BaggingClassifier

In [1]:
```
import pandas as pd
import seaborn as sns
```

In [2]:
```
iris=sns.load_dataset('iris')
```

In [3]:
```
iris
```

Out[3]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [4]:
```
x=iris[['sepal_length','sepal_width','petal_length','petal_width']]
```

In [5]:
```
y=iris['species']
```

In [6]:
```
from sklearn.model_selection import train_test_split
```

In [7]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=10)
```

In [8]:
```
#from sklearn.tree import BaggingClassifier
```

In [9]:
```
from sklearn.ensemble import BaggingClassifier
```

In [10]:
```
#model=BaggingClassifier(criterion='entropy',max_depth=3)
```

In [11]:
```
model=BaggingClassifier()
model.fit(x_train,y_train)
```

Out[11]:
```
BaggingClassifier()
```

In [12]:
```
y_pred=model.predict(x_test)
```

In [13]:
```
y_pred
```

Out[13]:
```
array(['versicolor', 'virginica', 'setosa', 'versicolor', 'setosa',
       'versicolor', 'versicolor', 'versicolor', 'setosa', 'versicolor',
       'versicolor', 'virginica', 'versicolor', 'setosa', 'setosa',
       'virginica', 'versicolor', 'setosa', 'setosa', 'setosa',
       'virginica', 'virginica', 'virginica', 'setosa', 'versicolor',
       'setosa', 'versicolor', 'versicolor', 'versicolor', 'virginica',
       'versicolor', 'versicolor', 'versicolor', 'virginica', 'virginica',
       'setosa', 'virginica', 'virginica', 'virginica', 'virginica',
       'setosa', 'setosa', 'versicolor', 'setosa', 'versicolor'],
      dtype=object)
```

In [14]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        14
  versicolor       0.94      1.00      0.97        17
   virginica       1.00      0.93      0.96        14

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

In [15]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [16]:
```
model = KNeighborsClassifier(n_neighbors=1)
model.fit(x_train,y_train)
```

Out[16]:
```
KNeighborsClassifier(n_neighbors=1)
```

In [17]:
```
from sklearn.linear_model import LogisticRegression
```

In [18]:
```
model = LogisticRegression()
model.fit(x_train, y_train)
```

Out[18]:
```
LogisticRegression()
```

In [19]:
```
y_pred=model.predict(x_test)
```

In [20]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        14
  versicolor       1.00      1.00      1.00        17
   virginica       1.00      1.00      1.00        14

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

In [21]:
```
model = KNeighborsClassifier(n_neighbors=1)
model.fit(x_train,y_train)
```

Out[21]:

KNeighborsClassifier

```
KNeighborsClassifier(n_neighbors=1)
```

In [22]:
```
y_pred=model.predict(x_test)
```

In [23]:
```
y_pred
```

Out[23]:
```
array(['versicolor', 'virginica', 'setosa', 'versicolor', 'setosa',
       'versicolor', 'virginica', 'versicolor', 'setosa', 'versicolor',
       'versicolor', 'virginica', 'versicolor', 'setosa', 'setosa',
       'virginica', 'versicolor', 'setosa', 'setosa', 'setosa',
       'virginica', 'virginica', 'virginica', 'setosa', 'versicolor',
       'setosa', 'versicolor', 'versicolor', 'versicolor', 'virginica',
       'versicolor', 'versicolor', 'virginica', 'virginica', 'virginica',
       'setosa', 'virginica', 'virginica', 'virginica', 'virginica',
       'setosa', 'setosa', 'versicolor', 'setosa', 'versicolor'],
      dtype=object)
```

In [24]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        14
  versicolor       1.00      0.94      0.97        17
   virginica       0.93      1.00      0.97        14

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

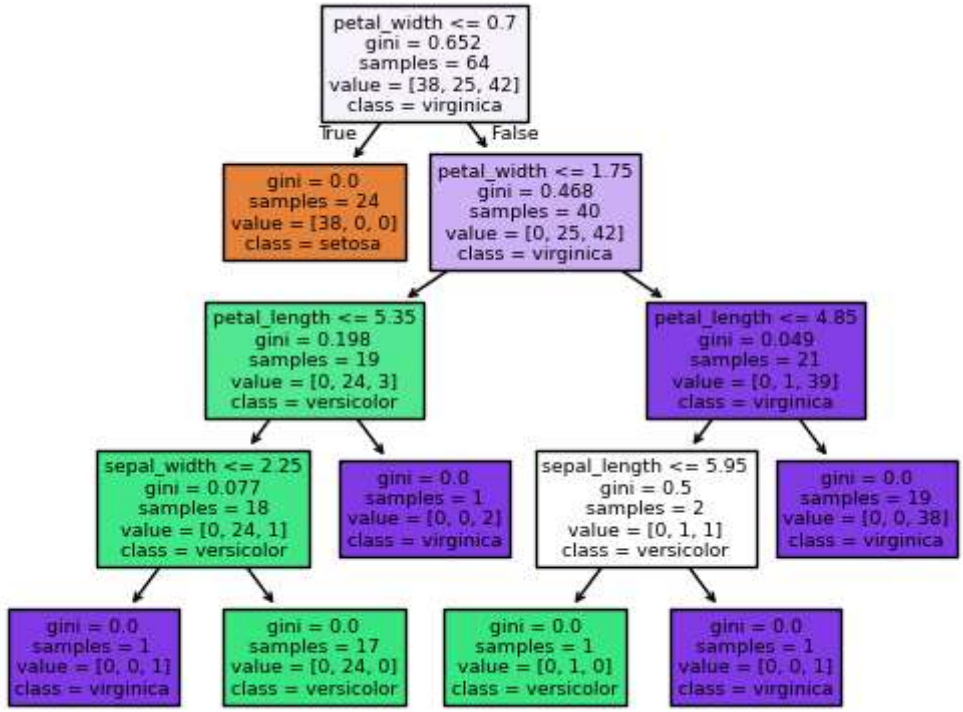
In [25]:
```
#from sklearn.tree import DecisionTreeClassifier
#base_model = DecisionTreeClassifier()
```

In [26]:
```
model=BaggingClassifier()
```

In [27]:
```
model.fit(x_train,y_train)
```

Out[27]:
```
BaggingClassifier()
```

In [35]:
```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
plot_tree(model.estimators_[0],feature_names=x.columns, class_names=model.classes_,
plt.show()
```



In [ ]:
```
```