Summer of Code

Artificial Intelligence

(Machine Learning & Deep Learning)

Instructor
Wajahat Ullah
- *Research Assistant* (DIP Lab)

Duration
03 Months
(September – November)

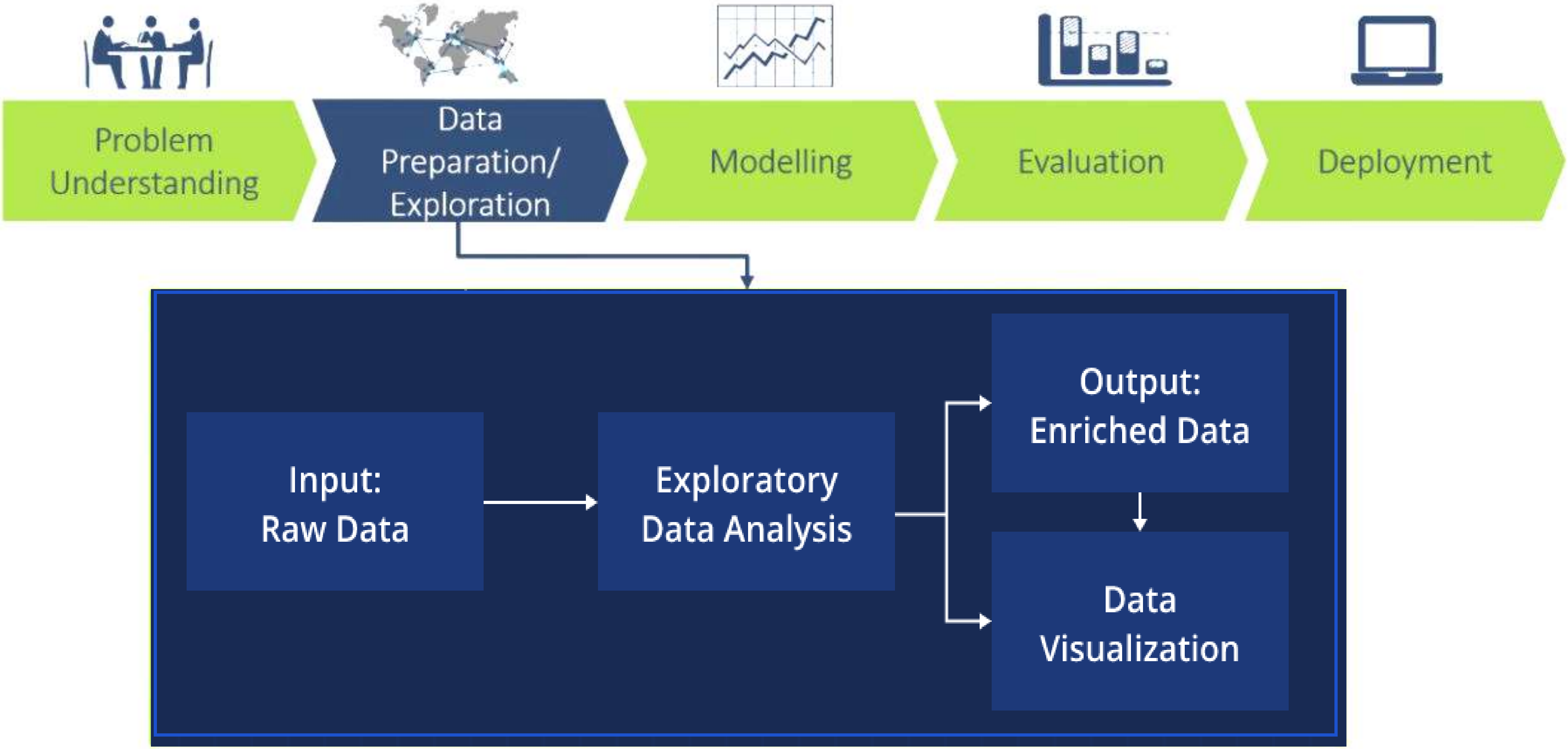
Day 01 – Exploratory Data Analysis
(Introduction to NumPy)

Objectives:

- ❖ What is Exploratory Data Analysis?
- ❖ Introduction to NumPy
- ❖ NumPy Arrays

Exploratory Data Analysis

The process of examining datasets to summarize their main characteristics. It is a crucial step in the data analysis workflow to gain a deep understanding of the dataset before modeling.



Exploratory Data Analysis

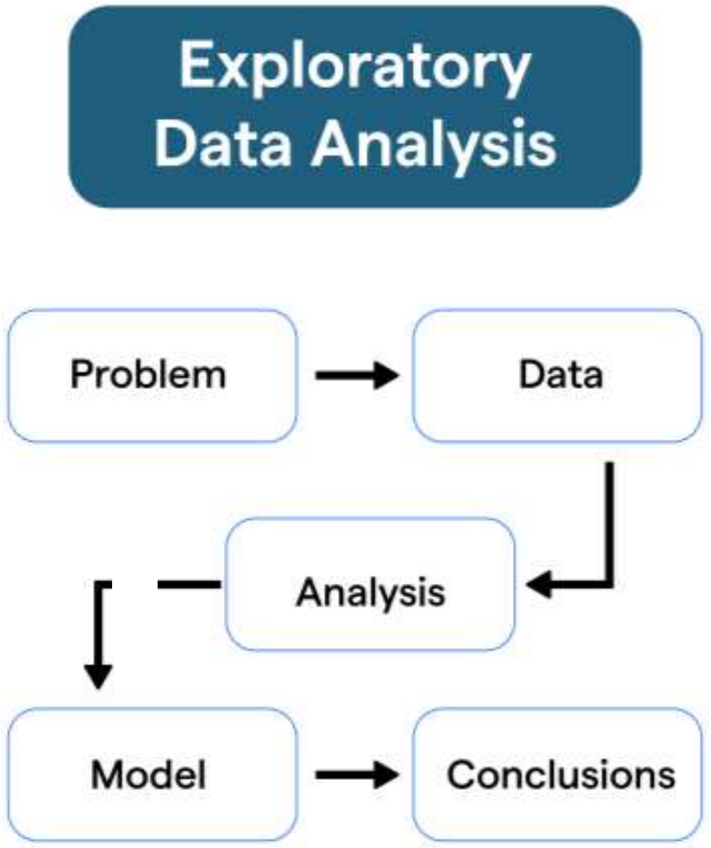
- The process of examining datasets – often with visual methods – to summarize their main characteristics.
- It is a crucial step in the data analysis workflow to gain a deep understanding of the dataset before modeling.

Objectives:

- Understand data structure and underlying patterns.
- Identify anomalies, missing values, and outliers.
- Detect trends and relationships between variables.
- Form hypothesis to inform further analysis or modeling.

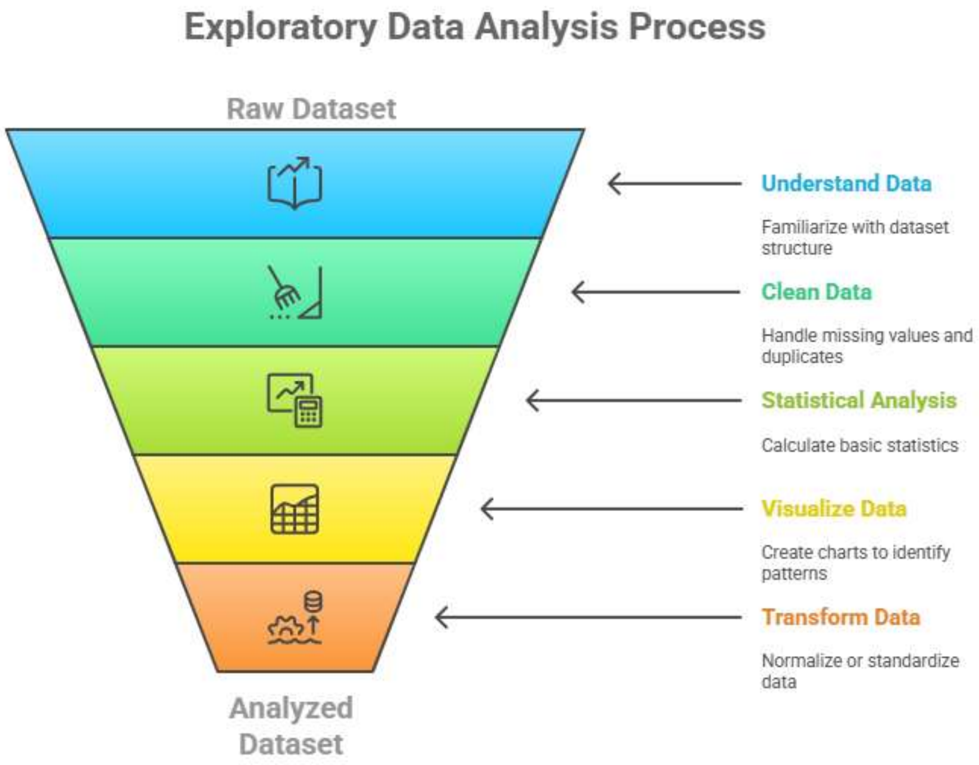
Importance:

- Provides insights for data-driven decision making.
- Improves predictive model quality by identifying issues early.
- Ensures data integrity and readiness for analysis.



Key Steps in EDA

- **Understanding the Data:** Get familiar with the dataset, check number of rows, columns, and data types.
- **Data Cleaning:** Handle missing values, duplicates, and inconsistencies.
- **Statistical Analysis:** Use basic statistics (mean, median, standard deviation) to summarize each variable.
- **Data Visualization:** Use charts to uncover patterns, trends and outliers.
- **Data Transformation** (if needed)**:** Normalize or standardize values, or convert data into a better format for further analysis or modeling.


Exploratory Data Analysis Process
Raw Dataset
Understand Data
Familiarize with dataset structure
Clean Data
Handle missing values and duplicates
Statistical Analysis
Calculate basic statistics
Visualize Data
Create charts to identify patterns
Transform Data
Normalize or standardize data
Analyzed Dataset

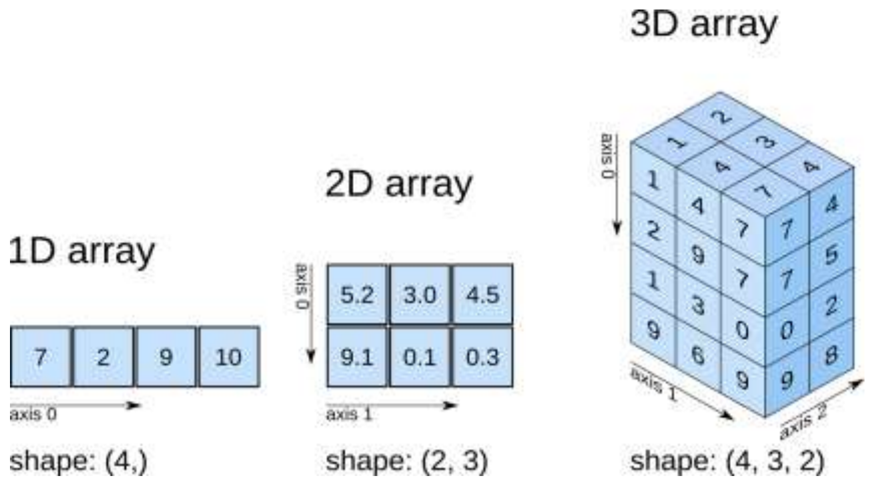
Python Libraries for EDA

- **`NumPy:`** Essential for numerical operations in Python, it provides support for multi-dimensional arrays, along with mathematical functions on these arrays.
- **`Pandas:`** Library for data manipulation and analysis. It makes it easy to clean, transform, and aggregate data.
- **`Matplotlib:`** A versatile plotting library used to create static, interactive, and animated visualizations in Python.
- **`sklearn:`** Primarily a machine learning library but includes many tools useful for data preprocessing and feature selection, which are key parts of EDA.

Introduction to NumPy

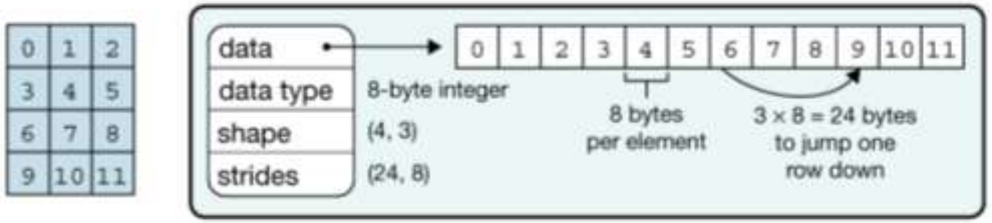
NumPy: “Numerical Python” – The foundation of scientific computing in Python

- **Initial release:** As **Numeric** (1995); as NumPy, (2006)
- Python library for efficient array operations and mathematical computations
- **Core Data Structure**: N-dimensional array (ndarray)



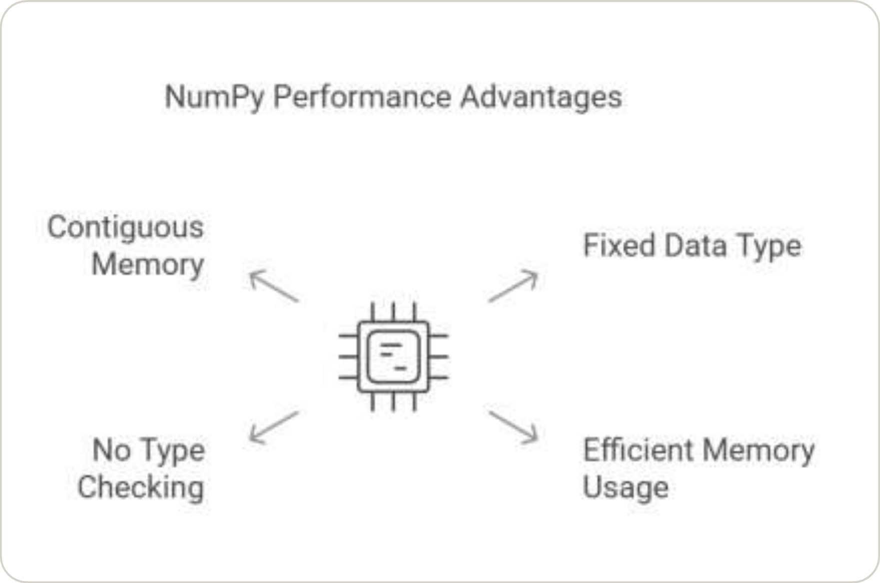
Why NumPy?

- **Speed:** Up to 50x faster than Python lists.
- **Memory Efficient:** Uses less memory than traditional Python data structures.
- **Foundation:** Backend for Pandas, SciPy, Matplotlib, and machine learning libraries
- **Versatile:** Supports 1D arrays, matrices, tensors, and higher dimensions.



Why NumPy Arrays are Faster Than Lists

1. **Fixed Data Type**:
   - NumPy arrays have a uniform data type, which eliminates the overhead of managing different data types like in Python lists.
2. **Contiguous Memory Storage**:
   - Data stored in continuous memory blocks enhances cache efficiency and reduces memory consumption.
3. **No Runtime Type Checking**:
   - Operations on NumPy arrays skip runtime type checks, making computations faster compared to Python lists.
4. **Optimized Implementation**:
   - Core operations written in C and Fortran for maximum performance.



Differences Between Python Lists and NumPy Arrays:

Feature	**Python Lists**	**NumPy Arrays**
Data Type	Mixed Types	Homogeneous types
Memory Efficiency	Low	High
Computation Speed	Slow	Fast

Thank You