Day 6 Training Report

30 June 2025

Data Visualization — Matplotlib Basics (Bar, Line, Scatter) + Hands-on Visualization

On **Day 6**, the focus was on **data visualization** — an essential step in understanding and interpreting data effectively. Students were introduced to **Matplotlib**, one of the most widely used Python libraries for creating static, animated, and interactive plots. Visualization helps in identifying patterns, trends, and anomalies in data, making it a crucial tool for **Exploratory Data Analysis (EDA)**.

---

1. Introduction to Matplotlib

Matplotlib is a Python library that provides a MATLAB-like interface for plotting. It supports various types of graphs, including bar charts, line charts, scatter plots, histograms, and more.

Installation:

```
pip install matplotlib
```

Basic Import:

```
import matplotlib.pyplot as plt
```

---

2. Line Plots

Line plots are used to **show trends over time or continuous data**. They are ideal for visualizing changes in data sequentially.

Example:

```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 12, 15, 20, 25]

plt.plot(x, y, color='blue', marker='o', linestyle='-')
plt.title('Line Plot Example')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.grid(True)
plt.show()
```

☑ **Key Points:**

- plot() is used for creating line charts.
- marker, color, and linestyle help customize the look.
- Useful for showing trends, growth patterns, or time-series data.

---

3. Bar Charts

Bar charts are useful for **comparing discrete categories or groups**. Each bar represents a category, and its height represents the corresponding value.

Example:

```
categories = ['A', 'B', 'C', 'D']
values = [20, 35, 30, 25]

plt.bar(categories, values, color='green')
plt.title('Bar Chart Example')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```

☑ **Key Points:**

- Ideal for **categorical comparisons** (e.g., sales by region).
- Can be horizontal using barh().
- Colors and labels make the chart more informative.

---

4. Scatter Plots

Scatter plots are great for **visualizing relationships between two variables**. They help identify correlations, clusters, and outliers.

Example:

```
x = [5, 7, 8, 5, 6, 7, 8, 6]
y = [5, 8, 7, 6, 7, 9, 10, 6]

plt.scatter(x, y, color='red')
plt.title('Scatter Plot Example')
plt.xlabel('X Values')
plt.ylabel('Y Values')
plt.show()
```

☑ **Key Points:**

- scatter() is used for creating scatter plots.
- Helps detect **positive, negative, or no correlation** between variables.

- Useful in exploratory data analysis to understand data distribution.

---

5. Hands-on Visualization Exercises

To strengthen learning, students performed practical exercises such as:

- Visualizing sales data using bar and line plots.
- Creating scatter plots to analyze the relationship between features like **Age vs. Salary**, **Hours Studied vs. Marks**, etc.
- Customizing plots by adding titles, labels, legends, and grids to make them more readable.

Sample Customization Example:

```
plt.figure(figsize=(8,5))
plt.bar(categories, values, color='orange')
plt.title('Customized Bar Chart')
plt.xlabel('Category')
plt.ylabel('Value')
plt.grid(axis='y')
plt.show()
```

Conclusion

By the end of **Day 6**, students gained practical knowledge of **creating and interpreting line, bar, and scatter plots using Matplotlib**. They understood how to represent different types of data visually, identify trends, and communicate insights effectively. These visualization skills will be essential for upcoming modules like **feature analysis and machine learning model evaluation**.