

A PYTHON PROGRAM TO IMPLEMENT DECISION TREE

EXP NO. 7

Name: MOHAMED FUZAIL A

Roll no: 241801162

Code:

```
from sklearn.datasets import load_iris  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.tree import DecisionTreeClassifier, plot_tree  
  
# Load dataset  
iris = load_iris()  
  
# Parameters  
n_classes = 3  
plot_colors = "ryb"  
plot_step = 0.02  
  
# Plot decision boundaries for all feature pairs  
plt.figure(figsize=(15, 10))  
for pairidx, pair in enumerate([[0, 1], [0, 2], [0, 3],
```

```
[1, 2], [1, 3], [2, 3]]):  
# Select the two features  
X = iris.data[:, pair]  
y = iris.target  
  
# Train the decision tree  
clf = DecisionTreeClassifier().fit(X, y)  
  
# Plot decision boundary  
plt.subplot(2, 3, pairidx + 1)  
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1  
y_min, y_max = X[:, 1].min() - 1, X[:, 1].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)  
plt.contourf(xx, yy, Z, cmap=plt.cm.RdYlBu)  
  
# Label axes
```

```
plt.xlabel(iris.feature_names[pair[0]])
plt.ylabel(iris.feature_names[pair[1]])

# Plot the training points
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,
        label=iris.target_names[i],
        cmap=plt.cm.RdYlBu,
        edgecolor="black",
        s=15
    )

plt.suptitle("Decision surface of Decision Trees trained on pairs of
features")
plt.legend(loc="lower right", borderpad=0, handletextpad=0)
plt.axis("tight")
plt.show()

# Plot full decision tree using all features
```

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

clf = DecisionTreeClassifier().fit(iris.data, iris.target)

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

plt.title("Decision Tree trained on all the Iris features")

plt.show()
```

output:

