Training Report – Day 6

Topic Covered Today:

- Installation of Python and its role in AI/ML
- Introduction to Jupyter Notebook
- Basics of Scikit-learn library

Key Learning:

Python Installation and Setup:

Today I learned how to properly install and set up **Python**, which is the primary programming language used in Artificial Intelligence and Machine Learning. Python is popular because it is simple, has a large number of libraries, and provides great community support.

I explored two ways of installing Python:

1. **Direct Python installation** from the official website (python.org).
2. **Anaconda distribution**, which comes preloaded with Python, Jupyter Notebook, and essential ML libraries.

After installation, I also learned how to check the version of Python using the command:

- `python --version` or `python3 --version`

Jupyter Notebook:

I was introduced to **Jupyter Notebook**, an interactive environment that allows writing and executing Python code in small blocks called "cells." It is widely used in data science because it allows mixing **code, output, and explanations** in one place.

Some basic Jupyter Notebook functions I learned:

- Running code cells (`Shift + Enter`)
- Adding markdown cells for explanations
- Saving notebooks with `.ipynb` extension

- Restarting kernel and clearing output

This tool makes experiments in Machine Learning easier and more organized.

Scikit-learn Basics:

I also learned the basics of **Scikit-learn (sklearn)**, one of the most powerful Python libraries for Machine Learning. Scikit-learn provides ready-made functions for:

- Data preprocessing (cleaning, splitting, scaling)
- Building ML models (classification, regression, clustering)
- Evaluating models (accuracy, precision, recall, confusion matrix)

Some important functions and modules I noted down:

- `train_test_split` → for splitting dataset into training and testing sets
- `fit()` → for training the model
- `predict()` → for making predictions
- `sklearn.linear_model` → for regression models
- `sklearn.tree` → for decision trees
- `sklearn.metrics` → for evaluating model performance

Activities / Assignments:

- Installed **Python** and verified the version.
- Installed **Anaconda** and explored Jupyter Notebook interface.
- Created a sample Jupyter Notebook and wrote simple Python programs (e.g., addition, printing, loops).
- Imported **scikit-learn** and tested a simple dataset using `train_test_split`.
- Noted down common sklearn functions and their uses for future practice.

Personal Reflection for Day 6:

Today's session gave me a clear idea of how Python, Jupyter Notebook, and Scikit-learn work together in Machine Learning projects. I realized that Python provides the language, Jupyter provides an easy interface for experiments, and Scikit-learn provides the tools for building ML models.

At first, installing packages and libraries seemed difficult, but using Anaconda made the process simpler. Working in Jupyter Notebook felt very comfortable compared to writing long scripts in other editors.

This session made me confident about setting up my development environment, which will be the foundation for building and testing AI/ML models in the upcoming days.