Knowledge Discovery & Data Mining Lab-07

Name: Kunal Sanjay Patil

PRN: 20190802025

AIM :-

To implement Decision Tree using pyrhon.

In [1]:

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import tree
```

In [2]:

```
data = pd.read_csv('tennis.csv')
data.head()
```

Out[2]:

	outlook	temp	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes

In [3]:

```
data.shape
```

Out[3]:

```
(14, 5)
```

In [4]:

```
data.isnull().sum()
```

Out[4]:

```
outlook     0
temp        0
humidity    0
windy       0
play        0
dtype: int64
```

In [5]:

```
X = data.iloc[:,0:4].values
y = data.iloc[:,4].values
```

In [6]:

```
labelencoder= LabelEncoder()
X[:,0] = labelencoder.fit_transform(X[:,0])
X[:,1] = labelencoder.fit_transform(X[:,1])
X[:,2] = labelencoder.fit_transform(X[:,2])
X[:,3] = labelencoder.fit_transform(X[:,3])
y = labelencoder.fit_transform(y)
```

In [7]:

```
X
```

Out[7]:

```
array([[2, 1, 0, 0],
       [2, 1, 0, 1],
       [0, 1, 0, 0],
       [1, 2, 0, 0],
       [1, 0, 1, 0],
       [1, 0, 1, 1],
       [0, 0, 1, 1],
       [2, 2, 0, 0],
       [2, 0, 1, 0],
       [1, 2, 1, 0],
       [2, 2, 1, 1],
       [0, 2, 0, 1],
       [0, 1, 1, 0],
       [1, 2, 0, 1]], dtype=object)
```

In [8]:

```
y
```

Out[8]:

```
array([0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0])
```

In [9]:

```
# splitting the dataset in training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [10]:

```
clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

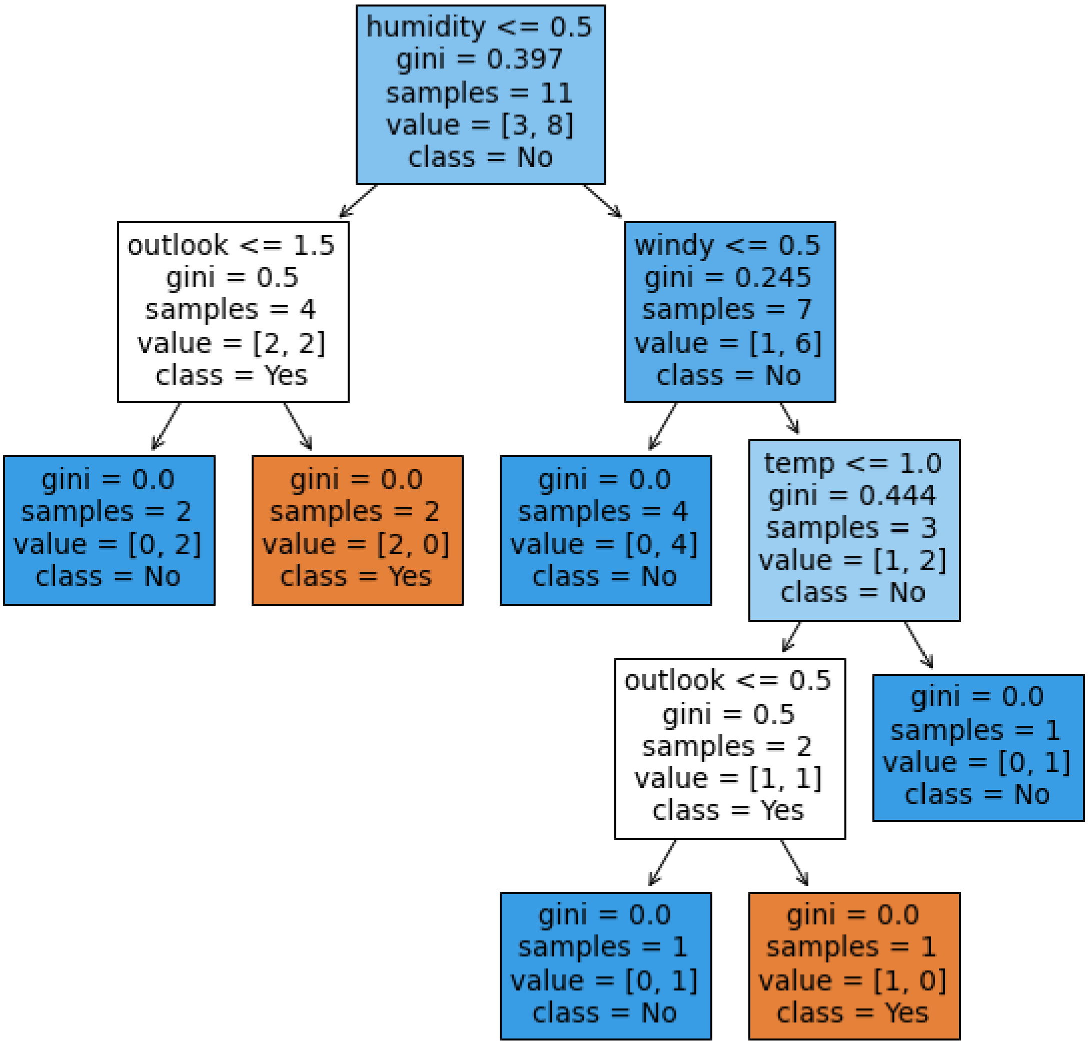
In [11]:

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6666666666666666
```

In [12]:

```
from matplotlib import pyplot as plt
fig = plt.figure(figsize=(10,10))
tree.plot_tree(clf,
                feature_names=['outlook', 'temp', 'humidity', 'windy'],
                class_names=['Yes', 'No'],
                filled=True);
```


humidity <= 0.5
gini = 0.397
samples = 11
value = [3, 8]
class = No
outlook <= 1.5
gini = 0.5
samples = 4
value = [2, 2]
class = Yes
windy <= 0.5
gini = 0.245
samples = 7
value = [1, 6]
class = No
gini = 0.0
samples = 2
value = [0, 2]
class = No
gini = 0.0
samples = 2
value = [2, 0]
class = Yes
gini = 0.0
samples = 4
value = [0, 4]
class = No
temp <= 1.0
gini = 0.444
samples = 3
value = [1, 2]
class = No
outlook <= 0.5
gini = 0.5
samples = 2
value = [1, 1]
class = Yes
gini = 0.0
samples = 1
value = [0, 1]
class = No
gini = 0.0
samples = 1
value = [0, 1]
class = No
gini = 0.0
samples = 1
value = [1, 0]
class = Yes