.

**🌳 Decision Tree Classifier on the Iris Dataset**

**📘 Overview**

This project demonstrates the implementation of a **Decision Tree Classifier** using the classic **Iris dataset**. The goal is to classify iris flowers into three species based on four measurable features: sepal length, sepal width, petal length, and petal width.

**📂 Dataset Description**

* **Source**: sklearn.datasets.load\_iris()
* **Samples**: 150
* **Classes**:
  + 0: Setosa
  + 1: Versicolor
  + 2: Virginica
* **Features**:
  + sepal length (cm)
  + sepal width (cm)
  + petal length (cm)
  + petal width (cm)

**🧪 Project Objectives**

* Train a **Decision Tree Classifier** on the Iris dataset.
* Evaluate the model using classification metrics.
* Visualize both the **confusion matrix** and the **decision tree structure**.

**⚙️ Tools and Technologies**

* **Python**
* **Scikit-learn** (for ML models and dataset)
* **Matplotlib / Seaborn** (for visualization)
* **Pandas** (for data handling)

**🧾 Code Breakdown**

**1. Data Loading and Preparation**

python

CopyEdit

from sklearn.datasets import load\_iris

import pandas as pd

data = load\_iris()

X = pd.DataFrame(data.data, columns=data.feature\_names)

y = data.target

**2. Train-Test Split**

python

CopyEdit

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.2, random\_state=42)

**3. Training the Model**

python

CopyEdit

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(criterion='gini', max\_depth=3, random\_state=42)

model.fit(X\_train, y\_train)

**4. Evaluation**

python

CopyEdit

from sklearn.metrics import classification\_report, confusion\_matrix

y\_pred = model.predict(X\_test)

print(classification\_report(y\_test, y\_pred, target\_names=data.target\_names))

**5. Confusion Matrix Visualization**

python

CopyEdit

import seaborn as sns

import matplotlib.pyplot as plt

conf\_mat = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(conf\_mat, annot=True, cmap='Greens',

xticklabels=data.target\_names,

yticklabels=data.target\_names)

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix - Iris Classification")

plt.show()

**6. Decision Tree Plot**

python

CopyEdit

from sklearn.tree import plot\_tree

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

plot\_tree(model, feature\_names=data.feature\_names,

class\_names=data.target\_names, filled=True, rounded=True)

plt.title("Decision Tree Visualization")

plt.show()

**📊 Results**

* **Model Accuracy**: High accuracy on test set (typically >90%)
* **Key Features**: Petal length and petal width are dominant decision points.
* **Interpretability**: The tree visualization helps explain model decisions in an intuitive way.

**📌 Conclusion**

This project highlights how a **Decision Tree Classifier** can be effectively used for **multi-class classification** problems. The Iris dataset serves as an ideal starting point due to its balanced size, clean features, and clearly defined classes. The decision tree structure makes the model easily interpretable — an important quality in many real-world applications.