Exno: 07

date : 04/10/2025

A python program to implement decision tree

Aim:-

To implement a decision tree using a python program

Algorithm :-

Step 1 :- Import the Iris Dataset

Step 2 : Import Necessary libraries

Step 3 :- Declare and Intialize parameters

Step 4 : Prepare Data for model training

Step 5 : Train the Model.

Step 6 : Initialize pair Index and plot Graph

Step 7 : Assign Axis Limits

Step 8 : Create Meshgrid

Step 9 : plot Graph with Tight Layout

Step 10 : predict And Reshape

Step 11 : plot Decision Boundaries

Step 12 : plot Features pairs

Step 13 : plot training points

Step 14 : plot final Decision tree

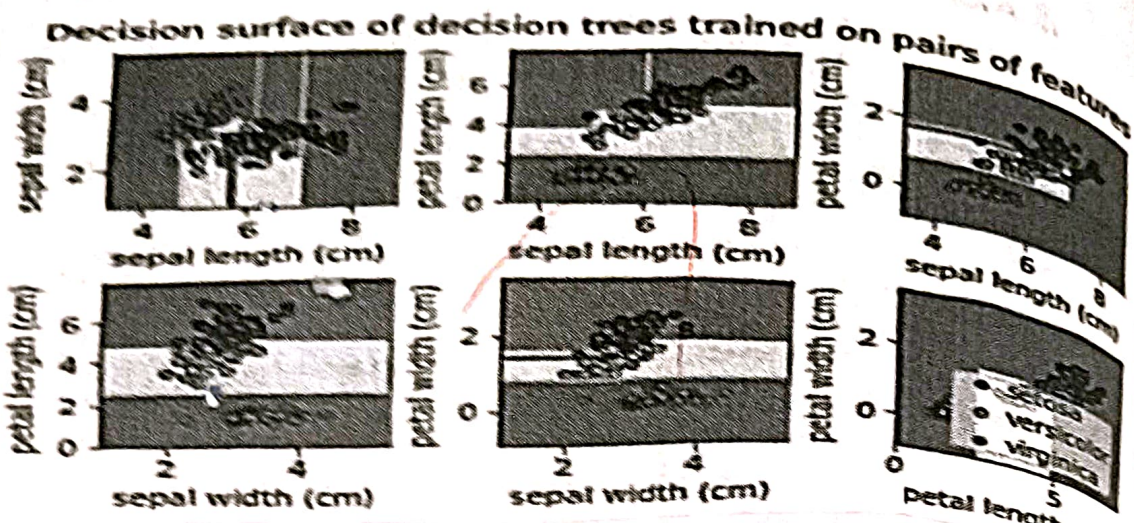
Program

```
from sklearn.dataset import load_iris
iris = load_iris()
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
n_classes = 3
plot_colors = "ryb"
plot_step = 0.02
for pairidx, pair in enumerate([[0,1],[0,2],[0,3],
[1,2],[1,3],[2,3]]):
x = iris.data[:, pair]
y = iris.target
clf = DecisionTreeClassifier().fit(x,y)
plt.subplot(2,3, pairidx+1)
x_min, x_max = x[:,0].min()-1, x[:,0].max()+1
xx, yy = np.meshgrid(np.arange(x_min, x_max, plot_step)
    np.arange(y_min, y_max, plot_step))
plt.tight_layout(h_pad=0.5, w_pad=0.5, pad=2.5)
Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
cs = plt.contourf(xx, yy, Z, cmap=plt.cm.RdYlBu)
plt.ylabel(iris.feature_names[pair[1]])
for i, color in zip(range(n_classes), plot_colors):
    idx = np.where(y==i)
    plt.scatter(x[idx,0], x[idx,1], c=color
```

```
plt.scatter(X[:, 0], X[:, 1], c=iris.target,
label=iris.target_names[i],
cmap=plt.cm.RdYlBu, edgecolor="black",
s=15)
plt.suptitle("Decision surface of
decision tree trained
on pairs of features")
plt.legend(loc="lower right", borderpad=0)
plt.axis("tight")
from sklearn.tree import plot_tree
plt.figure()
clf = DecisionTreeClassifier().fit(iris.data
, iris.target)
plot_tree(clf, filled=True)
plt.title("Decision tree trained on all
the iris features")
plt.show()
```

Result:

Thus the python program to implement Decision tree for the given dataset has been successfully executed.



	precision	recall	f1-score	support
0	0.67	1.00	0.80	2
1	1.00	0.50	0.67	2
accuracy			0.75	4
macro avg	0.83	0.75	0.73	4
weighted avg	0.83	0.75	0.73	4

Decision tree trained on all the iris features