Week 1: Introduction to Machine Learning

Task 1.1: Exploring Machine Learning with Python

Problem Statement:

The goal is to introduce fundamental machine learning concepts using Python, focusing on data handling, visualization, and basic statistical analysis using the Iris dataset.

Solution Overview:

1. **Environment Setup:**
    - Ensure Jupyter Notebook and Python are installed on your system.
    - Use pip to install necessary libraries (numpy, pandas, matplotlib, seaborn).

      ```
      !pip install pandas numpy matplotlib seaborn
      ```

2. **Data Loading and Exploration:**
    - Load the Iris dataset (Iris.csv) into a Pandas DataFrame.

      ```
      import pandas as pd
      # Load dataset
      data = pd.read_csv("Iris.csv")
      ```

    - **Data Overview:**
        - Check the dimensions (rows, columns) of the dataset.

          ```
          data.shape
          ```

        - Display the first few rows of the dataset to ensure correct loading.

          ```
          data.head()
          ```

        - Obtain summary statistics for numerical columns.

          ```
          data.describe()
          ```

        - Check data types and missing values.

          ```
          data.info()
          ```

        - Identify unique classes in the target variable (Species).

          ```
          data['Species'].unique()
          ```

3. **Data Visualization:**

  - **Scatter Plots:**
    - Visualize relationships between Sepal Length vs Petal Length and Sepal Width vs Petal Width.

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.scatter(data['SepalLengthCm'], data['PetalLengthCm'],
color='b', label='Sepal Length vs Petal Length')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Petal Length (cm)')
plt.title('Sepal Length vs Petal Length')
plt.legend()

plt.subplot(1, 2, 2)
plt.scatter(data['SepalWidthCm'], data['PetalWidthCm'],
color='r', label='Sepal Width vs Petal Width')
plt.xlabel('Sepal Width (cm)')
plt.ylabel('Petal Width (cm)')
plt.title('Sepal Width vs Petal Width')
plt.legend()

plt.tight_layout()
plt.show()
```

  - **Histograms:**
    - Visualize distributions of Sepal Length and Petal Length.

```
plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.hist(data['SepalLengthCm'], bins=10, color='blue',
edgecolor='black')
```

```
plt.xlabel('Sepal Length (cm)')

plt.ylabel('Frequency')

plt.title('Histogram of Sepal Length')


plt.subplot(1, 2, 2)

plt.hist(data['PetalLengthCm'], bins=10, color='red',
edgecolor='black')

plt.xlabel('Petal Length (cm)')

plt.ylabel('Frequency')

plt.title('Histogram of Petal Length')


plt.tight_layout()

plt.show()
```

Challenges and Resolutions:

- Initial setup of Python environment and library installations ere the main challenge because libraries would not install.
    - **Resolution:** Used virtual environments to manage dependencies and ensured all libraries were correctly installed using pip.
- Data cleaning and handling missing values as missing values can cause a disaster.
    - **Resolution:** Implemented data inspection techniques (data.info()) to identify missing values and handled them appropriately, ensuring data integrity.
- Plotting complex visualizations like histograms and scatter plots were quite confusing.
    - **Resolution:** Referred to documentation and online resources for syntax and best practices in matplotlib and seaborn libraries, improving plot clarity and aesthetics.