EXPT NO: 7 **A python program to implement Decision tree**

DATE: 04.10.2024

AIM:

To write a python program to implement a Decision tree.

PROCEDURE:

Implementing the decision tree using the Iris dataset involve the following steps:

Step 1: Import Necessary Libraries

First, import the libraries that are essential for data manipulation, visualization, and model building.

```
import numpy as np

import pandas as pd

from sklearn import datasets

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn import metrics

import matplotlib.pyplot as plt

from sklearn.tree import plot_tree
```

Step 2: Load the Iris Dataset

The Iris dataset can be loaded and display the first few rows of the dataset .

```
# Load the Iris dataset

iris = datasets.load_iris()

X = iris.data  # Features
```

```
y = iris.target  # Target variable
```

Step 3 : Split the data set into training and testing sets

```
# Split the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

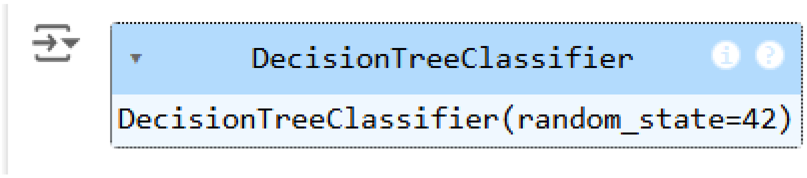
Step 4 : Create a decision tree classifier

```
# Create a Decision Tree classifier

clf = DecisionTreeClassifier(random_state=42)
```

Step 5 : Train the model :

```
# Train the model

clf.fit(X_train, y_train)
```

OUTPUT :



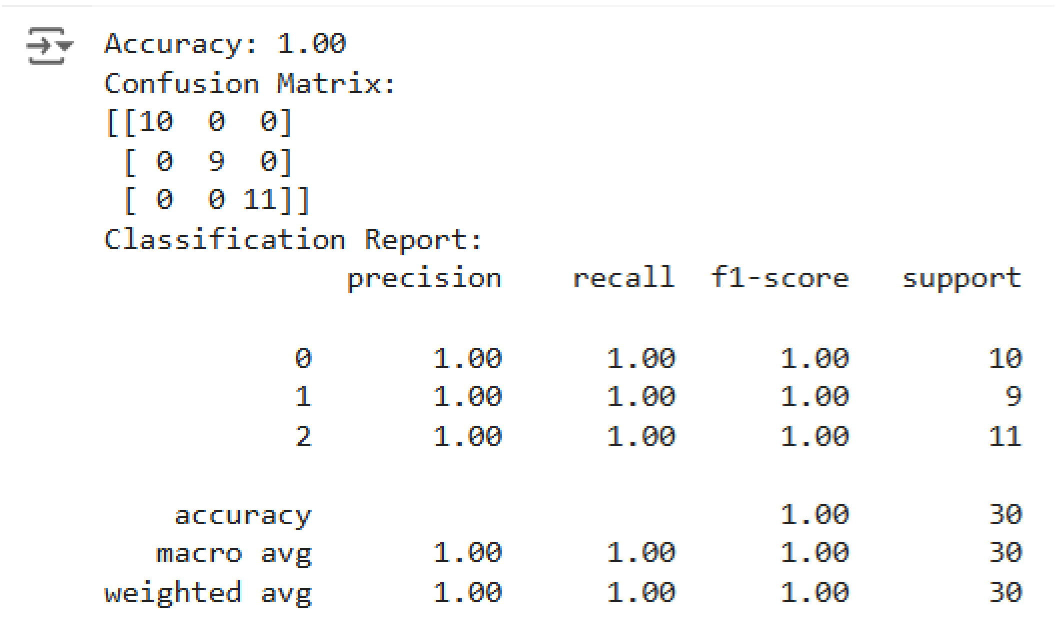
```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

Step 6 : Make the predictions and evaluate the model

```
# Make predictions

y_pred = clf.predict(X_test)



# Evaluate the model

accuracy = metrics.accuracy_score(y_test, y_pred)

confusion = metrics.confusion_matrix(y_test, y_pred)

classification_report = metrics.classification_report(y_test, y_pred)



print(f"Accuracy: {accuracy:.2f}")

print("Confusion Matrix:")
```

```python
print(confusion)

print("Classification Report:")

print(classification_report)
```

OUTPUT :

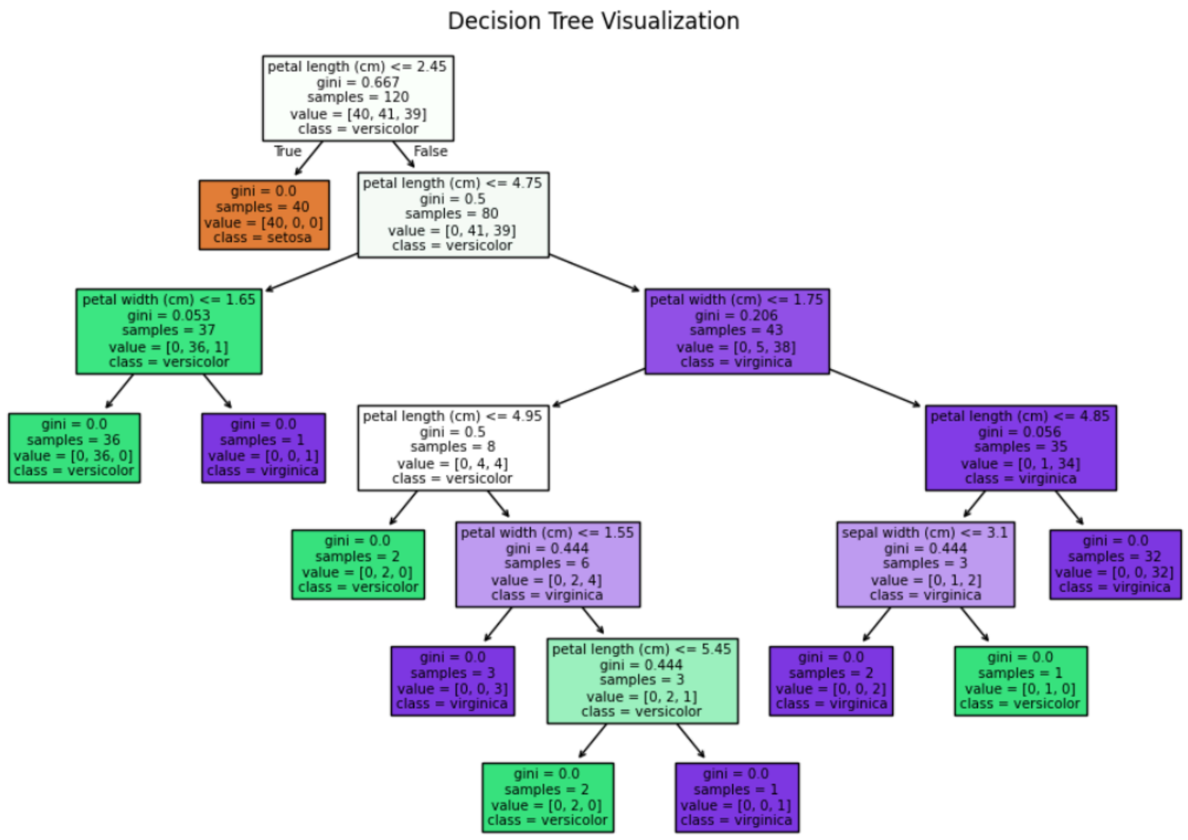


```
Accuracy: 1.00
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
Classification Report:
              precision    recall  f1-score   support

           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
```

Step 7 : Visualize the decision tree

```python
# Visualize the Decision Tree

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

plot_tree(clf, filled=True, feature_names=iris.feature_names,
class_names=iris.target_names)

plt.title("Decision Tree Visualization")

plt.show()
```

OUTPUT :



RESULT :

This process helps us to implement the decision tree using a python program.