from sklearn.datasets import load\_iris

This line imports the load\_iris function from the sklearn.datasets module. The Iris dataset is a classic dataset in machine learning that consists of features of iris flowers for classification tasks.

from sklearn.tree import DecisionTreeClassifier, plot\_tree

Here, we are importing the DecisionTreeClassifier class and the plot\_tree function from the sklearn.tree module. The DecisionTreeClassifier will be used to create a decision tree model, and plot\_tree is a utility that will allow us to visualize that model.

import matplotlib.pyplot as plt

This line imports the pyplot module from the matplotlib library and gives it the alias plt. pyplot is commonly used for creating various types of plots and visualizations in Python.

# Load the Iris dataset

iris = load\_iris()

Here, we call the load\_iris function to load the Iris dataset into the variable iris. This dataset is stored in a structured way, often as a simple object with attributes.

X = iris.data

This line assigns the feature data (the measurements of the iris flowers) from the iris object to the variable X. iris.data is usually a 2D array where each row represents a sample, and each column represents a feature.

y = iris.target

The target variable y, which contains the labels corresponding to the iris flower species (the classification), is assigned from iris.target. This will typically be a 1D array of integers, each representing the class of the corresponding sample in X.

class\_names = [str(name) for name in iris.target\_names]

In this line, we create a list class\_names that contains the names of the iris flower species (e.g., "setosa", "versicolor", "virginica"). iris.target\_names provides the names, and we convert them to strings, though they are likely already in string format.

# Initialize the Decision Tree Classifier

decision\_tree = DecisionTreeClassifier()

Here, we create an instance of the DecisionTreeClassifier and assign it to the variable decision\_tree. This initializes a decision tree model, which we will train with the dataset.

# Train the classifier on the entire dataset

decision\_tree.fit(X, y)

This line trains (fits) our decision tree model on the entire Iris dataset using the features X and the targets y. After this operation, the decision\_tree object will hold the learned decision tree structure based on the provided data.

# Visualize the Decision Tree

plt.figure(figsize=(12, 8))

Before plotting the decision tree, we create a new figure for the plot with a specified size (12 inches wide and 8 inches tall).

plot\_tree(decision\_tree, feature\_names=iris.feature\_names,

class\_names=class\_names, filled=True, rounded=True)

In this line, we use the plot\_tree function to visualize the structure of the trained decision tree. We specify several parameters:

* decision\_tree: The trained model we want to visualize.
* feature\_names: The names of the features (attributes) of the dataset, used in node labels.
* class\_names: The names of the target classes, used for labeling the leaf nodes.
* filled=True: Colors each node according to the majority class found in that node.
* rounded=True: Rounds the corners of the boxes in the plot for better aesthetics.

plt.title("Decision Tree Visualization")

We set the title for the plot to "Decision Tree Visualization" for clarity when the visualization is displayed.

plt.show()

Finally, we call plt.show() to display the figure with the decision tree plot on the screen.