In [7]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,classification_report
```

In [8]:
```
#Reading the file
gd=pd.read_excel(r"C:\Users\Agnel Sharon Jerald\OneDrive\Desktop\Machine learning\weather_forecast.xlsx")
gd
```

Out[8]:

	Outlook	Temperature	Humidity	Windy	Play
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
12	Overcast	Hot	Normal	Weak	Yes
13	Rain	Mild	High	Strong	No

In [9]:
```
gd.shape
```

Out[9]:
```
(14, 5)
```

In [10]:
```
gd.dtypes
```

Out[10]:
```
Outlook        object
Temperature    object
Humidity       object
Windy          object
Play           object
dtype: object
```

In [11]:
```
gd.isna().sum()
```

Out[11]:
```
Outlook        0
Temperature    0
Humidity       0
Windy          0
Play           0
dtype: int64
```

In [12]:
```
x=gd.iloc[:,:-1]
y=gd.iloc[:,-1]
```

In [21]:
```
ohe=OneHotEncoder(drop='first',sparse_output=False)
x_trns = ohe.fit_transform(x)
x_trns
```

Out[21]:
```
array([[0., 1., 1., 0., 0., 1.],
       [0., 1., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 1.],
       [1., 0., 0., 1., 0., 1.],
       [1., 0., 0., 0., 1., 1.],
       [1., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1., 0.],
       [0., 1., 0., 1., 0., 1.],
       [0., 1., 0., 0., 1., 1.],
       [1., 0., 0., 1., 1., 1.],
       [0., 1., 0., 1., 1., 0.],
       [0., 0., 0., 1., 0., 0.],
       [0., 0., 1., 0., 1., 1.],
       [1., 0., 0., 1., 0., 0.]])
```

In [26]:
```
x_trns=pd.DataFrame(x_trns,columns=ohe.get_feature_names_out(x.columns))
print(x_trns)
```

```
    Outlook_Rain  Outlook_Sunny  Temperature_Hot  Temperature_Mild  \
0            0.0            1.0              1.0               0.0
1            0.0            1.0              1.0               0.0
2            0.0            0.0              1.0               0.0
3            1.0            0.0              0.0               1.0
4            1.0            0.0              0.0               0.0
5            1.0            0.0              0.0               0.0
6            0.0            0.0              0.0               0.0
7            0.0            1.0              0.0               1.0
8            0.0            1.0              0.0               0.0
9            1.0            0.0              0.0               1.0
10           0.0            1.0              0.0               1.0
11           0.0            0.0              0.0               1.0
12           0.0            0.0              1.0               0.0
13           1.0            0.0              0.0               1.0

    Humidity_Normal  Windy_Weak
0               0.0         1.0
1               0.0         0.0
2               0.0         1.0
3               0.0         1.0
4               1.0         1.0
5               1.0         0.0
6               1.0         0.0
7               0.0         1.0
8               1.0         1.0
9               1.0         1.0
10              1.0         0.0
11              0.0         0.0
12              1.0         1.0
13              0.0         0.0
```

In [27]:
```
x_trns.dtypes
```

Out[27]:
```
Outlook_Rain        float64
Outlook_Sunny       float64
Temperature_Hot     float64
Temperature_Mild    float64
Humidity_Normal     float64
Windy_Weak          float64
dtype: object
```

In [29]:
```
le = LabelEncoder()
y_trns=le.fit_transform(y)
y_trns
```

Out[29]:
```
array([0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0])
```

In [45]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_trns,y_trns,test_size=0.2,random_state=42)
```

In [46]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(11, 6) (3, 6) (11,) (3,)
```

In [47]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [53]:
```
#Model Building
dt= DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_train_pred=dt.predict(x_train)
y_test_pred=dt.predict(x_test)
```

In [54]:
```
#train data
```

```
confusion_matrix(y_train,y_train_pred)
```

Out[54]:
```
array([[4, 0],
       [0, 7]])
```

In [56]:
```
#test data
confusion_matrix(y_test,y_test_pred)
```

Out[56]:
```
array([[1, 0],
       [1, 1]])
```

In [57]:
```
#classification report
print(classification_report(y_train,y_train_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       1.00      1.00      1.00         7

    accuracy                           1.00        11
   macro avg       1.00      1.00      1.00        11
weighted avg       1.00      1.00      1.00        11
```

In [59]:
```
print(classification_report(y_test,y_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       1.00      0.50      0.67         2

    accuracy                           0.67         3
   macro avg       0.75      0.75      0.67         3
weighted avg       0.83      0.67      0.67         3
```

Both are low, its an overfitting model

In [63]:
```
from sklearn import tree

plt.figure(figsize=(12,8))
tree.plot_tree(dt,feature_names=x_trns.columns, class_names=['No','Yes'],filled=True)
plt.show()
```