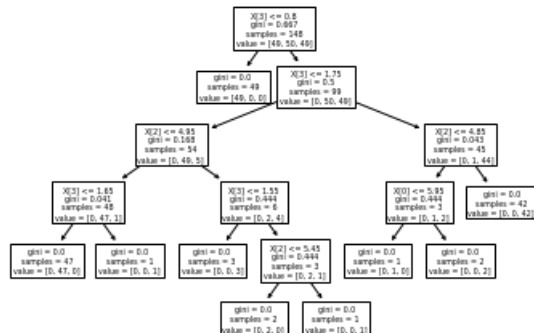


```
In [1]: from sklearn.datasets import load_iris
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

```
In [2]: data=load_iris()
x,y=data.data,data.target
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=2)
```

```
In [3]: dt=tree.DecisionTreeClassifier()
dt.fit(x_train,y_train)
tree.plot_tree(dt)
```

```
Out[3]: [Text(167.4, 199.32, 'X[3] <= 0.8\ngini = 0.667\nsamples = 148\nvalue = [49, 50, 49]'),
Text(141.64615384615385, 163.07999999999998, 'gini = 0.0\nsamples = 49\nvalue = [49, 0, 0]'),
Text(193.15384615384616, 163.07999999999998, 'X[3] <= 1.75\ngini = 0.5\nsamples = 99\nvalue = [0, 5, 49]'),
Text(103.01538461538462, 126.83999999999999, 'X[2] <= 4.95\ngini = 0.168\nsamples = 54\nvalue = [0, 49, 5]'),
Text(51.50769230769231, 90.6, 'X[3] <= 1.65\ngini = 0.041\nsamples = 48\nvalue = [0, 47, 1]'),
Text(25.753846153846155, 54.359999999999985, 'gini = 0.0\nsamples = 47\nvalue = [0, 47, 0]'),
Text(77.26153846153846, 54.359999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(154.52307692307693, 90.6, 'X[3] <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 2, 4]'),
Text(128.76923076923077, 54.359999999999985, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
Text(180.27692307692308, 54.359999999999985, 'X[2] <= 5.45\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),
Text(154.52307692307693, 18.119999999999976, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
Text(206.03076923076924, 18.119999999999976, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(283.2923076923077, 126.83999999999999, 'X[2] <= 4.85\ngini = 0.043\nsamples = 45\nvalue = [0, 1, 44]'),
Text(257.53846153846155, 90.6, 'X[0] <= 5.95\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
Text(231.7846153846154, 54.359999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
Text(283.2923076923077, 54.359999999999985, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
Text(309.04615384615386, 90.6, 'gini = 0.0\nsamples = 42\nvalue = [0, 0, 42]')]
```



```
In [4]: y_predict=dt.predict(x_test)
print(y_predict)
accuracy_score(y_test,y_predict)
```

```
[ 2  0]
```

```
Out[4]: 1.0
```