In [6]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
Iris=pd.read_csv(r'Downloads\Iris.csv')
Iris
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [7]:
```
Iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [10]:
```
Iris=Iris.drop(columns=['Id'])
Iris
```

Out[10]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [11]:
```
iris.isnull().sum()
```

Out[11]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [12]:
```
iris.shape
```

Out[12]:
```
(150, 6)
```

In [13]:
```
iris['Species'].value_counts()
```

Out[13]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [18]:
```
x=Iris.drop('Species',axis=1)
x
```

Out[18]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [21]:
```
y=Iris['Species']
y
```

Out[21]:
```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

In [22]:
```
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
y=label.fit_transform(y)
y
```

Out[22]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [23]:
```
Iris['Species']=y
Iris
```

Out[23]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

In [25]:
```
Iris['Species'].value_counts()
```

Out[25]:
```
0    50
1    50
2    50
Name: Species, dtype: int64
```

In [26]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
x_train.shape
```

Out[26]:
```
(120, 4)
```

In [27]:
```
x_test.shape
```

Out[27]:
```
(30, 4)
```

In [31]:
```
from sklearn.tree import DecisionTreeClassifier
c=DecisionTreeClassifier(criterion='entropy')
c.fit(x_train,y_train)
```

Out[31]:
```
DecisionTreeClassifier(criterion='entropy')
```

In [32]:
```
y_pred=c.predict(x_test)
y_pred
```

Out[32]:
```
array([2, 0, 2, 0, 2, 2, 0, 0, 2, 0, 0, 2, 0, 0, 2, 1, 1, 2, 2, 2, 2, 0,
       2, 0, 1, 2, 1, 0, 1, 2])
```

In [33]:
```
from sklearn.metrics import accuracy_score
print("Accuracy:",accuracy_score(y_pred,y_test)*100)
```

```
Accuracy: 96.66666666666667
```

In [36]:
```
from sklearn.metrics import classification_report
cr=classification_report(y_pred,y_test)
cr
```

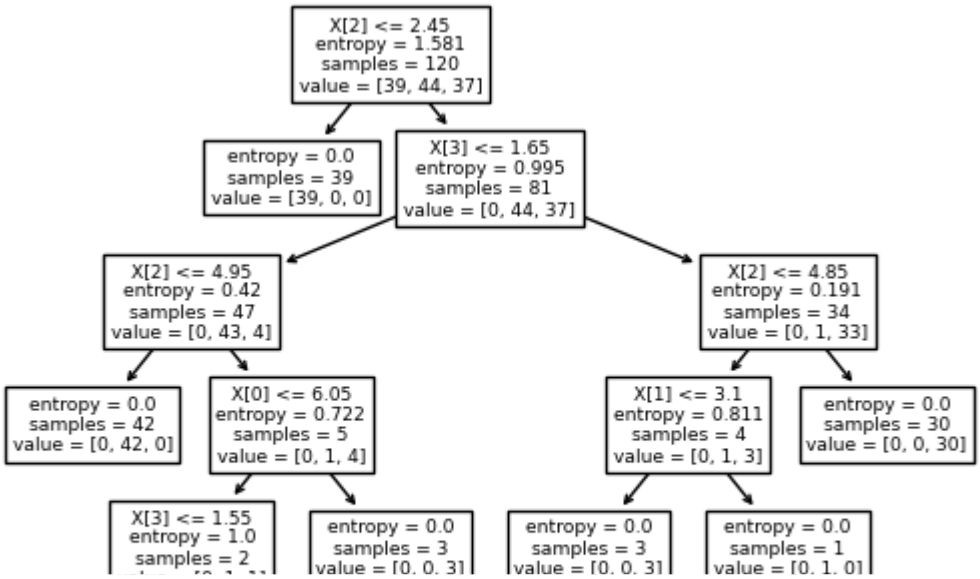
Out[36]:
```
'              precision    recall  f1-score   support\n\n           0
1.00      1.00      1.00        11\n           1       0.83      1.00      0.
91         5\n           2       1.00      0.93      0.96        14\n\n    ac
curacy                           0.97        30\n   macro avg       0.94
0.98      0.96        30\nweighted avg       0.97      0.97      0.97
30\n'
```

In [37]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_pred,y_test)
cm
```

Out[37]:
```
array([[11,  0,  0],
       [ 0,  5,  0],
       [ 0,  1, 13]], dtype=int64)
```

In [48]:
```
from sklearn import tree
tree.plot_tree(c)
plt.show
```

Out[48]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [51]:
```
x=[[5.2,3.6,1.5,0]]
y_pred=c.predict(x)
y_pred
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but DecisionTreeClassifier was fitted wi
th feature names
  warnings.warn(
```

Out[51]:
```
array([0])
```

In [ ]:
```

```