1.10.1. Classification

`DecisionTreeClassifier` is a class capable of performing multi-class classification on a dataset.

As with other classifiers, `DecisionTreeClassifier` takes as input two arrays: an array X, sparse or dense, of size `[n_samples, n_features]` holding the training samples, and an array Y of integer values, size `[n_samples]`, holding the class labels for the training samples:

```
from sklearn import tree
X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, Y)
```

After being fitted, the model can then be used to predict the class of samples:

```
 clf.predict([[2., 2.]])
out: array([1])
```

Alternatively, the probability of each class can be predicted, which is the fraction of training samples of the same class in a leaf:

```
 clf.predict\_proba([[2., 2.]])
out: array([[0., 1.]])
```

`DecisionTreeClassifier` is capable of both binary (where the labels are [-1, 1]) classification and multiclass (where the labels are [0, . . . , K-1]) classification.

Using the Iris dataset, we can construct a tree as follows:

```
 from sklearn.datasets import load_iris
from sklearn import tree
X, y = load_iris(return\_X\_y=True)
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, y)
```

Once trained, you can plot the tree with the `plot_tree` function:

```
 tree.plot\_tree(clf)
```