

LAB 1: INTRODUCTION TO PYTHON FOR MACHINE LEARNING

AIM:

To understand basic Python programming concepts and libraries used in Machine Learning.

INTRODUCTION:

Machine learning is a field where computers learn from data to make predictions. Python is popular for machine learning because it is easy to learn and has many helpful libraries.

The main libraries we use are NumPy for numerical calculations, Pandas for handling data, Matplotlib for creating graphs, and Scikit-learn for machine learning algorithms. These libraries help us write less code and get results faster.

In this lab, we will learn the basics of these libraries and see how they work with simple examples.

THEORY:

Python provides several key libraries that form the basis of machine learning work:

1. NumPy (Numerical Python):

NumPy is used for working with numbers and arrays. An array is like a list, but it can have multiple dimensions and supports fast mathematical operations. When we need to do calculations on thousands or millions of numbers, NumPy is much faster than regular Python.

Key features:

- Creates arrays of any dimension
- Performs element-wise operations
- Has functions for statistics like mean, median, standard deviation

- Supports linear algebra operations

2. Pandas (Panel Data):

Pandas gives us two main data structures - Series and DataFrame. A Series is like a single column, and a DataFrame is like a table with rows and columns.

Key features:

- Reads data from CSV, Excel, databases
- Handles missing data
- Filters and sorts data easily
- Groups data for analysis

3. Matplotlib:

Matplotlib is used for creating graphs and charts. It helps us visualize data to understand patterns better.

Key features:

- Line plots, scatter plots, bar charts
- Customizable labels, titles, colors
- Can save plots as image files
- Works well with NumPy and Pandas

4. Scikit-learn:

Scikit-learn provides simple tools for data analysis and modeling. It is built on NumPy, Pandas, and Matplotlib.

Key features:

- Classification algorithms (identifying categories)

- Regression algorithms (predicting numbers)
- Clustering algorithms (finding groups in data)
- Tools for splitting data into training and testing sets

PROCEDURE:

1. Install Python (version 3.7 or higher recommended)
2. Install required libraries using pip: `pip install numpy pandas matplotlib scikit-learn`
3. Open a Python editor or Jupyter Notebook
4. Import the libraries
5. Create simple examples using each library
6. Run the code and observe outputs
7. Create visualizations to understand the data

CODE EXAMPLES:

Example 1 - NumPy Arrays and Operations:

```
import numpy as np

# Arrays

arr1 = np.array([1, 2, 3, 4, 5])
arr2 = np.array([10, 20, 30, 40, 50])

print("Array 1:", arr1)
print("Array 2:", arr2)

# Operations

print("Sum:", arr1 + arr2)
```

```

print("Product:", arr1 * arr2)

# Statistics

print("Mean of arr1:", np.mean(arr1))
print("Standard deviation:", np.std(arr1))
print("Maximum value:", np.max(arr2))

# 2D array

matrix = np.array([[1, 2, 3],
                  [4, 5, 6],
                  [7, 8, 9]])

print("Matrix:")
print(matrix)
print("Shape:", matrix.shape)

```

Example 2 - Pandas DataFrames:

```

import pandas as pd

# DataFrame

student_data = {
    'Name': ['Rahul', 'Priya', 'Amit', 'Sneha', 'Vikram'],
    'Age': [21, 22, 20, 21, 23],
    'Marks': [85, 92, 78, 88, 95],
    'City': ['Mumbai', 'Delhi', 'Pune', 'Mumbai', 'Delhi']
}

```

```

df = pd.DataFrame(student_data)

print(df)

# Analysis

print("Average marks:", df['Marks'].mean())

print("Students from Mumbai:")

print(df[df['City'] == 'Mumbai'])

```

Example 3 - Matplotlib Visualizations:

```

import matplotlib.pyplot as plt

# Line plot

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']

temperature = [15, 17, 22, 28, 32, 35]

plt.plot(months, temperature, marker='o')

plt.xlabel('Month')

plt.ylabel('Temperature')

plt.title('Monthly Temperature')

plt.show()

```

Example 4 - Simple Machine Learning with Scikit-learn:

```

from sklearn.linear_model import LinearRegression

# Sample data

hours = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)

```

```

marks = np.array([50, 55, 65, 70, 80])

# Train model

model = LinearRegression()
model.fit(hours, marks)

# Predict

new_hours = np.array([[6]])
prediction = model.predict(new_hours)
print("Predicted marks for 6 hours:", prediction)

```

OUTPUT:

NumPy Output:

Array 1: [1 2 3 4 5]

Array 2: [10 20 30 40 50]

Sum: [11 22 33 44 55]

Mean of arr1: 3.0

Maximum value: 50

Matrix:

[[1 2 3]

[4 5 6]

[7 8 9]]

Shape: (3, 3)

Pandas Output:

Name	Age	Marks	City

```
0    Rahul    21     85  Mumbai
1    Priya    22     92  Delhi
2    Amit     20     78  Pune
3    Sneha    21     88  Mumbai
4    Vikram   23     95  Delhi
```

Average marks: 87.6

Students from Mumbai:

	Name	Age	Marks	City
0	Rahul	21	85	Mumbai
3	Sneha	21	88	Mumbai

Matplotlib Output:

A line graph is displayed showing temperature rising from 15°C in January to 35°C in June.

Scikit-learn Output:

Predicted marks for 6 hours: [85.]

CONCLUSION:

In this lab, I learned about the basic Python libraries used for machine learning. NumPy helps with fast calculations on arrays and matrices. Pandas makes it easy to work with data in table format and perform analysis operations. Matplotlib allows us to create graphs and visualize data patterns. Scikit-learn provides simple tools to build and train machine learning models.

These libraries work well together and form the foundation of most machine learning projects. NumPy handles the numerical computations, Pandas organizes and cleans the data, Matplotlib helps visualize the results, and Scikit-learn builds the predictive models.

The code examples showed practical uses of each library. NumPy performed array operations efficiently, Pandas filtered and analyzed student data, Matplotlib plotted temperature trends, and Scikit-learn predicted marks based on study hours. All libraries have simple syntax that makes them easy to learn and use.

Understanding these basics will help in future labs where we will use these libraries to build more complex machine learning models and solve real-world problems.