

PRACTICAL-1

Study Python ecosystem for Machine learning: Python, SciPy, Scikit-learn.

- To understand and implement the core libraries used in the Python Machine Learning , including NumPy, Pandas, Matplotlib, Seaborn, SciPy, and Scikit-learn.

1. Introduction to the Ecosystem

- Machine Learning in Python relies on a stack of specialized libraries that handle everything from data manipulation to model evaluation.
- **NumPy**: The foundation for numerical computing (N-dimensional arrays).
- **Pandas**: The primary tool for data manipulation and analysis (DataFrames).
- **Matplotlib/Seaborn**: Libraries for data visualization.
- **SciPy**: Used for advanced scientific and statistical operations.
- **Scikit-learn**: The industry standard for implementing machine learning algorithms.

2. NumPy (Numerical Python)

- **Purpose**: The fundamental package for scientific computing with Python. It provides high-performance multidimensional array objects.
- **Installation**: `pip install numpy`
- **Working**: It uses a contiguous block of memory and vectorized operations (written in C) to perform calculations much faster than standard Python lists.
- **Key Functions**:
 - `np.array()`: Creates an N-dimensional array.
 - `np.reshape()`: Changes the shape of data without changing its content.
 - `np.dot()`: Performs matrix multiplication.
- **Where Used**: Used in almost every ML algorithm to represent images, sound waves, and structured data as matrices.

3. Pandas (Python Data Analysis)

- **Purpose**: Provides high-level data structures (DataFrames) designed to make working with "relational" or "labeled" data easy and intuitive.
- **Installation**: `pip install pandas`

- **Working:** It acts like a "Programmatic Excel." It allows for data cleaning, merging, and filtering using simple syntax.
- **Key Functions:**
 - `pd.read_csv()`: Loads external data into a DataFrame.
 - `df.groupby()`: Aggregates data based on specific categories.
 - `df.fillna()`: Handles missing values in a dataset.
- **Where Used:** Used in the "Data Wrangling" stage to clean data before feeding it into a model.

4. Matplotlib

- **Purpose:** A low-level library for creating static, animated, and interactive visualizations in Python.
- **Installation:** `pip install matplotlib`
- **Working:** It works on a "Figure" and "Axes" logic. You create a figure (the canvas) and then add plots to the axes.
- **Key Functions:**
 - `plt.plot()`: Creates line graphs.
 - `plt.xlabel()` / `plt.ylabel()`: Adds labels to the axes.
- **Where Used:** Used for basic Exploratory Data Analysis (EDA) and checking the error rate of models over time.

5. Seaborn

- **Purpose:** A high-level interface based on Matplotlib for drawing attractive and informative statistical graphics.
- **Installation:** `pip install seaborn`
- **Working:** It integrates closely with Pandas DataFrames. It automates complex tasks like mapping colors to categories or drawing regression lines.
- **Key Functions:**
 - `sns.heatmap()`: Visualizes correlation matrices.
 - `sns.boxplot()`: Shows data distribution and identifies outliers.
- **Where Used:** Used for advanced statistical visualization and identifying patterns/clusters in data.

6. SciPy (Scientific Python)

- **Purpose:** Built on NumPy, it provides modules for optimization, linear algebra, integration, and statistics.
- **Installation:** pip install scipy
- **Working:** It provides highly optimized functions that are the "math engine" for complex scientific problems.
- **Key Functions:**
 - stats.mode(): Finds the most frequent value.
 - optimize.minimize(): Finds the best parameters for a function (used in training models).
 - scipy.linalg: Solves complex linear equations.
- **Where Used:** Used for signal processing, image manipulation, and calculating advanced statistical probabilities.

7. Scikit-learn (Machine Learning in Python)

- **Purpose:** The core library for classical machine learning. It features various classification, regression, and clustering algorithms.
- **Installation:** pip install scikit-learn
- **Working:** It follows a consistent "**Estimator API**." Every model is an object, and you interact with it using three standard methods: fit, predict, and score.
- **Key Functions:**
 - train_test_split(): Divides data into training and testing sets.
 - model.fit(): Learns the relationship between features and labels.
- **Where Used:** Used for building predictive models like Spam Detectors, House Price Predictors, and Customer Segmentation tools.

Summary of the ML Workflow

1. **Installation:** Run pip install numpy pandas matplotlib seaborn scipy scikit-learn.
2. **Data Loading:** Use **Pandas** to load your CSV.
3. **Data Cleaning:** Use **NumPy/Pandas** to handle missing values.
4. **Exploration:** Use **Matplotlib/Seaborn** to find trends.
5. **Math Prep:** Use **SciPy** for any complex statistical transforms.
6. **Modeling:** Use **Scikit-learn** to train and test your prediction engine.