Decision Tree—Assignment

Theoretical

1. What is a Decision Tree, and how does it work

ANS-

A Decision Tree is a supervised machine learning algorithm used for classification and regression tasks. It mimics human decision-making by splitting data into branches based on feature values, forming a tree-like structure.

🔹 Root Node – The first decision point (top of the tree).  
🔹 Internal Nodes – Decision points that split into branches.  
🔹 Leaf Nodes – The final outcome or prediction (no further splits).

How Does a Decision Tree Work?

The algorithm works by recursively splitting the dataset into subsets based on feature values to maximize data separation.

✅ Step-by-Step Process:

1️Select the Best Feature to Split

* Choose a feature that provides the most information gain (for classification) or minimizes variance (for regression).
* Common splitting criteria:
  + Gini Impurity (CART Algorithm)
  + Entropy & Information Gain (ID3 Algorithm)
  + Mean Squared Error (MSE) (for regression)

2️Split the Data

* Divide the dataset into subsets based on the selected feature.
* Repeat this step for each branch until a stopping condition is met (e.g., max depth, min samples per leaf).

3️ Assign Class Labels or Predictions

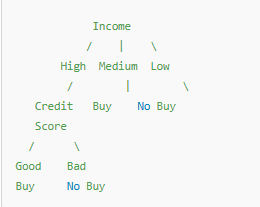
* Each leaf node represents the final output.
* For classification: The majority class in the node is assigned as the output.
* For regression: The average value in the node is used as the output.

Example of a Decision Tree (Classification)

Imagine a Decision Tree that predicts whether a person will buy a car based on:

* Income (High, Medium, Low)
* Credit Score (Good, Bad)

Tree Structure:

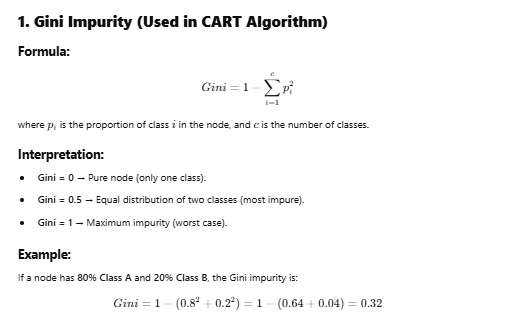


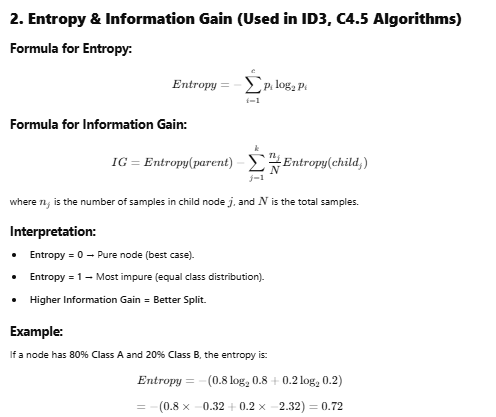
1. What are impurity measures in Decision Trees

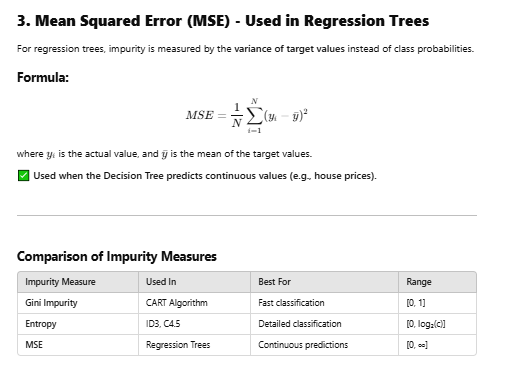
ANS-

### Impurity Measures in Decision Trees

Impurity measures help Decision Trees determine **how "pure"** a node is when splitting the dataset. A pure node contains only **one class**, while an impure node contains a mix of multiple classes.





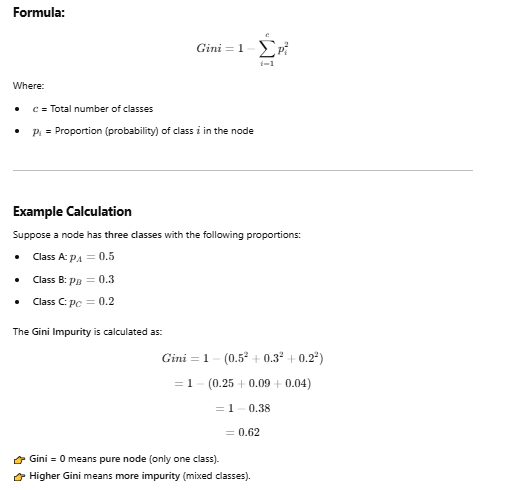


1. What is the mathematical formula for Gini Impurity

ANS-

### Mathematical Formula for Gini Impurity

The **Gini Impurity** measures how often a randomly chosen element from a set would be incorrectly classified if randomly labeled according to the class distribution in the set.

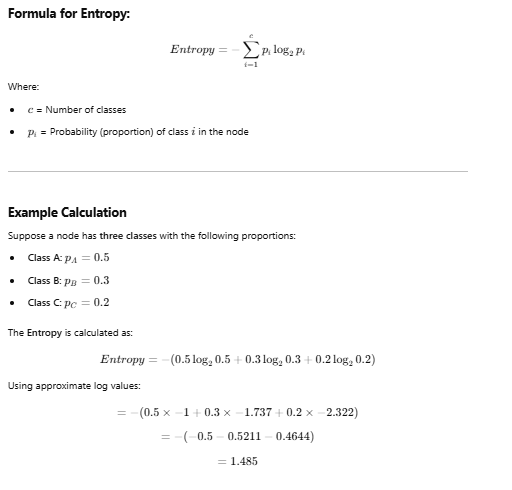


1. What is the mathematical formula for Entropy

ANS-

### Mathematical Formula for Entropy

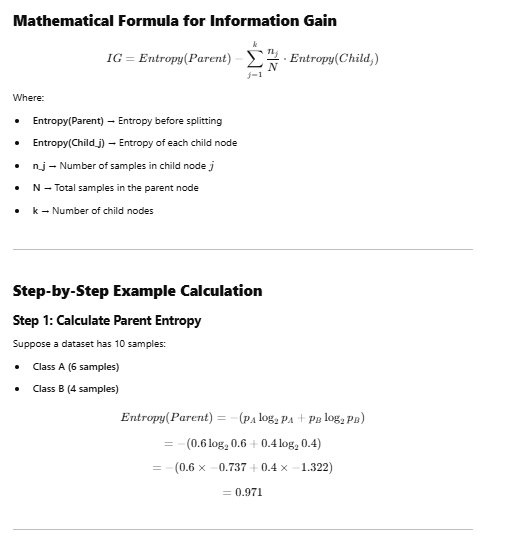
Entropy is a measure of impurity or randomness in a dataset. It is used in **Decision Trees** (ID3, C4.5) to determine the best feature to split the data.

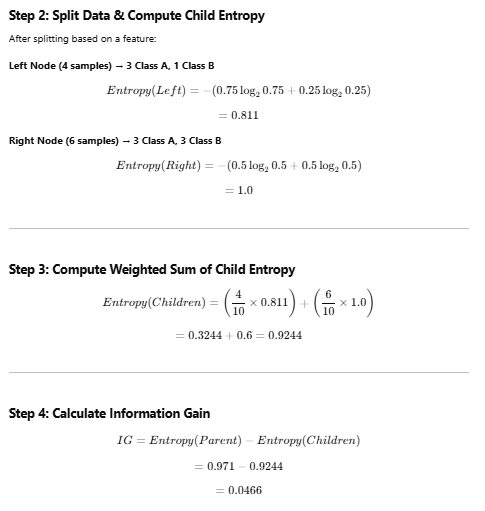


1. What is Information Gain, and how is it used in Decision Trees

ANS-

Information Gain (IG) measures how much a feature reduces entropy (impurity) in a dataset. It helps Decision Trees select the best feature to split the data. The higher the Information Gain, the better the feature for splitting.



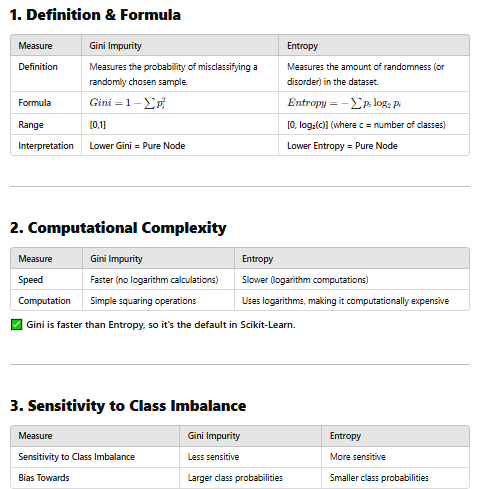


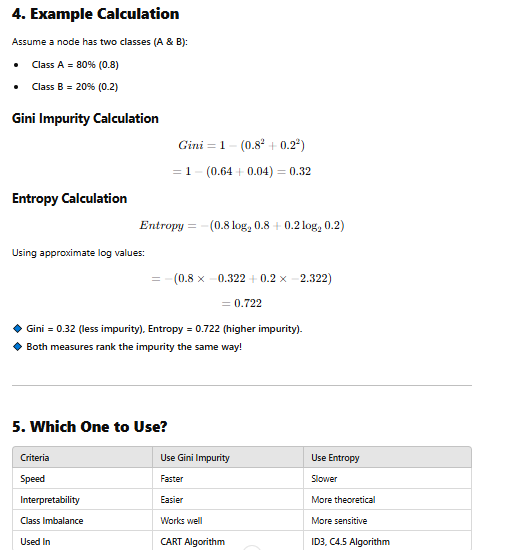
## **How is Information Gain Used in Decision Trees?**

**1️Compute Entropy for Parent Node  
2️Split the data based on a feature  
3️Compute Entropy for Child Nodes  
4️Calculate Weighted Average Entropy of Children  
5️Compute Information Gain  
6️ Choose the feature with the Highest IG for splitting  
7️Repeat until stopping criteria are met (pure nodes or max depth reached).**

1. What is the difference between Gini Impurity and Entropy

ANS-

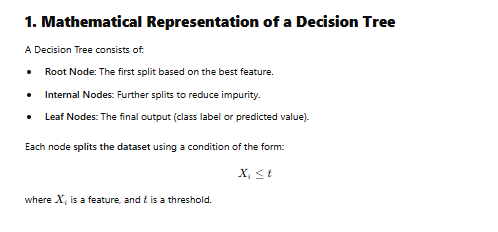


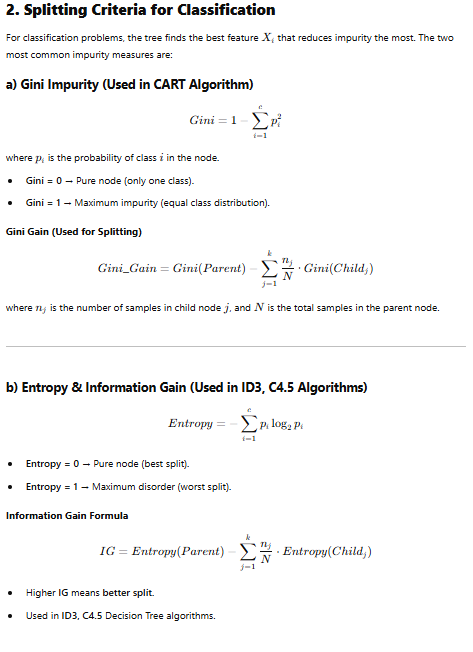


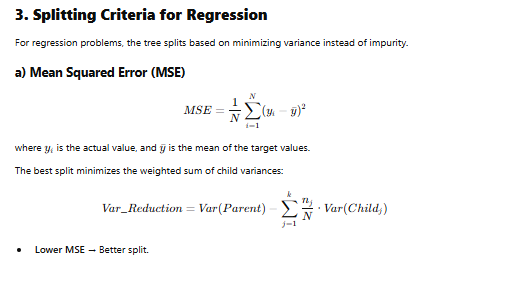
1. What is the mathematical explanation behind Decision Trees

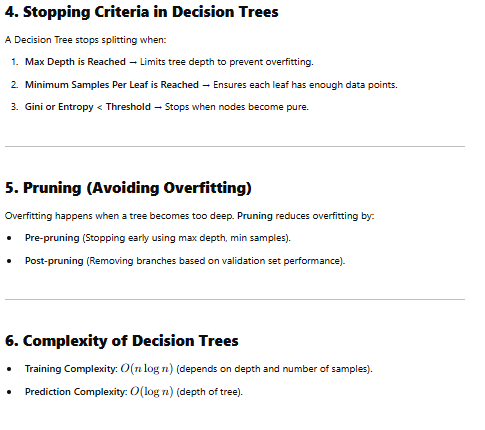
ANS-

Decision Trees are a **supervised learning algorithm** used for **classification and regression**. They split data into subsets using mathematical criteria to minimize impurity (for classification) or variance (for regression).









1. What is Pre-Pruning in Decision Trees

ANS-

### What is Pre-Pruning?

Pre-pruning (also called early stopping) is a technique used to prevent overfitting by stopping tree growth early before it becomes too complex. It sets constraints on tree expansion during training, rather than cutting branches after the tree is fully grown.

## Why is Pre-Pruning Needed?

A Decision Tree without pruning grows until:  
✔ Every leaf node is pure (only one class) OR  
✔ Each sample is in its own leaf node

This often leads to overfitting, where the tree memorizes training data but performs poorly on unseen data.

Pre-pruning prevents this by stopping the tree early.

## Pre-Pruning Techniques (Stopping Criteria)

### 1️ Maximum Depth (max\_depth)

Limits the number of levels in the tree.

✔ Shallow trees → Less overfitting, faster training  
❌ Too shallow → May lead to underfitting

🔹 Example: In Scikit-Learn, set max\_depth=5 to restrict the tree to 5 levels.

### 2 Minimum Samples Per Split (min\_samples\_split)

A node must have at least X samples to split further.

✔ Larger values → Prevents overfitting, reduces splits  
❌ Too large → May lead to underfitting

🔹 Example: min\_samples\_split=10 ensures a node must have at least 10 samples to be split.

### 3 Minimum Samples Per Leaf (min\_samples\_leaf)

A leaf node must contain at least X samples.

✔ Prevents nodes with very few samples  
✔ Helps with imbalanced data

🔹 Example: min\_samples\_leaf=5 ensures each leaf has at least 5 samples.

### 4 Maximum Number of Nodes (max\_leaf\_nodes)

Limits the total number of leaf nodes.

✔ Controls complexity  
❌ May underfit if too restrictive

🔹 Example: max\_leaf\_nodes=10 limits the tree to 10 leaf nodes.

### 5 Maximum Impurity Decrease (min\_impurity\_decrease)

Splitting only happens if impurity decrease exceeds a threshold.

✔ Prevents unnecessary splits  
✔ Used with Gini Impurity or Entropy

🔹 Example: min\_impurity\_decrease=0.01 ensures splits only happen if impurity drops by at least 0.01.

1. What is Post-Pruning in Decision Trees

ANS –

## **What is Post-Pruning?**

Post-pruning (also called cost-complexity pruning) is a technique used to prevent overfitting by removing unnecessary branches after the Decision Tree is fully grown.

## Why is Post-Pruning Needed?

A Decision Tree **without pruning** grows until:  
✔ Each leaf node is **pure** (only one class)  
✔ Each sample has **its own leaf node**

This often leads to **overfitting**, where the tree memorizes training data but **performs poorly on unseen data**.

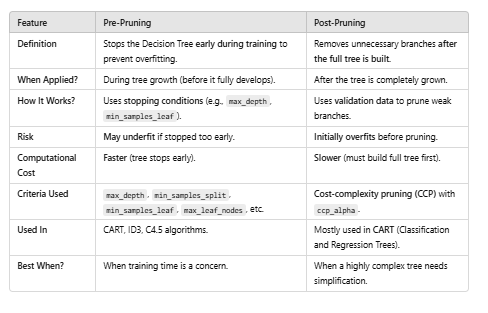
**Post-pruning removes weak branches after training to improve generalization.**

## **Steps in Post-Pruning**

1️ **Train the Decision Tree to its full depth**  
2️ **Evaluate performance on a validation set**  
3️ **Prune branches that do not improve validation accuracy**  
4️ **Repeat until no further improvement is seen**

1. What is the difference between Pre-Pruning and Post-Pruning

ANS-



1. What is a Decision Tree Regressor

ANS-

A Decision Tree Regressor is a type of supervised learning algorithm used for regression tasks. Unlike a classification tree, which predicts discrete labels, a Decision Tree Regressor predicts continuous numerical values.

## How Does a Decision Tree Regressor Work?

1️ Splitting the Data

* The algorithm splits the dataset recursively into different regions based on the feature that minimizes prediction error.
* It uses Mean Squared Error (MSE), Mean Absolute Error (MAE), or Variance Reduction as the splitting criterion.

2️ Stopping Criteria

* The tree stops growing when a stopping condition is met (e.g., max\_depth, min\_samples\_leaf).

3️ Prediction

* For a given test data point, the model traverses the tree and predicts the average value of samples in the corresponding leaf node.

1. What are the advantages and disadvantages of Decision Trees

ANS –

## **Advantages of Decision Trees**

### 1️ Easy to Understand & Interpret

* Decision trees are highly interpretable and easy to visualize.
* Even non-technical users can understand the decision-making process.

### 2️ Handles Both Classification & Regression

* Works well for both classification (DecisionTreeClassifier) and regression (DecisionTreeRegressor) problems.

### 3️ No Need for Feature Scaling

* Unlike SVM or KNN, Decision Trees do not require normalization or standardization of data.

### 4️ Handles Non-Linear Data

* Can capture non-linear relationships between features without needing transformations.

### 5️ Works with Missing Values

* Can handle missing values by assigning the most frequent class or mean value.

### 6️ Feature Selection is Automatic

* Decision trees automatically select important features during the training process.

## **Disadvantages of Decision Trees**

### 1️ Prone to Overfitting

* If the tree is too deep, it memorizes training data instead of generalizing.
* Solution → Use pre-pruning (max\_depth, min\_samples\_split) or post-pruning (ccp\_alpha).

### 2️ Sensitive to Noisy Data

* Small changes in training data can cause large changes in the tree structure.
* Solution → Use ensemble methods like Random Forest to stabilize predictions.

### 3️ Biased Towards Dominant Classes

* If one class is more frequent, the tree may be biased toward that class.
* Solution → Use balanced datasets or class weighting.

### 4️ Greedy Algorithm (Suboptimal Splits)

* Decision Trees use a greedy approach that finds local best splits, not necessarily the global best.
* Solution → Use ensemble techniques (Bagging, Boosting) to improve performance.

1. How does a Decision Tree handle missing values

ANS –

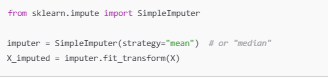
## **1️Handling Missing Values in Features (X)**

### (A) Ignore Missing Values (Sklearn Default)

* Scikit-learn's DecisionTreeClassifier does not support missing values (NaN) directly.
* If a feature contains NaN, you must fill it before training using imputation

### (B) Imputation Strategies

* Mean/Median Imputation (for numerical features)





### (C) Surrogate Splits (Used in Some Tree Algorithms)

### If a feature has missing values, some Decision Tree implementations create an alternative split using another feature.

### CART (Classification and Regression Trees) supports surrogate splits, but Scikit-learn does not implement this feature.

## **2️ Handling Missing Target Labels (y)**

* If the target label (y) has missing values, the Decision Tree simply **ignores those rows** during training.

## **3️ Advanced Handling Techniques**

* Create a “Missing” Category
  + Convert NaN values into a separate category (for categorical features)



* Use Predictive Imputation (KNN, Regression)
* Use models like KNN or Linear Regression to estimate missing values.

1. How does a Decision Tree handle categorical features

ANS –

Decision Trees naturally handle categorical features by splitting them into different branches. However, the way they handle categorical data depends on the implementation (e.g., Scikit-learn vs. other libraries like C4.5).

## **1️One-Hot Encoding (Used in Scikit-Learn)**

### Works with Scikit-Learn Decision Trees

Scikit-learn's DecisionTreeClassifier does not natively support categorical data. You need to convert categorical features into numerical values before training.

### Example: Convert Categorical to Numeric Using One-Hot Encoding



## **2️Label Encoding (For Ordinal Data)**

If the categorical feature has a natural order (e.g., "Low", "Medium", "High"), you can use Label Encoding.

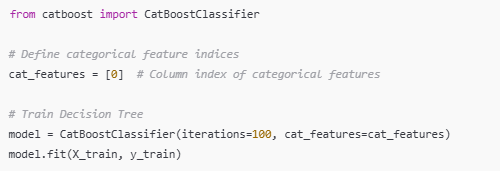
### Example: Label Encoding



## **3️Direct Handling of Categorical Features (C4.5, XGBoost, CatBoost)**

Some algorithms like C4.5, XGBoost, and CatBoost can handle categorical features directly without encoding.

Example using CatBoost:



1. What are some real-world applications of Decision Trees?

ANS –

## **1 Healthcare & Medical Diagnosis**

* **Disease Diagnosis** → Decision Trees help in diagnosing diseases like **diabetes, cancer, and heart disease** based on symptoms.
* **Treatment Recommendation** → Doctors use them to suggest treatments based on patient history.
* **Example**: A model predicts whether a patient has diabetes based on features like **blood sugar levels, age, and BMI**.

## **2 Banking & Finance**

* **Credit Risk Assessment** → Banks use Decision Trees to determine **whether to approve a loan** based on income, credit score, and employment status.
* **Fraud Detection** → Helps in detecting fraudulent transactions by analyzing spending patterns.
* **Investment Strategies** → Used to predict **stock market trends**.

## **3 E-commerce & Retail**

* **Customer Segmentation** → Groups customers based on purchase behavior for targeted marketing.
* **Churn Prediction** → Identifies customers likely to stop using a service.
* **Product Recommendation** → Suggests products based on customer preferences.

## **4 Manufacturing & Quality Control**

* **Defect Detection** → Classifies products as **defective or non-defective** based on quality parameters.
* **Supply Chain Optimization** → Predicts **demand and inventory needs** to optimize stock levels.

## **5 Marketing & Sales**

* **Lead Scoring** → Identifies high-value potential customers for businesses.
* **Ad Click Prediction** → Predicts whether a user will **click on an advertisement**.
* **Customer Support Automation** → Determines if a query needs **human intervention** or can be handled automatically.

## **6 Human Resources (HR) & Hiring**

* **Employee Attrition Prediction** → Identifies employees likely to leave the company.
* **Candidate Screening** → Helps in selecting the best candidates for a job role.

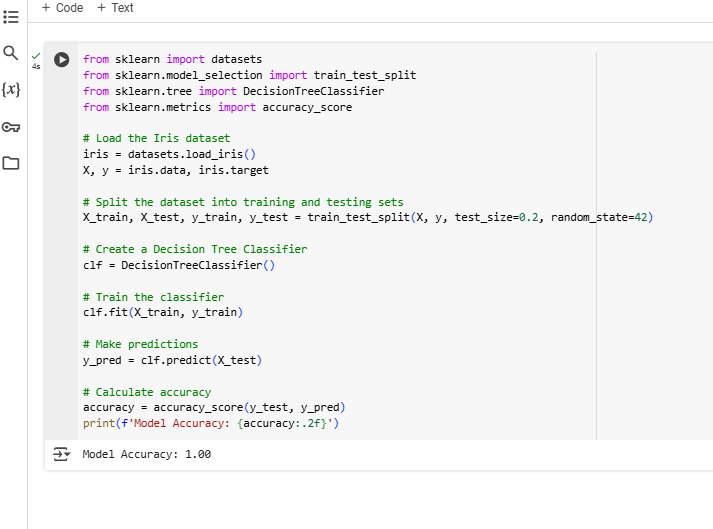
## **7 Cybersecurity**

* **Spam Detection** → Classifies emails as **spam or not spam**.
* **Intrusion Detection** → Identifies suspicious network activity for potential cyber threats.

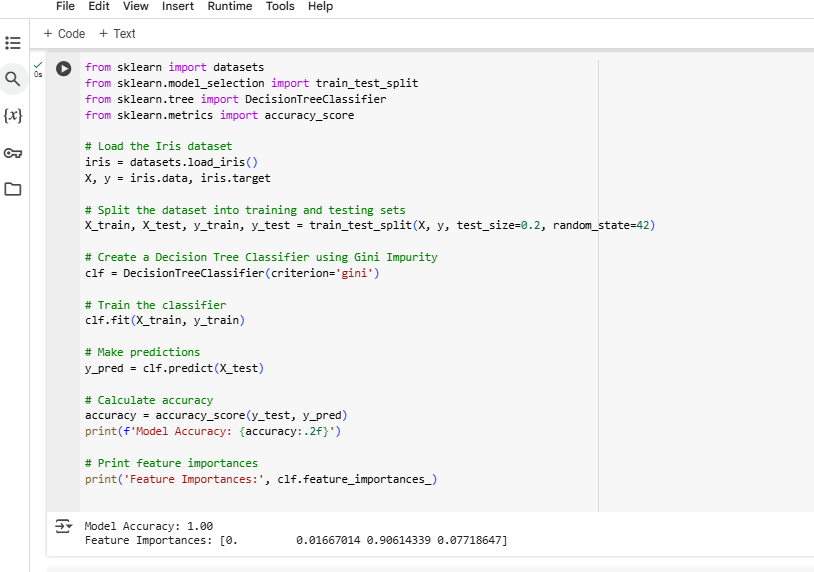
## **8 Agriculture & Weather Forecasting**

* **Crop Disease Detection** → Identifies diseases based on soil and plant conditions.
* **Yield Prediction** → Estimates crop yield based on factors like **rainfall, temperature, and soil fertility**.

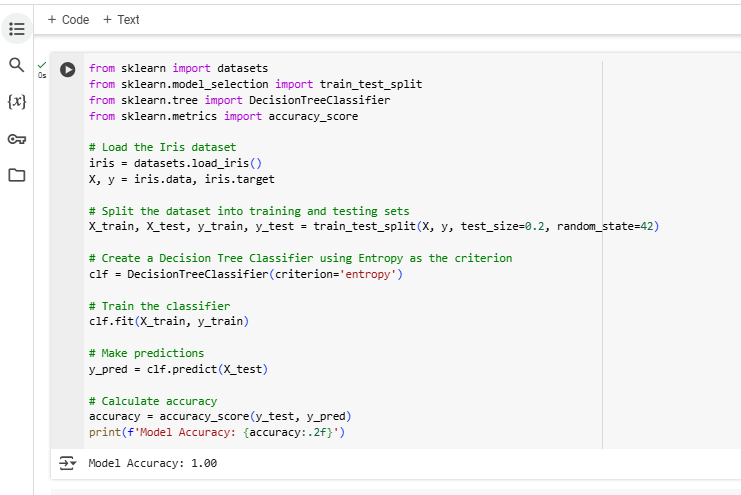
1. Write a Python program to train a Decision Tree Classifier on the Iris dataset and print the model accuracy



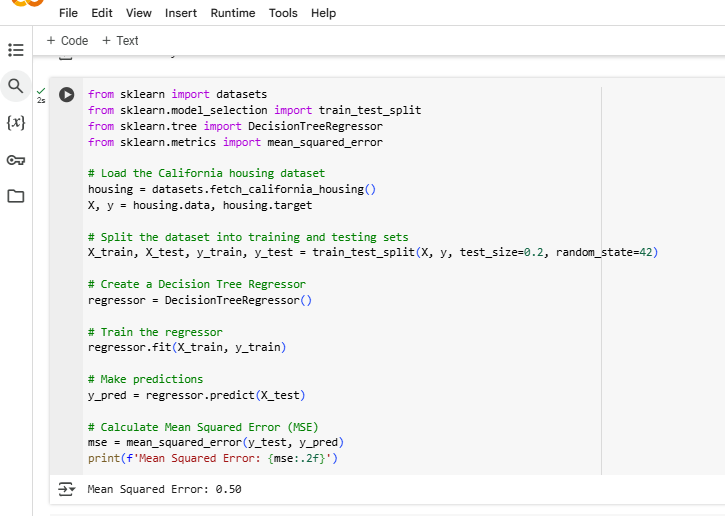
1. Write a Python program to train a Decision Tree Classifier using Gini Impurity as the criterion and print the feature importances



1. Write a Python program to train a Decision Tree Classifier using Entropy as the splitting criterion and print the model accuracy



1. Write a Python program to train a Decision Tree Regressor on a housing dataset and evaluate using Mean Squared Error (MSE)



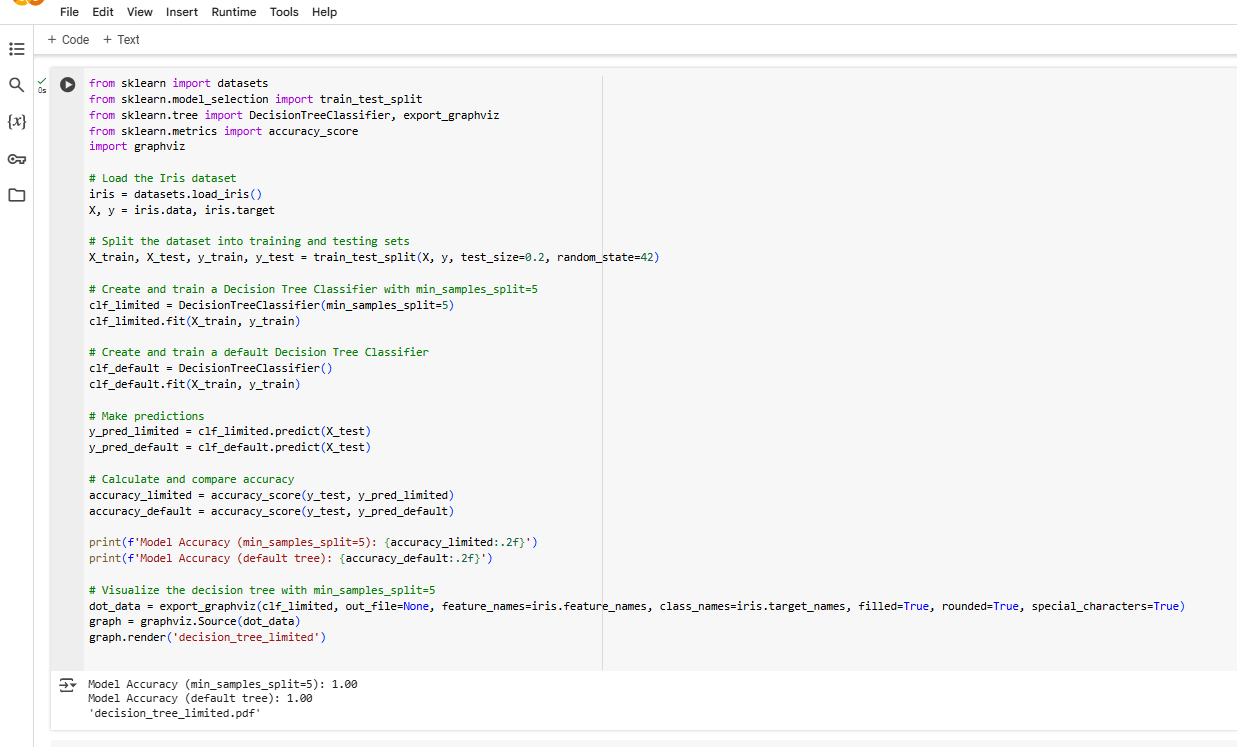
1. Write a Python program to train a Decision Tree Classifier and visualize the tree using graphviz



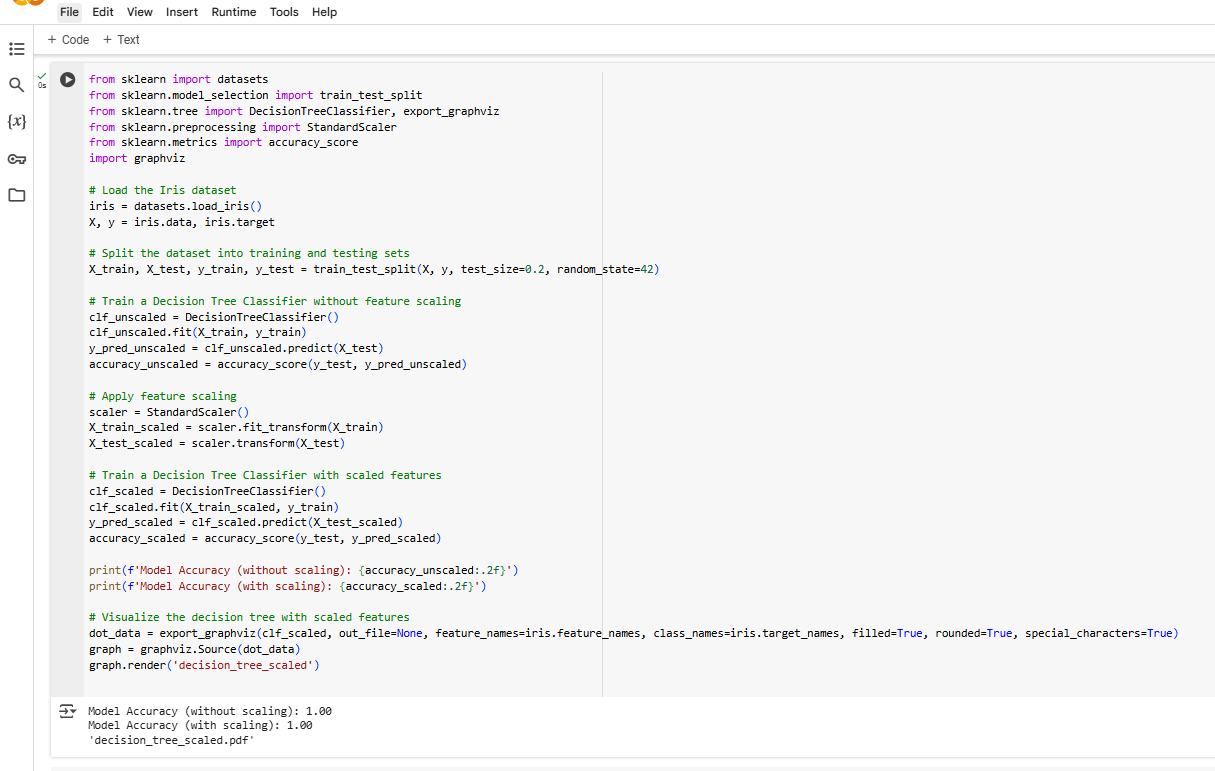
1. Write a Python program to train a Decision Tree Classifier with a maximum depth of 3 and compare its accuracy with a fully grown tree



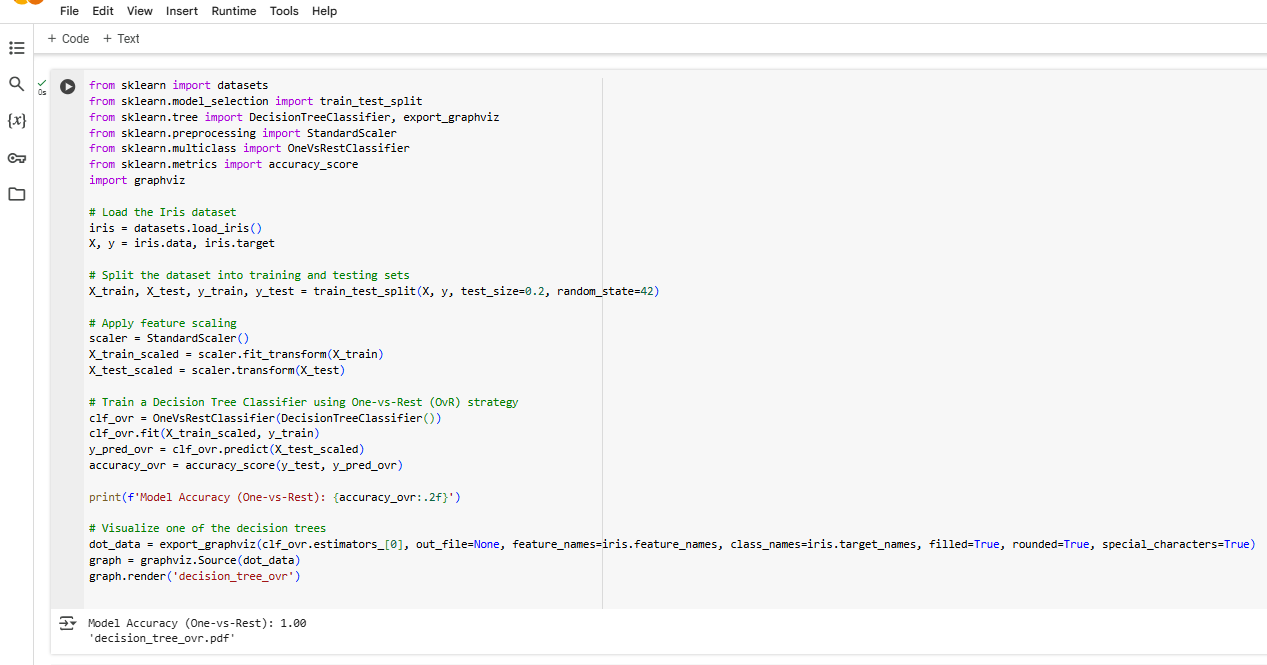
1. Write a Python program to train a Decision Tree Classifier using min\_samples\_split=5 and compare its accuracy with a default tree



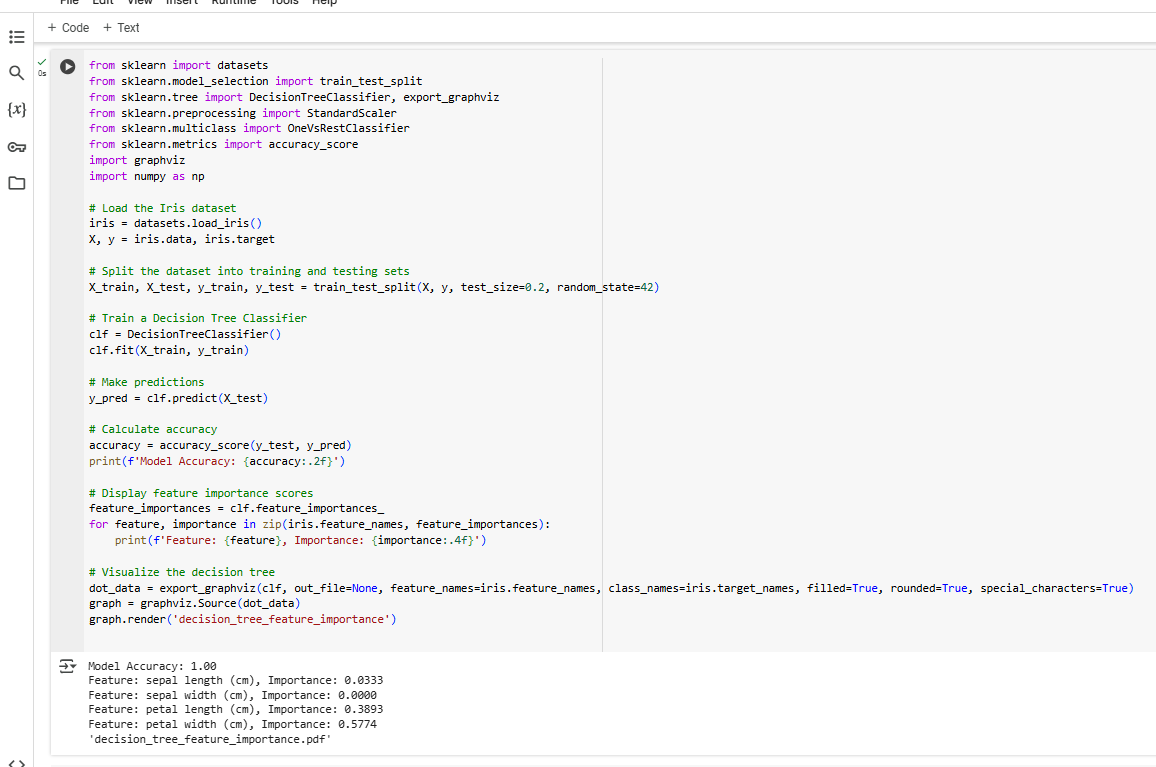
1. Write a Python program to apply feature scaling before training a Decision Tree Classifier and compare its accuracy with unscaled data



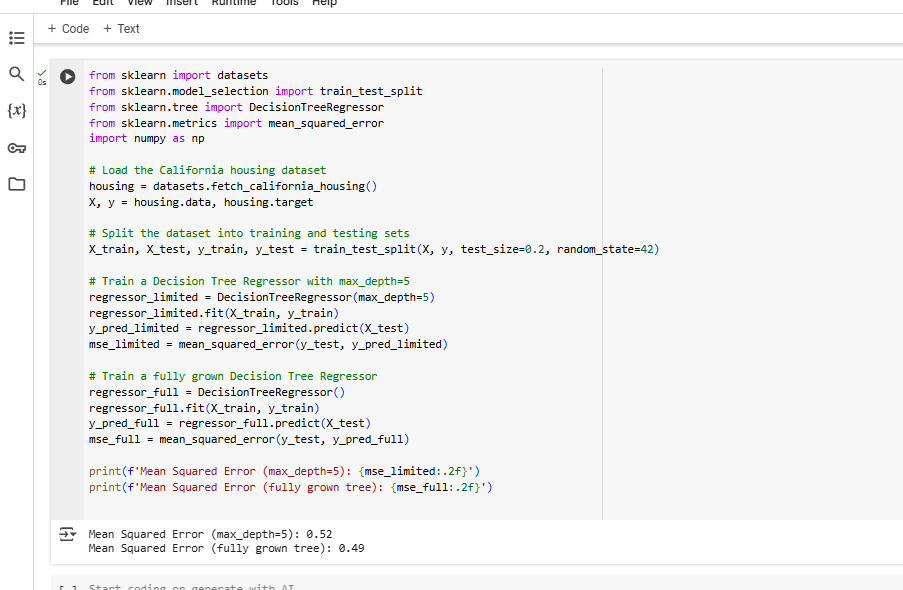
1. Write a Python program to train a Decision Tree Classifier using One-vs-Rest (OvR) strategy for multiclass classification



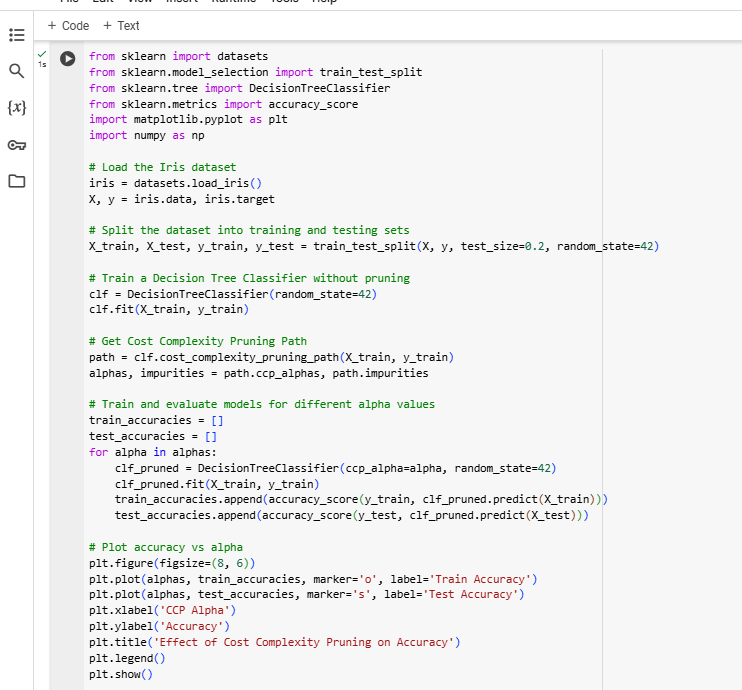
1. Write a Python program to train a Decision Tree Classifier and display the feature importance scores

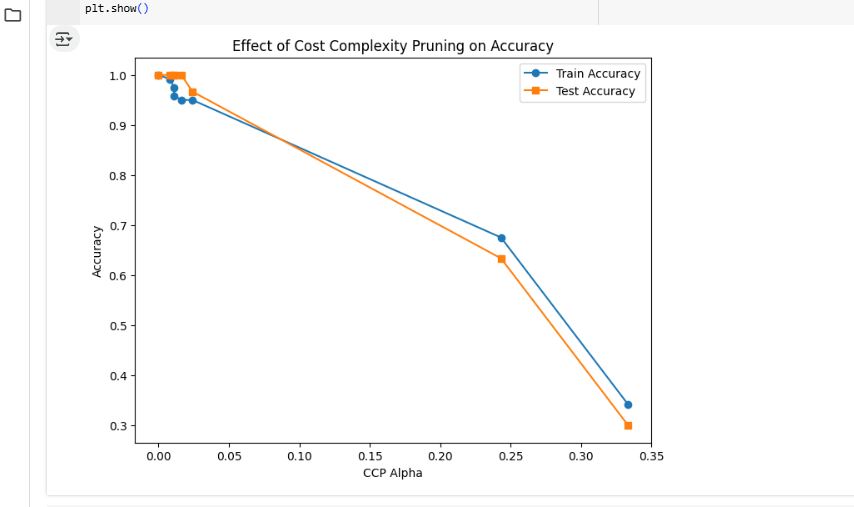


1. Write a Python program to train a Decision Tree Regressor with max\_depth=5 and compare its performance with an unrestricted tree



1. Write a Python program to train a Decision Tree Classifier, apply Cost Complexity Pruning (CCP), and visualize its effect on accuracy

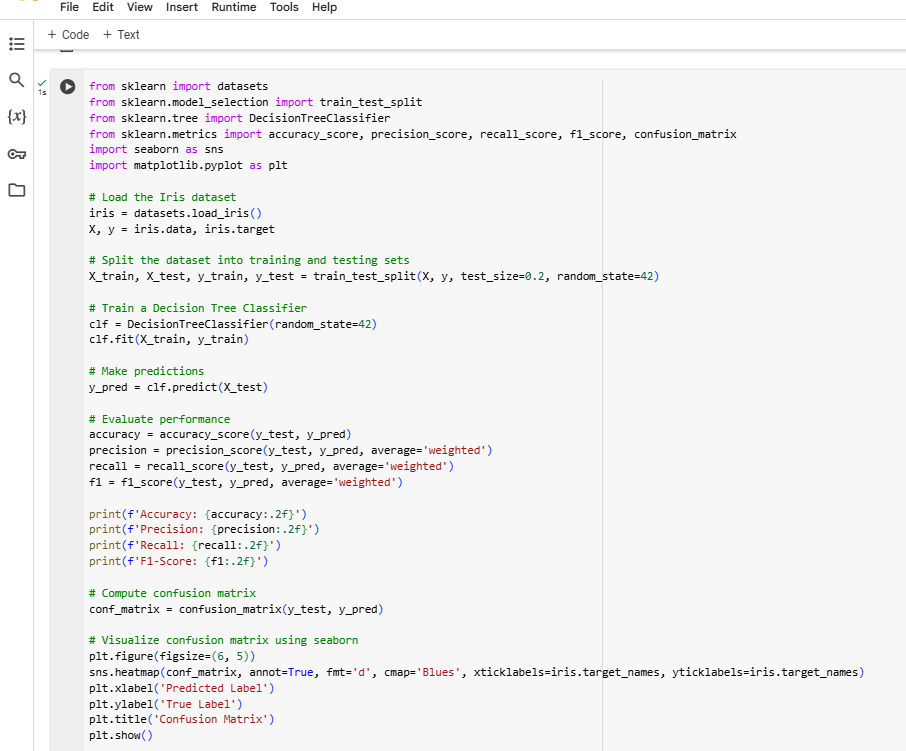


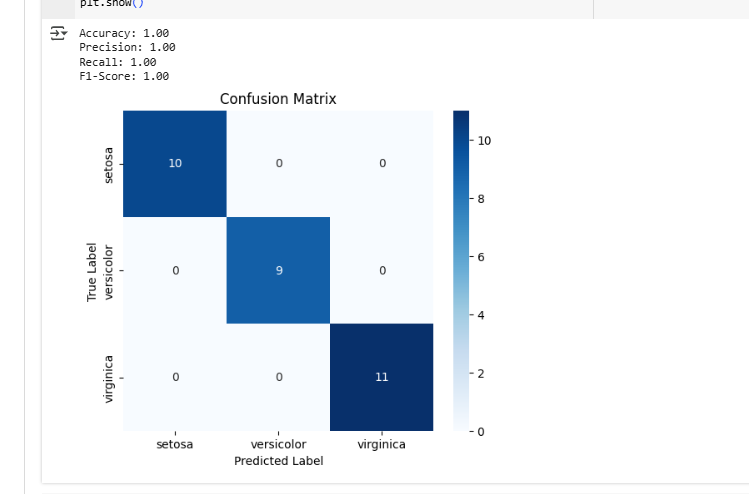


1. Write a Python program to train a Decision Tree Classifier and evaluate its performance using Precision, Recall, and F1-Score



1. Write a Python program to train a Decision Tree Classifier and visualize the confusion matrix using seaborn





1. Write a Python program to train a Decision Tree Classifier and use GridSearchCV to find the optimal values for max\_depth and min\_samples\_split.



