ITCP-25/AI-003 (Noor Fatima)

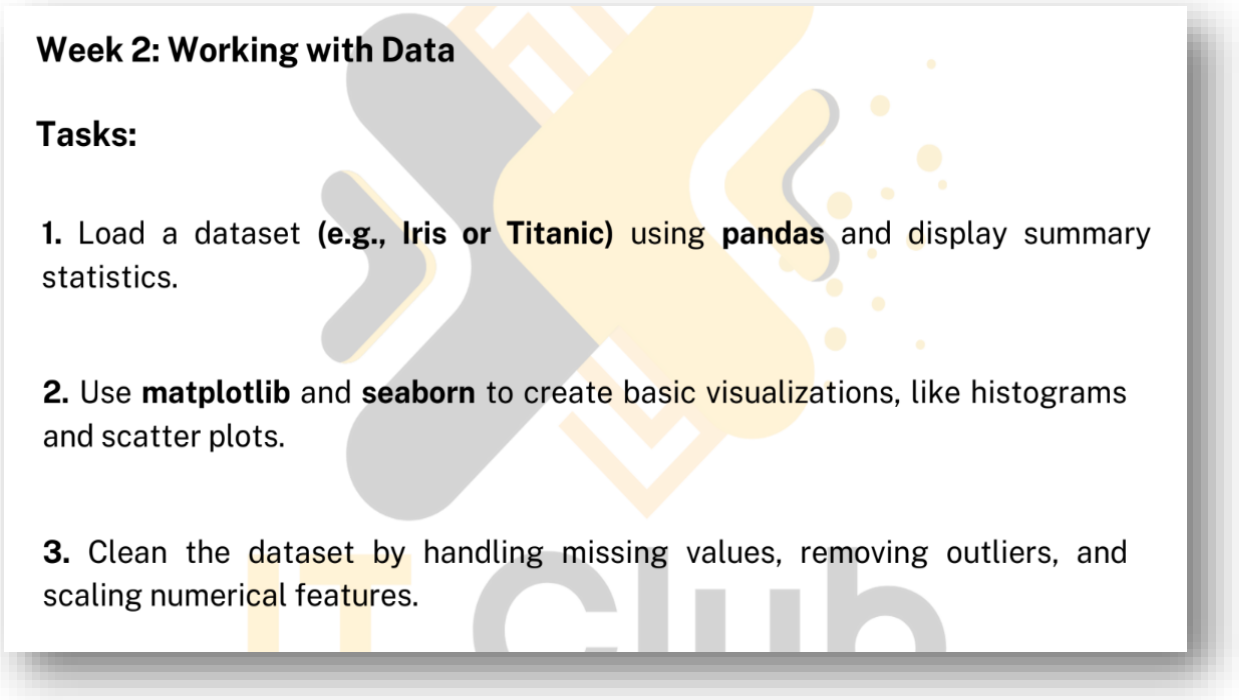
Week 2: Working with Data

Tasks:

1. Load a dataset (e.g., Iris or Titanic) using pandas and display summary statistics.

2. Use matplotlib and seaborn to create basic visualizations, like histograms and scatter plots.

3. Clean the dataset by handling missing values, removing outliers, and scaling numerical features.



Task 1:

Load a dataset (e.g., Iris or Titanic) using pandas and display summary statistics.

Loading a Dataset and Displaying Summary Statistics

1. Introduction: I was assigned the task of loading a dataset (such as **Iris** or **Titanic**) using **pandas** and displaying its summary statistics. This document outlines the step-by-step process followed to complete this task successfully.

2. Loading a Dataset: To load the dataset, the following steps were followed:

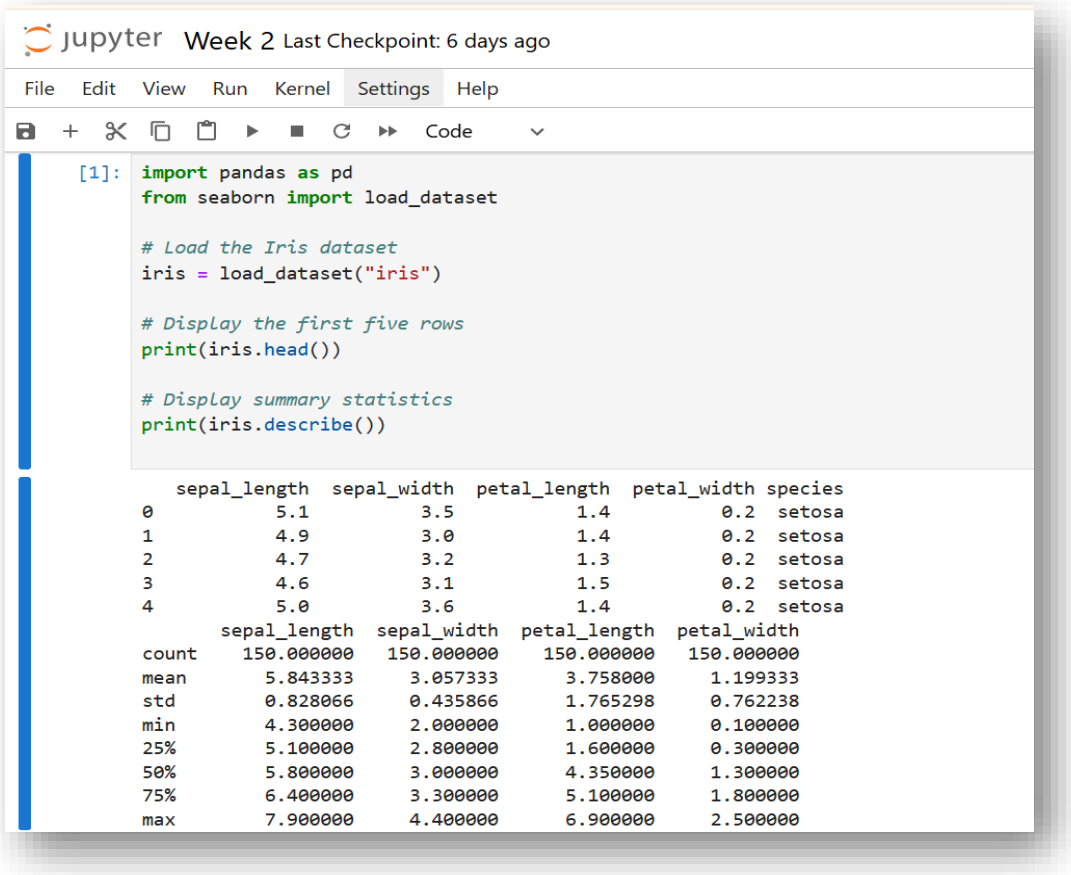
1. **Imported the necessary libraries.**

2. **Loaded the dataset** using Seaborn's built-in dataset repository.
3. **Displayed the first five rows** using the .head() function.

3. Displaying Summary Statistics : To analyze the dataset, summary statistics were computed using the .describe() function.

1. **For numerical columns**, .describe() provides key statistics such as:
    - Count
    - Mean
    - Standard deviation
    - Min, Max, and Quartiles
2. **For categorical columns**, .describe(include='all') displays:
    - Count
    - Unique values
    - Most frequent value (mode)

Code and Output :



```
import pandas as pd

from seaborn import load_dataset

# Load the Iris dataset

iris = load_dataset("iris")

# Display the first five rows

print(iris.head())

# Display summary statistics

print(iris.describe())
```

6. Conclusion

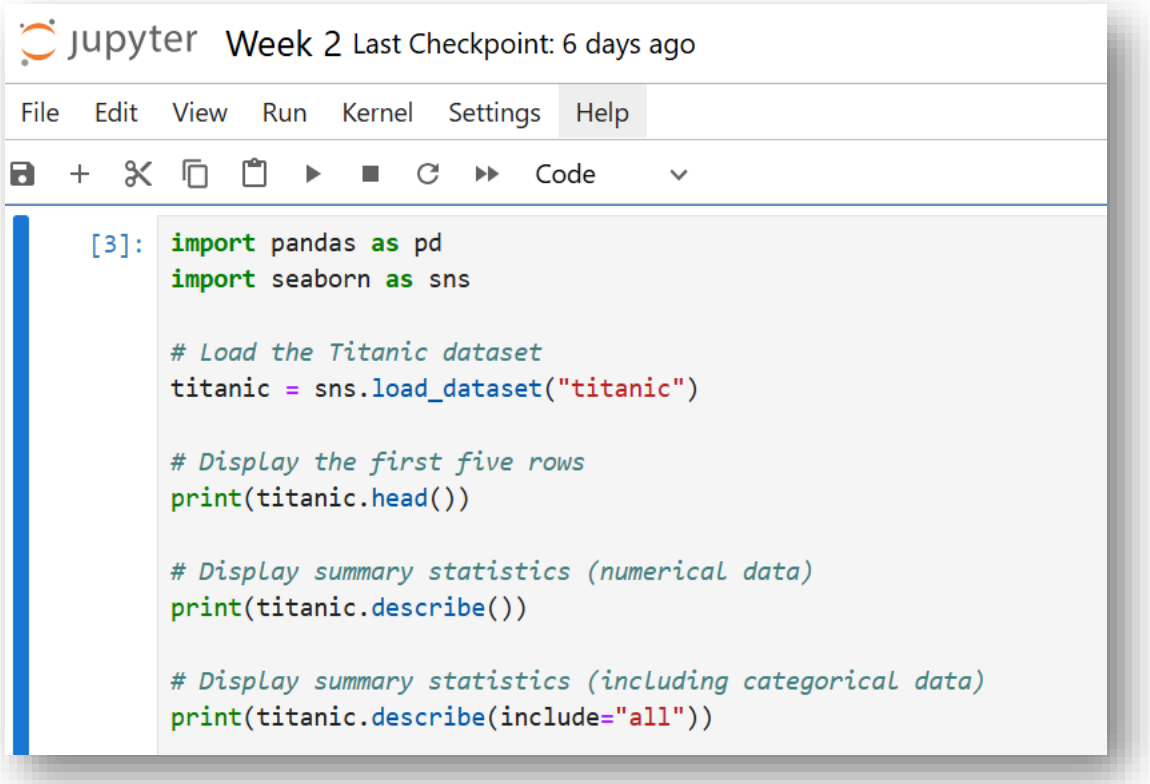
- Successfully loaded **Iris** and **Titanic** datasets using **Seaborn**.
- Displayed the **first five rows** to understand the dataset structure.
- Used **pandas** to compute **summary statistics** for numerical and categorical columns.
- These steps help in **exploratory data analysis (EDA)** before applying machine learning or visualization techniques.

TASK 2:

Use matplotlib and seaborn to create basic visualizations, like histograms and scatter plots.

Code:

```
import pandas as pd

import seaborn as sns

# Load the Titanic dataset

titanic = sns.load_dataset("titanic")

# Display the first five rows

print(titanic.head())

# Display summary statistics (numerical data)

print(titanic.describe())

# Display summary statistics (including categorical data)

print(titanic.describe(include="all"))
```



A **histogram** is used to visualize the distribution of a numerical variable. The following steps were followed:

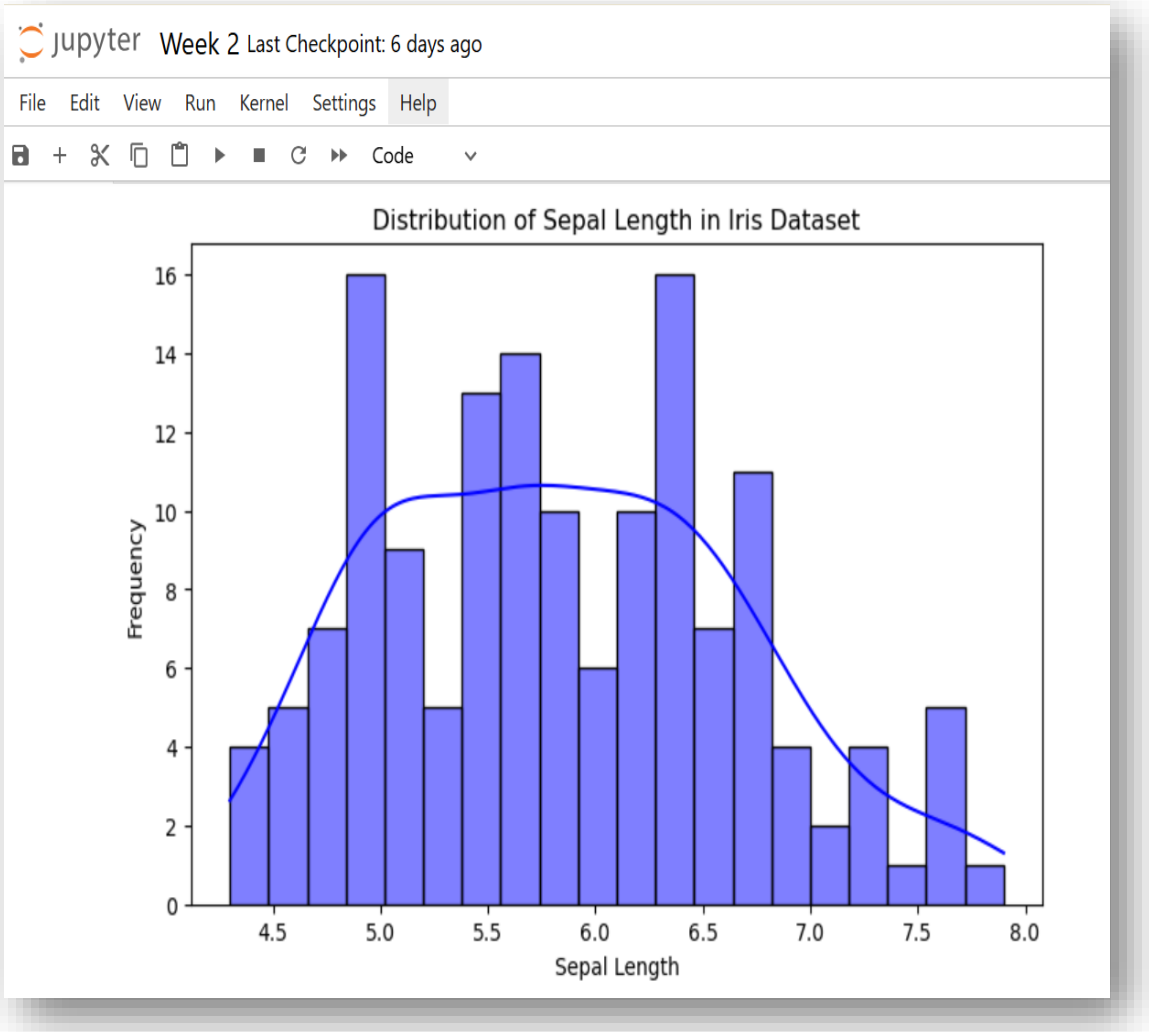
1. **Loaded the dataset** using Seaborn.
2. **Created a histogram** using sns.histplot().
3. **Added labels and titles** to improve readability.
4. **Displayed the plot** using plt.show().

Code:

```
import matplotlib.pyplot as plt

import seaborn as sns

from seaborn import load_dataset

# Load the Iris dataset

iris = load_dataset("iris")

# Create a histogram for Sepal Length

plt.figure(figsize=(8, 5))

sns.histplot(iris['sepal_length'], bins=20, kde=True, color='blue')
```

```
# Labels and title
plt.xlabel("Sepal Length")
plt.ylabel("Frequency")
plt.title("Distribution of Sepal Length in Iris Dataset")
# Show the plot
plt.show()
```

Output:



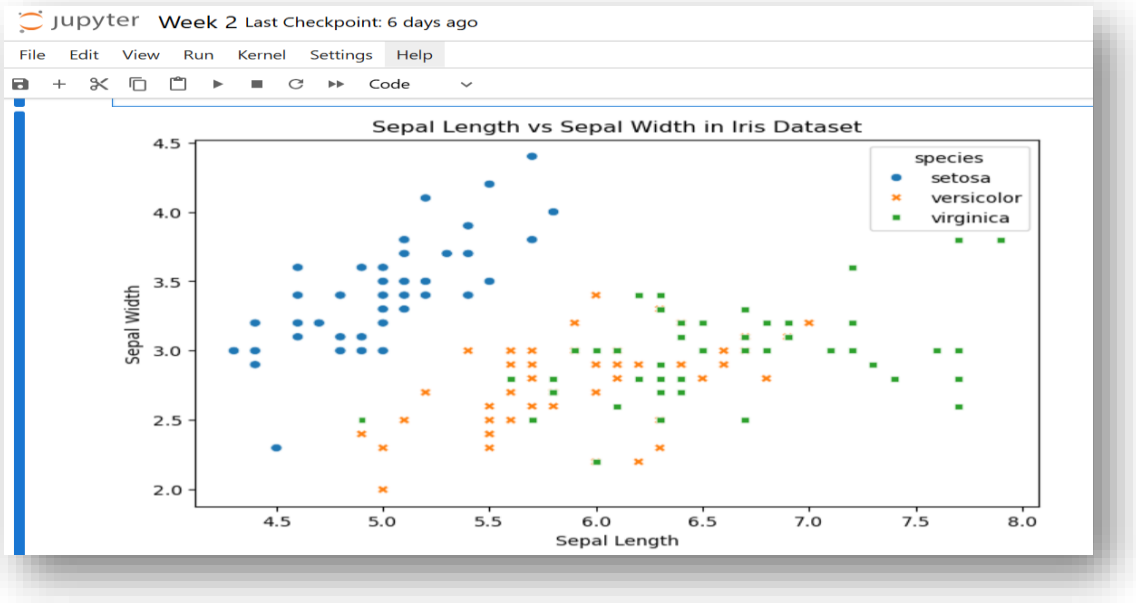
A **scatter plot** is used to visualize the relationship between two numerical variables. The following steps were followed:

1. **Loaded the dataset** using Seaborn.
2. **Created a scatter plot** using sns.scatterplot().
3. **Used color (hue) to differentiate categories** (e.g., species).
4. **Added labels and titles** to improve readability.
5. **Displayed the plot** using plt.show().

Code:

```
# Scatter plot for Sepal Length vs Sepal Width

plt.figure(figsize=(8, 5))

sns.scatterplot(data=iris, x="sepal_length", y="sepal_width", hue="species", style="species")

# Labels and title

plt.xlabel("Sepal Length")

plt.ylabel("Sepal Width")

plt.title("Sepal Length vs Sepal Width in Iris Dataset")

# Show the plot

plt.show()
```

Output:



TASK 3:

Clean the dataset by handling missing values, removing outliers, and scaling numerical features.

I was assigned the task of cleaning a dataset by performing three key data preprocessing steps:

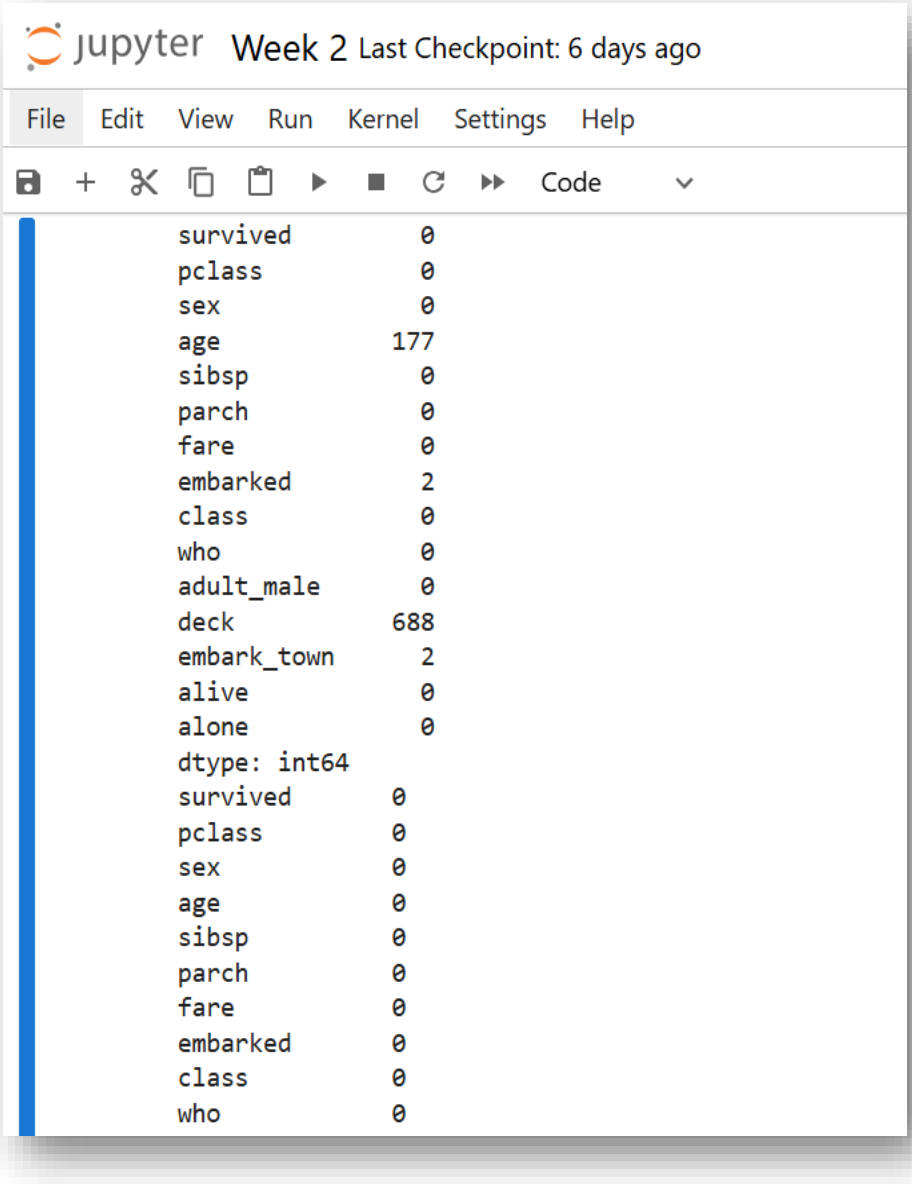
1. **Handling missing values** – Filling or removing missing data.
2. **Removing outliers** – Identifying and eliminating extreme values that may distort analysis.
3. **Scaling numerical features** – Normalizing or standardizing numerical variables for better model performance.

This document outlines the step-by-step process followed to complete this task successfully using the **pandas**, **seaborn**, and **scikit-learn** libraries.

Code:

```
import pandas as pd

import seaborn as sns

# Load Titanic dataset

titanic = sns.load_dataset("titanic")

# Check missing values

print(titanic.isnull().sum())

# Fill missing age values with median

titanic["age"].fillna(titanic["age"].median(), inplace=True)

# Fill missing embarked values with mode (most frequent value)

titanic["embarked"].fillna(titanic["embarked"].mode()[0], inplace=True)

# Drop the 'deck' column because it has too many missing values

titanic.drop(columns=["deck"], inplace=True)

# Verify missing values are handled

print(titanic.isnull().sum())
```

Output:


jupyter Week 2 Last Checkpoint: 6 days ago
File Edit View Run Kernel Settings Help
Code
survived 0
pclass 0
sex 0
age 177
sibsp 0
parch 0
fare 0
embarked 2
class 0
who 0
adult_male 0
deck 688
embark_town 2
alive 0
alone 0
dtype: int64
survived 0
pclass 0
sex 0
age 0
sibsp 0
parch 0
fare 0
embarked 0
class 0
who 0