**AI LAB – NumPy, Pandas & Matplotlib**

**Objective:** Understand and apply Python libraries for numerical computation, data handling, and visualization as a foundation for AI model development.

**1. NumPy – Numerical Python**

* NumPy helps you store and process numbers in arrays (like lists but faster).
* It’s used in AI for fast mathematical operations on data.
* Especially useful when working with matrix operations, like in neural networks.

**Code and Detailed Explanation**

# Step 1: Import the NumPy library

import numpy as np

We import NumPy and give it a nickname np to use it easily.

# Step 2: Create a basic list

my\_list = [1, 2, 3, 4]

This is a normal Python list storing 4 numbers.

# Step 3: Convert list into a NumPy array

my\_array = np.array(my\_list)

NumPy array is like a faster, more powerful list made for math operations.

# Step 4: Perform math on the array

print("Original array:", my\_array)

print("Add 10 to each number:", my\_array + 10)

print("Multiply each number by 2:", my\_array \* 2)

These operations are vectorized (no loop needed).

# Step 5: Create a 2D array (like a matrix)

my\_2d\_array = np.array([[1, 2], [3, 4]])

print("2D Array:\n", my\_2d\_array)

print("Sum of all values:", np.sum(my\_2d\_array))

print("Transpose of 2D array:\n", my\_2d\_array.T)

2D arrays are like tables and are used for AI inputs. .T transposes rows into columns.

**2. Pandas – Data Handling Library**

* Pandas is used for data analysis.
* Helps you work with tables, like rows and columns in Excel.
* In AI, we often clean, filter, and organize data before sending it to the model.

**Code and Detailed Explanation**

# Step 1: Import the Pandas library

import pandas as pd

We use pd as a short name to access Pandas.

# Step 2: Create sample data

data = {

'Name': ['Ali', 'Sara', 'John'],

'Marks': [85, 90, 78]

}

A Python dictionary holding student names and their marks.

# Step 3: Convert data into a DataFrame

df = pd.DataFrame(data)

print("Student Data:\n", df)

DataFrame is a Pandas table. It looks like a spreadsheet.

# Step 4: View only the names column

print("Names:\n", df['Name'])

This selects just one column from the table.

# Step 5: Filter students with marks > 80

print("Marks > 80:\n", df[df['Marks'] > 80])

Returns only the rows where the condition is true.

**3. Matplotlib – Drawing Graphs**

Matplotlib is used to visualize data through charts and graphs.

* Helps us “see” trends and patterns, which is important before making an AI model.

**Code and Detailed Explanation**

# Step 1: Import the plotting library

import matplotlib.pyplot as plt

We use plt as a short name to access plotting functions.

# Step 2: Sample data

names = ['Ali', 'Sara', 'John']

marks = [85, 90, 78]

These are same values as we used in Pandas.

# Step 3: Bar chart

plt.bar(names, marks)

plt.title("Student Marks")

plt.xlabel("Students")

plt.ylabel("Marks")

plt.show()

Shows student marks in vertical bars.

# Step 4: Line chart

plt.plot(names, marks, marker='o')

plt.title("Student Marks Line Chart")

plt.xlabel("Students")

plt.ylabel("Marks")

plt.show()

Draws a line joining the points (like progress tracking).

**Lab Task**

**Title: Student Marks Analyzer using NumPy, Pandas, and Matplotlib**

**Instructions:**

1. Use **Pandas** to create a table with 5 students and their marks in 3 subjects.
2. Use **NumPy** to:
   * Convert marks into arrays.
   * Calculate the average mark of each student.
3. Use **Matplotlib** to:
   * Draw a **bar chart** of total marks per student.
   * Draw a **line chart** showing the progress of each subject.