

```
In [3]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
In [4]: df = pd.read_csv("iris.csv")
print(df.head())
```

	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

```
In [5]: df.describe()
```

```
Out[5]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
In [6]: df.shape
```

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

```
In [19]: df1=df.drop(columns=["Id"], axis=1)
df1
df1.boxplot()
plt.show()
```

```
In [20]: df1.head()
```

```
Out[20]:
```

	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

```
In [21]: X = df1.drop(columns=['Species'], axis=1)
y = df['Species']
```

```
In [22]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
In [37]: X.shape , y.shape
```

```
Out[37]: ((150, 4), (150,))
```

```
In [23]: model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

```
Out[23]:
```

▼ DecisionTreeClassifier

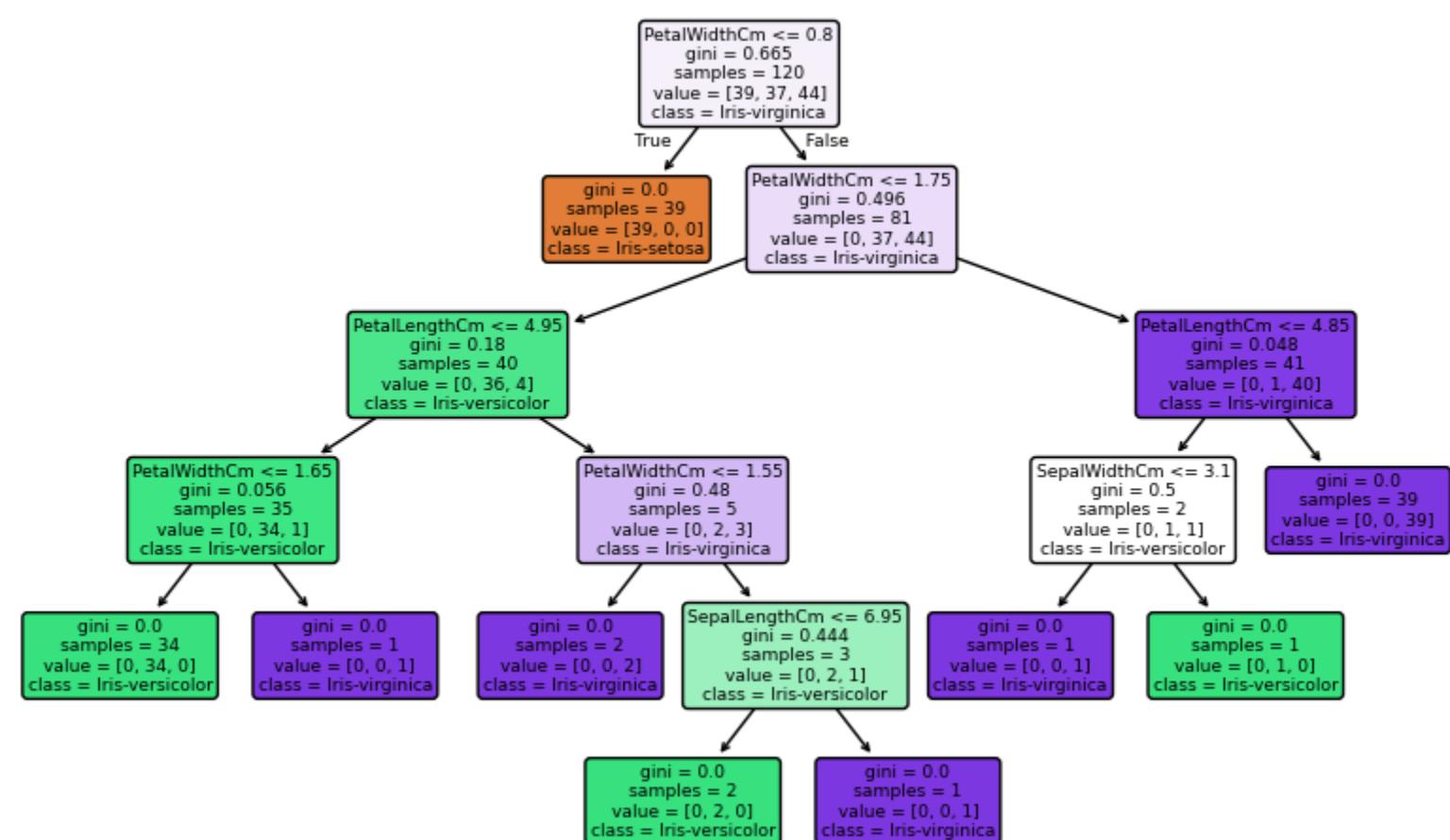
► Parameters

```
In [31]: y_pred = model.predict(X_test)
```

```
In [35]: print("Accuracy:", accuracy_score(y_test, y_pred))
Accuracy: 1.0
```

```
In [29]: from sklearn.tree import plot_tree
plt.figure(figsize=(10,6))
plot_tree(model,
          filled=True,
          feature_names=X.columns,
          class_names=model.classes_,
          rounded=True)
plt.title("Decision Tree for Iris Dataset")
plt.show()
```

Decision Tree for Iris Dataset



In [ ]: