Usama Arif Roll No 14

In [177]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder,MinMaxScaler
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

reading a csv file

In [178]:
```
iris=pd.read_csv('iris.csv')
iris.head(10)

```

Out[178]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

seprating the data

In [179]:
```
x=iris.drop('species',axis=1)
y=iris['species']
```

In [180]:
```
scaler=MinMaxScaler()
x_scaled=scaler.fit_transform(x)
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

fitting the model and training the data

In [186]:
```
clf=DecisionTreeClassifier()
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2)

clf.fit(x_train,y_train);
```

evaluating the model

In [187]:
```
accuracy_score=clf.score(x_test,y_test)
print('accuracy score',accuracy_score )
```

```
accuracy score 0.9666666666666667
```

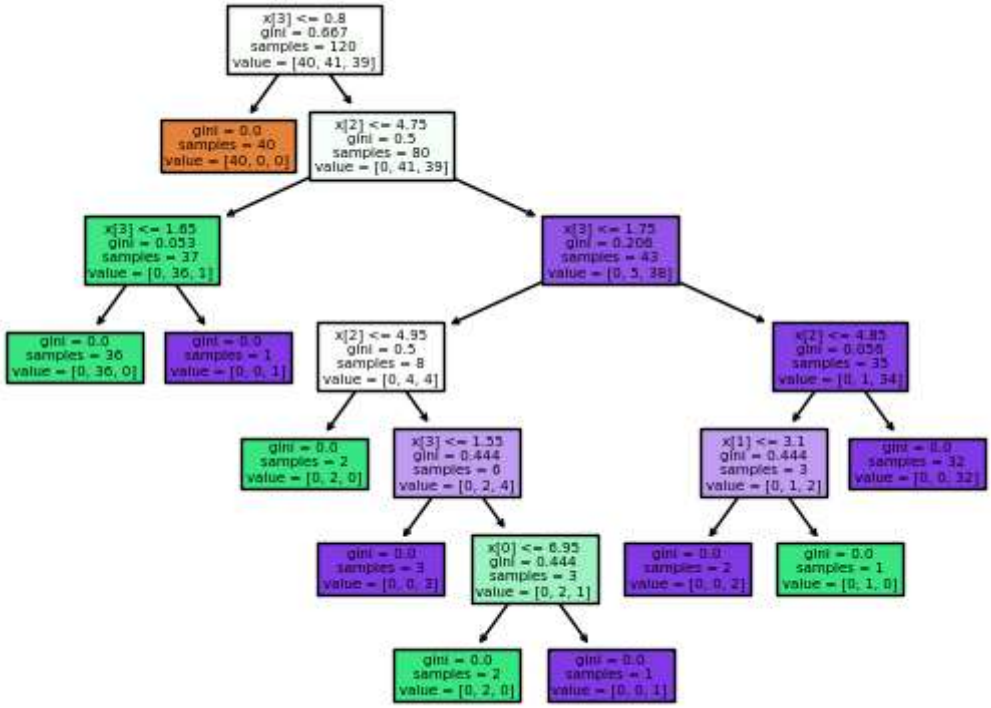
making the pridictions

In [188]:
```
y_preds=clf.predict(x_test)
y_preds
```

Out[188]:
```
array(['setosa', 'versicolor', 'setosa', 'versicolor', 'setosa', 'setosa',
       'setosa', 'virginica', 'versicolor', 'setosa', 'versicolor',
       'virginica', 'versicolor', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'versicolor', 'setosa', 'setosa', 'virginica', 'setosa',
       'versicolor', 'virginica', 'setosa', 'versicolor', 'setosa',
       'setosa', 'setosa', 'virginica'], dtype=object)
```

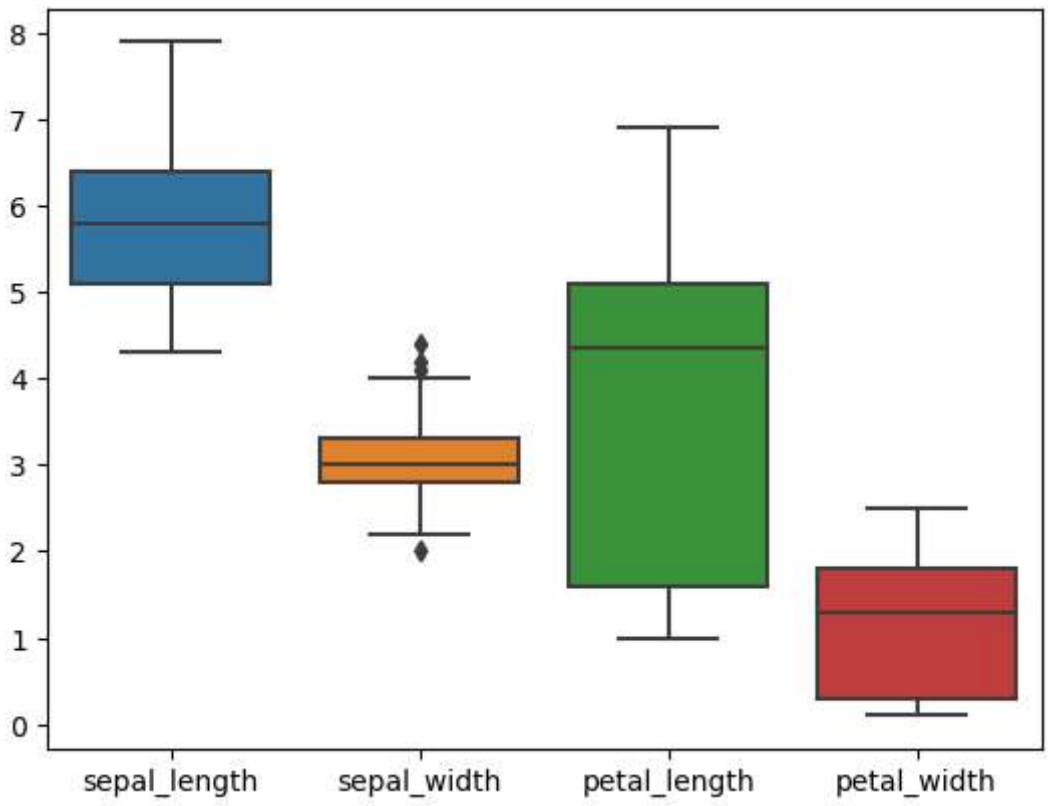
Plotting the tree at large

In [88]:
```
1 plot_tree(clf,filled=True);
2
```



Identify outliers using a box plot

In [100]:
```
1 sns.boxplot(data=iris,saturation=0.75,width=0.8);
2
```



Entropy

In [77]:
```
1 total_species=len(y)
2 class_counts=y.value_counts()
3 entropy=0
4 for count in class_counts:
5     probability=count/total_species
6     entropy=entropy-probability*math.log2(probability)
7 print(f'Entropy: {entropy}')
```

```
Entropy: 1.584962500721156
```

Gini Calculation

In [172]:

```python
import numpy as np

def gini_index(labels):
    classes, count = np.unique(labels, return_counts=True)

    prob = count / len(labels)
    gini = 1 - np.sum(prob ** 2)
    return gini

gini = gini_index(y)
print("Gini Index:", gini)
```

```
Gini Index: 0.6666666666666667
```

In [ ]:

```python

```