Decision Trees

Iris Example

sklearn.tree.DecisionTreeClassifier

- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Example

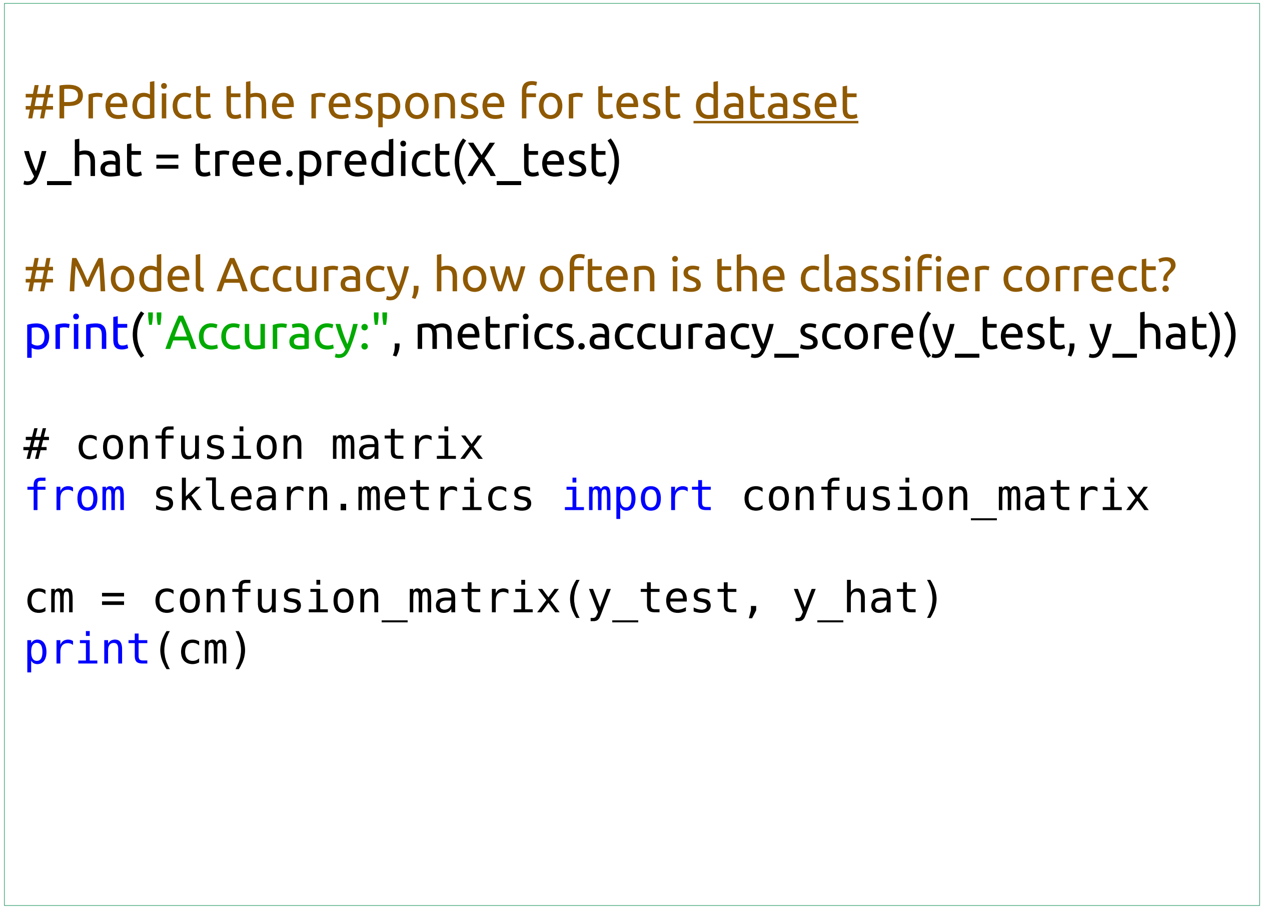
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import pandas as pd

df = pd.read_csv('data/iris.csv')
df.head()

X = df[["sepal_length","sepal_width","petal_length","petal_width"]]
y = df["species"]

# 125 training and 25 test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=25,
                                        random_state=1, stratify=y)
tree = DecisionTreeClassifier()
tree.fit(X_train,y_train)
```

Example (cont)



```
#Predict the response for test dataset
y_hat = tree.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_hat))

# confusion matrix
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_hat)
print(cm)
```

Output

```
Accuracy: 0.92

[[8 0 0]
 [0 8 1]
 [0 1 7]]
```

Example

```
# conda install pydotplus
# (installs graphviz)
from sklearn.tree import export_graphviz
from io import StringIO
from IPython.display import Image
import pydotplus

feature_names = ["sepal_length","sepal_width","petal_length","petal_width"]
target_names = ["setosa", "versicolor", "virginica"]

dot_data = StringIO()
export_graphviz(tree, out_file=dot_data,
            filled=True, rounded=True,
            special_characters=True, feature_names = feature_names,
            class_names = target_names)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('plots/iris.png')
Image(graph.create_png())
```

train_test_split

- In train_test_split, random_state=1 is sets the seed of the random number generator.
- This ensures reproducability of results.
- stratify = y means that the split maintains the proportions of each value of y between the training and test data.
- This is important, especially for small data sets.

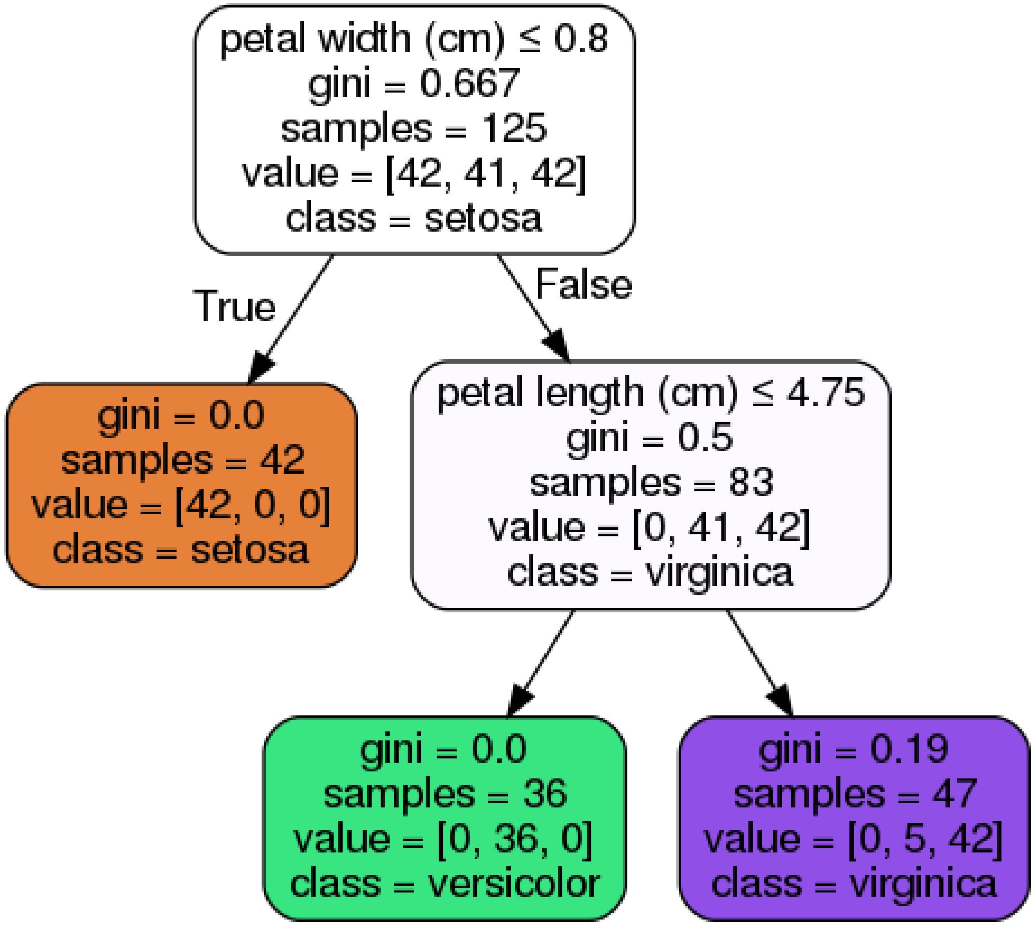
Confusion Matrix

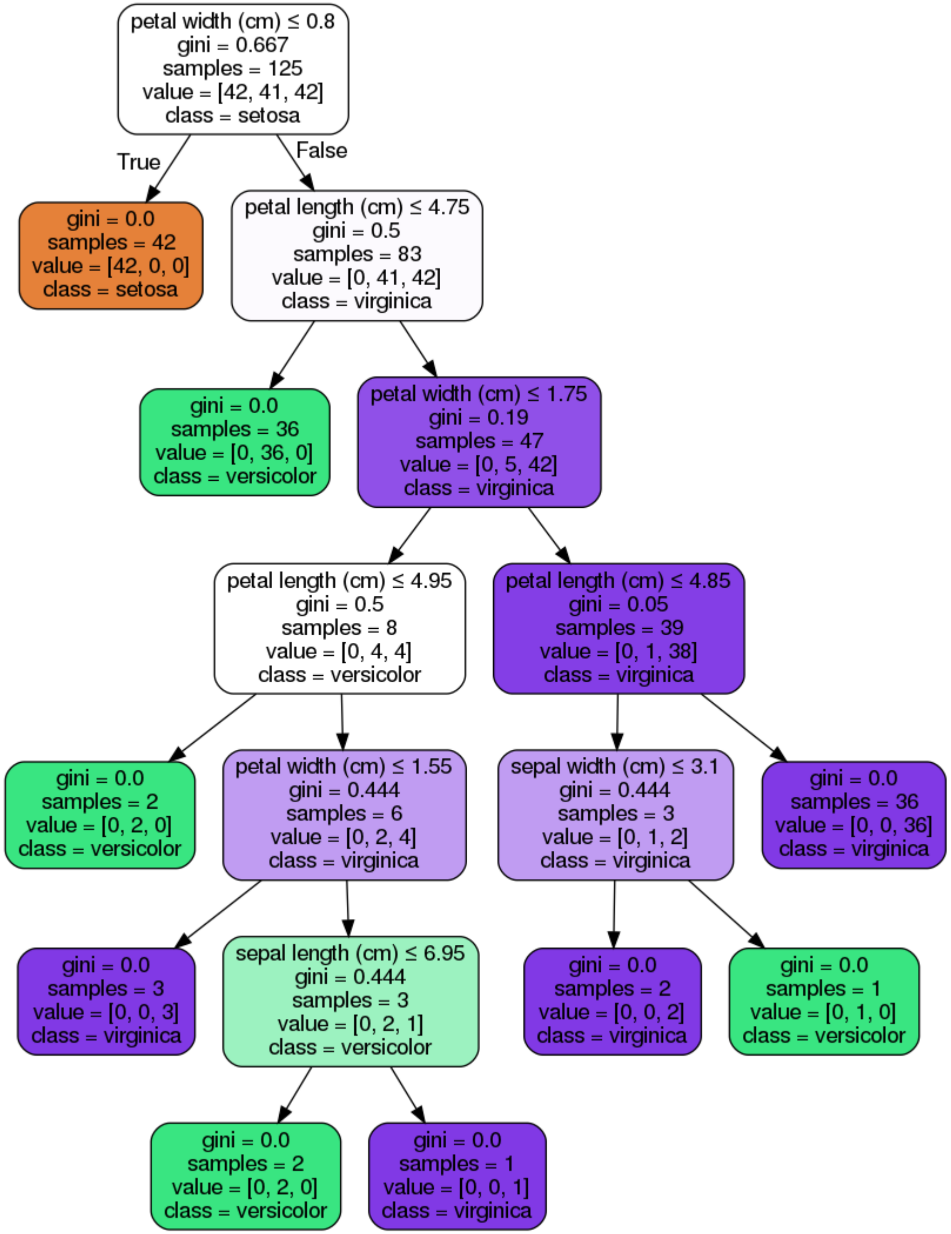
- It is always easier to split setosa, and harder to classify virginica and versicolor.
- This is seen in the confusion matrix.
- Two from 25 instances have been classified incorrectly. This is an error rate of 8% giving an accuracy of 92%.
-

The Decision Tree

- The max_depth parameter controls the depth of the decision tree.
- It defaults to None (no maximum depth)
- max-depth=None can lead to overfitting.
- If max_depth is too small, this can lead to underfitting.
- This is an example of a meta-parameter. It can only be set by looking at the performance on unseen validation (test) data.

max_depth = 2


petal width (cm) ≤ 0.8
gini = 0.667
samples = 125
value = [42, 41, 42]
class = setosa
True
False
gini = 0.0
samples = 42
value = [42, 0, 0]
class = setosa
petal length (cm) ≤ 4.75
gini = 0.5
samples = 83
value = [0, 41, 42]
class = virginica
gini = 0.0
samples = 36
value = [0, 36, 0]
class = versicolor
gini = 0.19
samples = 47
value = [0, 5, 42]
class = virginica


petal width (cm) ≤ 0.8
gini = 0.667
samples = 125
value = [42, 41, 42]
class = setosa
True
False
gini = 0.0
samples = 42
value = [42, 0, 0]
class = setosa
petal length (cm) ≤ 4.75
gini = 0.5
samples = 83
value = [0, 41, 42]
class = virginica
gini = 0.0
samples = 36
value = [0, 36, 0]
class = versicolor
petal width (cm) ≤ 1.75
gini = 0.19
samples = 47
value = [0, 5, 42]
class = virginica
petal length (cm) ≤ 4.95
gini = 0.5
samples = 8
value = [0, 4, 4]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.05
samples = 39
value = [0, 1, 38]
class = virginica
gini = 0.0
samples = 2
value = [0, 2, 0]
class = versicolor
petal width (cm) ≤ 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
sepal width (cm) ≤ 3.1
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
gini = 0.0
samples = 36
value = [0, 0, 36]
class = virginica
gini = 0.0
samples = 3
value = [0, 0, 3]
class = virginica
sepal length (cm) ≤ 6.95
gini = 0.444
samples = 3
value = [0, 2, 1]
class = versicolor
gini = 0.0
samples = 2
value = [0, 0, 2]
class = virginica
gini = 0.0
samples = 1
value = [0, 1, 0]
class = versicolor
gini = 0.0
samples = 2
value = [0, 2, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica