, we can pass a **floating point number** as **step-size**.

Code

```
arr3 = np.arange(1, 5, 0.5)
arr3
```

Output

```
array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5])
```

# Indexing

- Similar to Python lists

```
a = np.arange(100)
a[ ]
```

↳ index / -ve index

list of indexes

slicing

## Fancy Indexing (Masking)

- Numpy arrays can be indexed with boolean arrays (masks).
- This method is called **fancy indexing** or **masking**.

# Use Case: NPS (Net Promoter Score)

**Imagine you are a Data Analyst @ Airbnb**

You've been asked to analyze user survey data and report NPS to the management.

**But, what exactly is NPS?**

Have you all seen that every month, you get a survey form from Scaler?

- This form asks you to fill in feedback regarding how you are liking the services of Scaler in terms of a numerical score.
- This is known as the **Likelihood to Recommend Survey**.
- It is widely used by different companies and service providers to evaluate their performance and customer satisfaction.

---

How likely is it that you would recommend [company X] to a friend or colleague?

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Not at all likely | | | | | | | | | | Extremely likely |

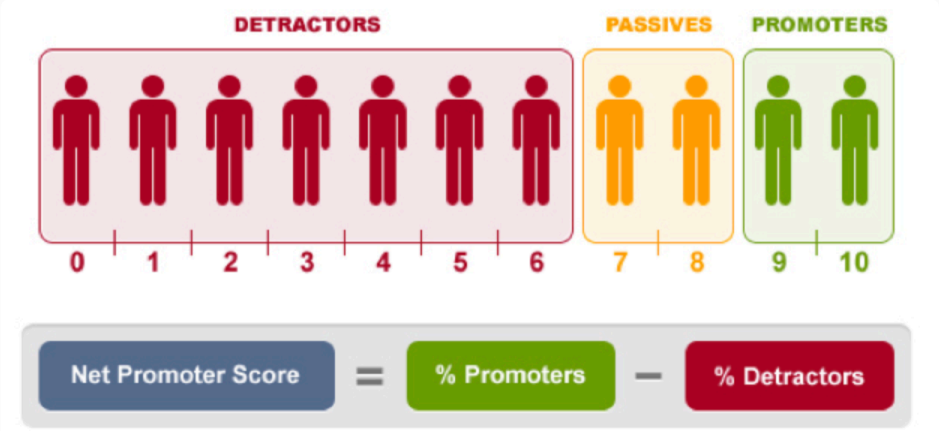
---

Responses are given a scale ranging from 0–10,

- with 0 labeled with “Not at all likely,” and
- 10 labeled with “Extremely likely.”

Based on this, we calculate the **Net Promoter Score**.

## How to calculate NPS score?



## Range of NPS

- If all people are promoters **(rated 9-10)**, we get **100 NPS**.
- Conversely, if all people are detractors **(rated 0-6)**, we get **– 100 NPS**.
- Also, if all people are neutral **(rated 7-8)**, we get a **0 NPS**.

Therefore, the range of NPS lies between **[–100,100]**

Generally, each company targets to get at least a threshold NPS.

- For Scaler, this is a score of 60.
- This means that if ***NPS*** > 60, it is great performance of the company.

Naturally, this varies from business to business.

## How is NPS helpful?

**Why would we want to analyse the survey data for NPS?**

NPS helps a brand in gauging its brand value and sentiment in the market.

- Promoters are highly likely to recommend your product or sevice. Hence, bringing in more business.
- whereas, Detractors are likely to recommend against your product or service's usage. Hence, bringing the business down.

These insights can help business make customer oriented decision along with product improvisation.