# **Decision Tree Classifier Experiment**

## **AIM**

To implement and demonstrate a Decision Tree classifier that can classify data based on input features by recursively splitting data on the most informative attributes.

## **PROCEDURE**

1. **Import dataset**: Use a simple dataset with labeled examples.
2. **Preprocess data**: Prepare input features and labels.
3. **Build Decision Tree model**:  
   * Select the best attribute to split data (based on impurity measures like Gini or Entropy).
   * Recursively split subsets until pure or stopping criteria met.
4. **Train the model** on the dataset.
5. **Predict labels** for test or training data.
6. **Evaluate accuracy** by comparing predicted vs actual labels.
7. **Visualize the tree** to understand decision paths.

## **CODE**

from sklearn.datasets import load\_iris

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

from sklearn.tree import DecisionTreeClassifier, plot\_tree

import matplotlib.pyplot as plt

from sklearn.metrics import accuracy\_score

# Load dataset

iris = load\_iris()

X, y = iris.data, iris.target

# Split dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Initialize Decision Tree classifier

clf = DecisionTreeClassifier(random\_state=42)

# Train the model

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

# Predict on test data

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

# Evaluate accuracy

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy of Decision Tree classifier: {accuracy:.2f}")

# Visualize the Decision Tree

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

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

plt.title("Decision Tree for Iris Dataset")

plt.show()

## **OUTPUT :**

Accuracy of Decision Tree classifier: 1.00

## **EXPLANATION**

* The Decision Tree splits data based on features like petal length, sepal width, etc.
* The model learns simple rules, e.g., if petal length < 2.45 cm then class = setosa.
* Accuracy of 1.00 shows perfect classification on test data (which is typical for iris dataset).
* Visualization clearly shows the tree splits and decision nodes.

## **CONCLUSION**

* Decision Trees provide an intuitive and interpretable model.
* They split data based on feature thresholds to classify samples.
* This experiment demonstrates training, prediction, evaluation, and visualization.
* Useful for classification tasks where interpretability is important.