KIG4068 : Machine Learning

Week 7: Decision Trees

Semester 2, Session 2023/2024

Decision Trees

What are decision trees?

- Versatile ML algorithm: supports both classification and regression, and multioutput tasks.
- Powerful ML algorithm: capable of fitting complex datasets with lots of non-linear relationships.
- Robust ML algorithm: often requiring very little data preparation.
- Used in many ensemble modeling pipelines, including random forests.

What are we going to learn today?

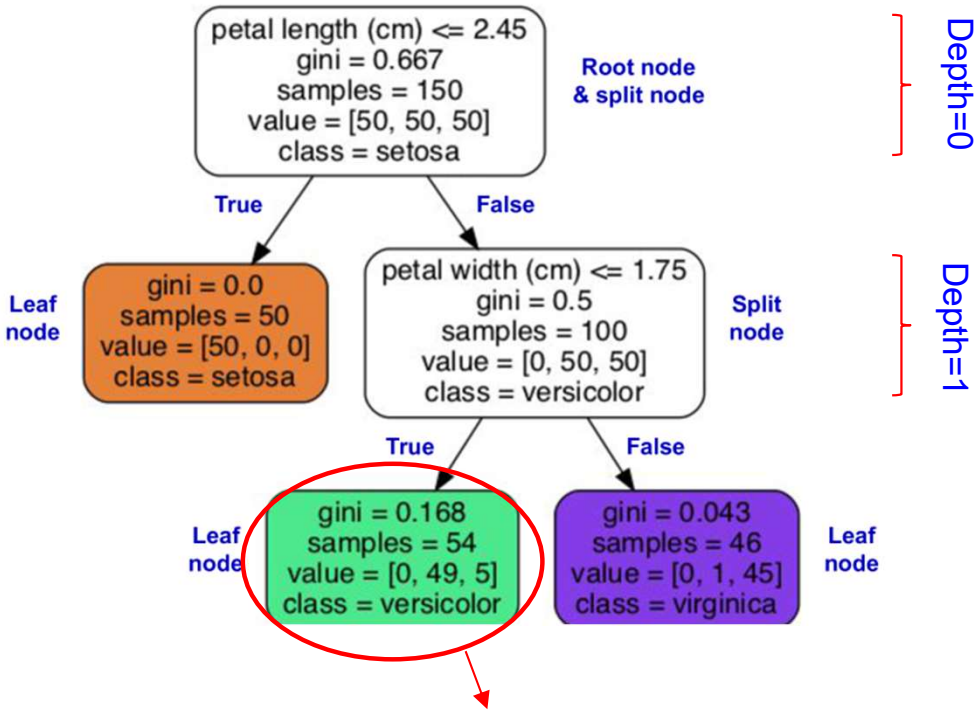
- How to train, visualize, and make predictions with decision tree
- Go through Classification And Regression Tree (CART) training algorithm used by Scikit-Learn
- Explore how to regularize trees and use them for regression tasks
- Discuss some of the limitation of decision trees.

Decision Trees (Cont'd)

How to make predictions?

Start at the top and work down!

- To make predictions start at the root node and work down to a leaf node using each node's decision criterion.
- **samples**: number of training instances that node contains.
- **value**: number of training instances per class that node contains.
- **gini/entropy**: impurity measure; node is "pure" (gini=0) when all training instances belong to same class.
- Can also use **value** and **samples** to estimate class probabilities.



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

G_i = Gini impurity of the i^{th}node
$P_{i,k}$= ratio of class k instances among the training instances in the i^{th}node

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

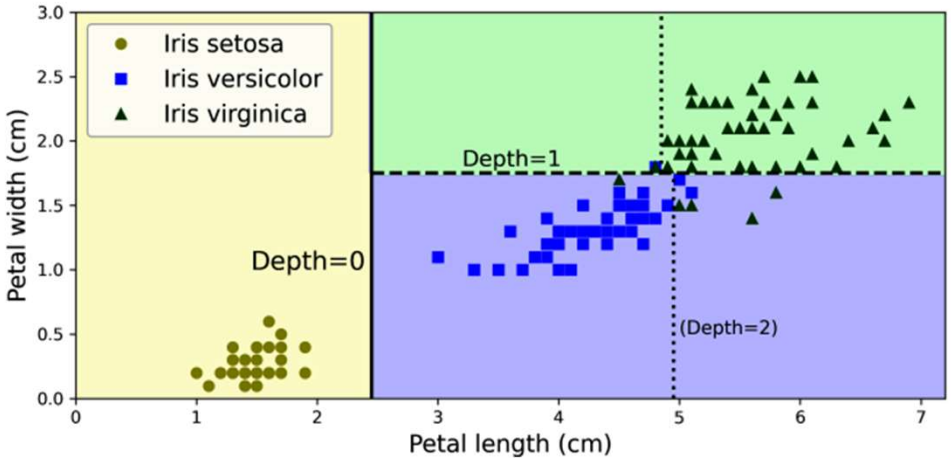
Iris decision tree

Decision Trees (Cont'd)

Understanding decision boundaries

- Depth=0 boundary is determined by root node decision criterion.
- Depth=1 boundary is determined by decision criteria of nodes at depth=1 in the tree.
- Max depth of this tree was 2, but if max depth was 3 then Depth=2 boundary is determined by decision criteria of nodes at depth=2 in the tree.

Typical decision boundaries



Pure area, only Setosa, no more split required

Impure area, we use petal width feature in the depth-1 split node. Since max_depth was set to 2, the decision tree stops here.

Decision tree decision boundaries

The CART Algorithm

Scikit-Learn uses Classification And Regression Tree (CART) algorithm; a greedy algorithm that trains decision trees by "growing" them.

1. Splitting Process
   - CART starts by dividing the training set into two subsets based on a single feature k and threshold t_k(e.g., “petal length ≤ 2.45 cm).
   - It searches for the feature k and threshold t_k pair that produces the purest subsets, weighted by their size.

2. Cost Function
   - Use algorithm aims to minimize the cost function

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

$G_{left/right}$ measures the impurity of the left/ right subset

$m_{left/right}$ is the number of instances in the left/ right subset

The CART Algorithm

3. Recursive Splitting
   - After successfully splitting the training set, CART repeats the process recursively on the subsets.
   - It stops when reaching the maximum depth or if no further split reduces impurity

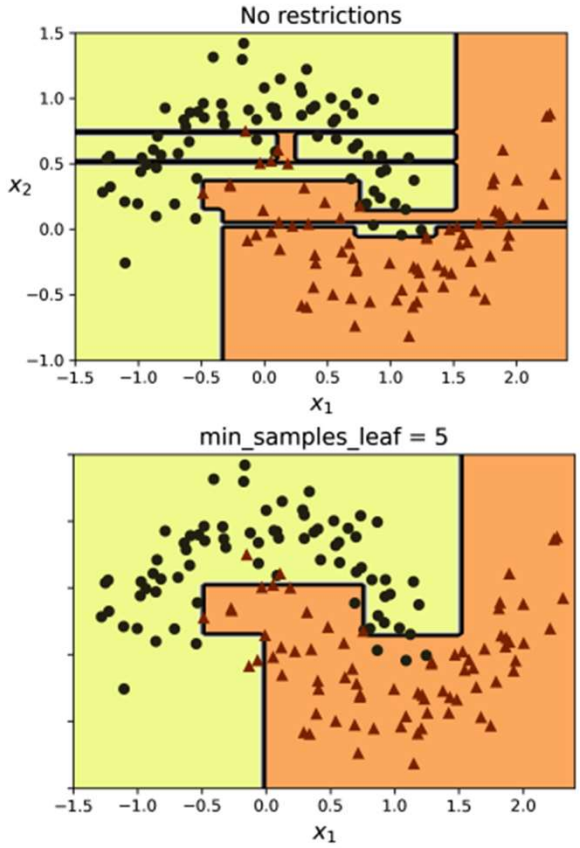
4. Hyperparameters
   - Additional stopping conditions are controlled by hyperparameters:
     I. ‘max_depth’
     II. ‘min_samples_split’
     III. ‘min_samples_leaf’
     IV. ‘min_weight_fraction_leaf’
     V. ‘max_leaf_nodes’

Decision Trees (Cont'd)

Regularization Hyperparameters

- Decision trees make almost zero assumptions about the data.
- Few assumptions => few constraints, few constraints => prone to overfit.
- Lots of tuning options. E.g., max_depth, max_features, max_leaf_nodes, min_samples_split, min_weight_fraction_leaf etc.
- Increasing min_* hyperparameters or reducing max_* hyperparameters will increase regularization.

Unregularized vs regularized

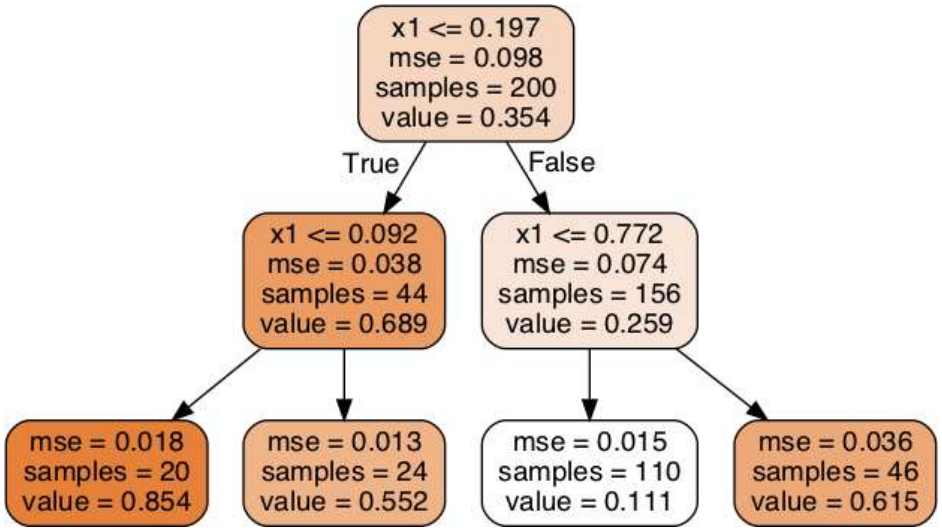


Decision boundaries of an unregularized tree (top) and a regularized tree (bottom)

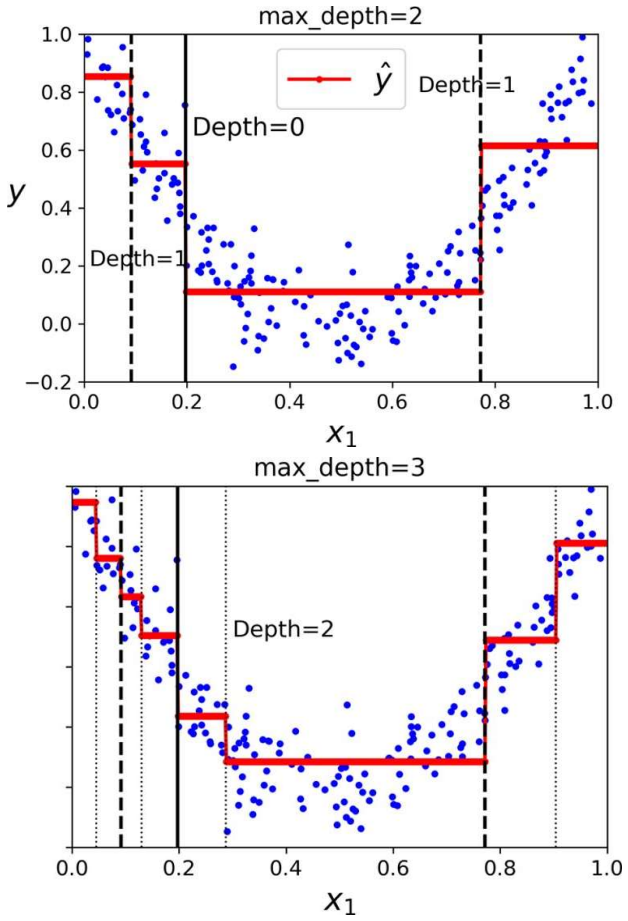
Regression

Predicting values rather than classes

Predicts average of target values in a region



A decision tree for regression



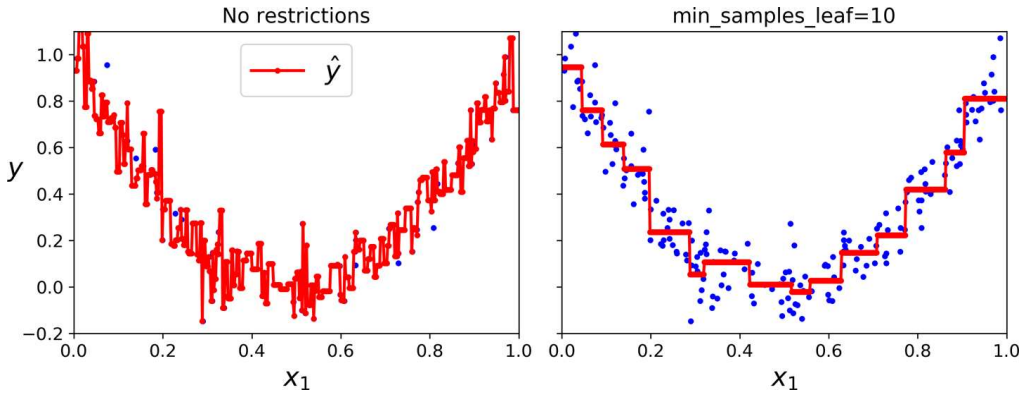
Predictions of two decision tree regression models

Regression(Cont'd)

Prone to overfit – Requires regularization

- Tree on left is what you get using default settings for hyperparameters.
- Tree on the right is what you get when you require a minimum number of samples in each leaf node.
- Use same hyperparameter tuning strategies as when tuning decision trees for classification tasks.
- Increasing min_* hyperparameters or reducing max_* hyperparameters will increase regularization.

Unregularized vs regularized



Predictions of an unregularized reg. tree (left) and a regularized tree (right)

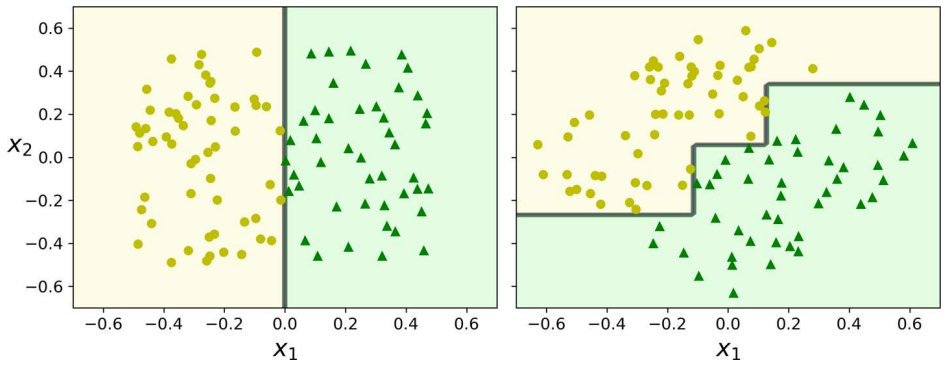
$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \dfrac{\sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2}{m_{\text{node}}} \\ \hat{y}_{\text{node}} =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

CART cost function for regression

Challenges

Sensitivity to training data rotations

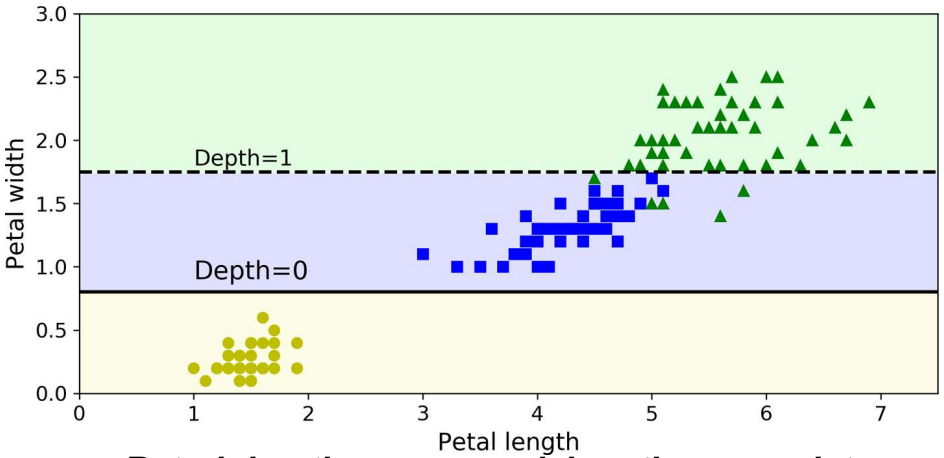
- Decision trees love orthogonal decision boundaries (all splits are perpendicular to an axis).
- Right data plot represents dataset after 45º rotation; decision boundary unnecessarily convoluted, and very likely not generalize well.



Sensitivity to training set rotation

Sensitivity to individual datapoints

- Decision trees have high variance – small changes to hyperparameters or data can result in significantly different models.
- Stochastic training algo in Scikit-Learn – Randomly selects the set of features to evaluate at each node, resulting different models for the same dataset (unless random_state is set)



Retraining the same model on the same data may produce a very different model