Hands on Machine Learning 2nd Edition

Chapter 6 – Decision Trees

Prepared by Glenn Miller for the San Diego Machine Learning Meetup™ group

Decision Tree (DT) + / -

ADVANTAGES

- Simple to understand and interpret (*'White Box'* model)
- Little data prep (e.g. no scaling)
- Versatile (classification and regression)
- Cost of using the tree is logarithmic in # of data points used to train the tree
- Can handle multi-output problems
- Can validate using statistical tests
- Performs well even if its assumptions are somewhat violated

DISADVANTAGES

- Prone to overfitting (must restrict degrees of freedom)
- Can be unstable (small data variations produce big changes to the tree)
- Predictions are piecewise constant approximations (not smooth or continuous)
- DT learners create biased trees if some classes dominate
- Practical DT algos cannot guarantee to return the globally optimal DT b/c learning an optimal DT is <u>NP-complete</u>

Source: Scikit-Learn documentation

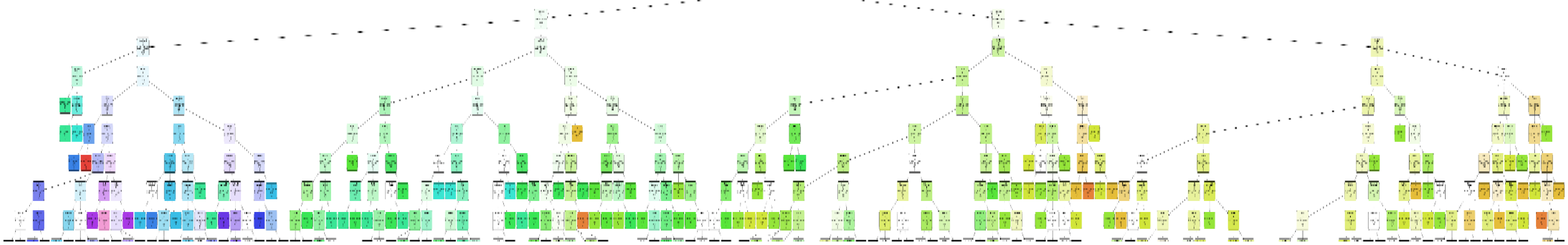
Computational Complexity

- Predictions are fast, even with large training sets
- Algorithm compares all *available** features on all samples at each node
- Big O – $O(n \text{ x } m\log_2(m))$
- Presorting the data (presort=True) can speed up training for small data sets

**If max_features is set, the algorithm will consider max_features features at each split, subject to a minimum one valid partition of the node samples*

Regularization and Pruning

Regularize: prevent the tree from growing too large (like the tree below) by limiting parameter(s) before growing the tree (e.g., set max_depth or min_samples_leaf)



Prune: let the tree grow and then replace irrelevant nodes with leaves

Impurity

Gini impurity index

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

Where $P_{i,k}$ is the ratio of *k* instances among the training instances in the i^{th} node

Entropy / Information Gain

- $H_i = - \sum_{k=1}^{n} P_{i,k} log_2(P_{i,k})$

CART Algorithm (Scikit-Learn)

Splits the training set into two subsets using a single feature (k) and a threshold (t_k)

Classification (DecisionTreeClassifier)

- Predict a *class* in each node
- Minimize impurity
- Cost function:

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

Regression (DecisionTreeRegressor)

- Predict a *value* in each node
- Minimize MSE
- Cost function:

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

G is impurity of the subset; m is number of instances in the subset

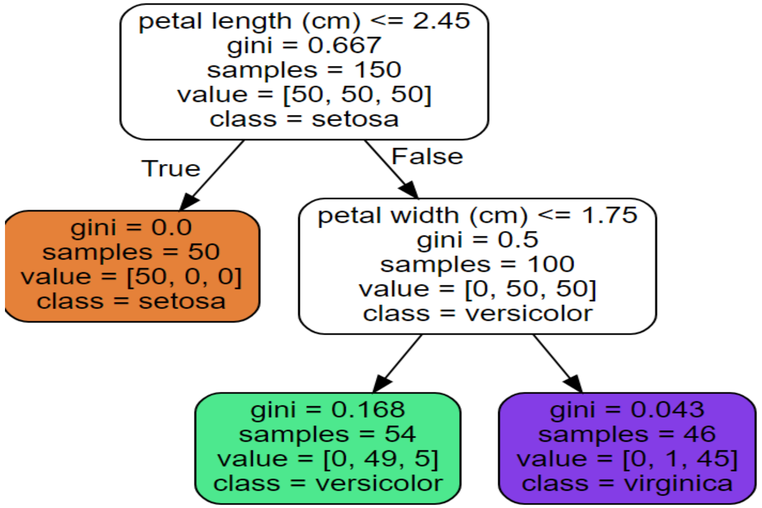
(Some) Iris Species

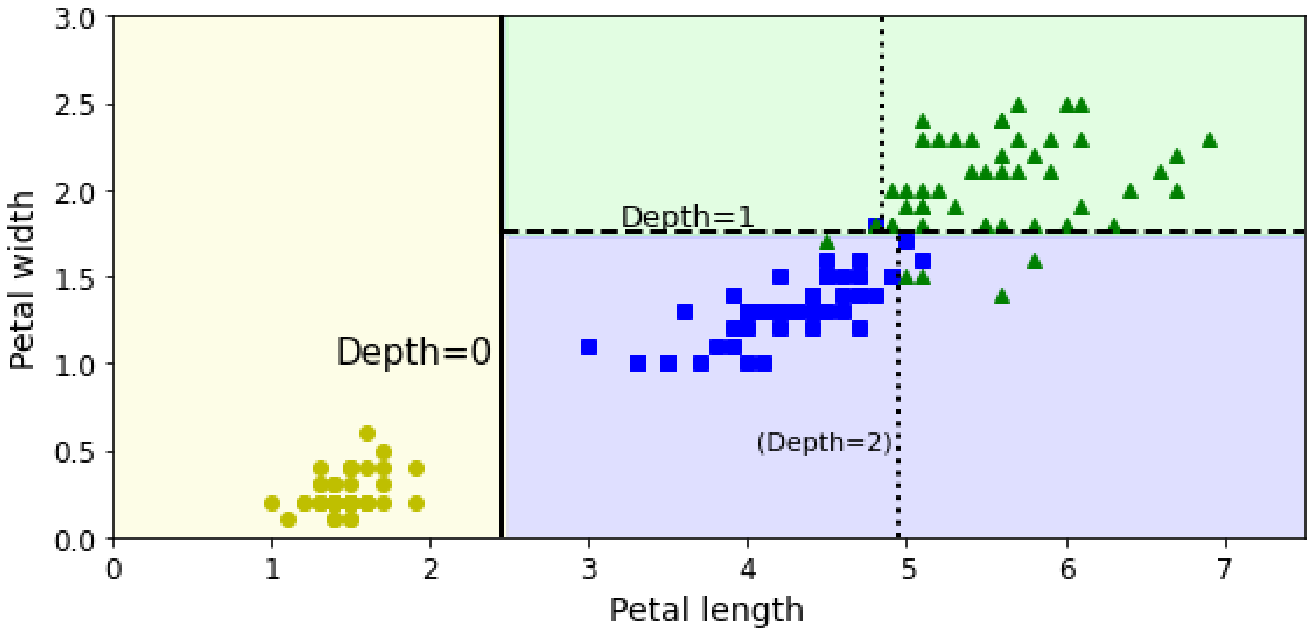
- Setosa

- Versicolor

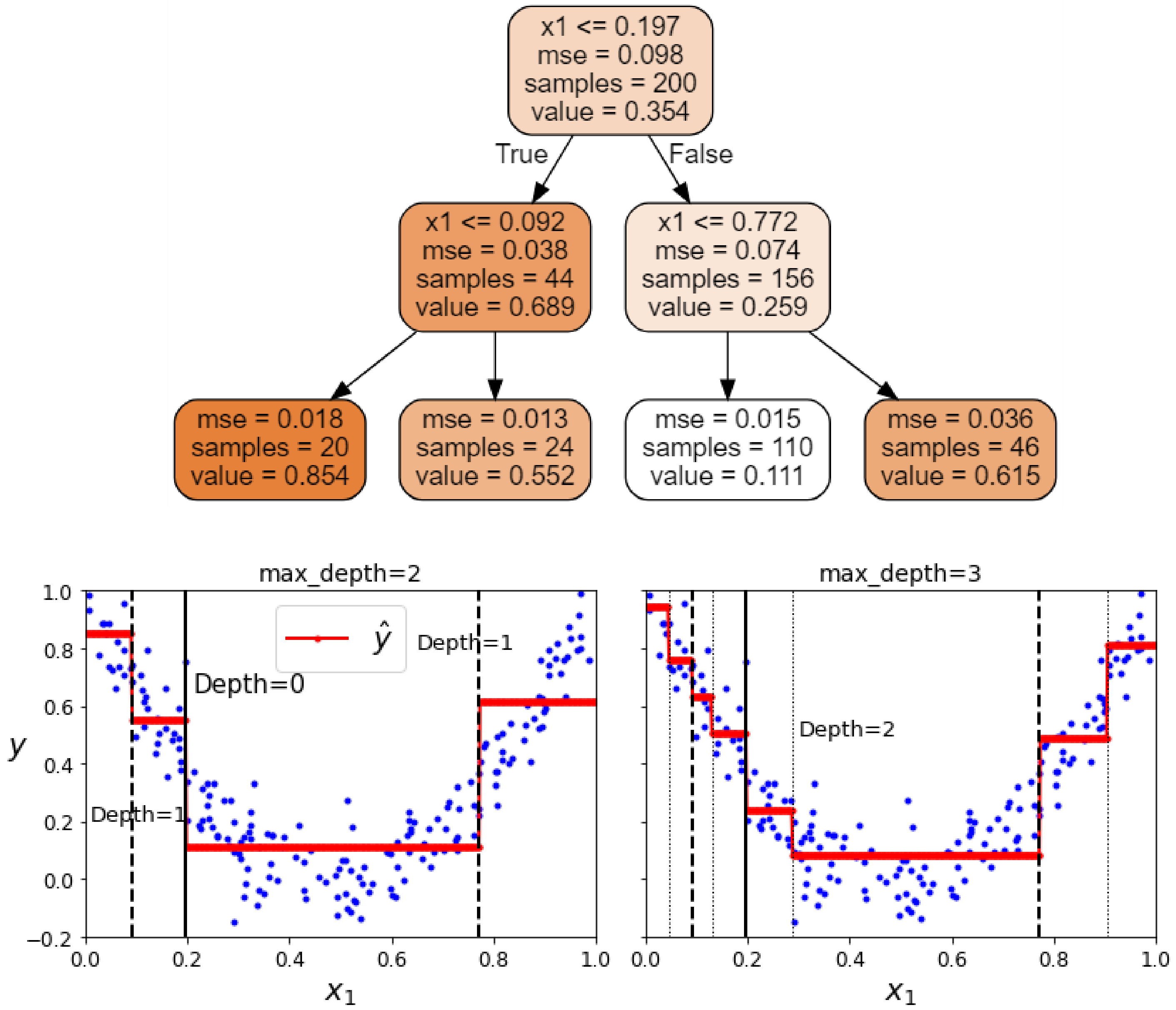
- Versicolor

Classification



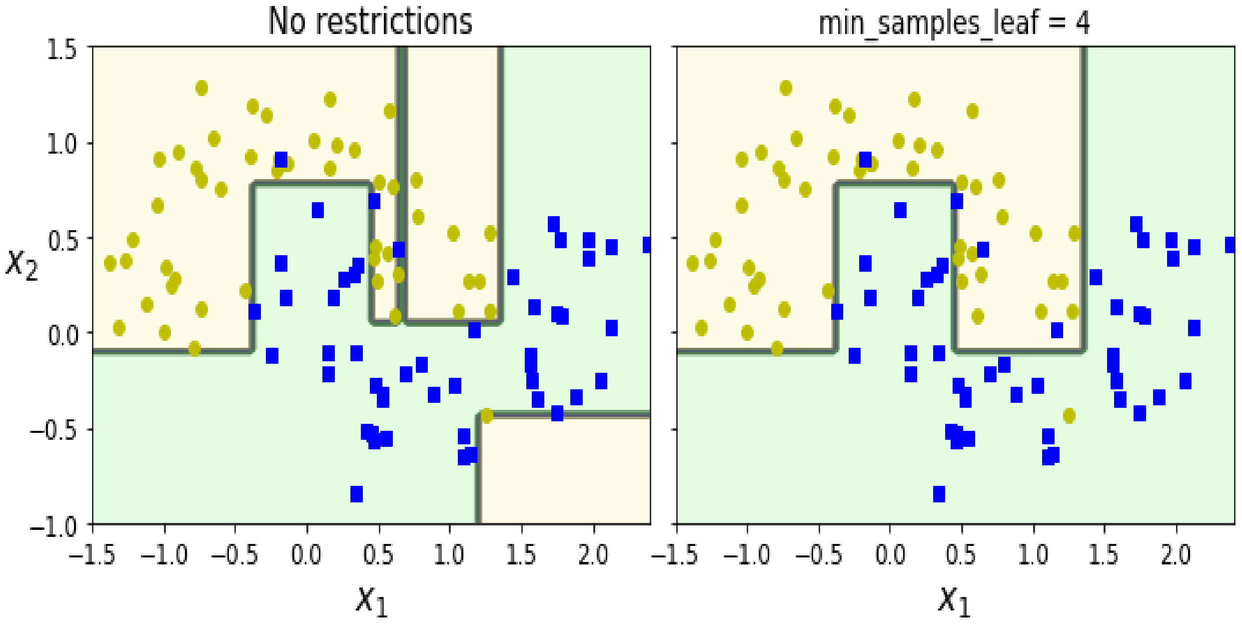


Regression

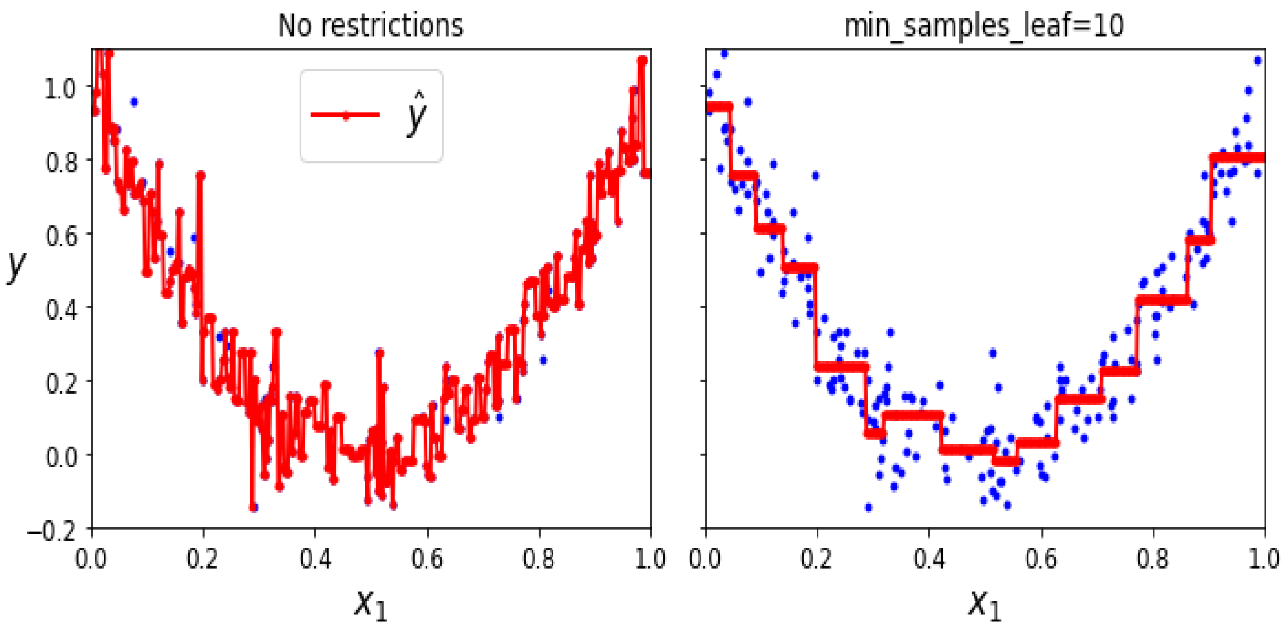

x1 <= 0.197
mse = 0.098
samples = 200
value = 0.354
True
False
x1 <= 0.092
mse = 0.038
samples = 44
value = 0.689
x1 <= 0.772
mse = 0.074
samples = 156
value = 0.259
mse = 0.018
samples = 20
value = 0.854
mse = 0.013
samples = 24
value = 0.552
mse = 0.015
samples = 110
value = 0.111
mse = 0.036
samples = 46
value = 0.615
max_depth=2
max_depth=3
ŷ
Depth=0
Depth=1
Depth=1
Depth=2
y
x1
x1

Regularization / Pruning

Classification



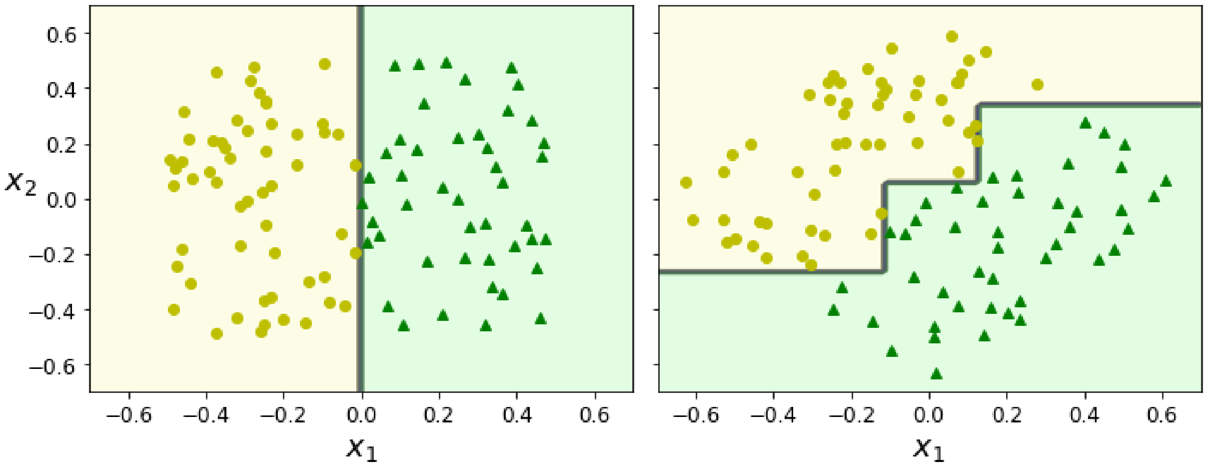
Regression



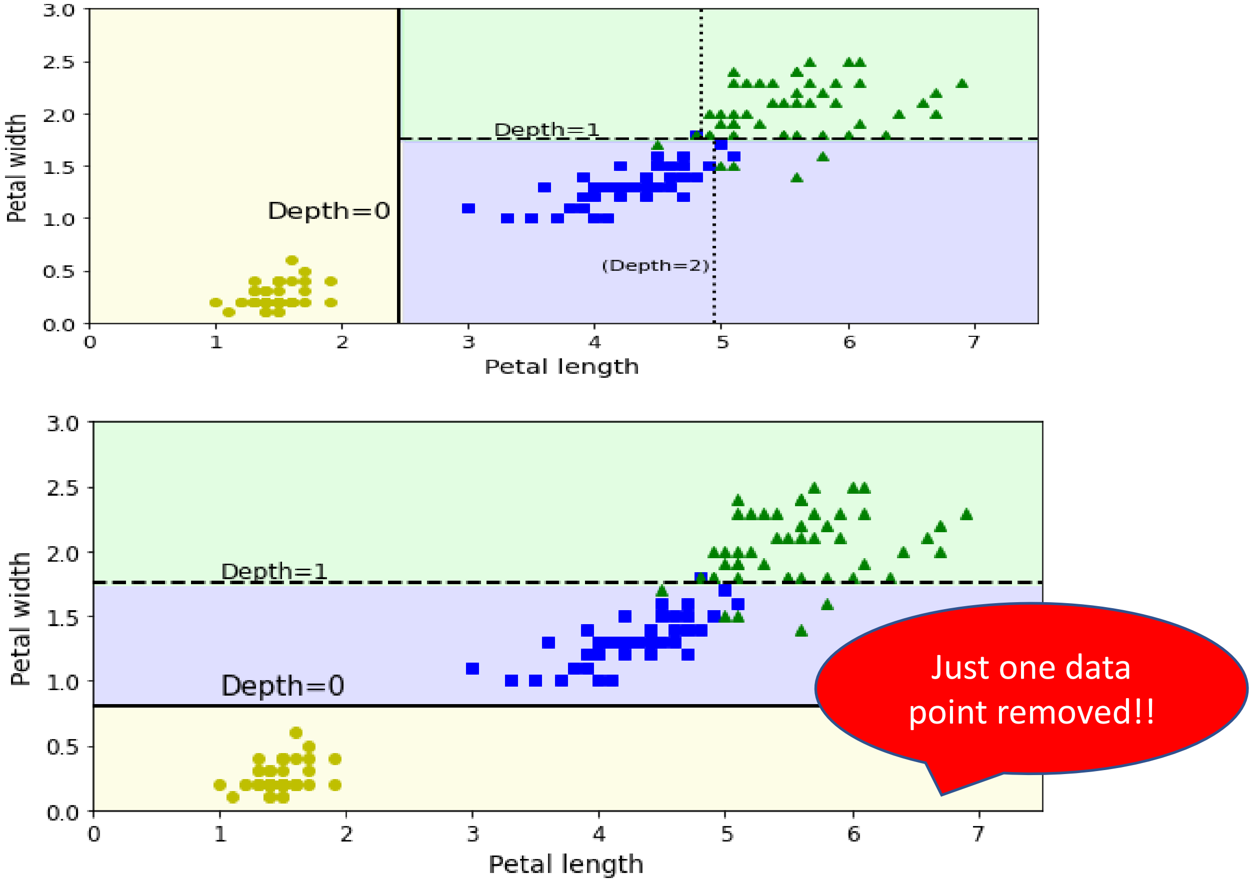
Source: HOML 2nd edition pp. 182,184 / https://github.com/ageron/handson-ml2/blob/master/06_decision_trees.ipynb

Instability

Sensitivity to training set rotation



Sensitivity to training set details



Source: HOML 2nd edition pp. 185,186 / https://github.com/ageron/handson-ml2/blob/master/06_decision_trees.ipynb

Conclusion

- Decision trees are powerful, versatile, and easy to understand
- They have limitations, which can be addressed – e.g. averaging trees reduces instability (random forests)
- More on this in the chapter 7, ‘Ensemble Learning and Random Forests’