

```
import pandas as pd
```

```
df = pd.read_csv("https://raw.githubusercontent.com/AmenaNajeeb/Data/master/Iris.csv")
```

```
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

```
df.shape
```

```
(150, 6)
```

```
df = df.drop(["Id"],axis=1)
```

```
df.isnull().sum()
```

```
SepalLengthCm    0
SepalWidthCm      0
PetalLengthCm     0
PetalWidthCm      0
Species           0
dtype: int64
```

```
df.dtypes
```

```
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species          object
dtype: object
```

```
from sklearn import preprocessing
```

```
le = preprocessing.LabelEncoder()
```

```
df["Species"]=le.fit_transform(df["Species"])
```

```
df.dtypes
```

```
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species          int64
dtype: object
```

```
y = df["Species"]
```

```
x = df.drop(["Species"],axis=1)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.15,random_state=3)
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
model = DecisionTreeClassifier(max_leaf_nodes=10)
```

```
model.fit(x_train,y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_leaf_nodes=10)
```

```
y_pred=model.predict(x_test)
```

```
model.score(x_train,y_train)
```

```
1.0
```

```
model.score(x_test,y_test)
```

```
0.9565217391304348
```

```
from sklearn.metrics import confusion_matrix
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[9, 0, 0],
       [0, 5, 1],
       [0, 0, 8]])
```

```
from sklearn.metrics import classification_report
```

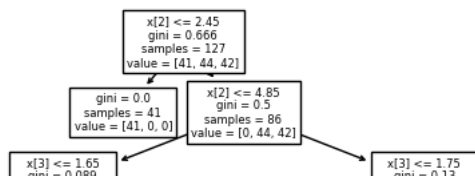
```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	9
1	1.00	0.83	0.91	6
2	0.89	1.00	0.94	8
accuracy			0.96	23
macro avg	0.96	0.94	0.95	23
weighted avg	0.96	0.96	0.96	23

```
from sklearn import tree
```

```
tree.plot_tree(model)
```

```
[Text(0.36363636363636365, 0.9285714285714286, 'x[2] <= 2.45\ngini = 0.666\nsamples = 127\nvalue = [41, 44, 42]'),
 Text(0.2727272727272727, 0.7857142857142857, 'gini = 0.0\nsamples = 41\nvalue = [41, 0, 0]'),
 Text(0.45454545454545453, 0.7857142857142857, 'x[2] <= 4.85\ngini = 0.5\nsamples = 86\nvalue = [0, 44, 42]'),
 Text(0.18181818181818182, 0.6428571428571429, 'x[3] <= 1.65\ngini = 0.089\nsamples = 43\nvalue = [0, 41, 2]'),
 Text(0.09090909090909091, 0.5, 'gini = 0.0\nsamples = 40\nvalue = [0, 40, 0]'),
 Text(0.2727272727272727, 0.5, 'x[1] <= 3.1\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
 Text(0.18181818181818182, 0.35714285714285715, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.36363636363636365, 0.35714285714285715, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.7272727272727273, 0.6428571428571429, 'x[3] <= 1.75\ngini = 0.13\nsamples = 43\nvalue = [0, 3, 40]'),
 Text(0.6363636363636364, 0.5, 'x[1] <= 2.65\ngini = 0.49\nsamples = 7\nvalue = [0, 3, 4]'),
 Text(0.5454545454545454, 0.35714285714285715, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.7272727272727273, 0.35714285714285715, 'x[2] <= 5.05\ngini = 0.48\nsamples = 5\nvalue = [0, 3, 2]'),
 Text(0.6363636363636364, 0.21428571428571427, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
 Text(0.8181818181818182, 0.21428571428571427, 'x[0] <= 6.15\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
 Text(0.7272727272727273, 0.07142857142857142, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.9090909090909091, 0.07142857142857142, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.8181818181818182, 0.5, 'gini = 0.0\nsamples = 36\nvalue = [0, 0, 36]')]
```



✓ 0s completed at 12:29 PM

