import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_iris

import pandas as pd

* **import matplotlib.pyplot as plt**: This imports the pyplot module from the matplotlib library, which is used for creating visualizations in Python. It is commonly aliased as plt.
* **import seaborn as sns**: This imports the Seaborn library, a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics, and is commonly aliased as sns.
* **from sklearn.datasets import load\_iris**: This imports the load\_iris() function from the sklearn.datasets module. The function loads the Iris dataset, which is a well-known dataset for classification tasks that contains measurements of iris flower species.
* **import pandas as pd**: This imports the Pandas library, which is used for data manipulation and analysis, particularly for handling tabular data with DataFrames. It is commonly aliased as pd.

iris = load\_iris()

This line calls the load\_iris() function, which loads the Iris dataset into the variable iris. The dataset consists of 150 samples from three species of iris flowers (Setosa, Versicolor, and Virginica), with four features: sepal length, sepal width, petal length, and petal width.

I

ris\_df = pd.DataFrame(data=iris.data, columns=iris.feature\_names)

iris\_df['target'] = iris.target

* **iris\_df = pd.DataFrame(data=iris.data, columns=iris.feature\_names)**: Here, a Pandas DataFrame named iris\_df is created containing the features of the dataset (iris.data) with the column names taken from the feature names (iris.feature\_names).
* **iris\_df['target'] = iris.target**: This adds a new column called 'target' to the DataFrame, which contains the species labels (targets) corresponding to each sample in the dataset, represented numerically (0, 1, 2 for the three species).

plt.figure(figsize=(8, 6))

plt.scatter(iris\_df['sepal length (cm)'], iris\_df['sepal width (cm)'],

c=iris\_df['target'], cmap='viridis', s=80, alpha=0.7)

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

plt.ylabel('Sepal Width (cm)')

plt.title('Scatter Plot of Sepal Length vs Sepal Width')

plt.colorbar(label='Species')

plt.show()

* **plt.figure(figsize=(8, 6))**: This initializes a new figure for the plot with specified dimensions (8 inches wide and 6 inches tall).
* **plt.scatter(...)**: This generates a scatter plot:
  + iris\_df['sepal length (cm)']: The x-axis data (sepal length).
  + iris\_df['sepal width (cm)']: The y-axis data (sepal width).
  + c=iris\_df['target']: The color of the points is based on the species labels (target).
  + cmap='viridis': The color map used for differentiating the species.
  + s=80: The size of the scatter points.
  + alpha=0.7: The transparency level of the points.
* **plt.xlabel('Sepal Length (cm)')**: This sets the label for the x-axis.
* **plt.ylabel('Sepal Width (cm)')**: This sets the label for the y-axis.
* **plt.title('Scatter Plot of Sepal Length vs Sepal Width')**: This sets the title of the scatter plot.
* **plt.colorbar(label='Species')**: This adds a color bar to the side of the plot to provide a reference for the species represented by different colors.
* **plt.show()**: This displays the plot.

plt.figure(figsize=(8, 6))

sns.countplot(x='target', data=iris\_df, palette='viridis')

plt.xlabel('Species')

plt.ylabel('Count')

plt.title('Bar Chart: Count of Each Species')

plt.show()

* **plt.figure(figsize=(8, 6))**: Similar to before, this initializes a new figure for the bar chart.
* **sns.countplot(x='target', data=iris\_df, palette='viridis')**: This generates a count plot (bar chart) using Seaborn:
  + x='target': Indicates that the x-axis will represent the target column (species).
  + data=iris\_df: The source DataFrame for the data.
  + palette='viridis': Specifies the color palette for the bars.
* **plt.xlabel('Species')**: This sets the label for the x-axis (in this case, the species).
* **plt.ylabel('Count')**: This labels the y-axis, indicating that it shows the count of samples for each species.
* **plt.title('Bar Chart: Count of Each Species')**: This sets the title of the bar chart.
* **plt.show()**: This displays the bar chart.