

In [13]:

```
import numpy as np
import sklearn
from sklearn import tree
import pandas as pd
```

In [14]:

```
data=pd.read_csv("./play_tennis.csv")
data
```

Out[14]:

| | day | outlook | temp | humidity | wind | play |
|----|-----|----------|------|----------|--------|------|
| 0 | D1 | Sunny | Hot | High | Weak | No |
| 1 | D2 | Sunny | Hot | High | Strong | No |
| 2 | D3 | Overcast | Hot | High | Weak | Yes |
| 3 | D4 | Rain | Mild | High | Weak | Yes |
| 4 | D5 | Rain | Cool | Normal | Weak | Yes |
| 5 | D6 | Rain | Cool | Normal | Strong | No |
| 6 | D7 | Overcast | Cool | Normal | Strong | Yes |
| 7 | D8 | Sunny | Mild | High | Weak | No |
| 8 | D9 | Sunny | Cool | Normal | Weak | Yes |
| 9 | D10 | Rain | Mild | Normal | Weak | Yes |
| 10 | D11 | Sunny | Mild | Normal | Strong | Yes |
| 11 | D12 | Overcast | Mild | High | Strong | Yes |
| 12 | D13 | Overcast | Hot | Normal | Weak | Yes |
| 13 | D14 | Rain | Mild | High | Strong | No |

In [15]:

```
data.play
```

Out[15]:

```
0      No
1      No
2      Yes
3      Yes
4      Yes
5      No
6      Yes
7      No
8      Yes
9      Yes
10     Yes
11     Yes
12     Yes
13     No
```

Name: play, dtype: object

In [16]:

```
d=np.array(data)[:,-1]
print(d)
```

```
[['Sunny' 'Hot' 'High' 'Weak']
 ['Sunny' 'Hot' 'High' 'Strong']
 ['Overcast' 'Hot' 'High' 'Weak']
 ['Rain' 'Mild' 'High' 'Weak']
 ['Rain' 'Cool' 'Normal' 'Weak']
 ['Rain' 'Cool' 'Normal' 'Strong']
 ['Overcast' 'Cool' 'Normal' 'Strong']
 ['Sunny' 'Mild' 'High' 'Weak']
 ['Sunny' 'Cool' 'Normal' 'Weak']
 ['Rain' 'Mild' 'Normal' 'Weak']
 ['Sunny' 'Mild' 'Normal' 'Strong']
 ['Overcast' 'Mild' 'High' 'Strong']
 ['Overcast' 'Hot' 'Normal' 'Weak']
 ['Rain' 'Mild' 'High' 'Strong']]
```

In [17]:

```
targets=np.array(data)[:,-1]
print(targets)
```

```
['No' 'No' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes'
 'No']
```

In [18]:

```
data['outlook'].replace(['Sunny', 'Overcast', 'Rain'],[0,1,2], inplace=True)
data['temp'].replace(['Hot', 'Mild', 'Cool'],[0,1,2], inplace=True)
data['humidity'].replace(['High', 'Normal'],[0,1], inplace=True)
data['wind'].replace(['Weak', 'Strong'],[0,1], inplace=True)
data['play'].replace(['No', 'Yes'],[0,1], inplace=True)
data
```

Out[18]:

| | day | outlook | temp | humidity | wind | play |
|----|-----|---------|------|----------|------|------|
| 0 | D1 | 0 | 0 | 0 | 0 | 0 |
| 1 | D2 | 0 | 0 | 0 | 1 | 0 |
| 2 | D3 | 1 | 0 | 0 | 0 | 1 |
| 3 | D4 | 2 | 1 | 0 | 0 | 1 |
| 4 | D5 | 2 | 2 | 1 | 0 | 1 |
| 5 | D6 | 2 | 2 | 1 | 1 | 0 |
| 6 | D7 | 1 | 2 | 1 | 1 | 1 |
| 7 | D8 | 0 | 1 | 0 | 0 | 0 |
| 8 | D9 | 0 | 2 | 1 | 0 | 1 |
| 9 | D10 | 2 | 1 | 1 | 0 | 1 |
| 10 | D11 | 0 | 1 | 1 | 1 | 1 |
| 11 | D12 | 1 | 1 | 0 | 1 | 1 |
| 12 | D13 | 1 | 0 | 1 | 0 | 1 |
| 13 | D14 | 2 | 1 | 0 | 1 | 0 |

In [19]:

```
d=np.array(data)[:,-1]
print(d)
d1=d.tolist()
print(d1)
```

```
[[0 0 0 0]
 [0 0 0 1]
 [1 0 0 0]
 [2 1 0 0]
 [2 2 1 0]
 [2 2 1 1]
 [1 2 1 1]
 [0 1 0 0]
 [0 2 1 0]
 [2 1 1 0]
 [0 1 1 1]
 [1 1 0 1]
 [1 0 1 0]
 [2 1 0 1]]
[[0, 0, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [2, 1, 0, 0], [2, 2, 1, 0], [2,
2, 1, 1], [1, 2, 1, 1], [0, 1, 0, 0], [0, 2, 1, 0], [2, 1, 1, 0], [0, 1,
1, 1], [1, 1, 0, 1], [1, 0, 1, 0], [2, 1, 0, 1]]
```

In [20]:

```
targets=np.array(data)[:,-1]
print(targets)
target=targets.tolist()
print(target)
```

```
[0 0 1 1 1 0 1 0 1 1 1 1 1 0]
[0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0]
```

In [21]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(d1,target,test_size=0.25,random_state=No
ne)
```

In [27]:

```
clf = tree.DecisionTreeClassifier()
clf = clf.fit(x_train,y_train)
clf.predict([[0.,0.,0.,1.]])
```

Out[27]:

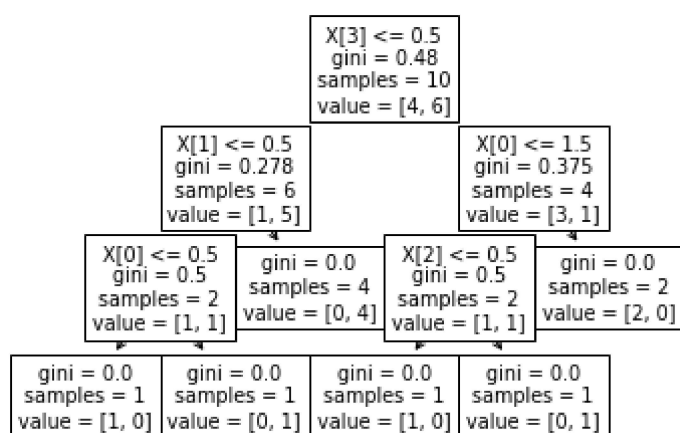
```
array([0])
```

In [28]:

```
tree.plot_tree(clf)
```

Out[28]:

```
[Text(0.5555555555555556, 0.875, 'X[3] <= 0.5\ngini = 0.48\nsamples = 10\nvalue = [4, 6]'),  
Text(0.3333333333333333, 0.625, 'X[1] <= 0.5\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),  
Text(0.2222222222222222, 0.375, 'X[0] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.1111111111111111, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.3333333333333333, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.4444444444444444, 0.375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),  
Text(0.7777777777777778, 0.625, 'X[0] <= 1.5\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.6666666666666666, 0.375, 'X[2] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.5555555555555556, 0.125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.7777777777777778, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8888888888888888, 0.375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]')]
```



In [24]:

```
from sklearn import metrics
```

```
y_pred=clf.predict(x_test)
```

```
print("Accuracy ",metrics.accuracy_score(y_test,y_pred))
```

Accuracy 0.5