

# **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING TRAINING TR-102 REPORT DAY 6 30 JUNE 2025**

## **Overview:**

The sixth day of training focused on Jupyter Notebook, one of the most powerful and widely used tools in the field of Artificial Intelligence and Data Science. We learned how to create, edit, and execute code within Jupyter, explore its user interface, and understand how it helps in developing and documenting AI and Machine Learning projects. This session emphasized the importance of Jupyter Notebook for interactive coding, data visualization, and experimentation. It provided us with hands-on experience working in an environment where we could write and test Python code efficiently.

## **Learning Objectives:**

- Understand what Jupyter Notebook is and why it is used.
- Learn how to install, launch, and use Jupyter Notebook.
- Explore its interface, cells, and toolbar features.
- Practice writing, running, and saving Python code.
- Learn how to use Markdown, comments, and visualizations within a notebook.
- Understand its importance in AI and Machine Learning projects

## **Introduction to Jupyter Notebook**

Jupyter Notebook is an open-source interactive web application that allows users to create and share documents containing live code, visualizations, and text. It supports multiple programming languages, but Python is the most commonly used language in AI and Data Science projects.

Jupyter provides an interactive environment where users can:

- Write and execute code in small chunks (cells).
- Document code using Markdown text.
- Visualize data using graphs and charts.
- Combine code, output, and explanations in one file.

## Why Jupyter Notebook is Important

### 1. Interactive Coding Environment:

You can run code step-by-step and view outputs immediately.

### 2. Data Visualization:

Supports visualization libraries like Matplotlib, Seaborn, and Plotly for graphical analysis.

### 3. Documentation and Explanation:

Markdown cells allow writing text, mathematical equations, and notes alongside code.

### 4. Reproducibility:

All code, output, and explanations are stored in one `.ipynb` file, making it easy to reproduce experiments.

### 5. Integration with AI Tools:

Jupyter integrates seamlessly with libraries like NumPy, Pandas, TensorFlow, and Scikit-learn, making it ideal for Machine Learning workflows.

# Common Jupyter Notebook Commands and Shortcuts:

## 1. Basic Commands:

- **Shift + Enter** – Run the current cell and move to the next one.
- **Ctrl + Enter** – Run the current cell but stay in the same cell.
- **Alt + Enter** – Run the current cell and insert a new cell below.
- **A** – Insert a cell above.
- **B** – Insert a cell below.
- **DD** – Delete the selected cell.

## 2. Mode Switching:

- Command Mode (Blue border): Used for notebook-level commands.
- Edit Mode (Green border): Used for editing the content inside a cell.
- Press **Esc** to switch to Command Mode.
- Press **Enter** to switch to Edit Mode.

## 3. File Operations:

- Save Notebook: **Ctrl + S**
- Download Notebook: From the “File” menu → *Download as* → *Notebook (.ipynb)* or *Python (.py)*.
- Rename Notebook: Click the notebook title at the top.
- Designed to perform specific tasks within a limited context.
- Operates under predefined rules and cannot go beyond its programming.

## **Conclusion:**

Day 6 introduced us to the Jupyter Notebook environment, which plays a crucial role in AI and Machine Learning development. We learned how to execute Python code interactively, document our work, and visualize data effectively. This session demonstrated how Jupyter Notebook serves as both a development tool and a presentation platform — helping researchers, data scientists, and engineers write, test, and share their work efficiently. By mastering Jupyter, we have taken a significant step toward practical implementation of AI and ML concepts.