In [1]:

```
#Import necessary libraries

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
from sklearn import tree
```

Desicion Tree Classification

In [2]:

```
#Load the dataset

url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperS
df = pd.read_csv(url)
```

In [3]:

```
df.head()
```

Out[3]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [5]:

```
df = pd.get_dummies(df, columns = ['Sex','BP','Cholesterol'],
            drop_first=True)
```

In [6]:

```
df.head()
```

Out[6]:

	Age	Na_to_K	Drug	Sex_M	BP_LOW	BP_NORMAL	Cholesterol_NORMAL
0	23	25.355	drugY	0	0	0	0
1	47	13.093	drugC	1	1	0	0
2	47	10.114	drugC	1	1	0	0
3	28	7.798	drugX	0	0	1	0
4	61	18.043	drugY	0	1	0	0

In [7]:

```
#Select relevant features

X = df.drop('Drug', axis=1)
y = df['Drug']
```

In [8]:

```
#Split data into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4
```

In [9]:

```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(160, 6)
(160,)
(40, 6)
(40,)
```

In [10]:

```
#Fit the Decision Tree Classifier

model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[10]:

```
DecisionTreeClassifier(random_state=42)
```

In [11]:

```
#Make predictions

y_pred = model.predict(X_test)
```

In [12]:

```
#Evaluate the model

accuracy = accuracy_score(y_test, y_pred)
```

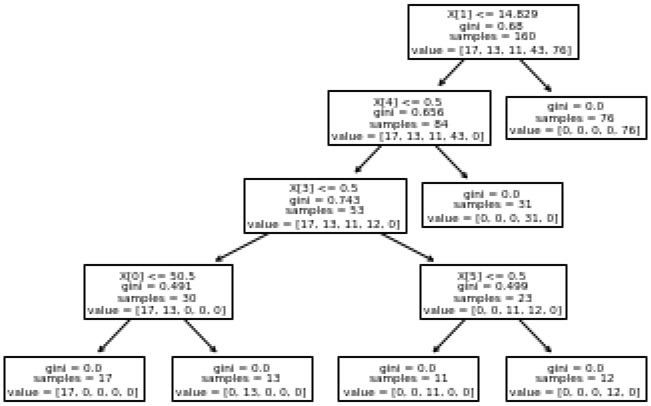
In [13]:

```
print(f"\nAccuracy on the whole dataset: {accuracy}")
```

```
Accuracy on the whole dataset: 1.0
```

In [14]:

```
tree.plot_tree(model)
plt.show()
```



Random Forest Classification

In [15]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_moons
```

In [16]:
```
#Fit the Random Forest Classifier

forest = RandomForestClassifier(n_estimators=100, random_state=0)
forest.fit(X_train, y_train)
```

Out[16]:
```
RandomForestClassifier(random_state=0)
```

In [17]:
```
#Make predictions

y_pred = model.predict(X_test)
```

In [18]:
```
#Evaluate the model

accuracy = accuracy_score(y_test, y_pred)
```

In [19]:
```
print(f"\nAccuracy on the whole dataset: {accuracy}")
```

```
Accuracy on the whole dataset: 1.0
```