Lecture 7
Decision Trees

GEOL 4397: Data analytics and machine learning for geoscientists

Jiajia Sun, Ph.D.

Feb. 19th, 2019

UNIVERSITY of HOUSTON
YOU ARE THE PRIDE
EARTH AND ATMOSPHERIC SCIENCES

Agenda

- Decision Trees: concepts
- Decision Tree regression
- Implementation in Scikit-learn

An example

Suppose you walk into a cell phone store.

Shopkeeper asks,"How can I help you Ma'am?"

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

Shopkeeper asks,"How can I help you Ma'am?"
"I am looking for a cell phone"

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

Shopkeeper asks,"How can I help you Ma'am?"
"I am looking for a cell phone"
"You are at the right place, we have over 300 different types of cell phones, what kind of phone would you like to buy today?"
Decision paralysis hits you, totally confused among so many choices of phones you go blank!

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

Shopkeeper asks,"How can I help you Ma'am?"
"I am looking for a cell phone"
"You are at the right place, we have over 300 different types of cell phones, what kind of phone would you like to buy today?"
Decision paralysis hits you, totally confused among so many choices of phones you go blank!

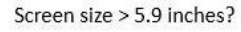
Let us see how a decision tree can help you!

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

"Let me help you choose a phone ma'am. What screen size would you like?"

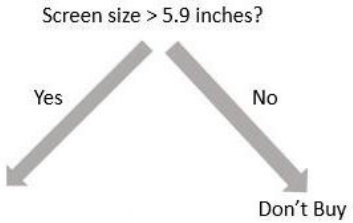


https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

"Let me help you choose a phone ma'am. What screen size would you like?"
"Umm… larger than 5.9 inches"

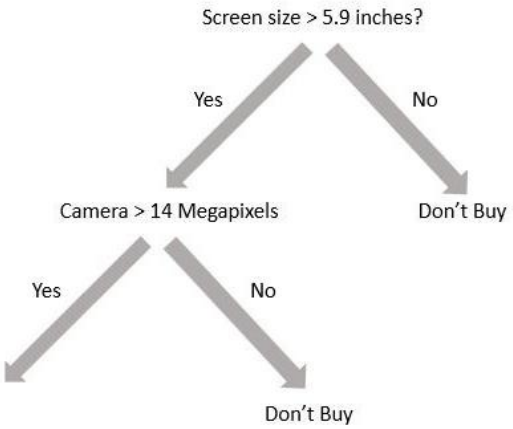


https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

"Let me help you choose a phone ma'am. What screen size would you like?"
"Umm… larger than 5.9 inches"
"Perfect, and how about the camera?"
"Definitely more than 14 Megapixels"

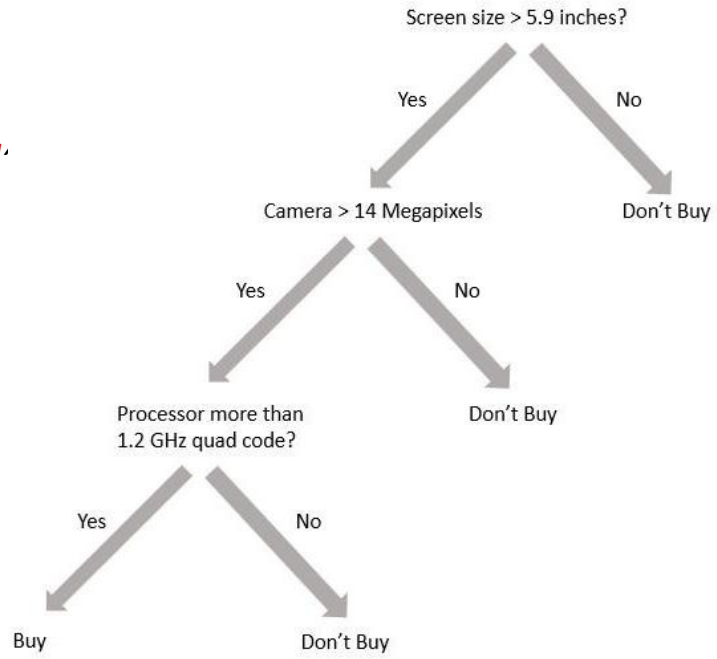


https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

An example

Suppose you walk into a cell phone store.

"Let me help you choose a phone ma'am. What screen size would you like?"
"Umm… larger than 5.9 inches"
"Perfect, and how about the camera?"
"Definitely more than 14 Megapixels"
"Alright, and any preferences on the processor?"
"I want a quad core processor with at least 1.2 GHz speed"



You can add more branches by answering/asking more questions.
This is called a decision tree.

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- Decision trees are everywhere. You use them every day knowingly or unknowingly.
- When you say, 'if it is raining, I will bring an umbrella', you've just constructed a simple decision tree.

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- Decision trees are everywhere. You use them every day knowingly or unknowingly.
- When you say, 'if it is raining, I will bring an umbrella', you've just constructed a simple decision tree.

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- What if it is windy and raining outside? You would rather take a rain jacket instead. You decision tree would look like

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- What if it is windy and raining outside? You would rather take a rain jacket instead. You decision tree would look like

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- What if it is windy and raining outside? You would rather take a rain jacket instead.
- What if it is snowing? Or the wind is too strong?

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- What if it is windy and raining outside? You would rather take a rain jacket instead.

- What if it is snowing? Or the wind is too strong? You can keep adding conditions to this tree and it will keep growing with more branches to handle more situations.

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Another example

- What if it is windy and raining outside? You would rather take a rain jacket instead.

- What if it is snowing? Or the wind is too strong? You can keep adding conditions to this tree and it will keep growing with more branches to handle more situations.
- It also allows you to see exactly how a decision is reached.

https://medium.com/x8-the-ai-community/decision-trees-an-intuitive-introduction-86c2b39c1a6c?cHa=true

Decision Trees

- Ask a simple yes/no question at each level (or, depth) Very intuitive

Decision Trees

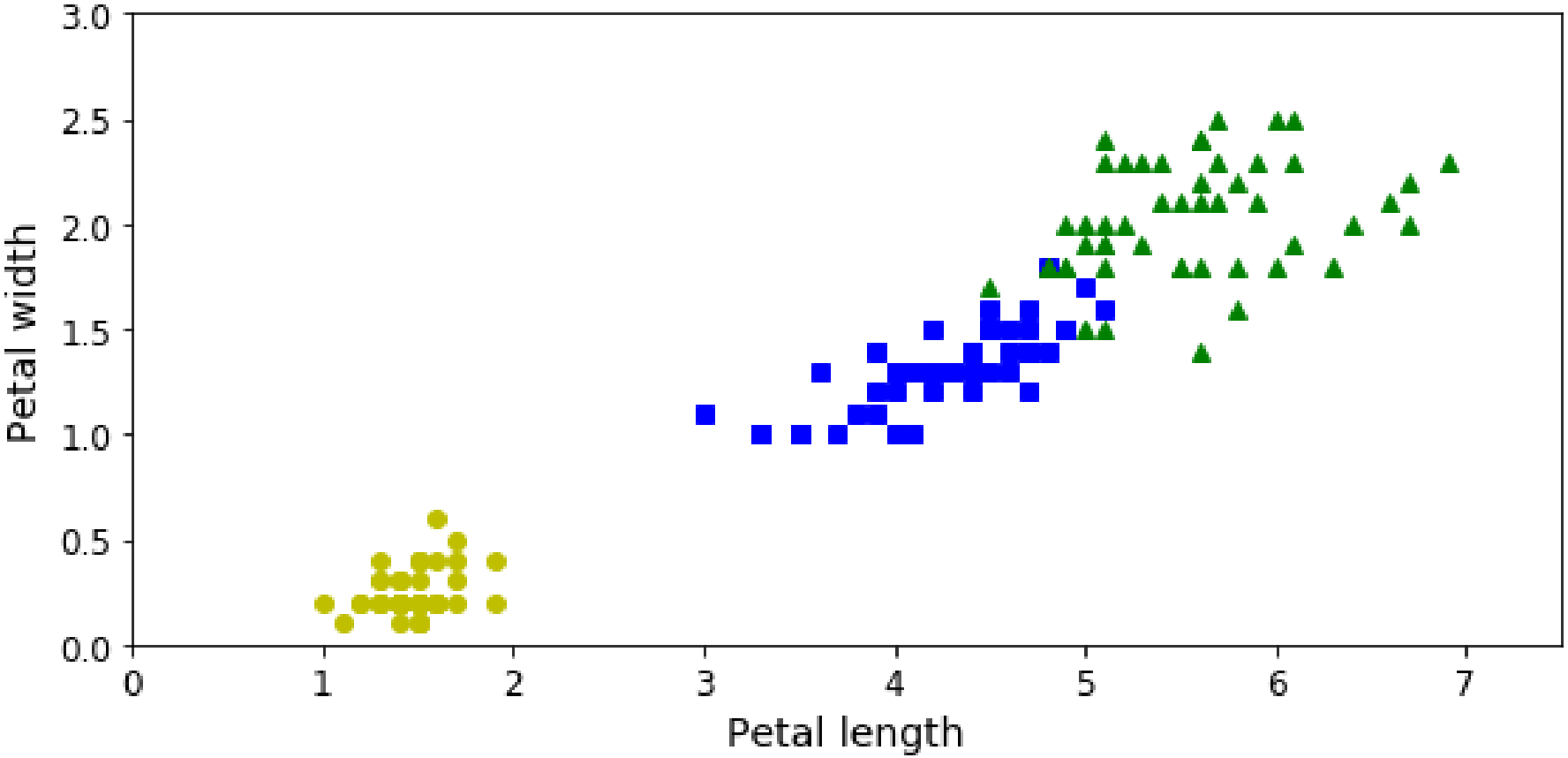
- Ask a simple yes/no question at each level (or, depth) Very intuitive

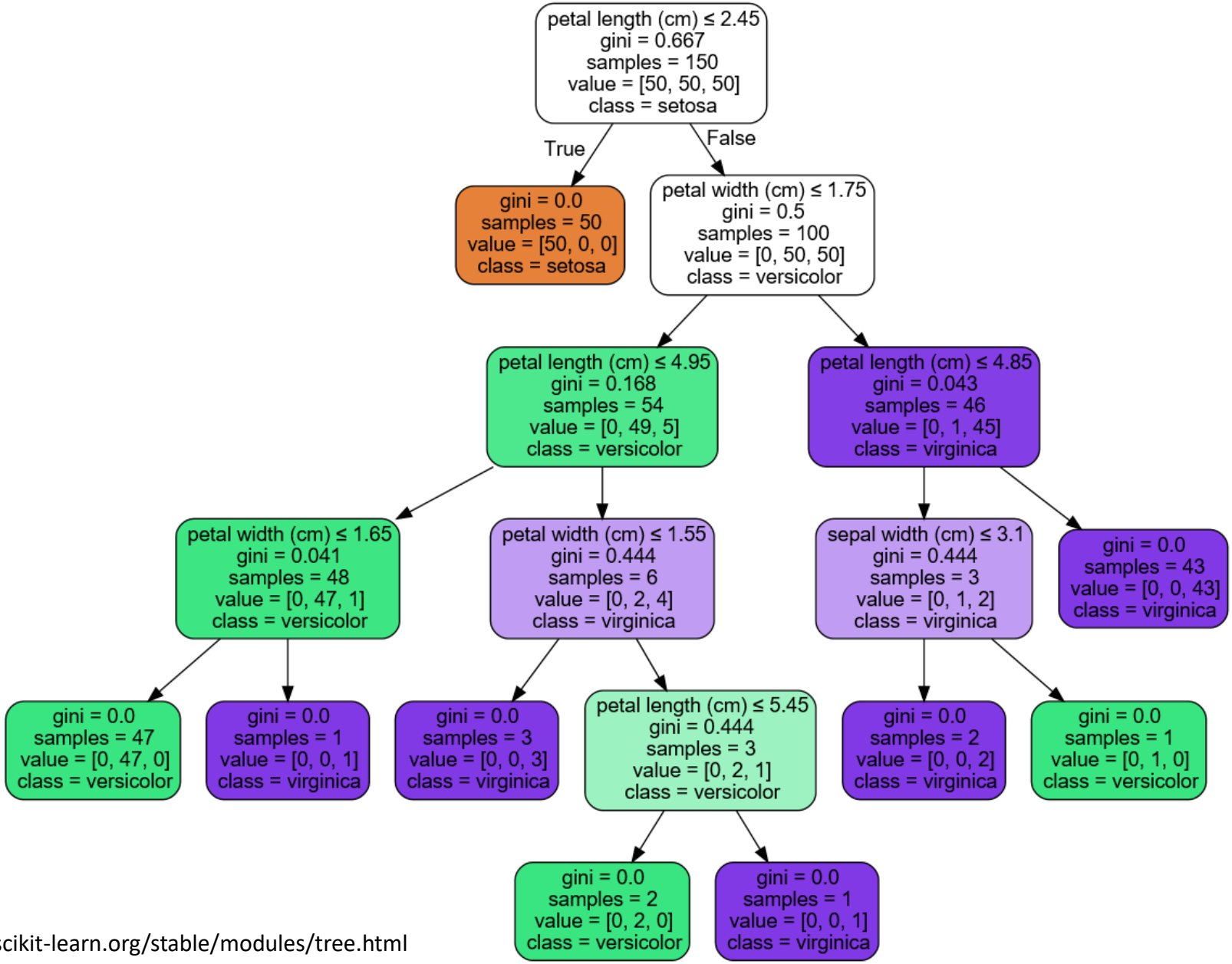
- The answer to each question will cut the number of options/possibilities by about half Very efficient

Decision Trees

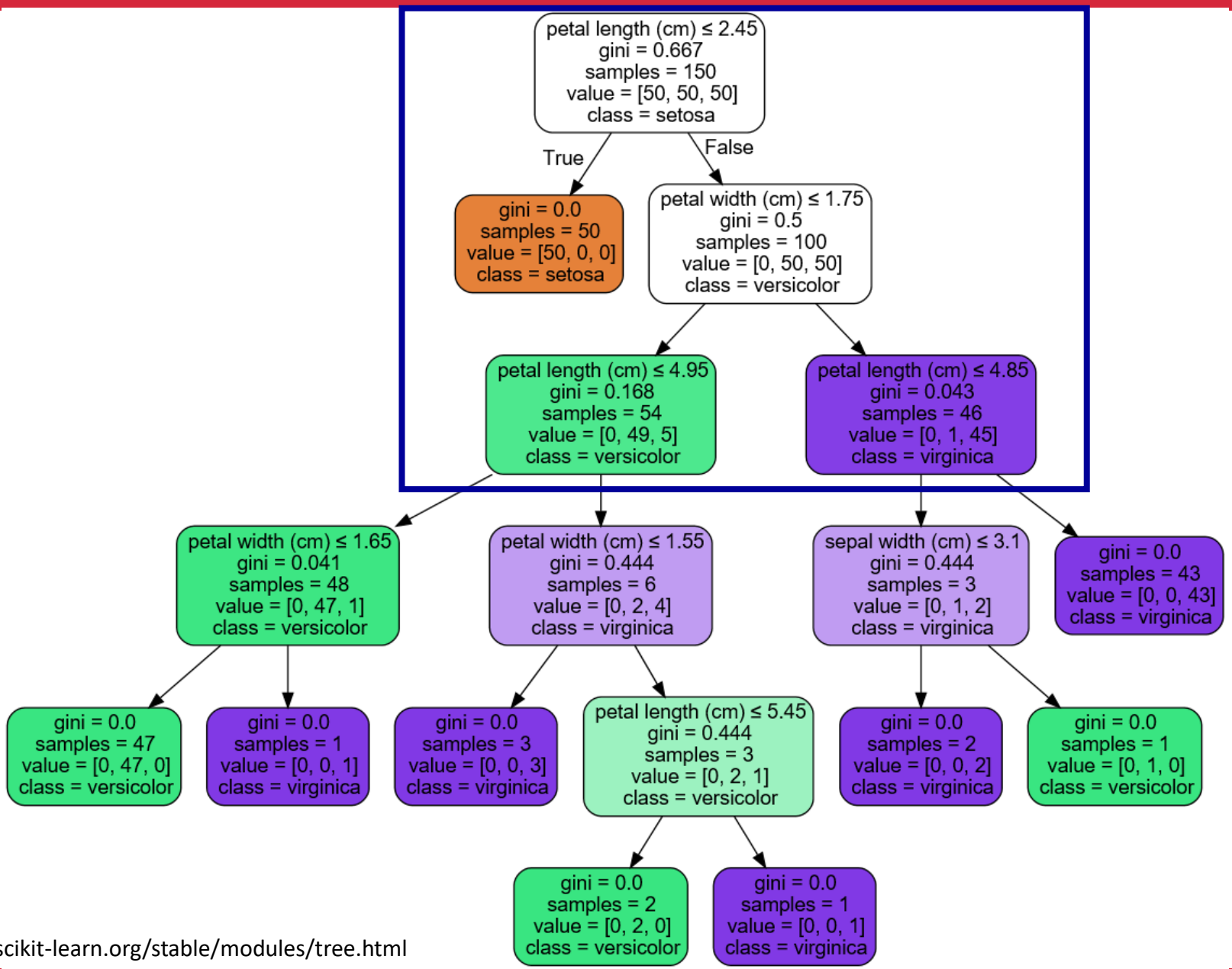
- Ask a simple yes/no question at each level (or, depth) Very intuitive
- The answer to each question will cut the number of options/possibilities by about half Very efficient
- Building blocks of random forests (which is one of the most powerful ML algorithms available today)

Iris data

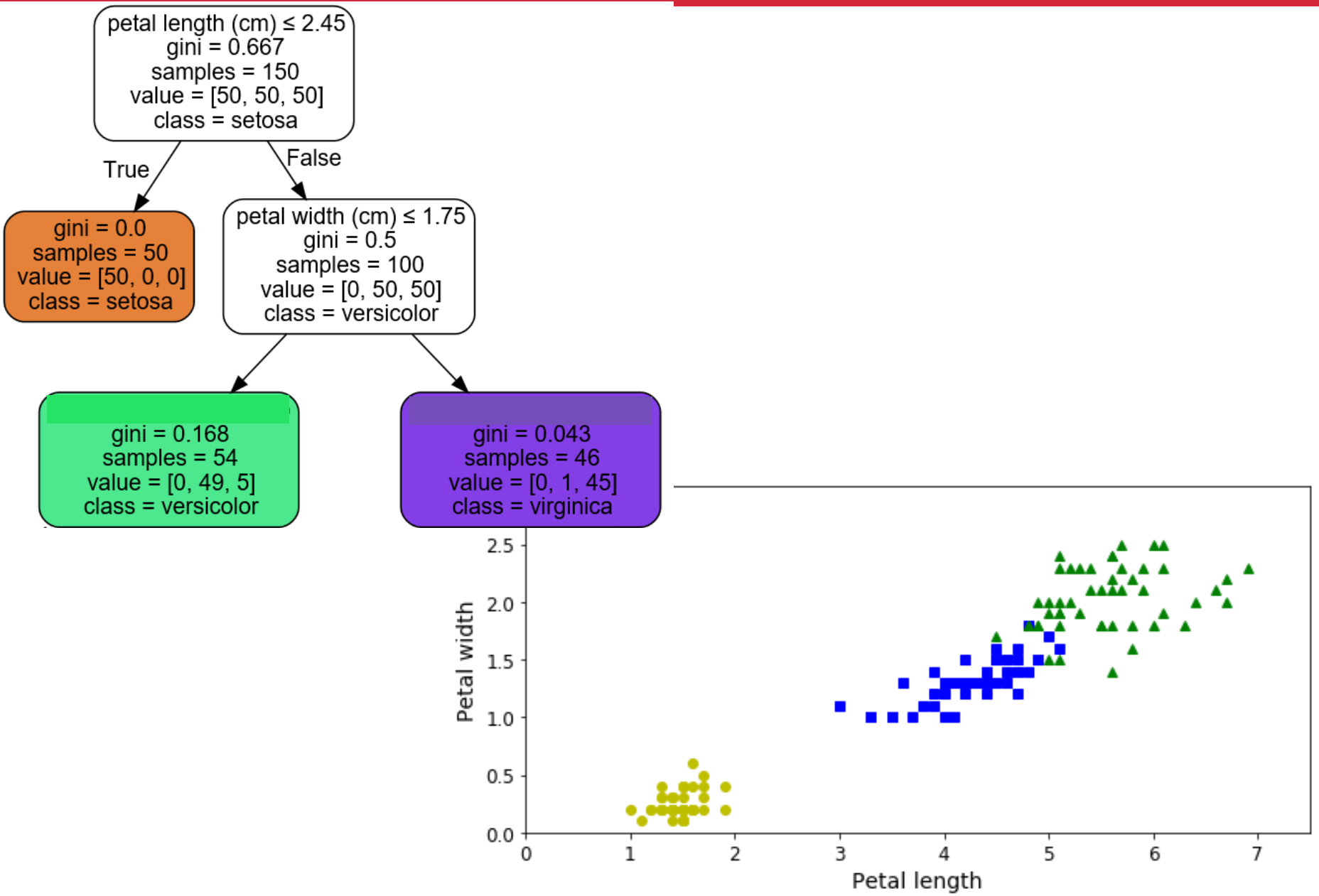


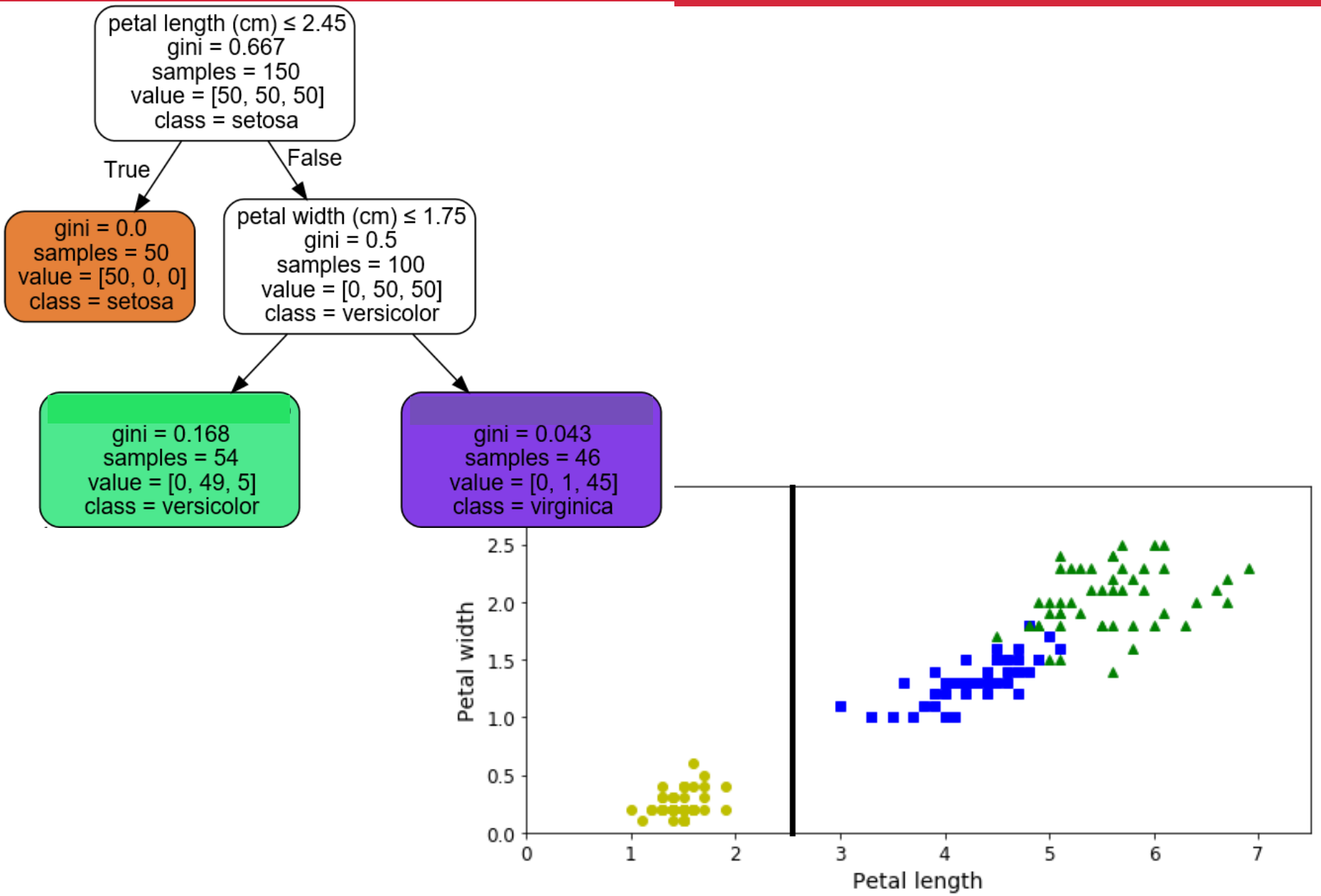


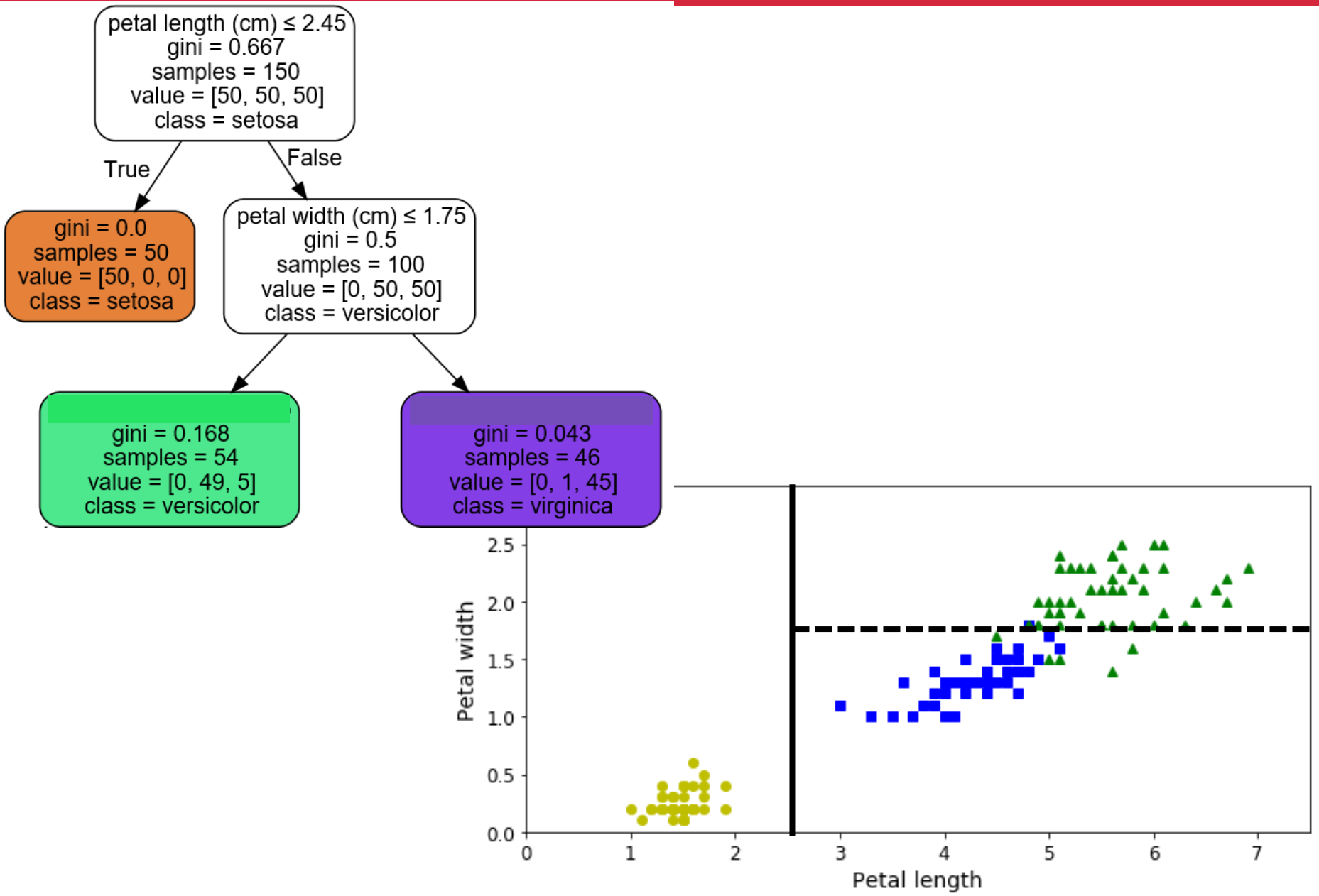
http://scikit-learn.org/stable/modules/tree.html

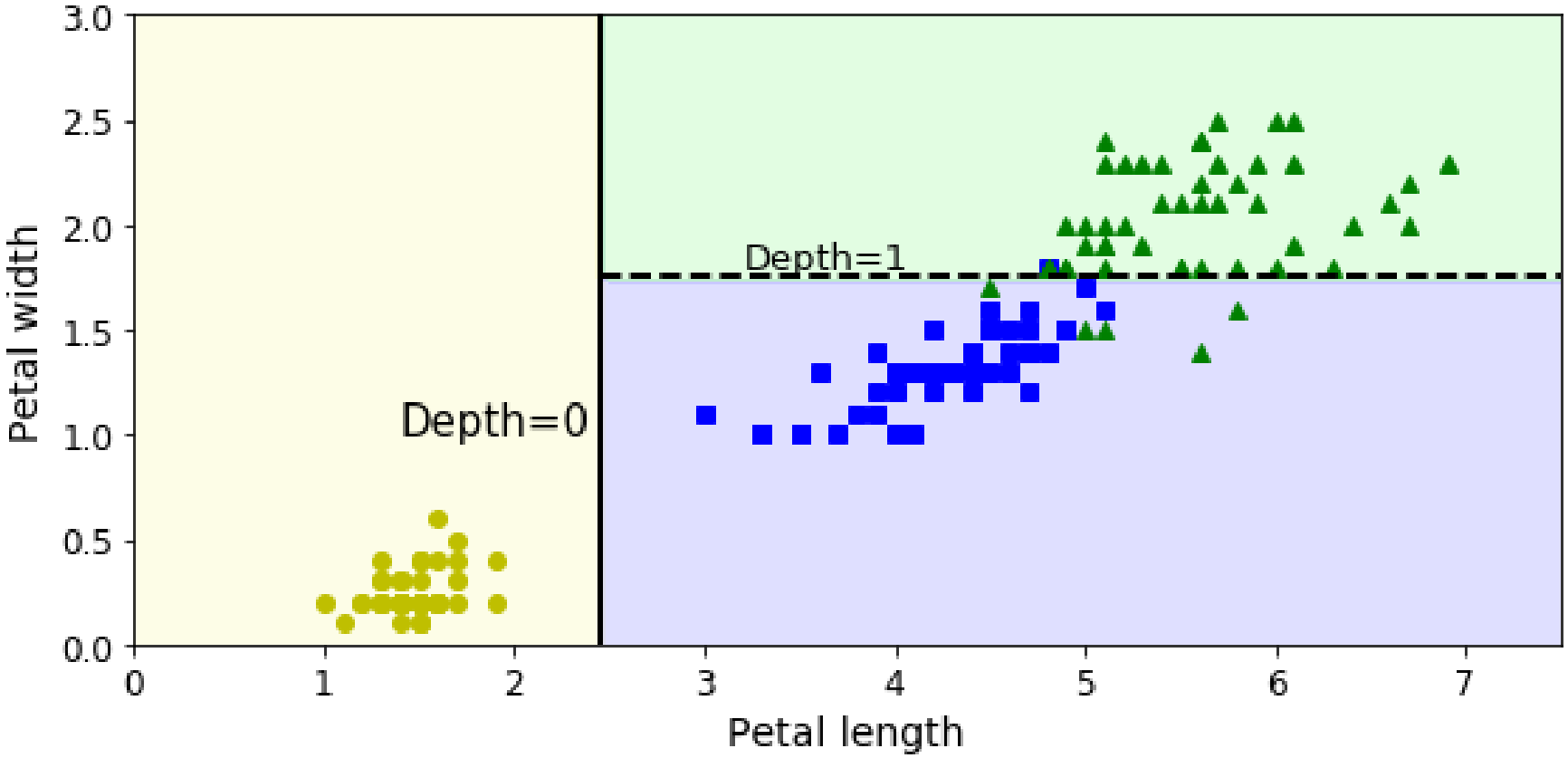


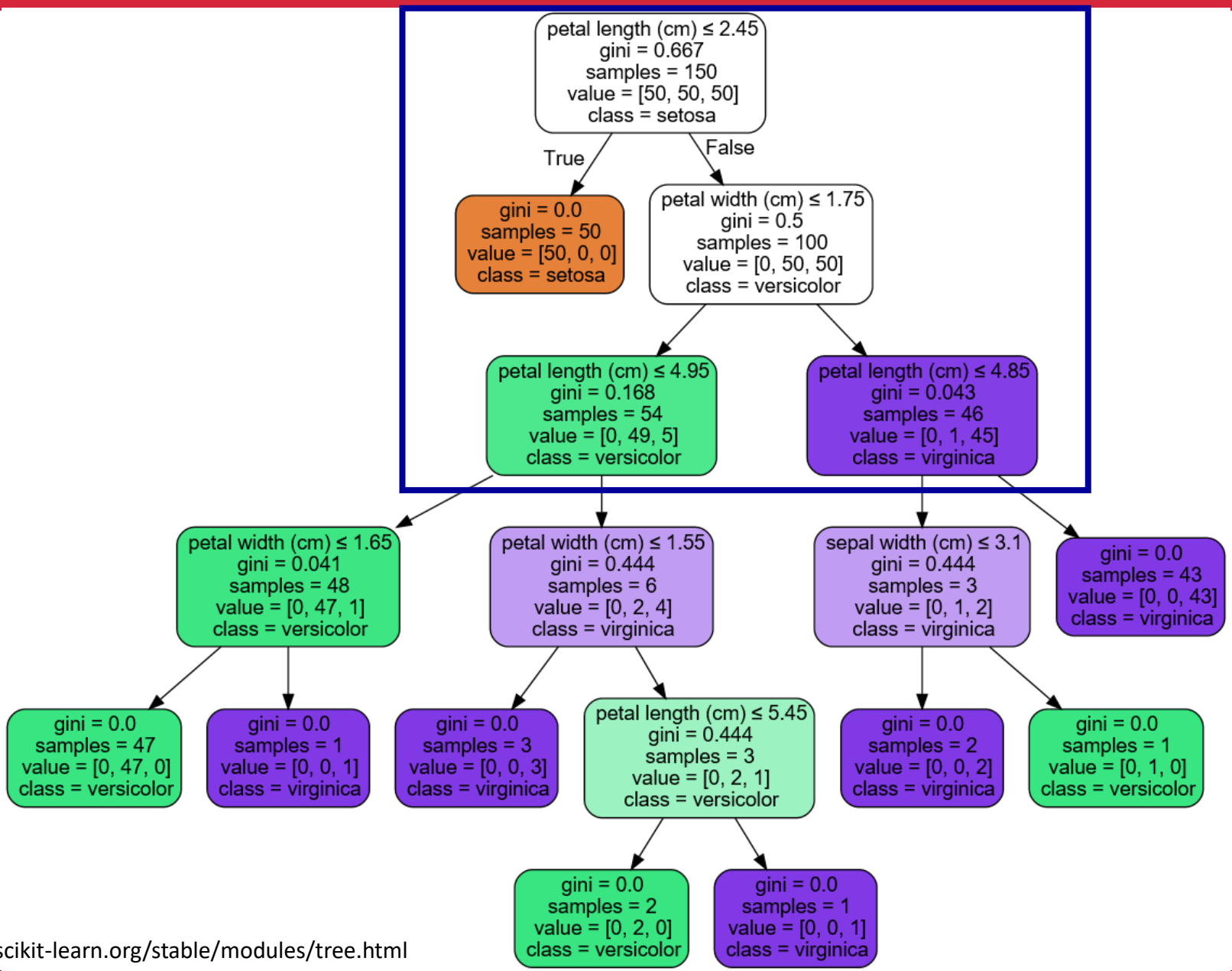
http://scikit-learn.org/stable/modules/tree.html


petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6
7

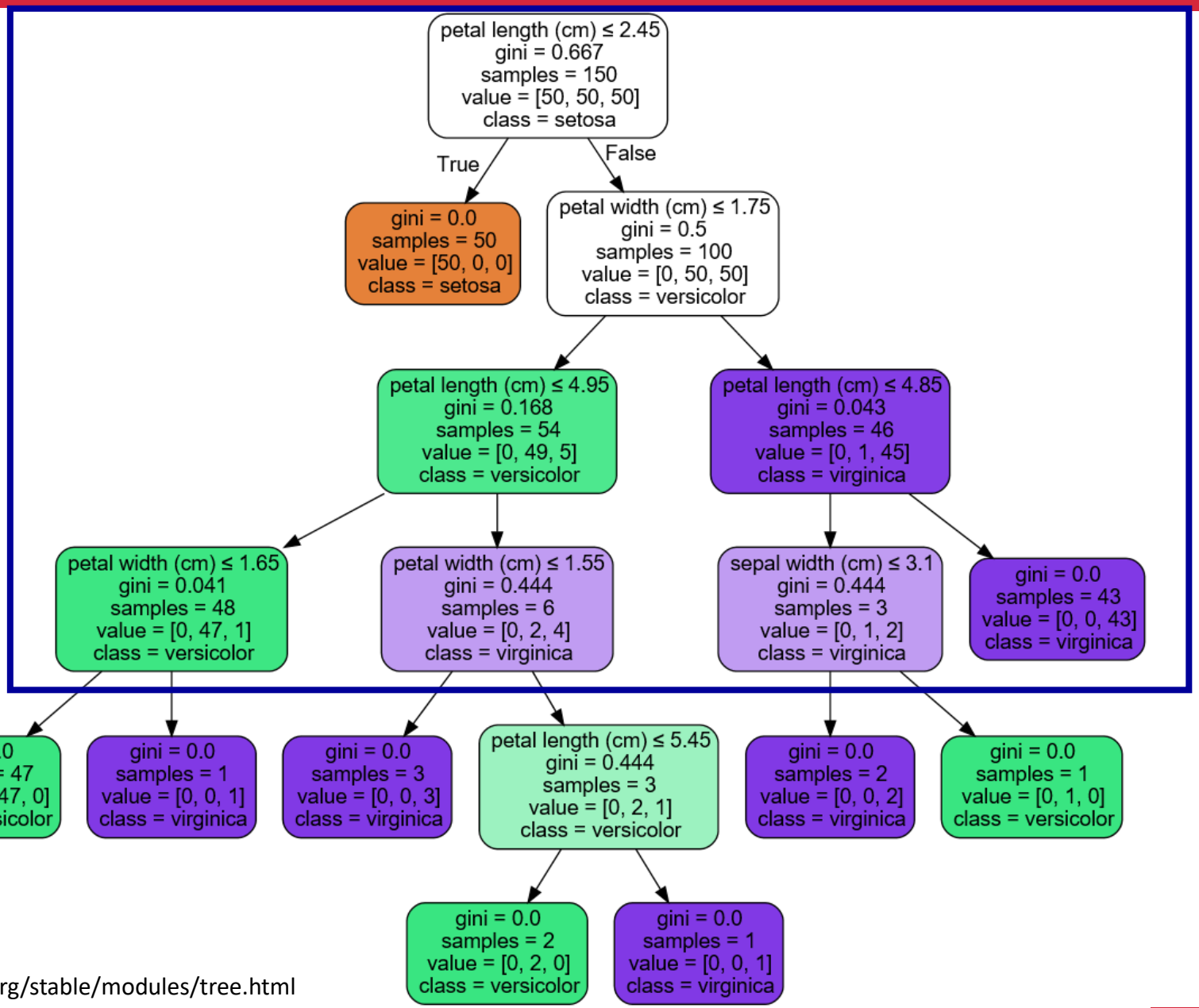

petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6
7


petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6
7

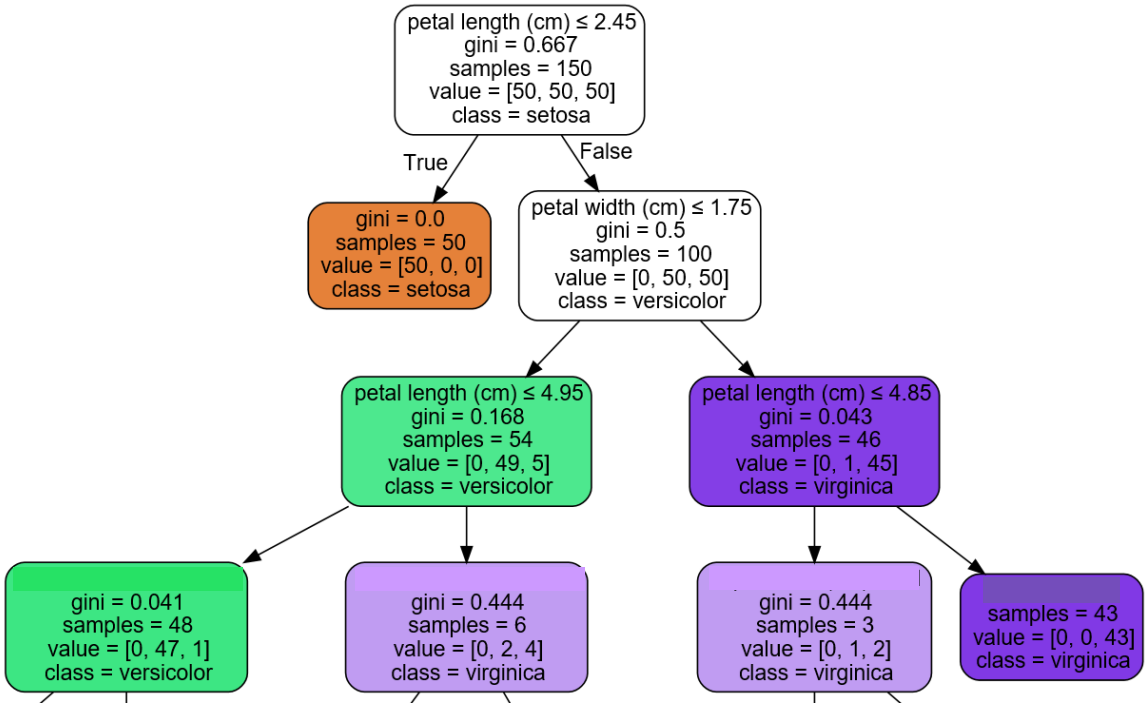

Depth=1
Depth=0
Petal width
Petal length
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7

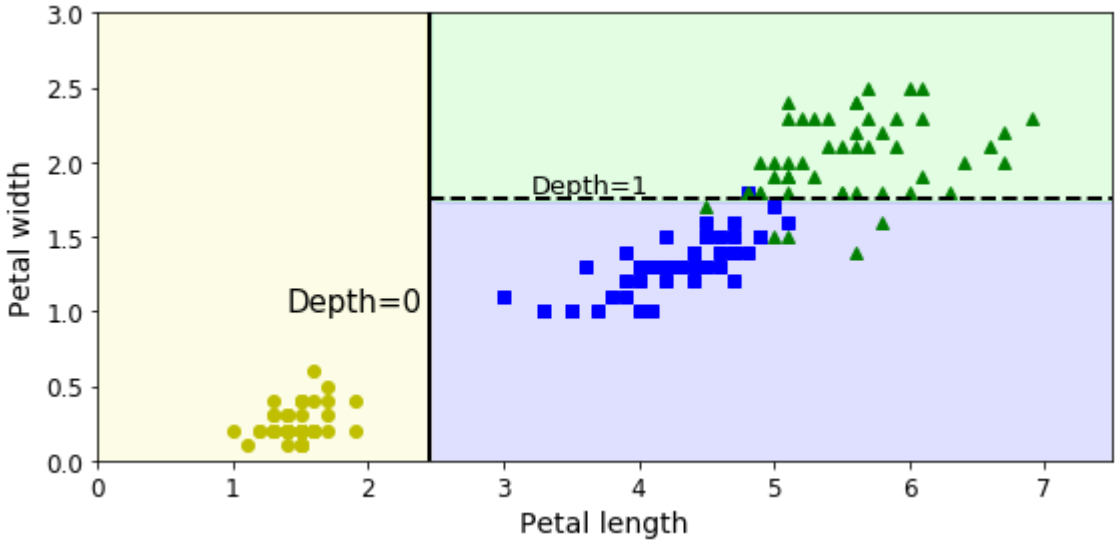


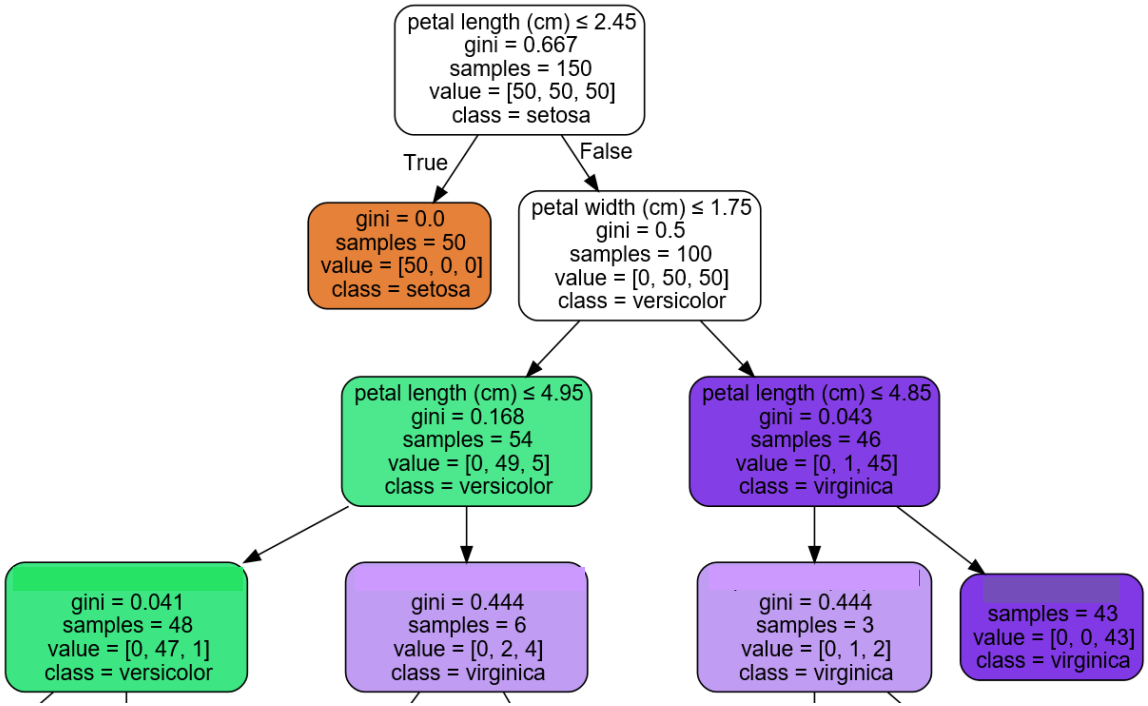
http://scikit-learn.org/stable/modules/tree.html

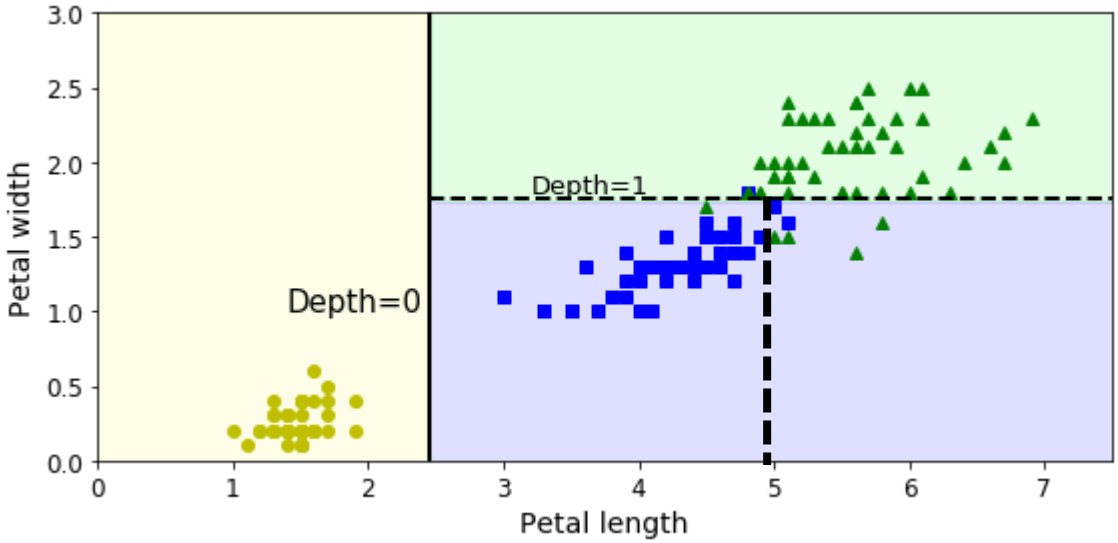


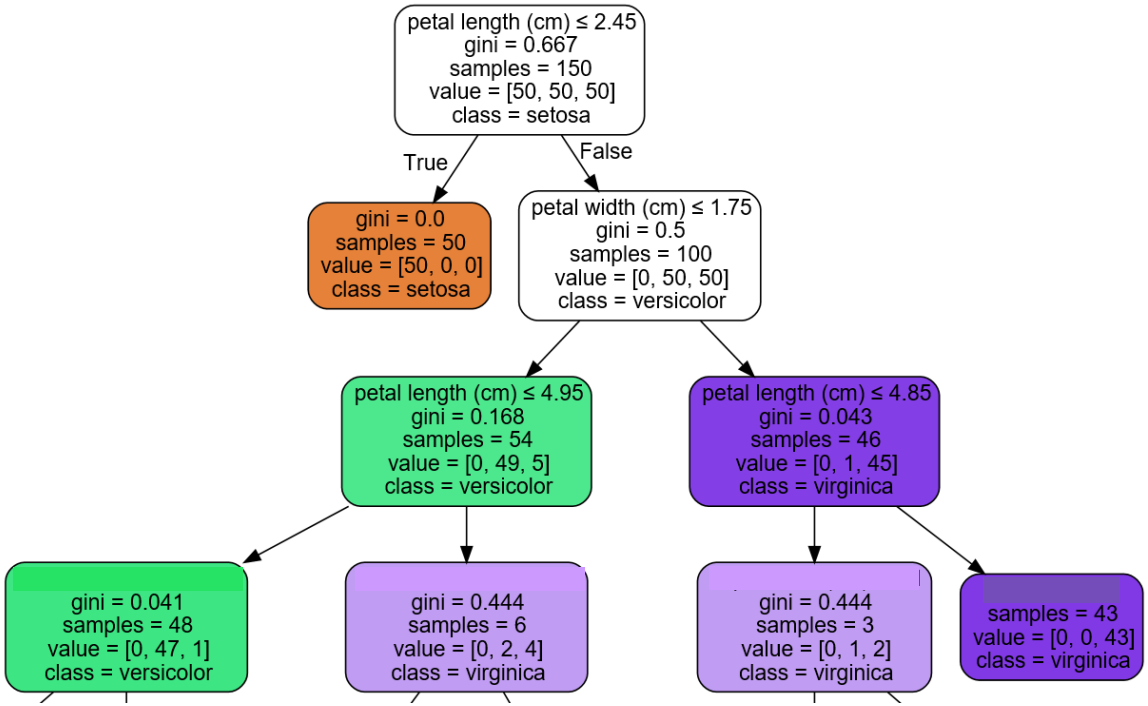
http://scikit-learn.org/stable/modules/tree.html

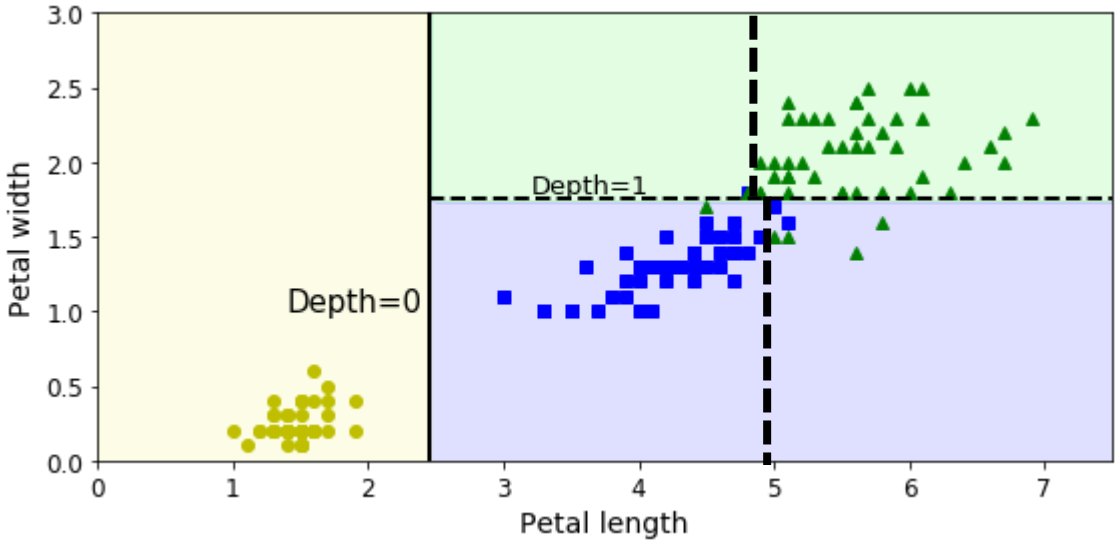

petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
petal length (cm) ≤ 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
gini = 0.041
samples = 48
value = [0, 47, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
samples = 43
value = [0, 0, 43]
class = virginica

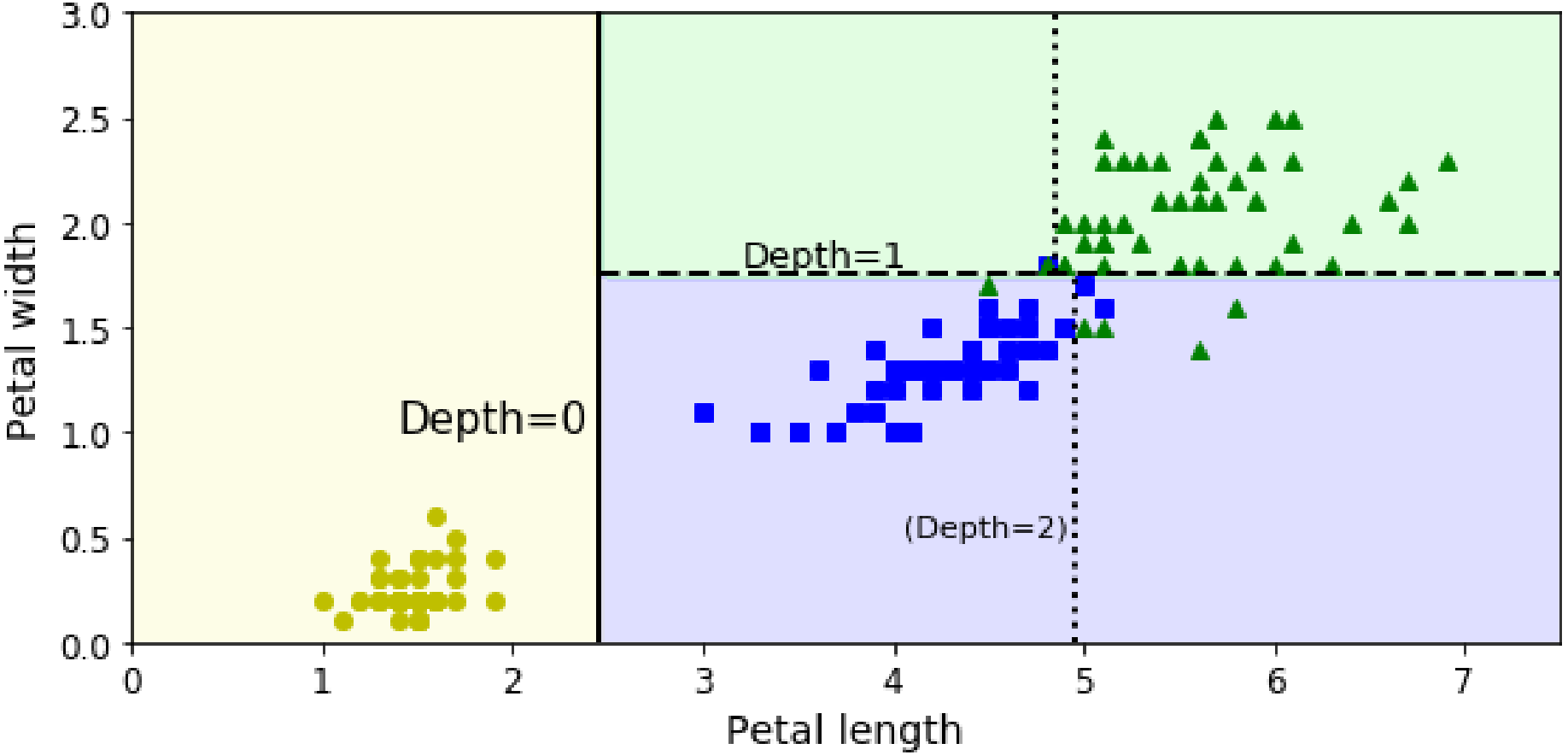

Depth=1
Depth=0
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7

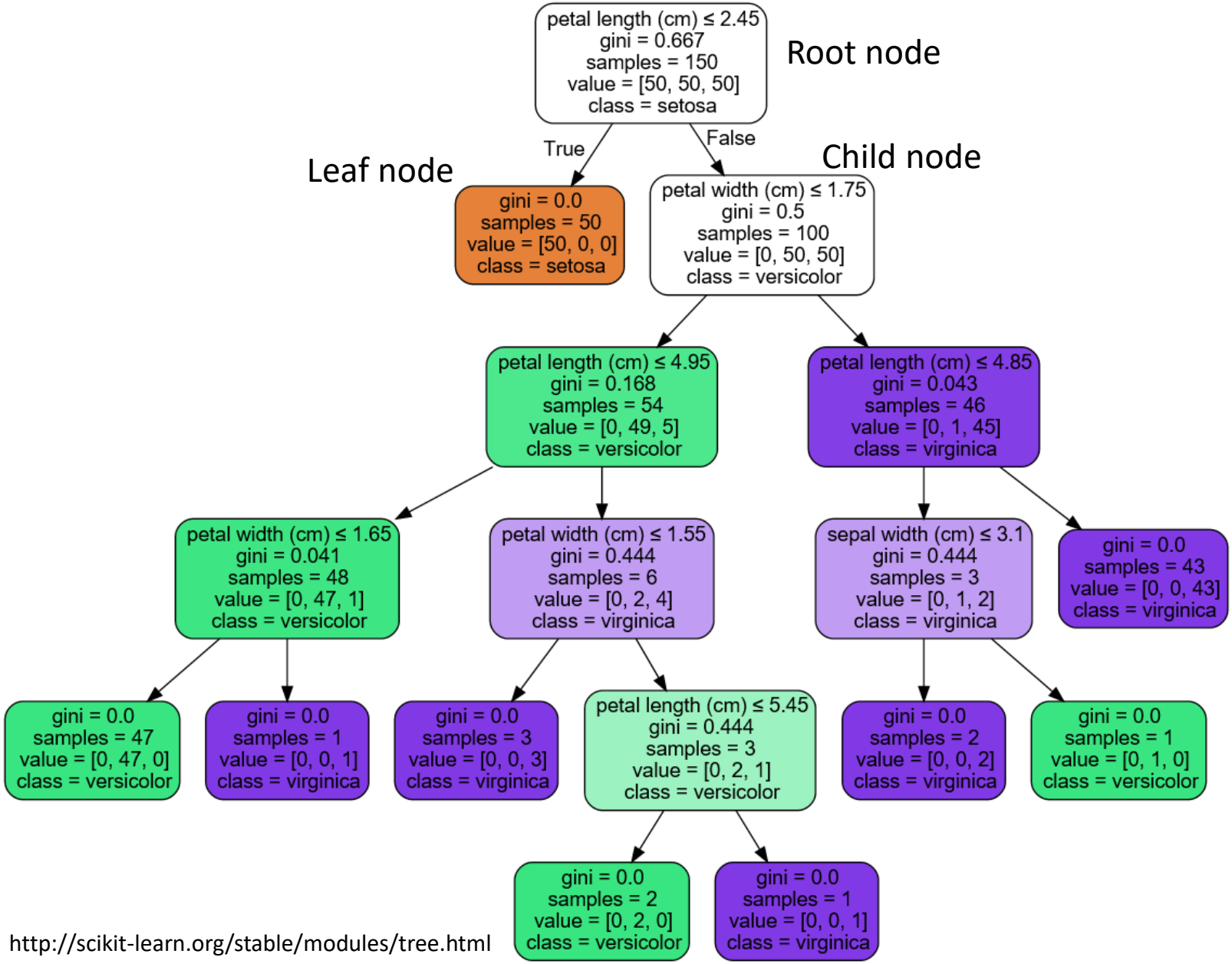

petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
petal length (cm) ≤ 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
gini = 0.041
samples = 48
value = [0, 47, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
samples = 43
value = [0, 0, 43]
class = virginica


Depth=1
Depth=0
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7


petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
petal length (cm) ≤ 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
gini = 0.041
samples = 48
value = [0, 47, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
samples = 43
value = [0, 0, 43]
class = virginica


Depth=1
Depth=0
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7


Depth=1
Depth=0
(Depth=2)
Petal width
Petal length
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7



http://scikit-learn.org/stable/modules/tree.html

Observation

- As the depth increases, we are able to fit the data better and better
- Eventually, at depth 5, we perfectly fit our data*.

*: We cannot really visualize the classification results because there are four features, i.e., 4D feature space.

Observation

- As the depth increases, we are able to fit the data better and better
- Eventually, at depth 5, we perfectly fit our data*.
- Decision tree is a very powerful classification algorithm.

Decision Tree: Iterative split

- The decision tree iteratively splits the data along one or the other axis according to answers to a question.
- The question is about a feature and a threshold value.

Decision Tree: Iterative split

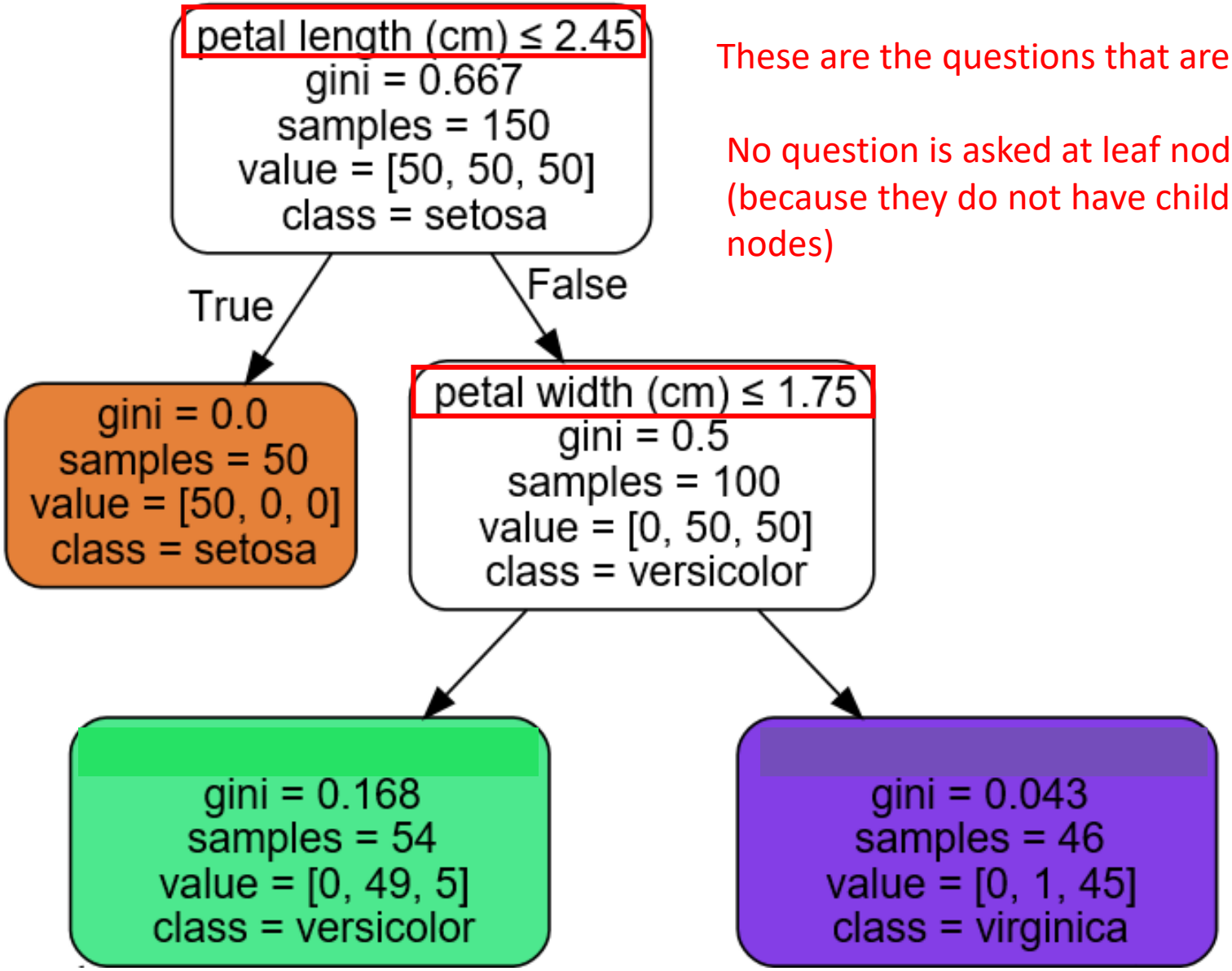
- The decision tree iteratively splits the data along one or the other axis according to answers to a question.
- The question is about a feature and a threshold value.

- Scikit-Learn uses CART algorithm which only asks questions with yes/no answers.
- Therefore, each non-leaf node only has two children.
- Binary tree

Problem

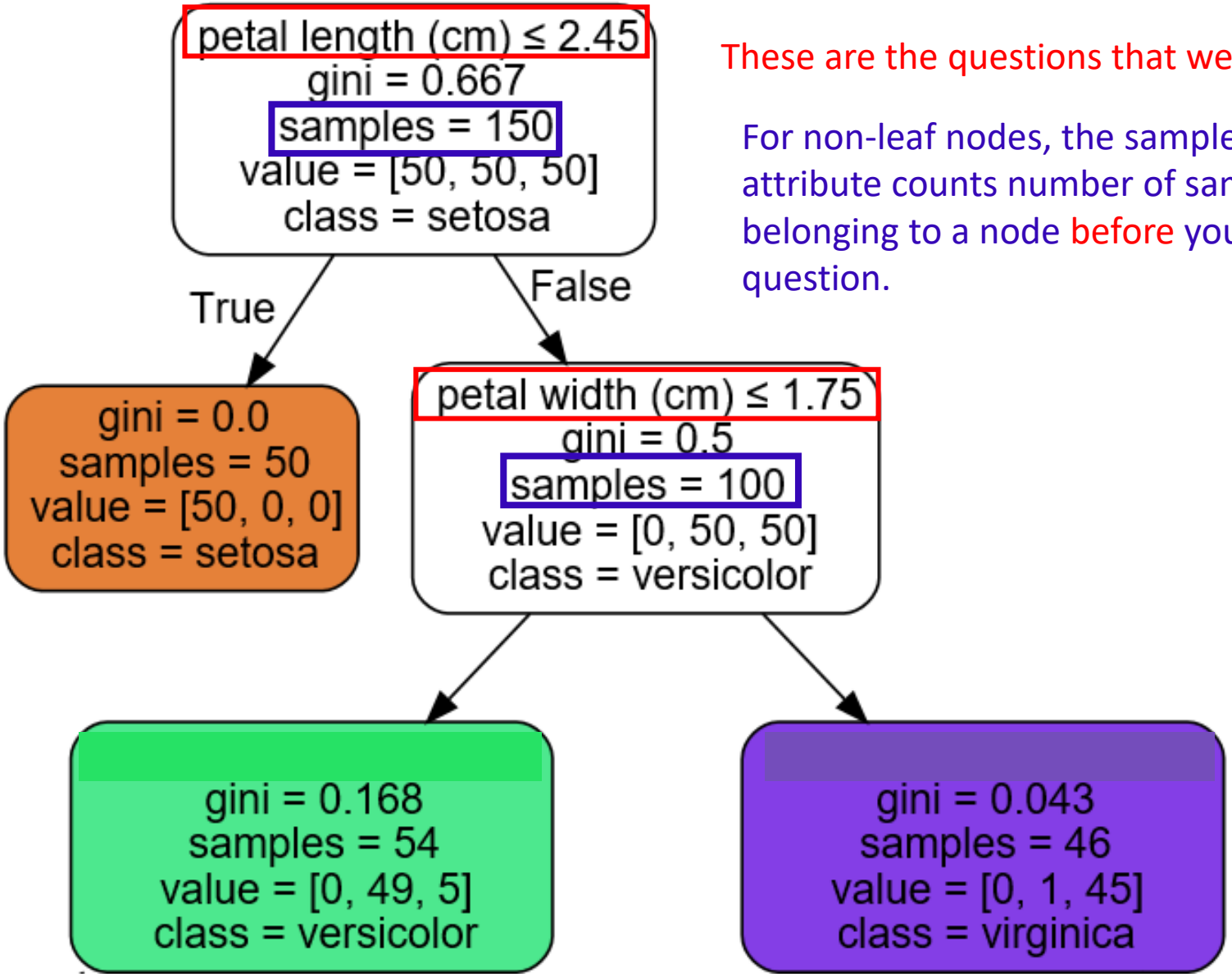
- How to choose which feature and which threshold value to use?

- You need to know what question to ask at each node.



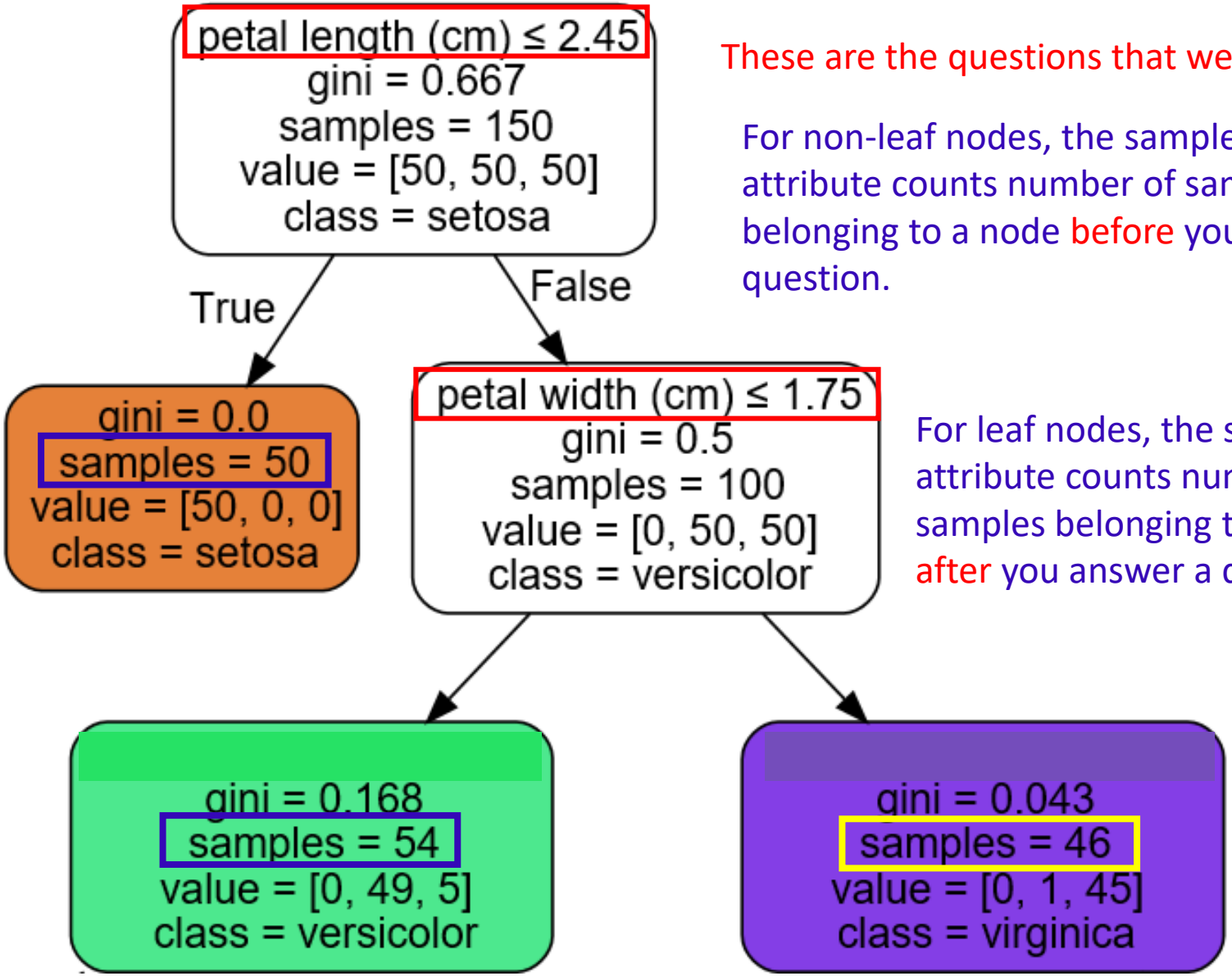
These are the questions that are asked.

No question is asked at leaf nodes (because they do not have children nodes)



These are the questions that were asked.

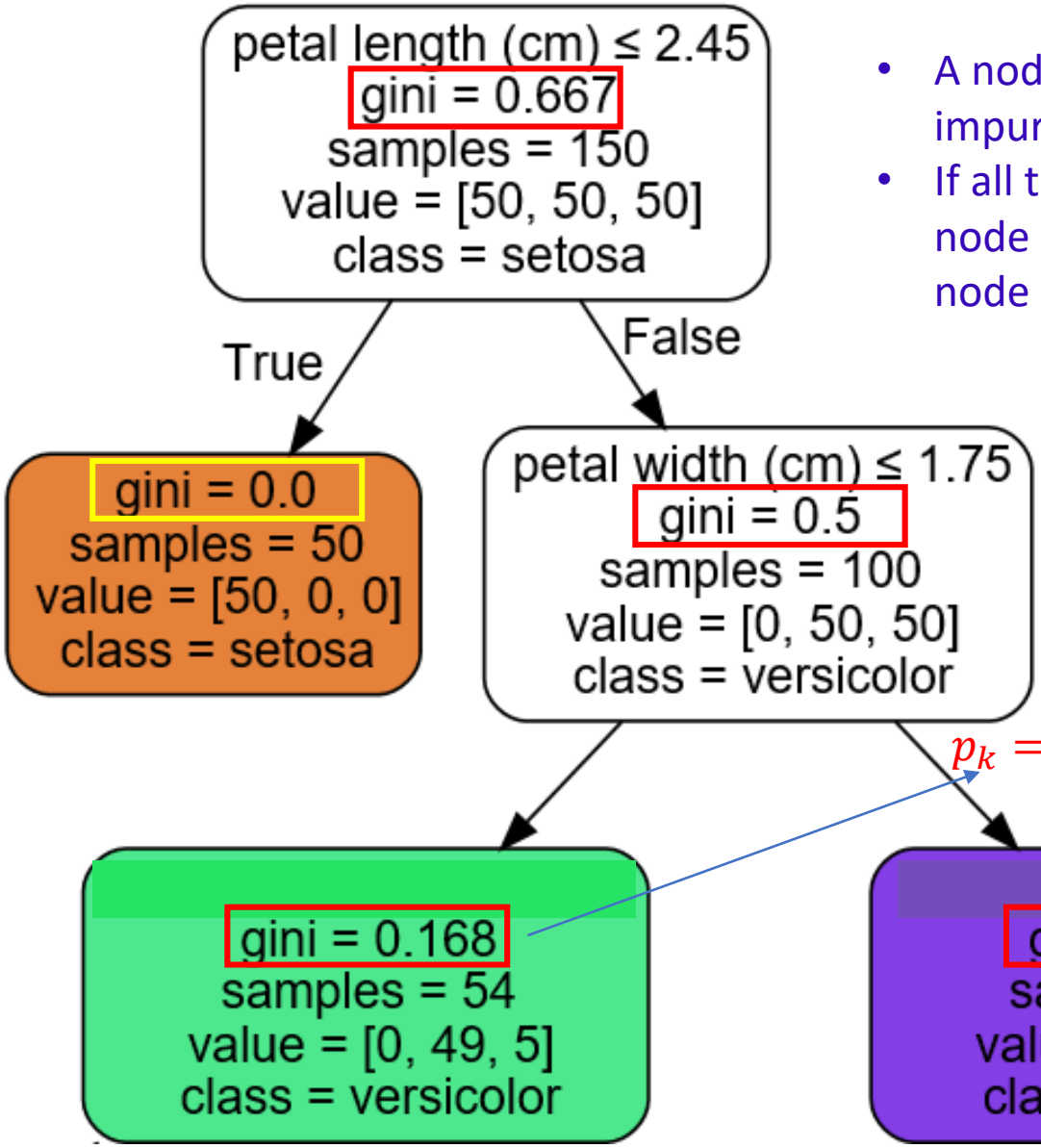
For non-leaf nodes, the sample attribute counts number of samples belonging to a node before you ask a question.



These are the questions that were asked.

For non-leaf nodes, the sample attribute counts number of samples belonging to a node before you ask a question.

For leaf nodes, the sample attribute counts number of samples belonging to a node after you answer a question.



- A node's gini attribute measures its impurity.
- If all the training examples in a node belong to the same class, this node is 'pure', and gini = 0.

$$gini = 1 - \sum_{k=1}^{n} p_k^2$$

p_k: the ratio of the class k examples.

$$p_k = 1 - \left(\frac{0}{54}\right)^2 - \left(\frac{49}{54}\right)^2 - \left(\frac{5}{54}\right)^2 = 0.168$$

CART algorithm

- Scikit-Learn uses Classification And Regression Tree (CART) to grow (or train) decision trees.
- The idea is simple: the algorithm splits the training set in two subsets using a single feature k and a threshold t_k (e.g., petal length <= 2.45 cm)

CART algorithm

- Scikit-Learn uses Classification And Regression Tree (CART) to grow (or train) decision trees.
- The idea is simple: the algorithm splits the training set in two subsets using a single feature k and a threshold t_k (e.g., petal length <= 2.45 cm)
- It searches for the pair (k, t_k) that produces the *purest* subsets*

*: Remember our goal with classification is to split or partition our data into as pure sets as possible.

CART algorithm

- Scikit-Learn uses Classification And Regression Tree (CART) to grow (or train) decision trees.
- The idea is simple: the algorithm splits the training set in two subsets using a single feature k and a threshold t_k (e.g., petal length <= 2.45 cm)
- It searches for the pair (k, t_k) that produces the *purest* subsets*
- by minimizing a cost function …

*: Remember our goal with classification is to split or partition our data into as pure sets as possible.

CART: cost function

$$J(k, t_k) = \frac{m_{left}}{m} g_{left} + \frac{m_{right}}{m} g_{right}$$

g_{left}: the impurity of the left subset

g_{right}: the impurity of the left subset

The tree stops growing once it reaches the maximum depth (max_depth), or if it cannot find a split that will reduce impurity.

Regularization

- Keep your model simple
- Avoid overfitting

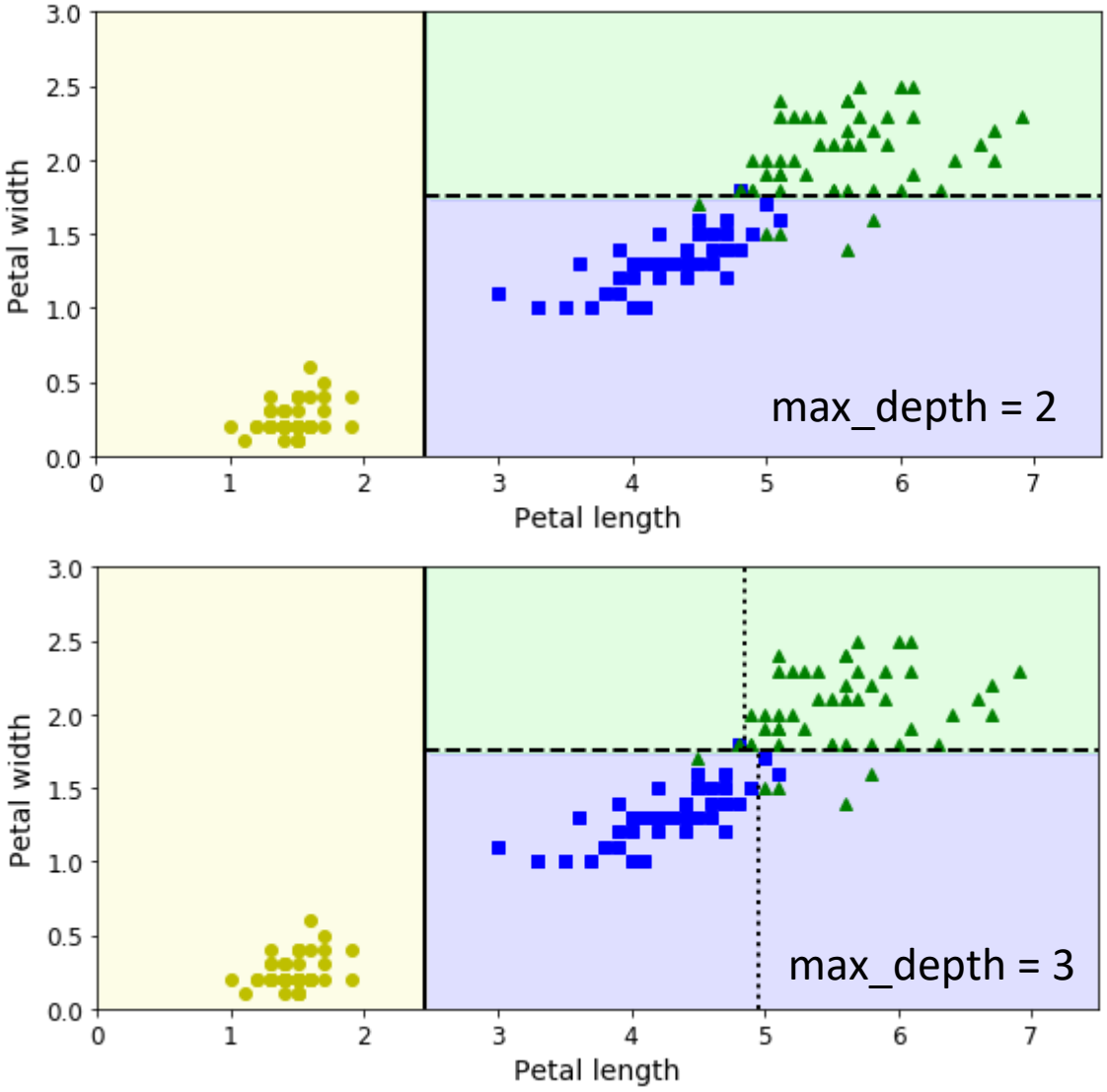
Regularization

- Keep your model simple
- Avoid overfitting

- Decision trees are so powerful that, if left unconstrained, the tree structure will adapt itself to the training data, fitting it very closely, and most likely overfitting it.

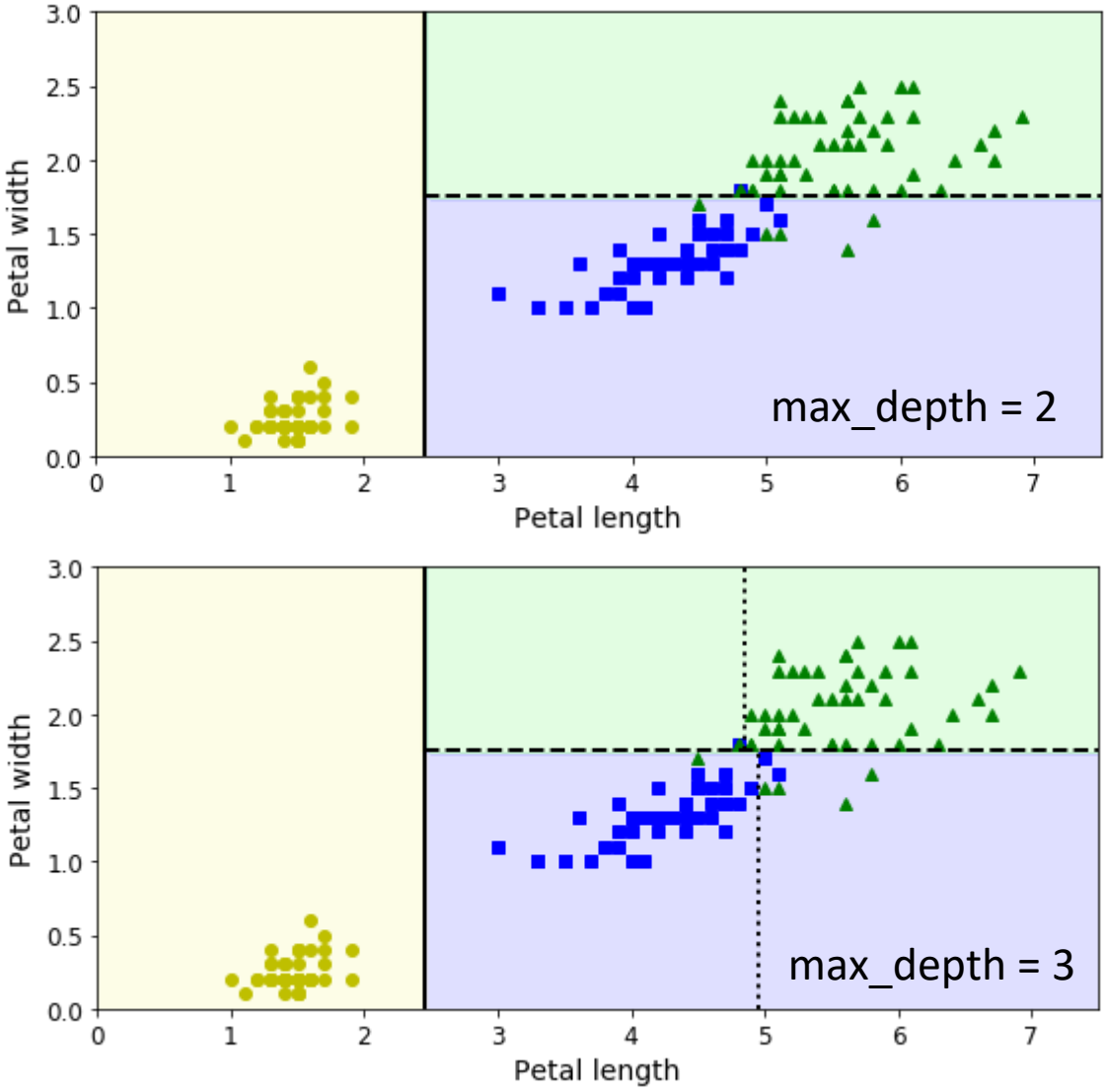
Regularization hyperparameters

- max_depth



Regularization hyperparameters

- max_depth

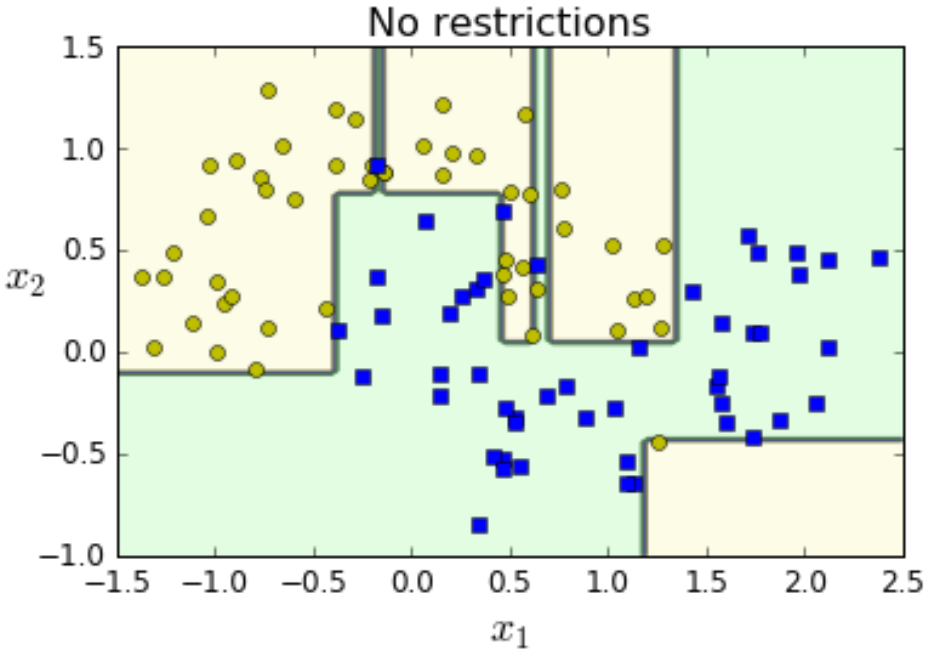


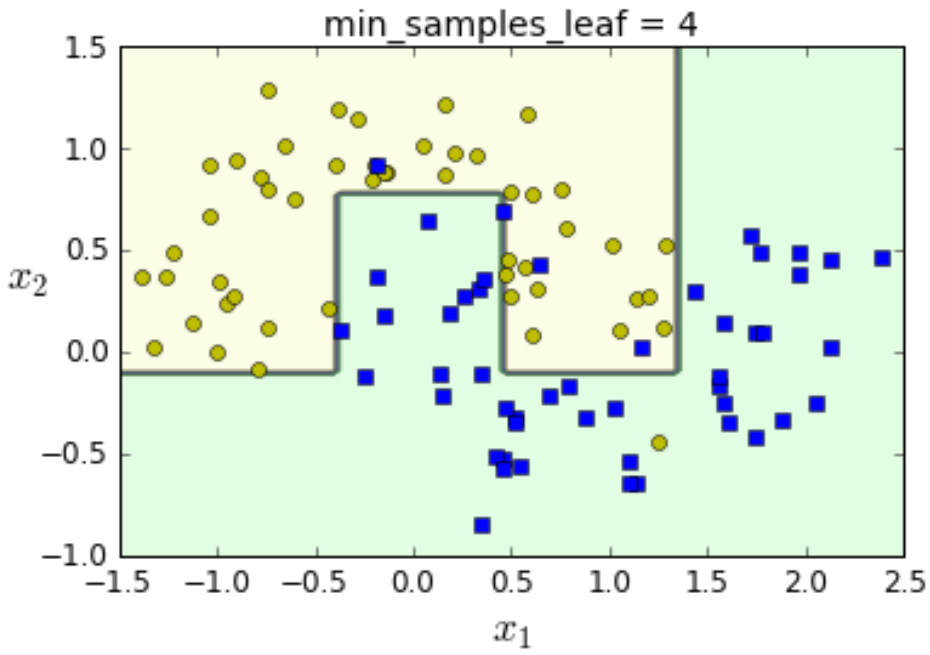
The default value is None, which means unlimited.

Reduing max_depth will regularize the model and reduce the risk of overfitting.

Regularization hyperparameters

- min_samples_leaf: the minimum number of samples a leaf node must have.





The default value is 1.

Regularization hyperparameters

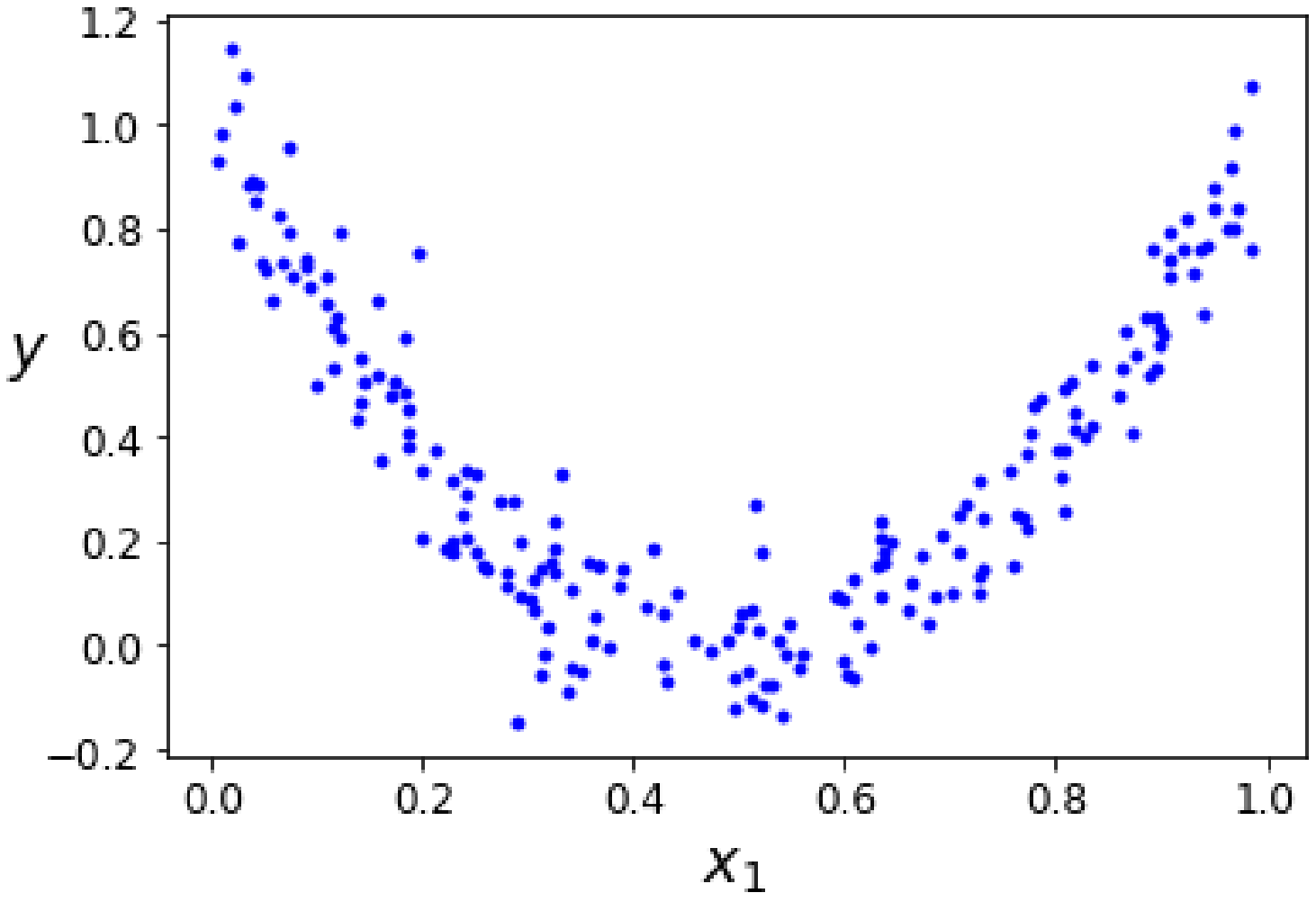
- min_samples_split: the minimum number of samples a node must have before it can be split.
- min_weight_fraction_leaf: same as *min_samples_leaf* but expressed as a fraction of the total number of weighted instances
- max_leaf_nodes: maximum number of leaf nodes
- max_features: maximum number of features that are evaluated for splitting at each node.

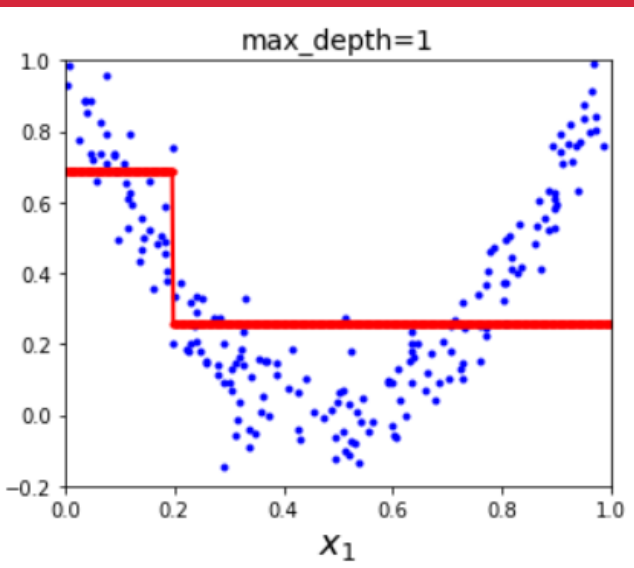
To learn more: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

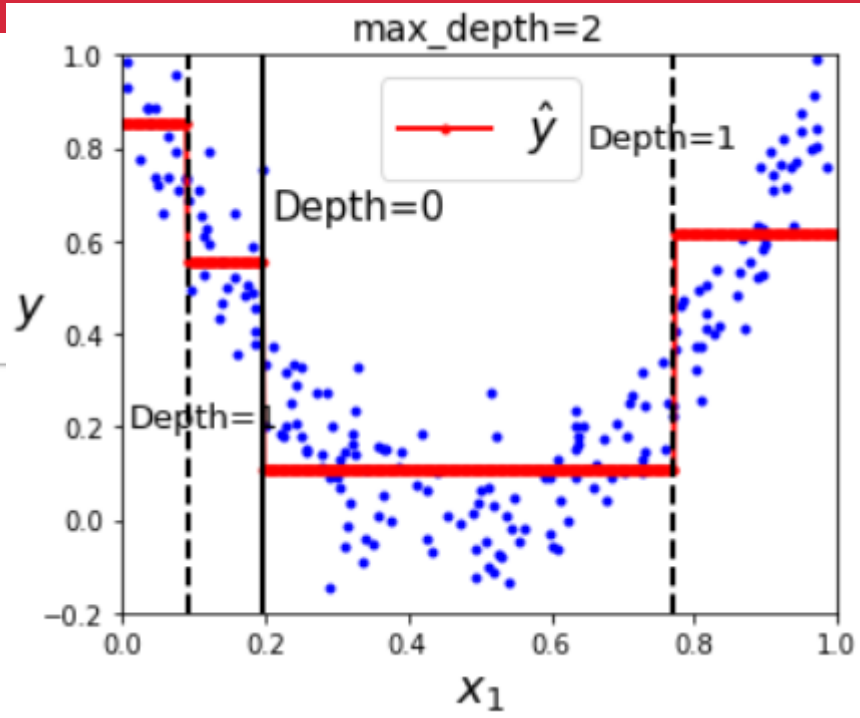
Regression

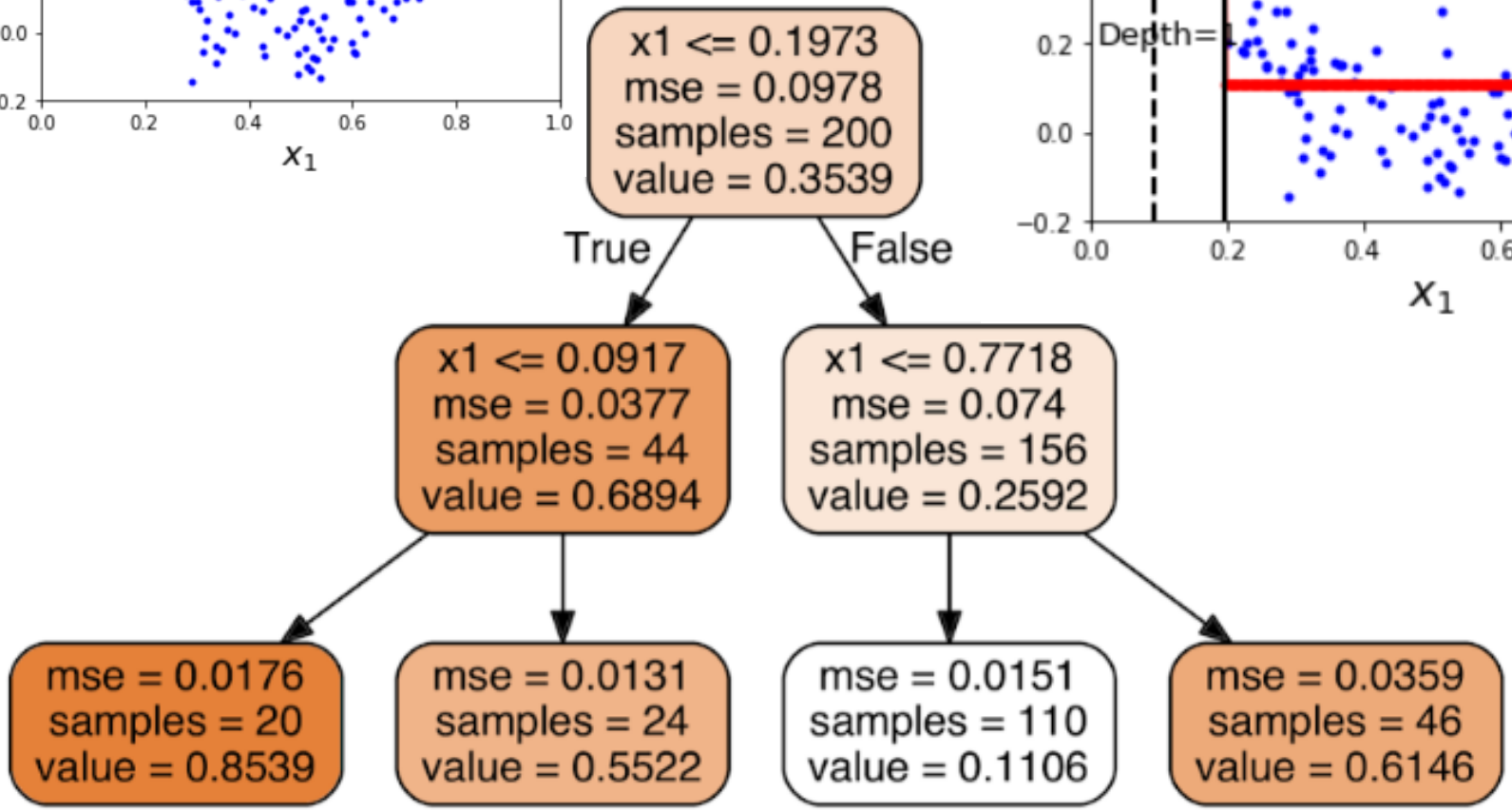
- Decision tress can also perform regression tasks.
- Instead of predicting a class in each node, it predicts a value.

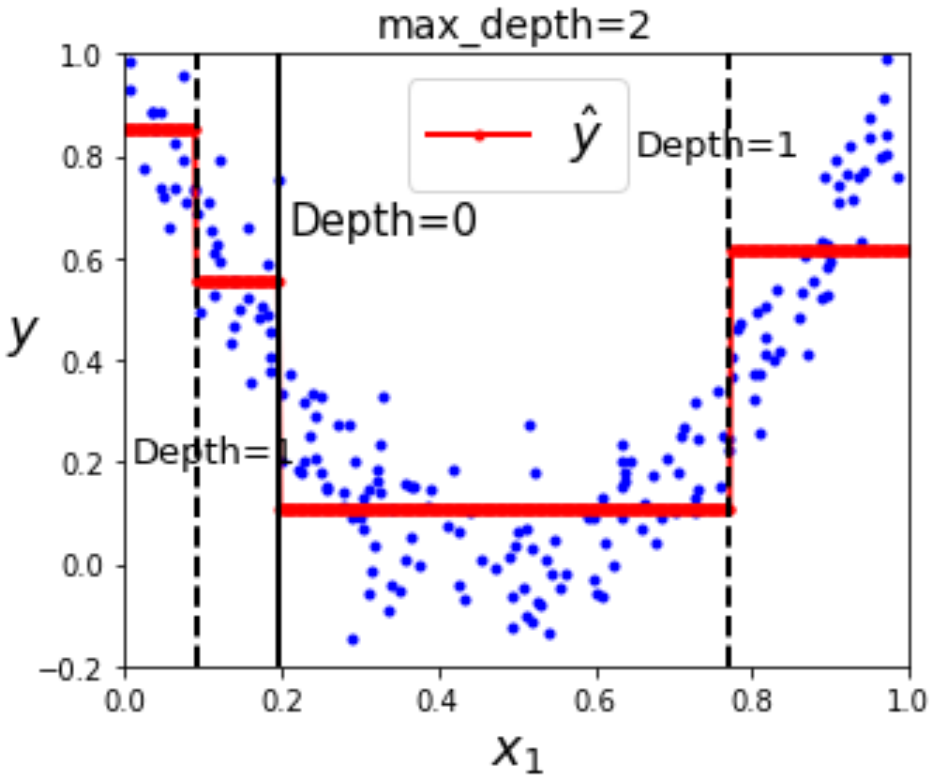
Regression example

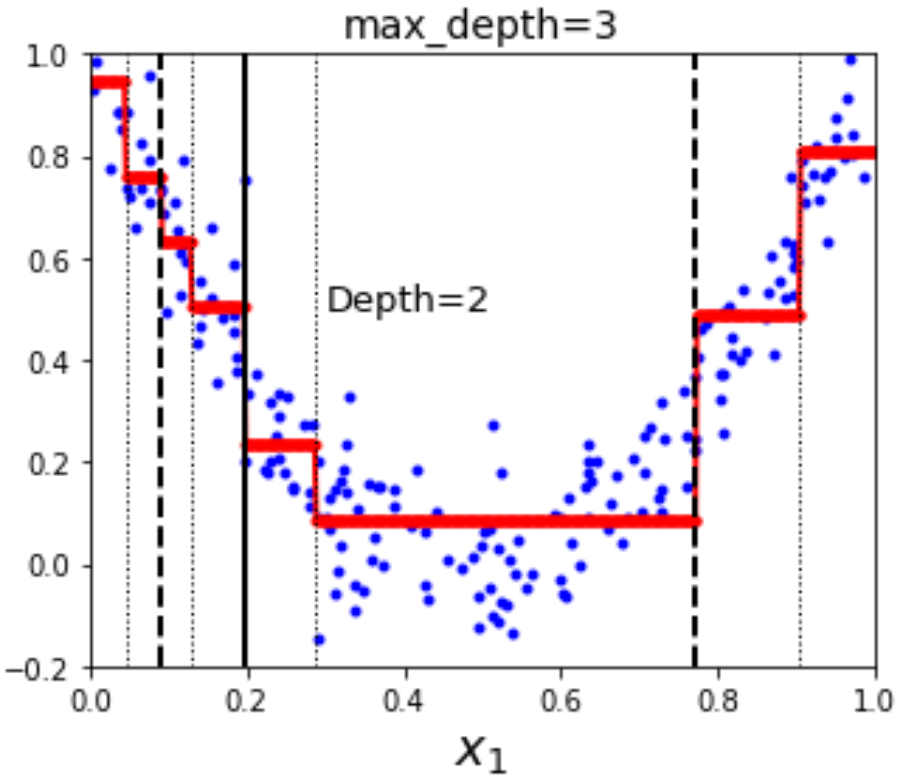


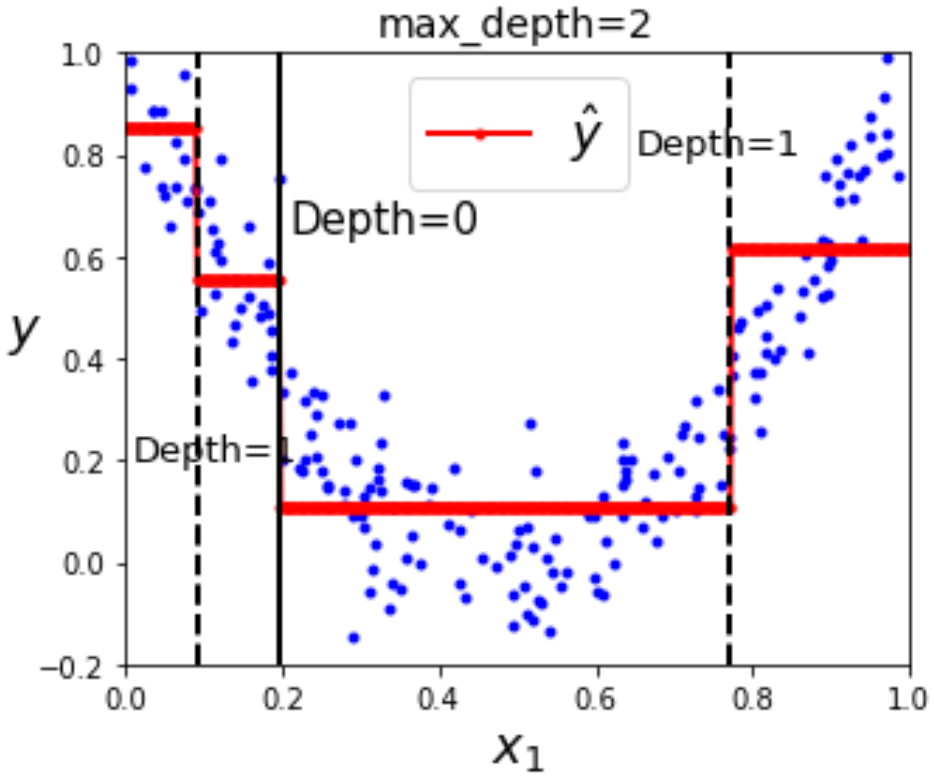

max_depth=1
x_1

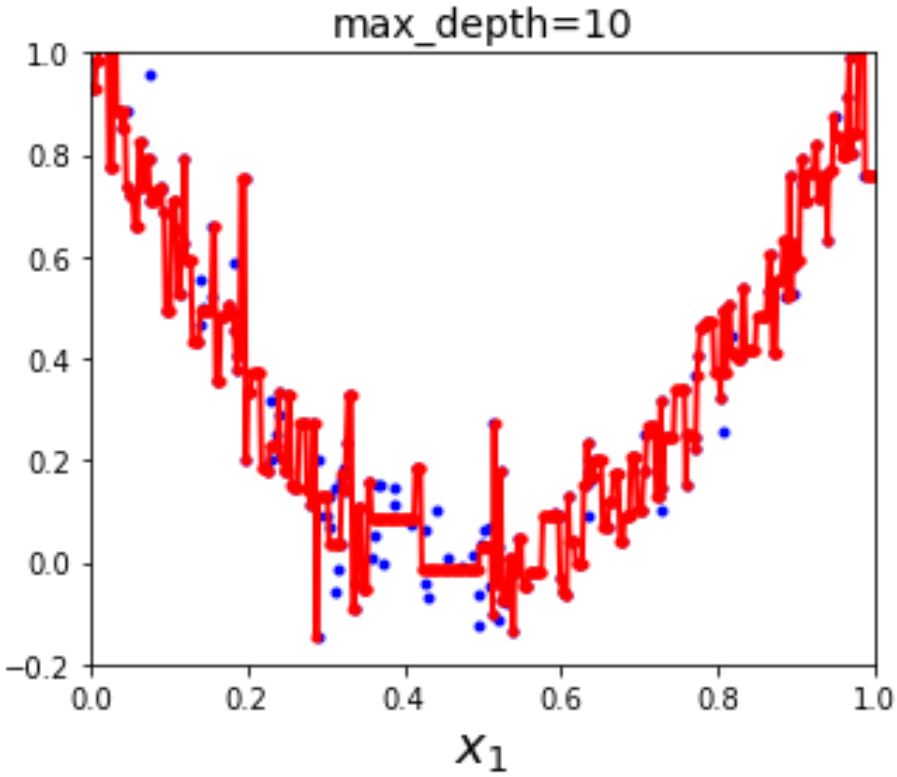

max_depth=2
$\hat{y}$
Depth=1
Depth=0
Depth=1
y
x_1


x1 <= 0.1973
mse = 0.0978
samples = 200
value = 0.3539
True
False
x1 <= 0.0917
mse = 0.0377
samples = 44
value = 0.6894
x1 <= 0.7718
mse = 0.074
samples = 156
value = 0.2592
mse = 0.0176
samples = 20
value = 0.8539
mse = 0.0131
samples = 24
value = 0.5522
mse = 0.0151
samples = 110
value = 0.1106
mse = 0.0359
samples = 46
value = 0.6146

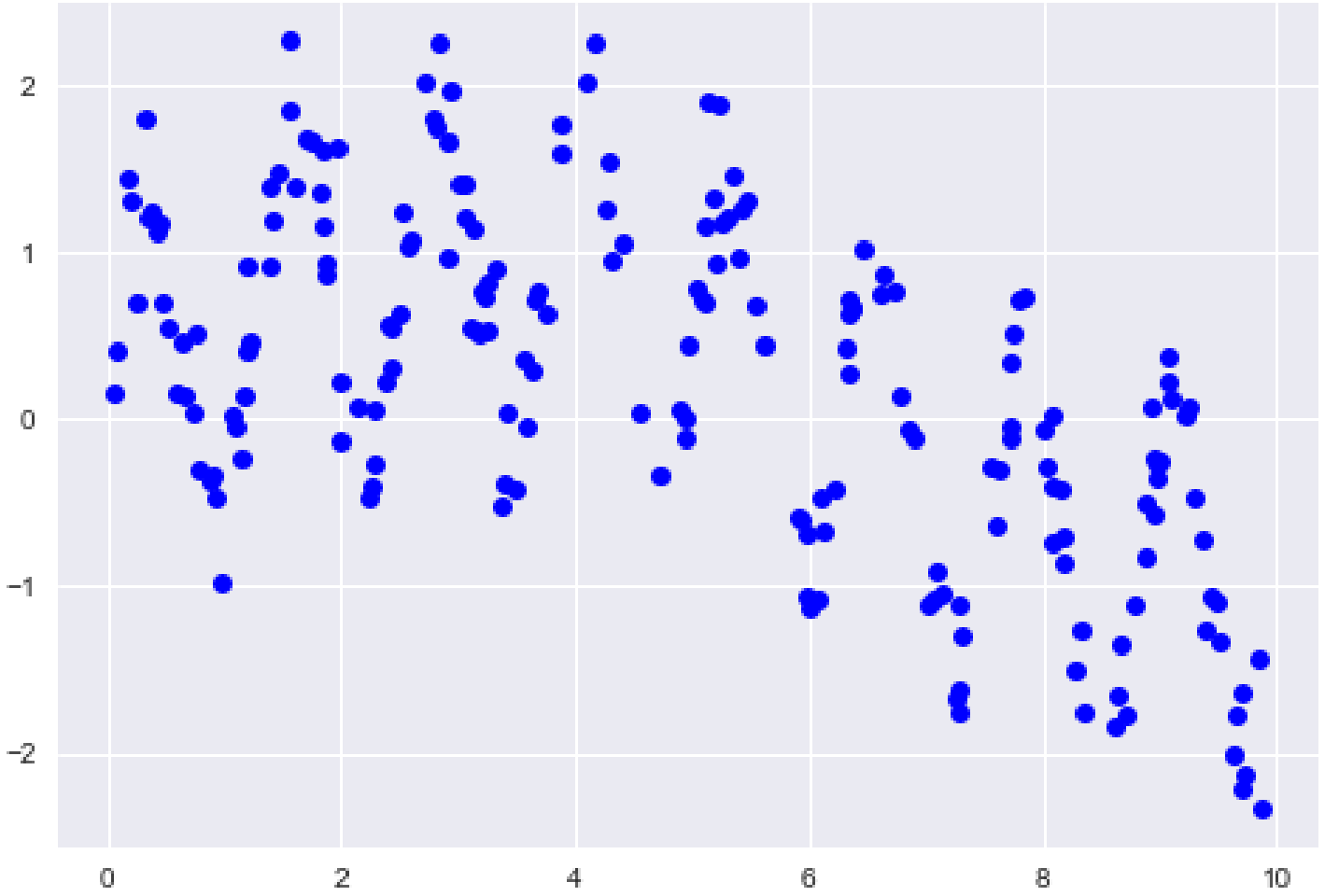

max_depth=2
$\hat{y}$
Depth=1
Depth=0
Depth=1
y
x_1

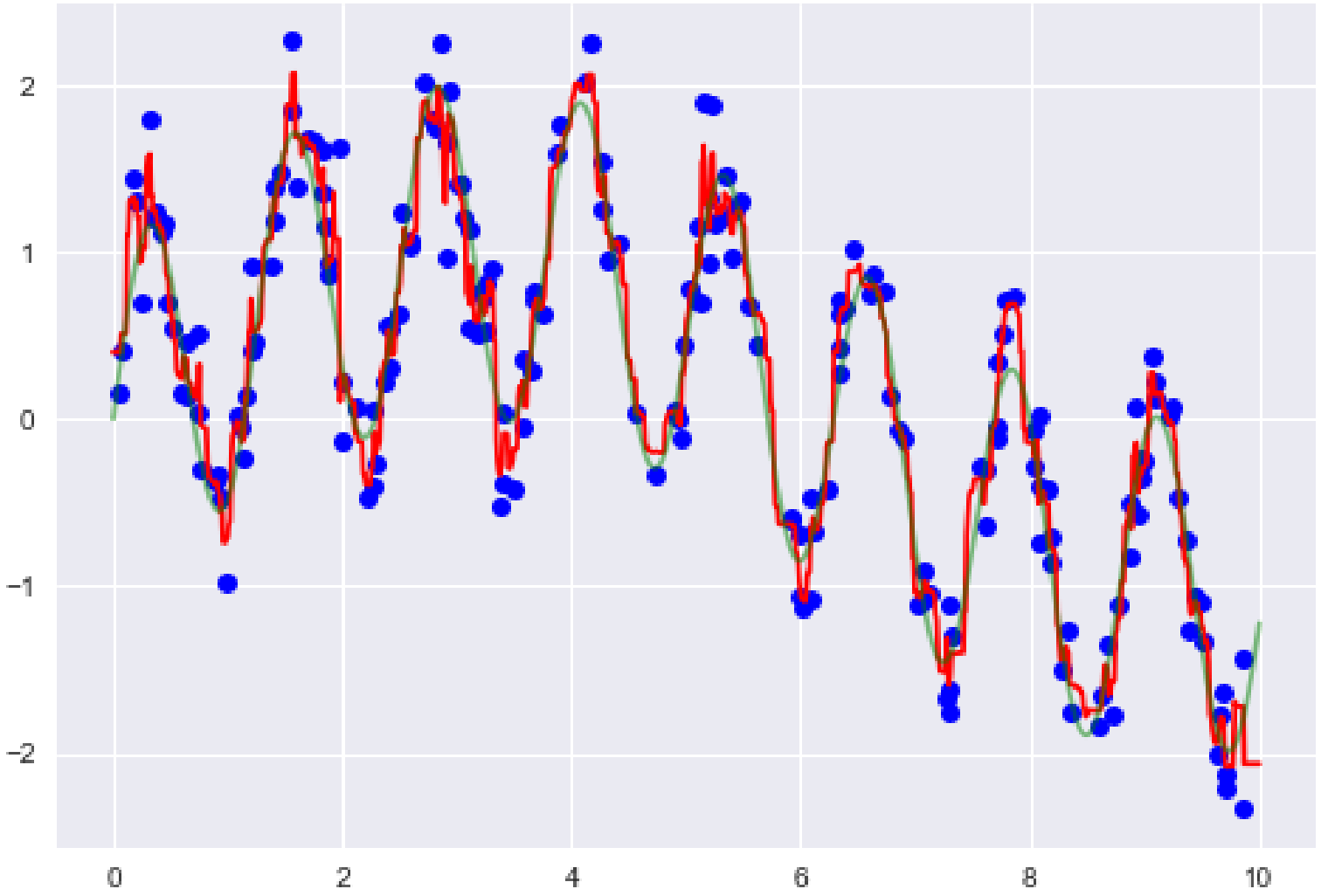

max_depth=3
Depth=2
x_1

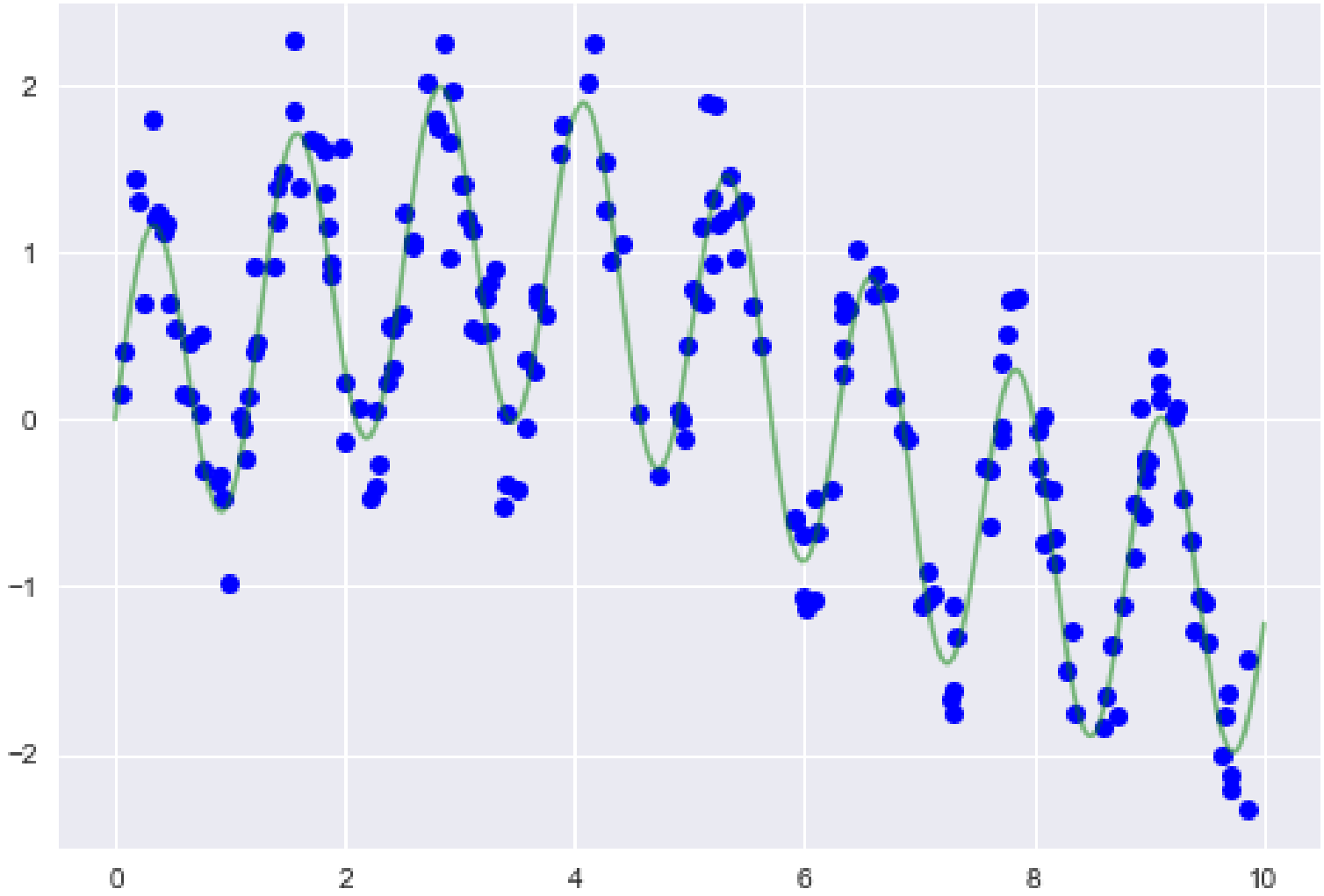

max_depth=2
$\hat{y}$
Depth=0
Depth=1
Depth=1
y
x_1


max_depth=10
x_1

A second regression example




2
1
0
−1
−2
0
2
4
6
8
10


2
1
0
−1
−2
0
2
4
6
8
10

Implementation in Scikit-Learn: code

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # petal length and width
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X, y)
```

Implementation in Scikit-Learn: results