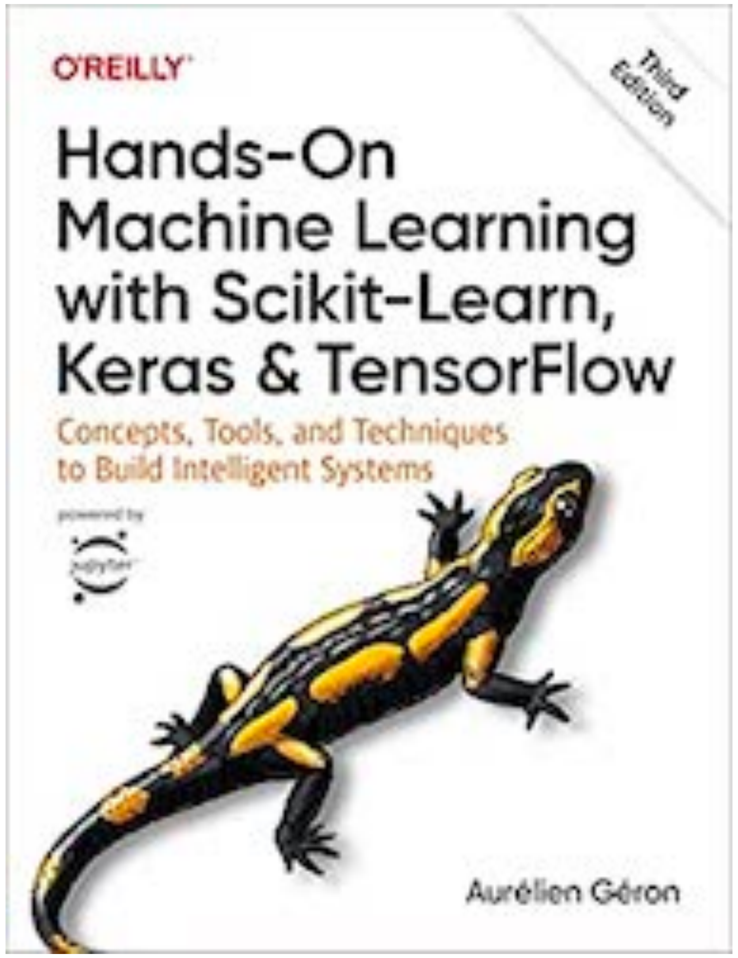
Machine Learning Security

6 Decision Trees



Updated 9-30-23

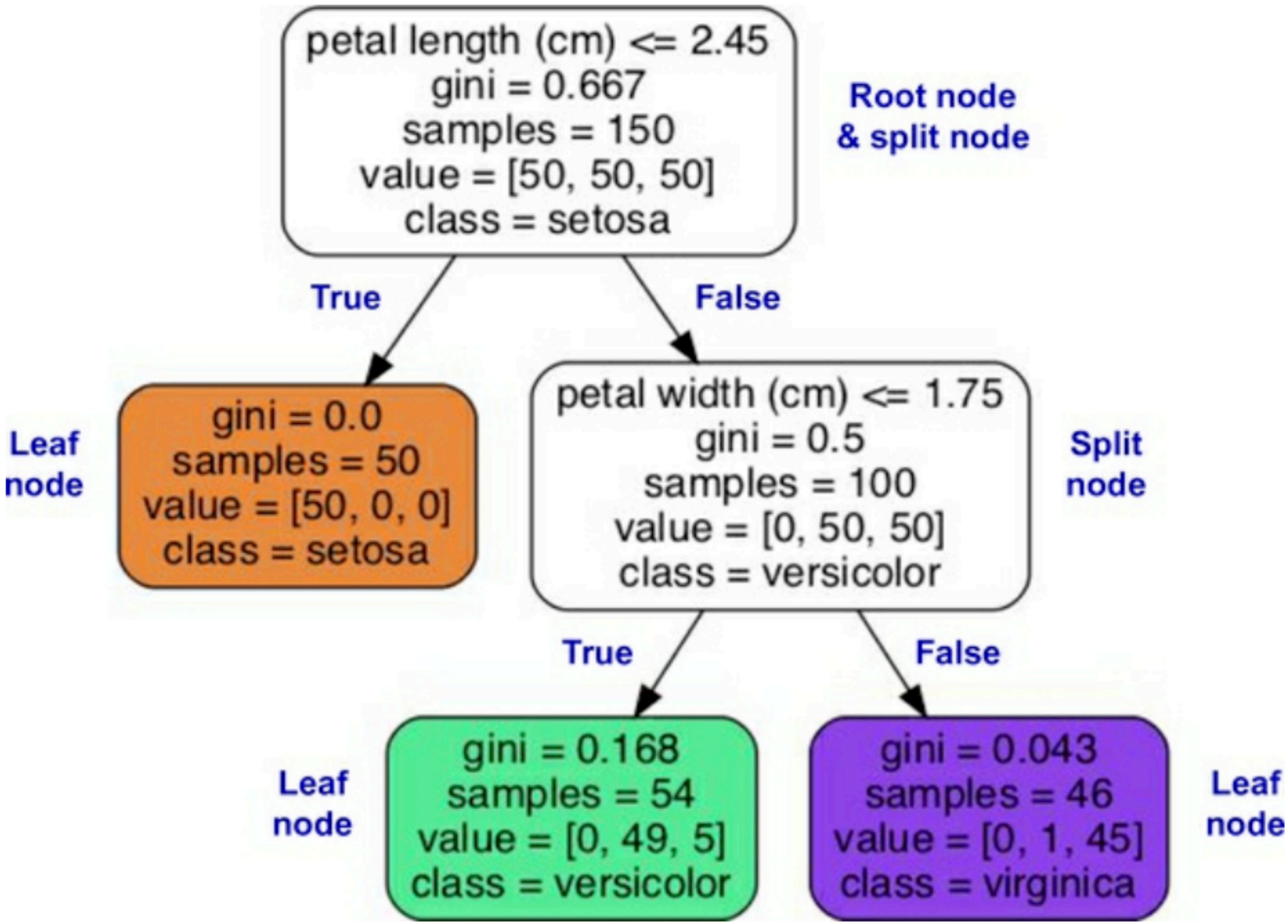
Topics

- Training and Visualizing a Decision Tree
- Making Predictions
- Estimating Class Probabilities
- The CART Training Algorithm
- Computational Complexity
- Gini Impurity or Entropy?
- Regularization Hyperparameters
- Regression
- Sensitivity to Axis Orientation
- Decision Trees Have a High Variance

Decision Trees

- A series of "if" statements
- Predictions are very fast
- Decisions are interpretable
- Can be combined to form powerful **random forests**

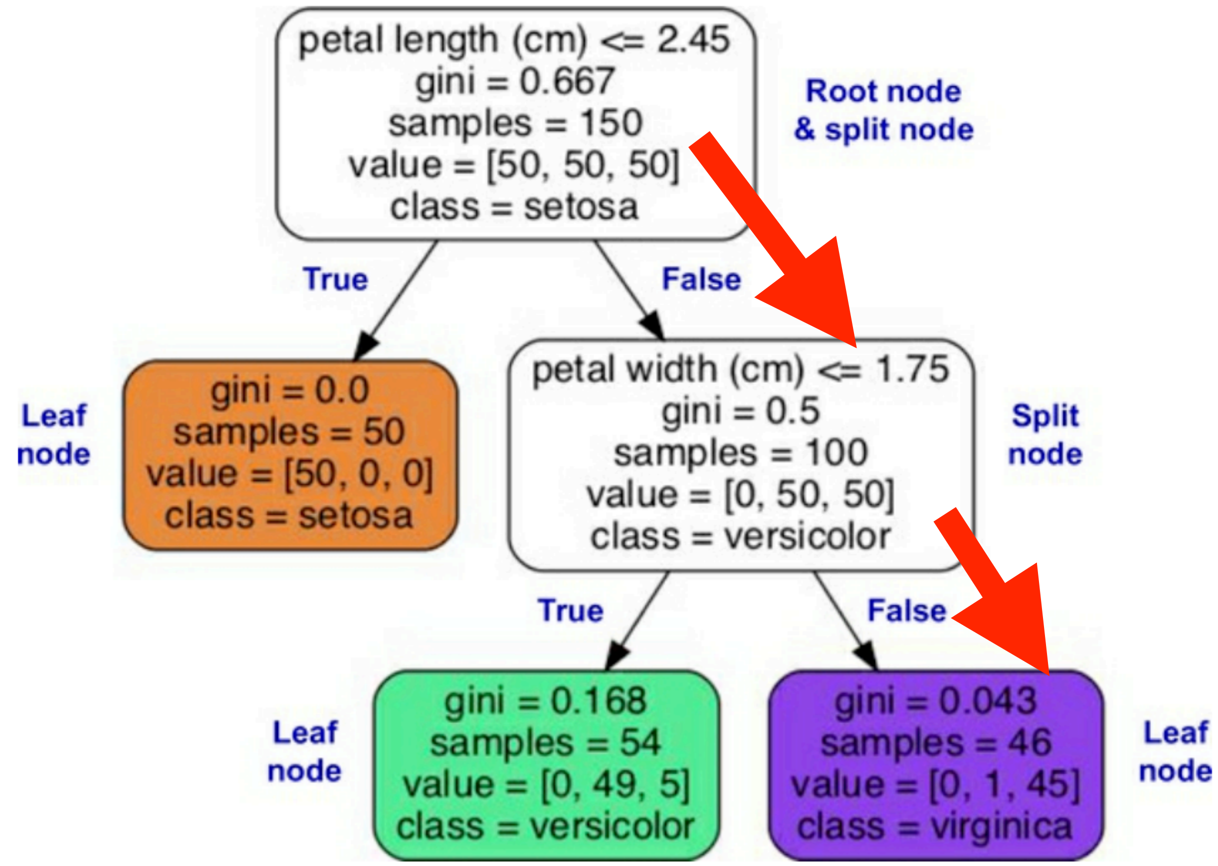
Training and Visualizing a Decision Tree


petal length (cm) <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
Root node
& split node
True
False
Leaf
node
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
Split
node
True
False
Leaf
node
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
Leaf
node

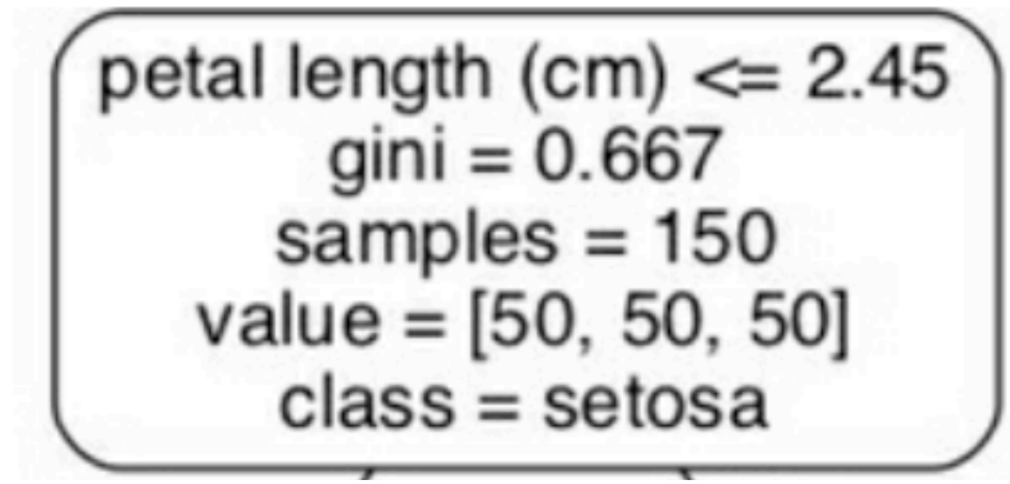
Making Predictions

Traverse the Tree

- Example
  - **petal length** = 3.0 cm
  - **petal width** = 2.0 cm
  - Result: **virginica**



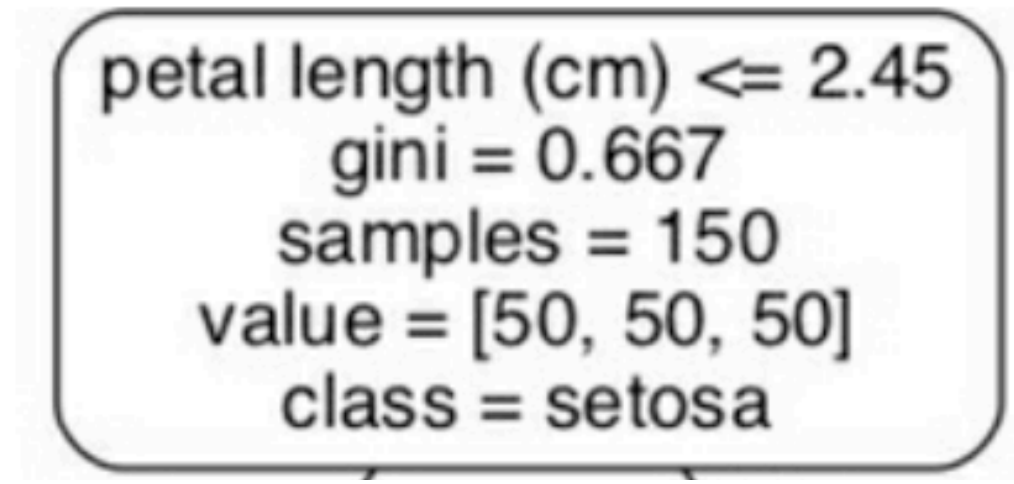
Samples and Value



- Samples
  - The number of training instances this node applies to
- Value
  - Count of instances in each class
- This node has 150 samples, 50 from each class

Gini Impurity

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

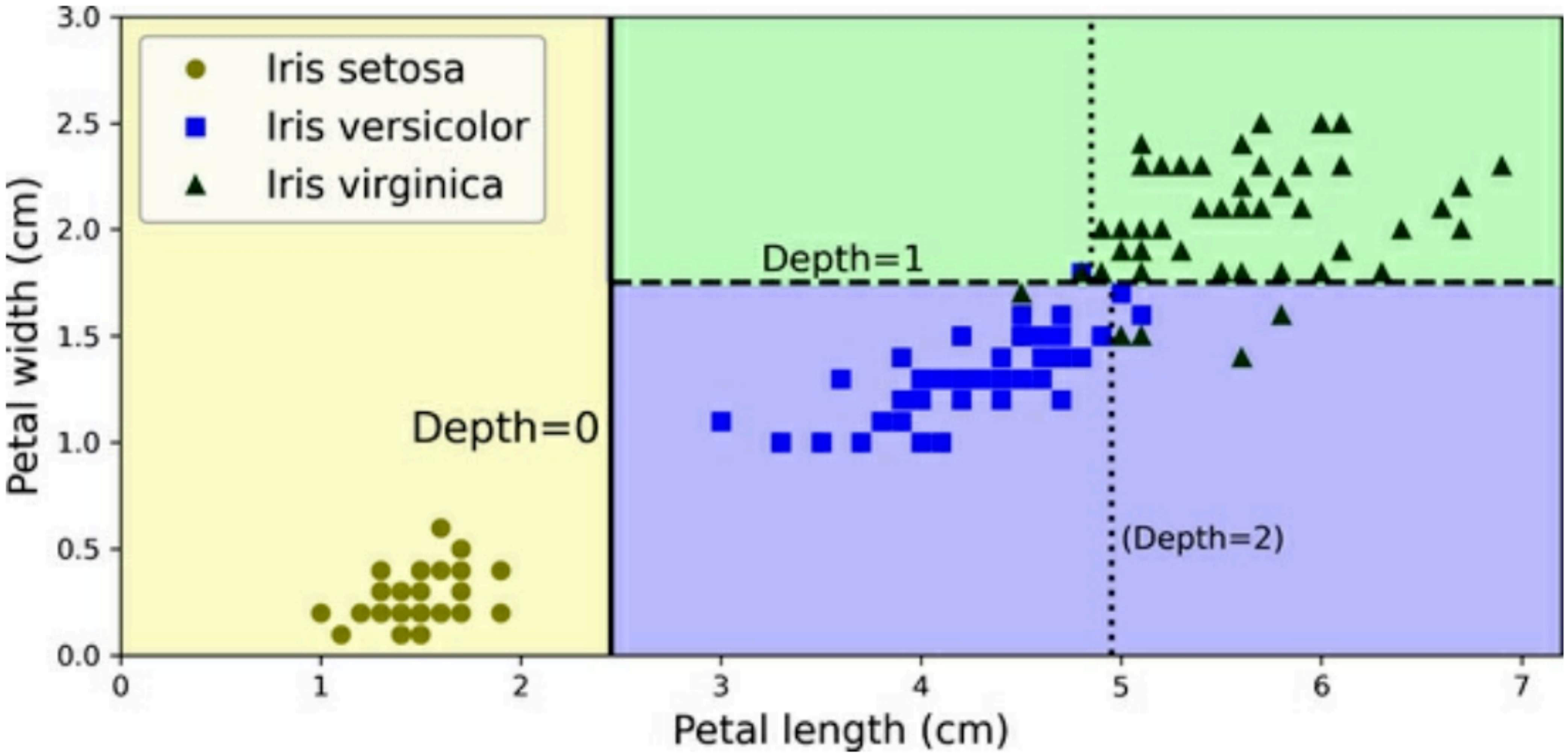


- Gini Impurity
  - 0 if all instances in this node are in the same class
  - Approaches 1 if many classes are present in this node with low probability
- This node has 50 samples from each class

  p = 1/3

  G = 1 - $(1/3)^2$ - $(1/3)^2$ - $(1/3)^2$ = 2/3

Decision Tree Boundaries



- The "Depth=2" lines are not present in our model

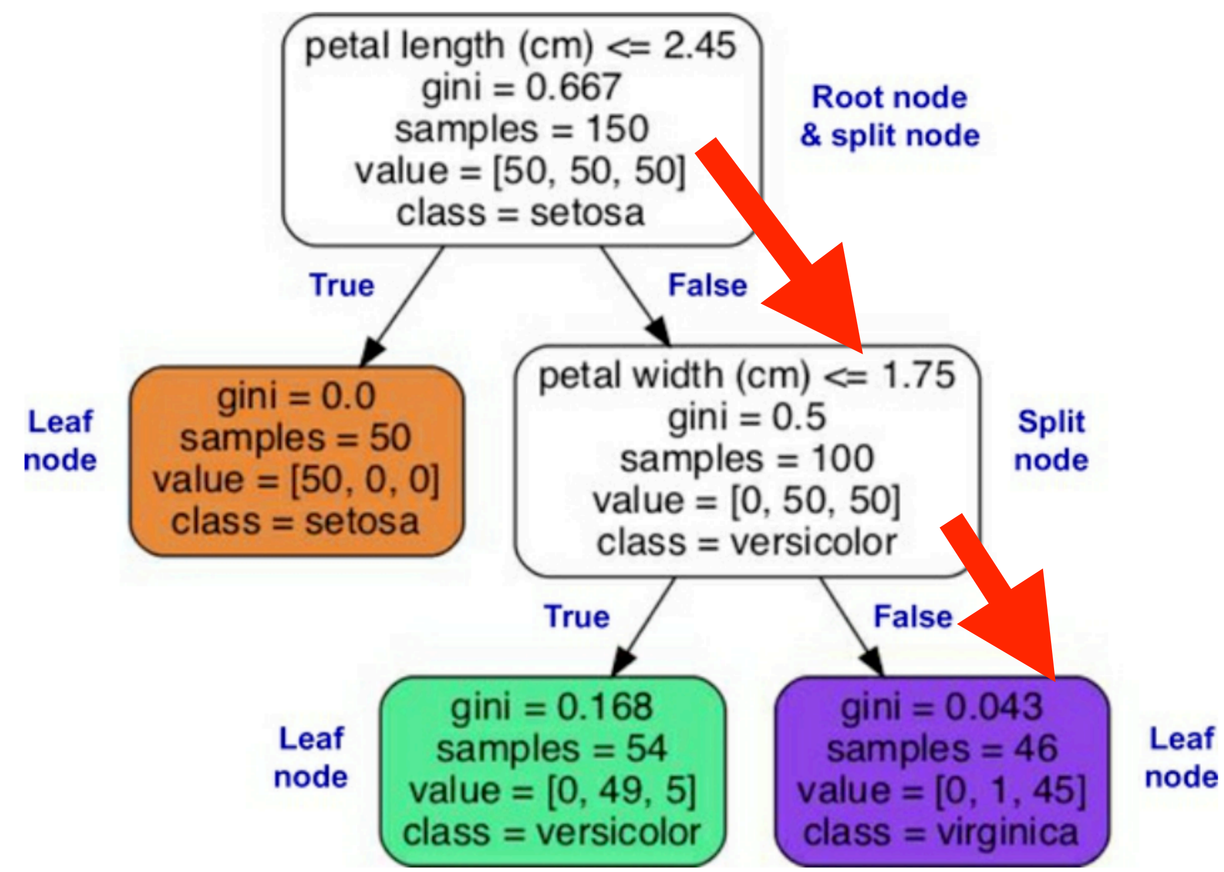
Interpretable ML

- Decision trees are **white box models**
  - It's easy to understand why they made their decisions
- Neural networks are **black box models**
  - No easy way to understand its decisions

Estimating Class Probabilities

Estimating Class Probabilities

- Example
  - **petal length** = 3.0 cm
  - **petal width** = 2.0 cm
  - Result: **virginica**
- What is the probability that this is actually virginica?
- Look in purple node
  - 45/46 instances were virginica
  - Probability = 45/46 = 98%



The CART Training Algorithm

The CART Training Algorithm

- Classification and Regression Tree (CART)
- First split the training set on a single feature ***k*** and threshold ***t***
  - Decision: ***k*** ≤ ***t*** ?
  - E. g. "petal length ≤ 2.45 cm"
- Choosing ***k*** and ***t***
  - Find values that produce the purest subsets
    - Weighted by size

CART Cost Function for Classification

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

$$\text{where} \begin{cases} G_{\text{left/right}} & \text{measures the impurity of the left/right subset} \\ m_{\text{left/right}} & \text{is the number of instances in the left/right subset} \end{cases}$$

- After splitting the root node, it splits those nodes, and their children, and so on
- Stops when it reaches maximum depth
  - Or when it cannot reduce impurity
- It's a **greedy** algorithm--it only maximizes the value of the current split. It does not look ahead to future splits.

Computational Complexity

Computational Complexity

- Traversing the decision tree
  - $\boldsymbol{O}(\log_2(\boldsymbol{m}))$ where there are $\boldsymbol{m}$ training instances
  - The number of features, $\boldsymbol{n}$, doesn't matter
- Training
  - $\boldsymbol{O}(\boldsymbol{n} \times \boldsymbol{m} \log_2(\boldsymbol{m}))$ where there are $\boldsymbol{m}$ training instances

Kahoot!

Ch 6a

Gini Impurity or Entropy?

Shannon Entropy

$$H_i = -\sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} \log_2(p_{i,k})$$

- Another measure of impurity
- In practice, either Gini impurity or entropy can be used
- The trees will be similar

Regularization Hyperparameters

Nonparametric Model

- Consider a linear or polynomial model
  - It makes an assumption about the data
  - Has a fixed number of parameters
  - These are **parametric models**
- Decision trees don't assume a shape for the data
  - Don't have a fixed number of parameters
  - Can grow as complex as needed
  - Can overfit the data
  - Can be regularized with hyperparameters

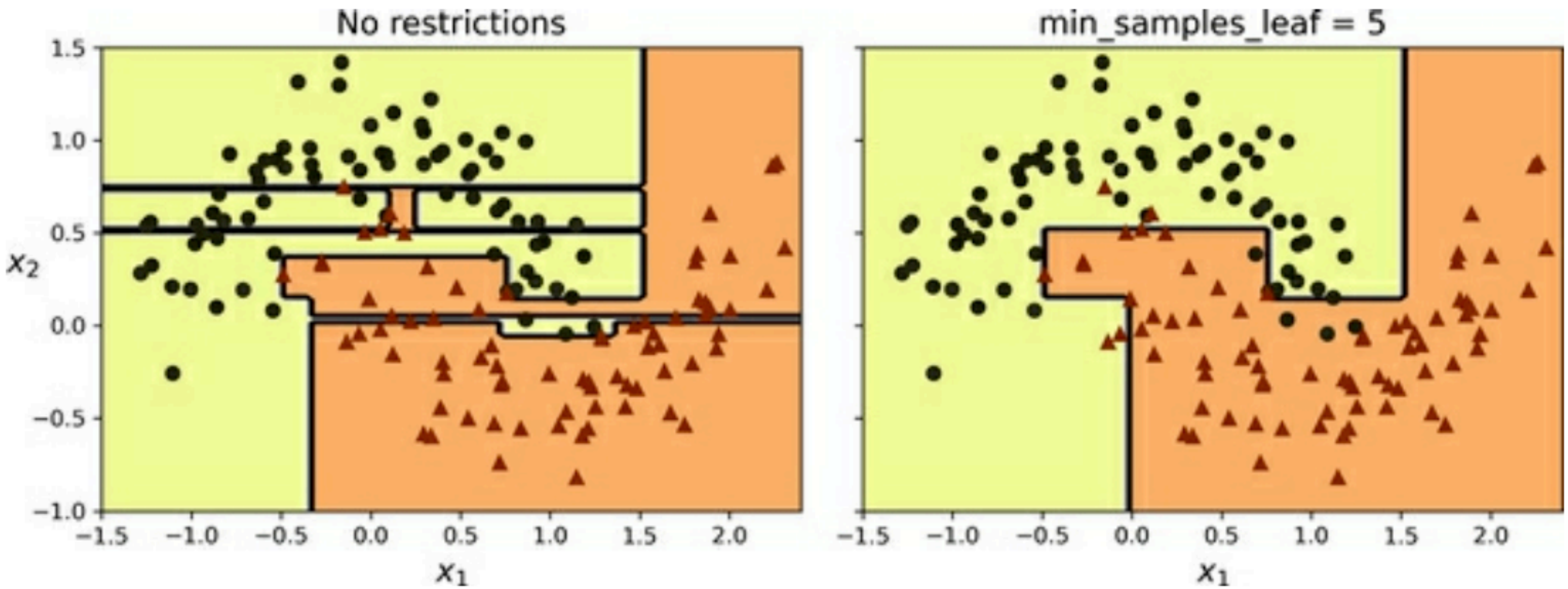
CART Hyperparameters

- **max_depth**
- **max_features**
  - Maximum number of features evaluated for splitting at each node
- **max_leaf_nodes**
- **min_samples_split**
  - Minimum number of samples a node must have before it can split
- **min_samples_leaf**
  - Minimum number of samples a leaf node must have to be created
- **min_weight_fraction_leaf**
  - Same as **min_weight_samples_leaf** but expressed as a fraction

Pruning

- Other algorithms first train the decision tree without restrictions
- Then **prune** it, deleting unnecessary nodes
- A node is unnecessary if
  - The purity improvement it provides is not statistically significant
  - Using standard statistical tests, like chi-squared

Effect of Regularization



Regression

Decision Tree for Regression

- Instead of predicting a class, it predicts a value

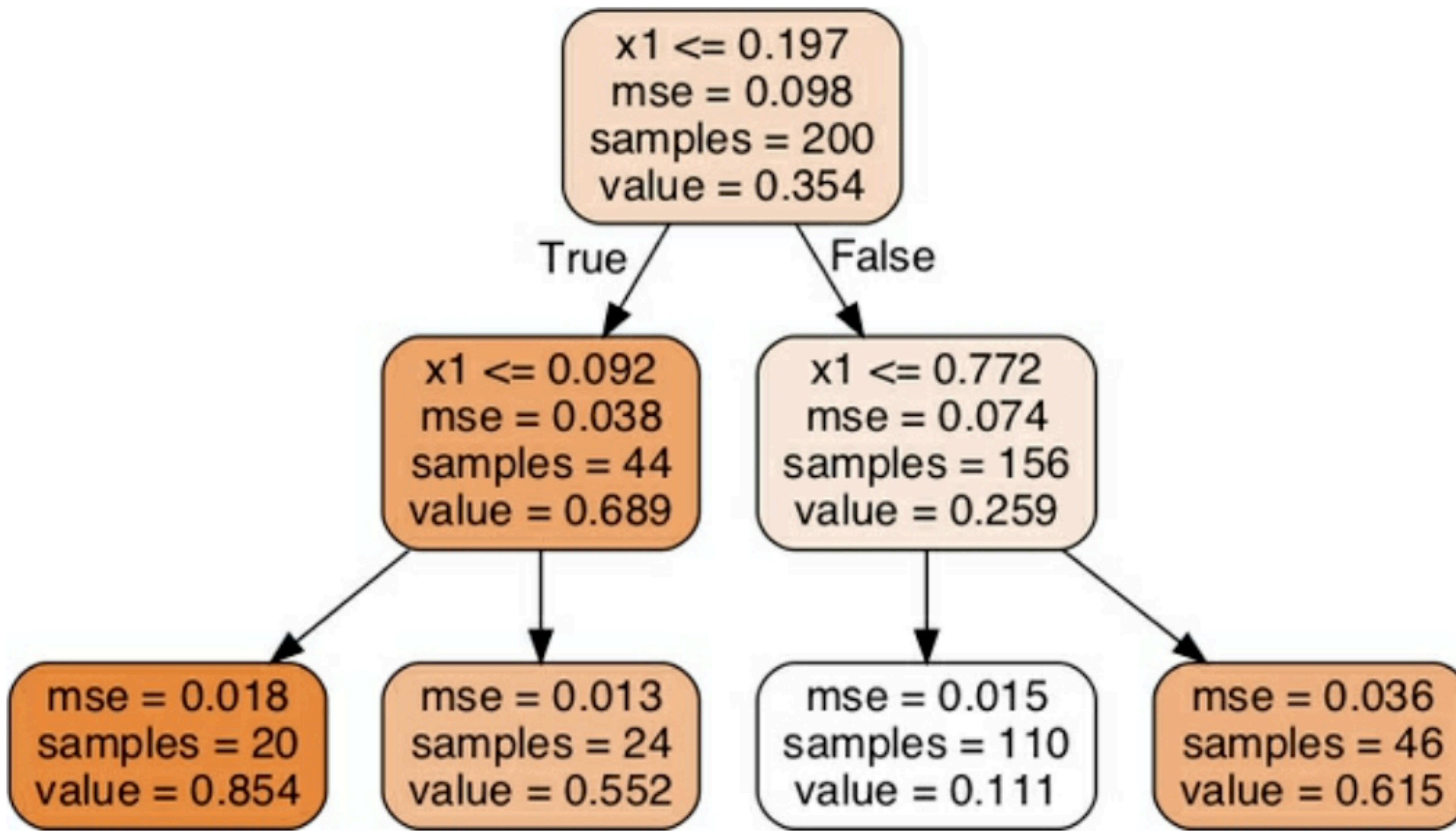
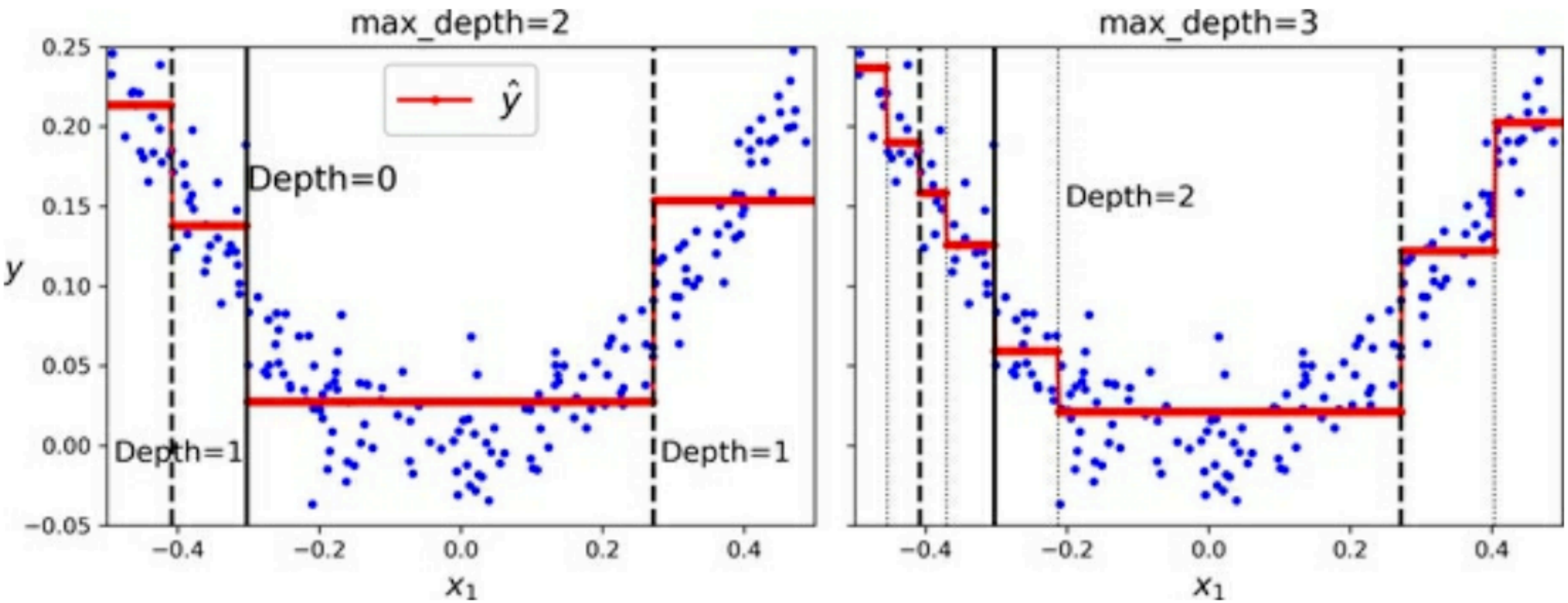


Figure 6-4. A decision tree for regression

Decision Tree for Regression



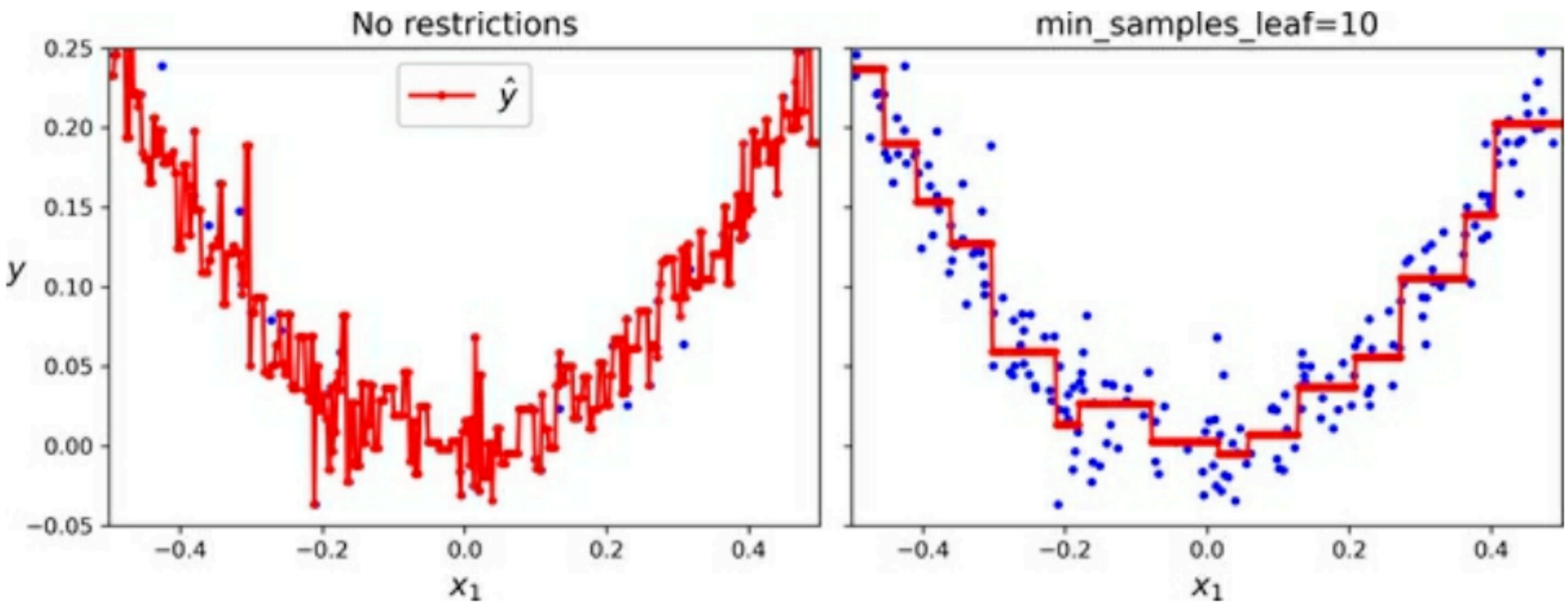
- Models curve as a series of horizontal lines

CART Cost Function for Regression

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \frac{\sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2}{m_{\text{node}}} \\ \hat{y}_{\text{node}} = \frac{\sum_{i \in \text{node}} y^{(i)}}{m_{\text{node}}} \end{cases}$$

- Uses Mean Squared Error instead of impurity

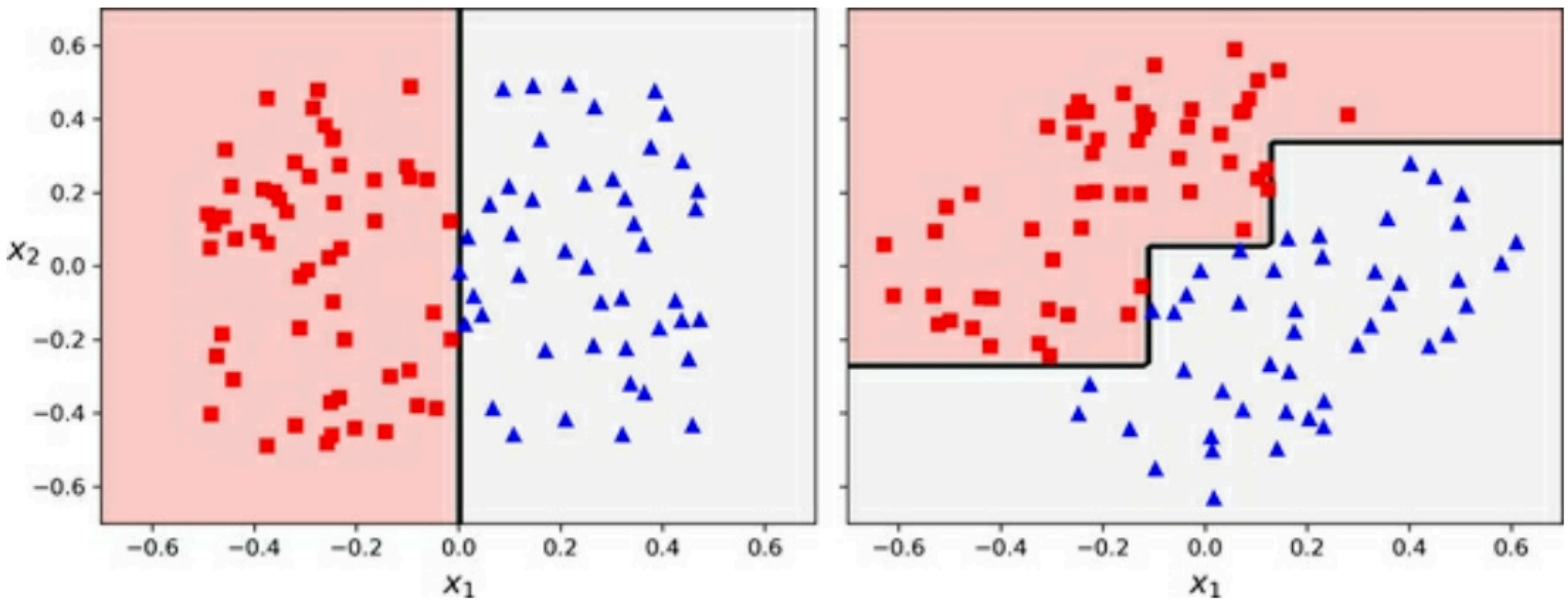
Overfitting



- Regularization is needed

Sensitivity to Axis Orientation

Axis Orientation

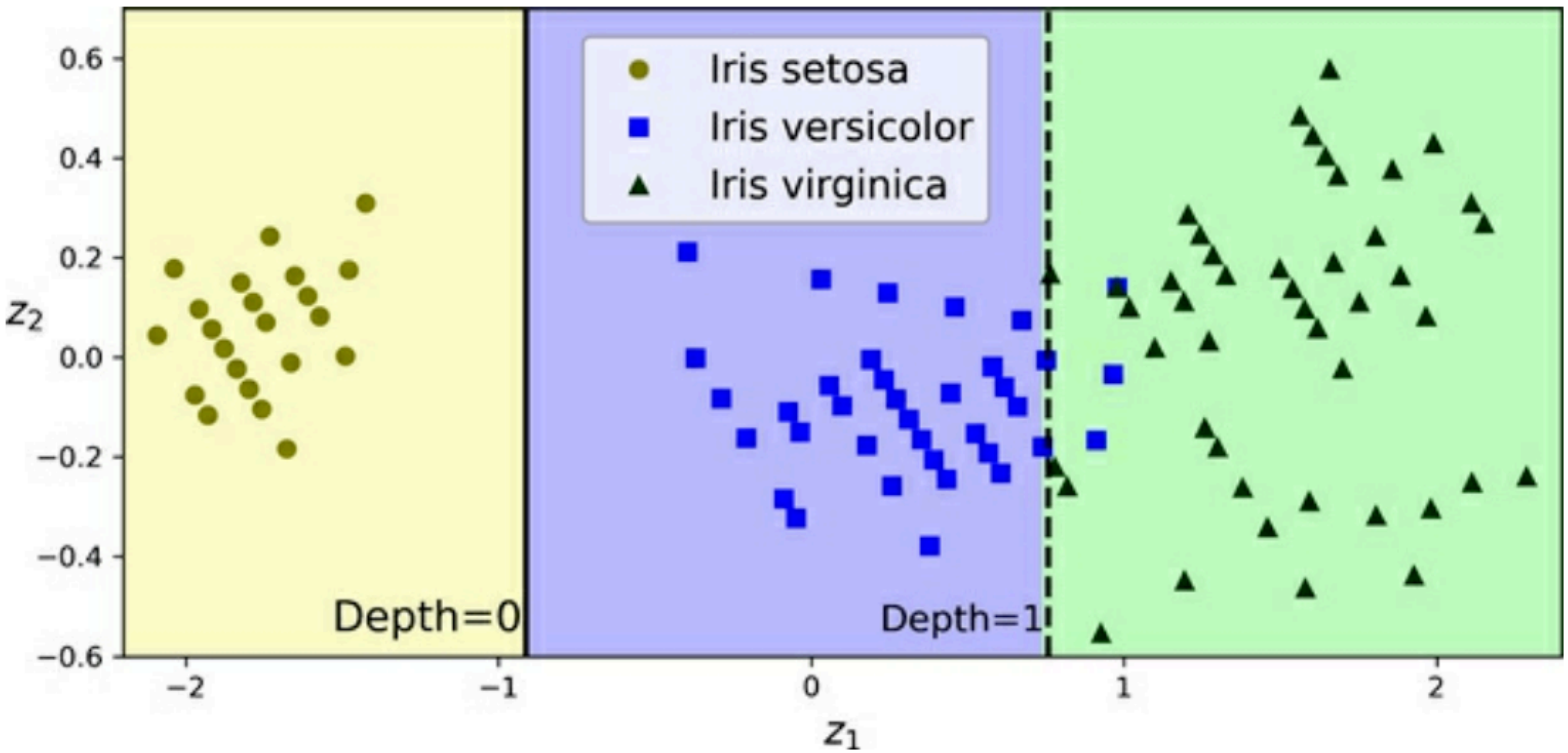


- It can only use horizontal or vertical lines
- Rotating data by 45 degrees makes the model less efficient

PCA Transformation

- Principal Component Analysis Transformation
  - Rotates data in a way that
    - Reduces the correlation between the features
- Usually makes things easier for decision trees

Result of PCA Rotation



- After scaling and PCA rotating, the iris dataset can be fit with a single feature

Decision Trees Have a High Variance

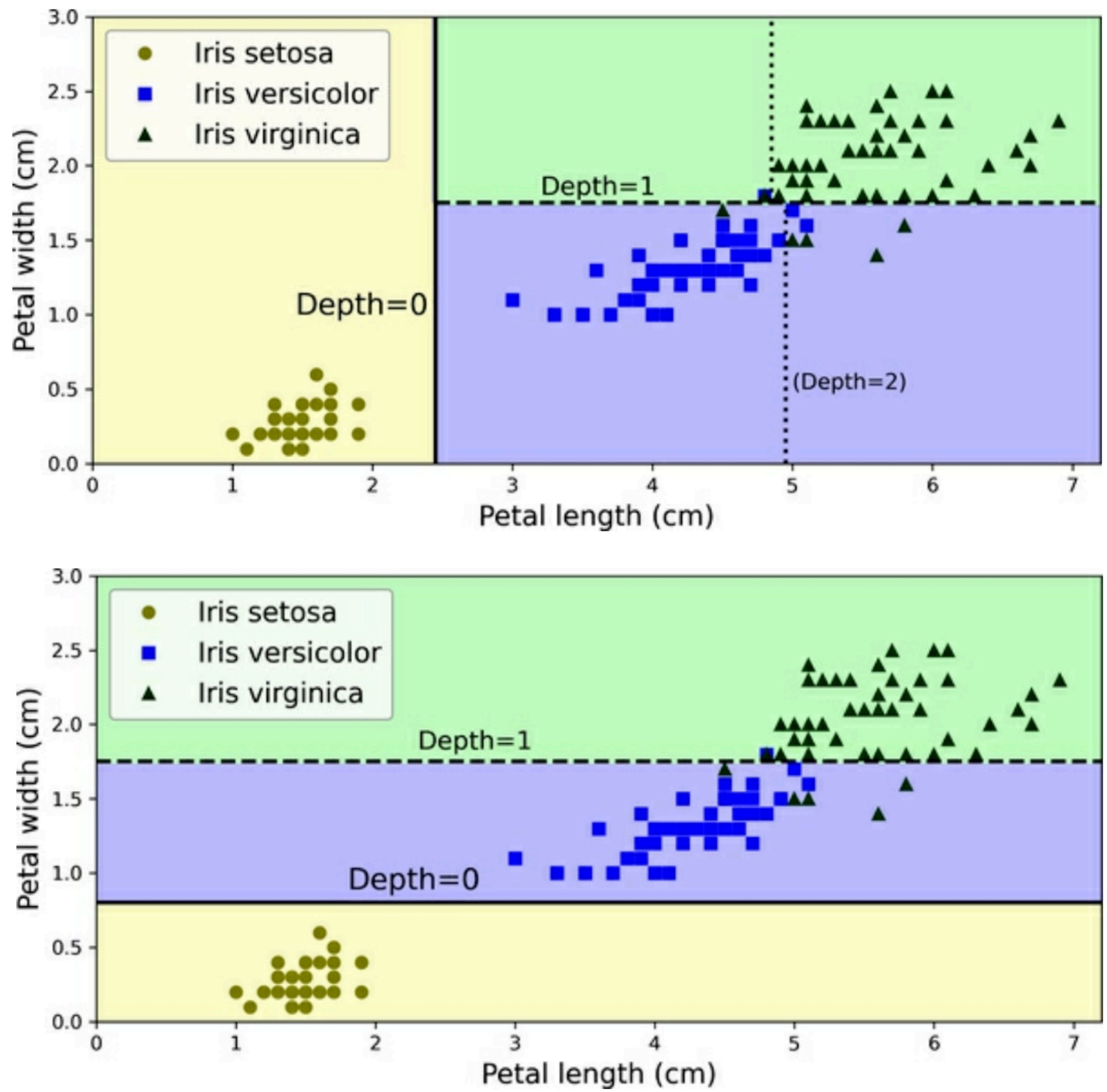
Variance

- Small changes to hyperparameters or data
  - May produce very different decision tree models
- Even repeating the Scikit-learn fit can come out different
  - Because it chooses features to evaluate randomly

Random Forest

- Average predictions over many decision trees
- Reduces variance
- One of the most powerful models available today

Retraining the Same Model



Kahoot!

Ch 6b