Prediction using Decision Tree

Done by

KEVAN P JOHN

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from matplotlib import pyplot as plt
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
iris=load_iris()
df=pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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

Shape

In [8]:
```
df.shape
```

Out[8]:
```
(150, 4)
```

Preparing Data for creating a model

In [9]:
```
x=iris.data
y=iris.target
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
print("x train data", x_train.shape)
print("x test data", x_test.shape)
print("y train data", y_train.shape)
print("y test data", y_test.shape)
```

```
x train data (120, 4)
x test data (30, 4)
y train data (120,)
y test data (30,)
```

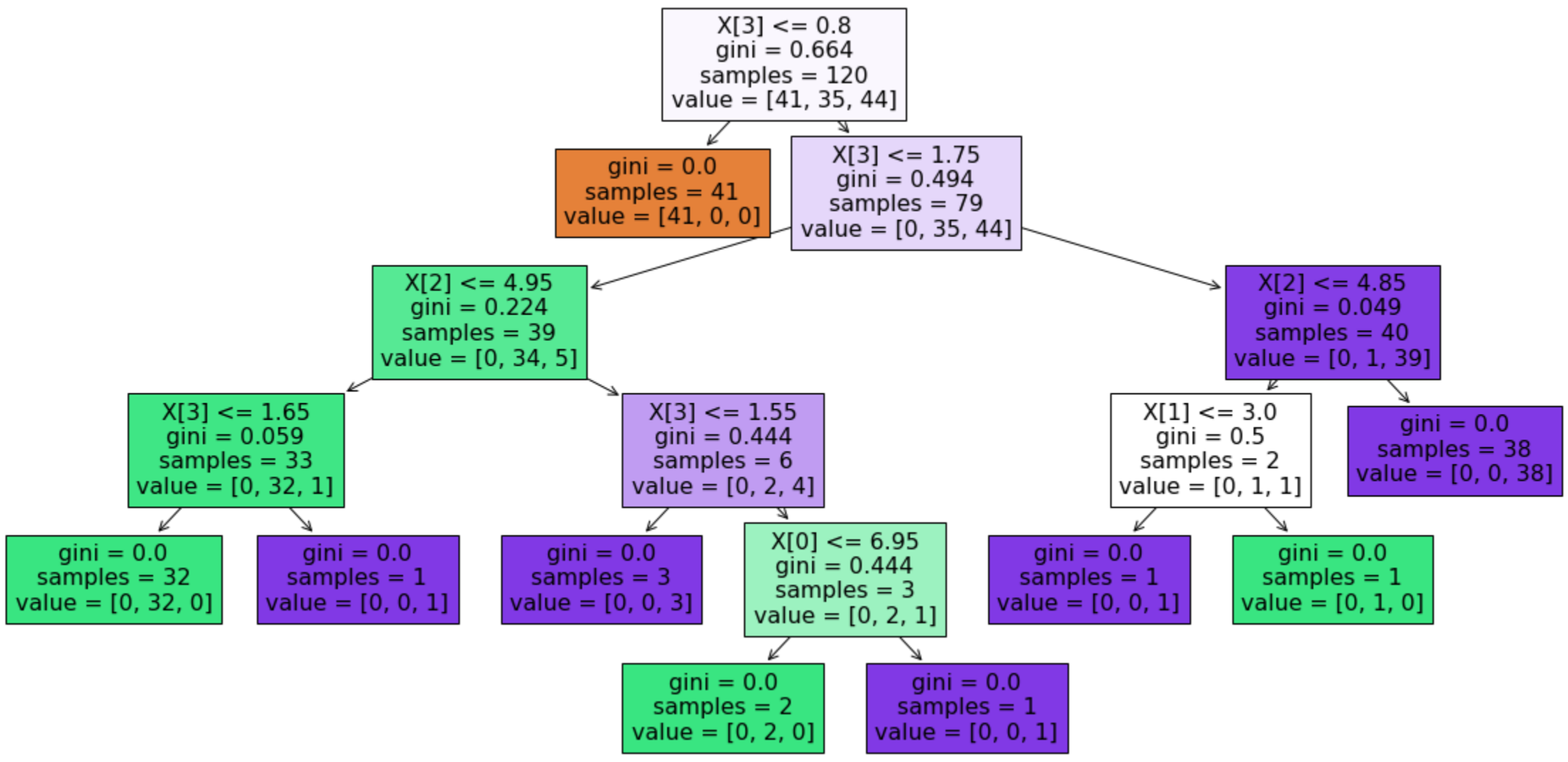
In [10]:
```
model=DecisionTreeClassifier()
model.fit(x_train,y_train)
print("model is trained")
```

```
model is trained
```

visualizing

In [11]:
```
plt.figure(figsize=(20,10))
tree.plot_tree(model, filled=True)
```

Out[11]:
```
[Text(558.0, 498.3, 'X[3] <= 0.8\ngini = 0.664\nsamples = 120\nvalue = [41, 35, 44]'),
 Text(472.15384615384613, 407.70000000000005, 'gini = 0.0\nsamples = 41\nvalue = [41, 0, 0]'),
 Text(643.8461538461538, 407.70000000000005, 'X[3] <= 1.75\ngini = 0.494\nsamples = 79\nvalue = [0, 35, 44]'),
 Text(343.38461538461536, 317.1, 'X[2] <= 4.95\ngini = 0.224\nsamples = 39\nvalue = [0, 34, 5]'),
 Text(171.69230769230768, 226.5, 'X[3] <= 1.65\ngini = 0.059\nsamples = 33\nvalue = [0, 32, 1]'),
 Text(85.84615384615384, 135.89999999999998, 'gini = 0.0\nsamples = 32\nvalue = [0, 32, 0]'),
 Text(257.53846153846155, 135.89999999999998, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(515.0769230769231, 226.5, 'X[3] <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 2, 4]'),
 Text(429.23076923076917, 135.89999999999998, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
 Text(600.9230769230769, 135.89999999999998, 'X[0] <= 6.95\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),
 Text(515.0769230769231, 45.299999999999955, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
 Text(686.7692307692307, 45.299999999999955, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(944.3076923076923, 317.1, 'X[2] <= 4.85\ngini = 0.049\nsamples = 40\nvalue = [0, 1, 39]'),
 Text(858.4615384615383, 226.5, 'X[1] <= 3.0\ngini = 0.5\nsamples = 2\nvalue = [0, 1, 1]'),
 Text(772.6153846153845, 135.89999999999998, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(944.3076923076923, 135.89999999999998, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(1030.1538461538462, 226.5, 'gini = 0.0\nsamples = 38\nvalue = [0, 0, 38]')]
```



Testing the model created

In [12]:
```
prediction=model.predict(x_test)
prediction
df=pd.DataFrame({'predicted value':prediction,'actual value':y_test})
df['label']=df['predicted value'].replace(dict(enumerate(iris.target_names)))
df.head()
```

Out[12]:

	predicted value	actual value	label
0	2	2	virginica
1	0	0	setosa
2	2	2	virginica
3	1	1	versicolor
4	1	1	versicolor

Accuracy

In [13]:
```
score=accuracy_score(prediction,y_test)
print(f"Accuracy score is {score} i.e. {score*100}%")
```

```
Accuracy score is 1.0 i.e. 100.0%
```