Experiment No. 4

Title: Study and Implement Decision Tree.

Objectives:

1. Implement a Decision Tree classifier to classify the Iris dataset into three species.
2. Evaluate model performance using accuracy, classification report, and confusion matrix.
3. Visualize the decision tree structure and feature importance.

Problem Statement:

The problem requires:

- Developing a supervised machine learning model (Decision Tree).
- Evaluating the performance and interpretability of the model.
- Visualizing the decision-making process.

Outcomes:

1. A trained Decision Tree classifier with optimal parameters.
2. A performance evaluation of the model using accuracy, confusion matrix, and classification report.
3. A graphical representation of the decision tree to illustrate the decision-making process.

Tools Required: 4GB RAM, Anaconda, Notebook

Theory:

A Decision Tree is a supervised learning algorithm used for classification and regression tasks. It works by recursively splitting data based on feature values to maximize information gain (entropy reduction or Gini impurity). The key concepts include:

- Gini Impurity: Measures how often a randomly chosen element would be incorrectly classified.
- Entropy: Measures the disorder or impurity of the dataset.
- Information Gain: The reduction in entropy after splitting on a particular feature.
- Pruning: Reducing overfitting by limiting the depth or complexity of the tree.

Algorithm:

Step 1: Load the Dataset

- Import the load_iris() dataset from sklearn.datasets.
- Extract feature variables (X) and target labels (y).

Step 2: Data Preprocessing

- Split the dataset into training and testing sets using train_test_split().
- Normalize or scale features if necessary (not required for Decision Trees).

Step 3: Train Decision Tree Model

- Initialize the Decision Tree classifier with criterion='gini' and max_depth=3.
- Train the model using fit(X_train, y_train).

Step 4: Make Predictions

- Use the trained model to predict labels for X_test.

Step 5: Evaluate Model Performance

- Compute accuracy using accuracy_score(y_test, y_pred).
- Generate a classification report with precision, recall, and F1-score.
- Plot a confusion matrix to analyze misclassifications.

Step 6: Visualize the Model

- Use tree.plot_tree() to graphically represent the decision tree.
- Plot feature importance to identify which features contribute most.

Source Code:

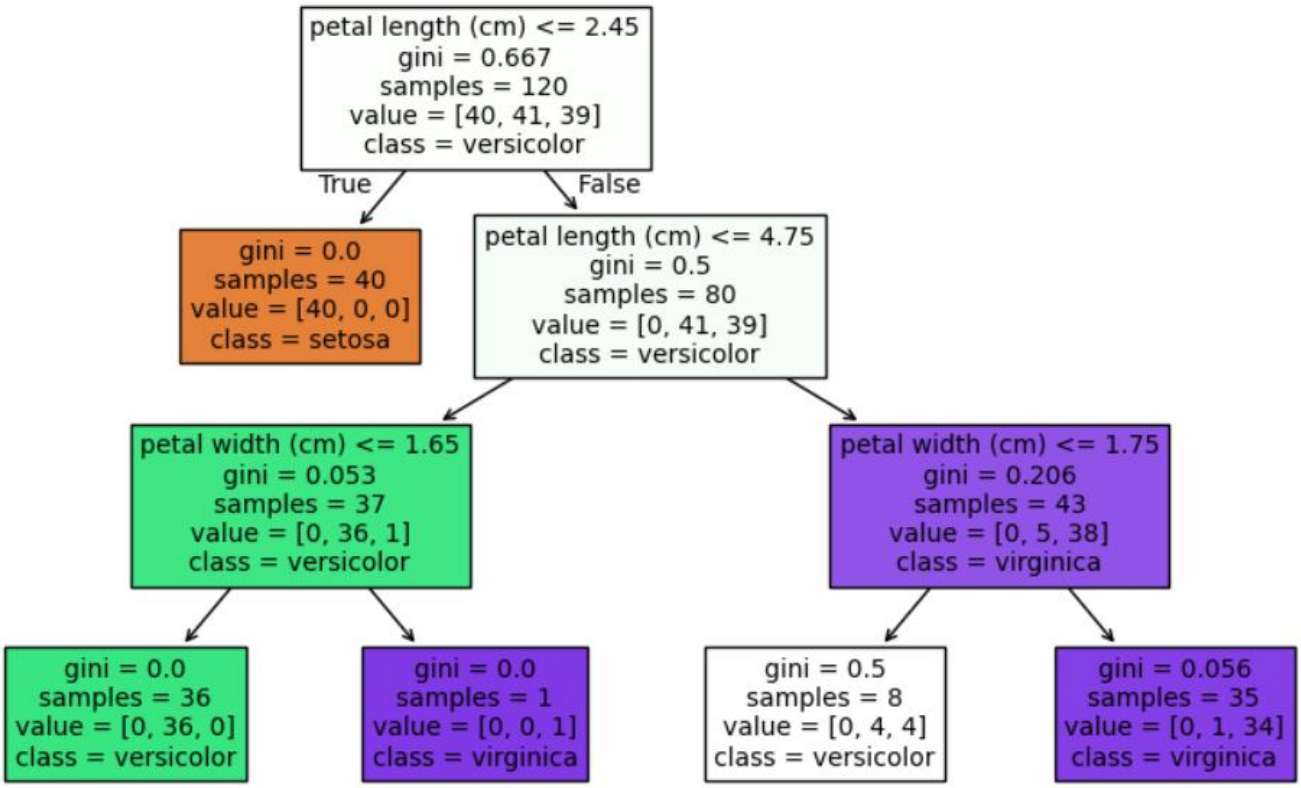
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.datasets import load_iris
data = load_iris()
X = data.data
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
dt_model = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=42)
dt_model.fit(X_train, y_train)
y_pred = dt_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))
tree.plot_tree(dt_model, feature_names=data.feature_names,
  class_names=data.target_names, filled=True)
plt.show()
```

Output:

Accuracy: 1.0

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	9
2	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30


petal length (cm) <= 2.45
gini = 0.667
samples = 120
value = [40, 41, 39]
class = versicolor
True
False
gini = 0.0
samples = 40
value = [40, 0, 0]
class = setosa
petal length (cm) <= 4.75
gini = 0.5
samples = 80
value = [0, 41, 39]
class = versicolor
petal width (cm) <= 1.65
gini = 0.053
samples = 37
value = [0, 36, 1]
class = versicolor
petal width (cm) <= 1.75
gini = 0.206
samples = 43
value = [0, 5, 38]
class = virginica
gini = 0.0
samples = 36
value = [0, 36, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
gini = 0.5
samples = 8
value = [0, 4, 4]
class = versicolor
gini = 0.056
samples = 35
value = [0, 1, 34]
class = virginica

Conclusion:

_____

_____

_____

_____