This program demonstrates “**supervised machine learning**” using the “**Decision Tree Classifier**” to classify iris flowers based on their physical characteristics. The goal is to train the model to recognize patterns in the data and then use that knowledge to predict the species of new flowers.

***Step-by-Step Breakdown:***

**1. Loading the Iris Dataset**

The “Iris dataset” is a well-known dataset in machine learning. It contains 150 records of iris flowers, categorized into three species:

Setosa

Versicolor

Virginica

Each flower in the dataset has four numerical features (measurements in centimeters):

Sepal length

Sepal width

Petal length

Petal width

**2. Splitting the Dataset**

The dataset is divided into “training data (80%)” and “testing data (20%)” using train\_test\_split().

The training data is used to “train” the model, and the test data is used to “evaluate” its performance.

**3. Training the Decision Tree Model**

The DecisionTreeClassifier is used to create the model.

The classifier learns from the training data by identifying patterns that separate the three iris species.

**4. Making Predictions**

The trained model makes predictions on:

Test data (to evaluate performance).

A new flower with specific measurements ([5.1, 3.5, 1.4, 0.2]).

The predicted species name is retrieved from the dataset’s target names.

**5. Evaluating Accuracy**

accuracy\_score() calculates how many predictions were correct.

The accuracy score is displayed (e.g., 1.00 means 100% accuracy).

**6. Visualizing Predictions with a Bar Chart**

The program counts the actual vs. predicted species in the test data.

A bar chart is plotted to compare them, helping to visually analyze how well the model performed.

***Additional Insights:***

**What is a Decision Tree?**

A Decision Tree is a flowchart-like structure that splits data into smaller groups based on conditions.

At each step, it asks yes/no questions about a feature (e.g., "Is petal length > 2.5 cm?") and makes a decision.

**Why Use a Decision Tree?**

Simple & Easy to Understand– Works like a series of if-else statements.

Fast Training & Prediction – Ideal for small datasets like Iris.

Interpretable – The logic behind predictions can be visualized using plot\_tree().

**Possible Enhancements**

Visualizing the Decision Tree

**python**

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

plot\_tree(iris\_classifier, feature\_names=iris\_data.feature\_names, class\_names=iris\_data.target\_names, filled=True)

plt.show()