Life Sciences and
Facility Management

Institute of
Computational Life Sciences

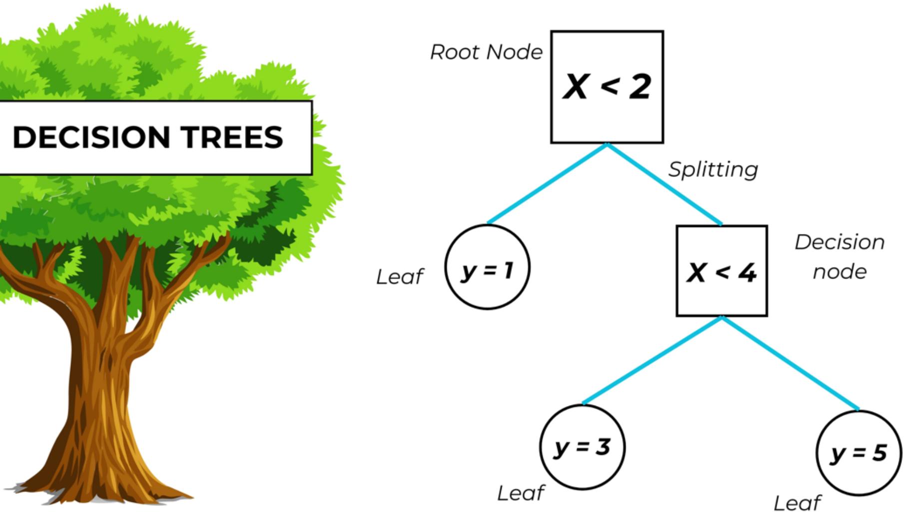
Decision trees

Machine Learning

Norman Juchler

Outline

- Decision trees are a very simple method for achieving complex, non-linear mappings between feature and target.

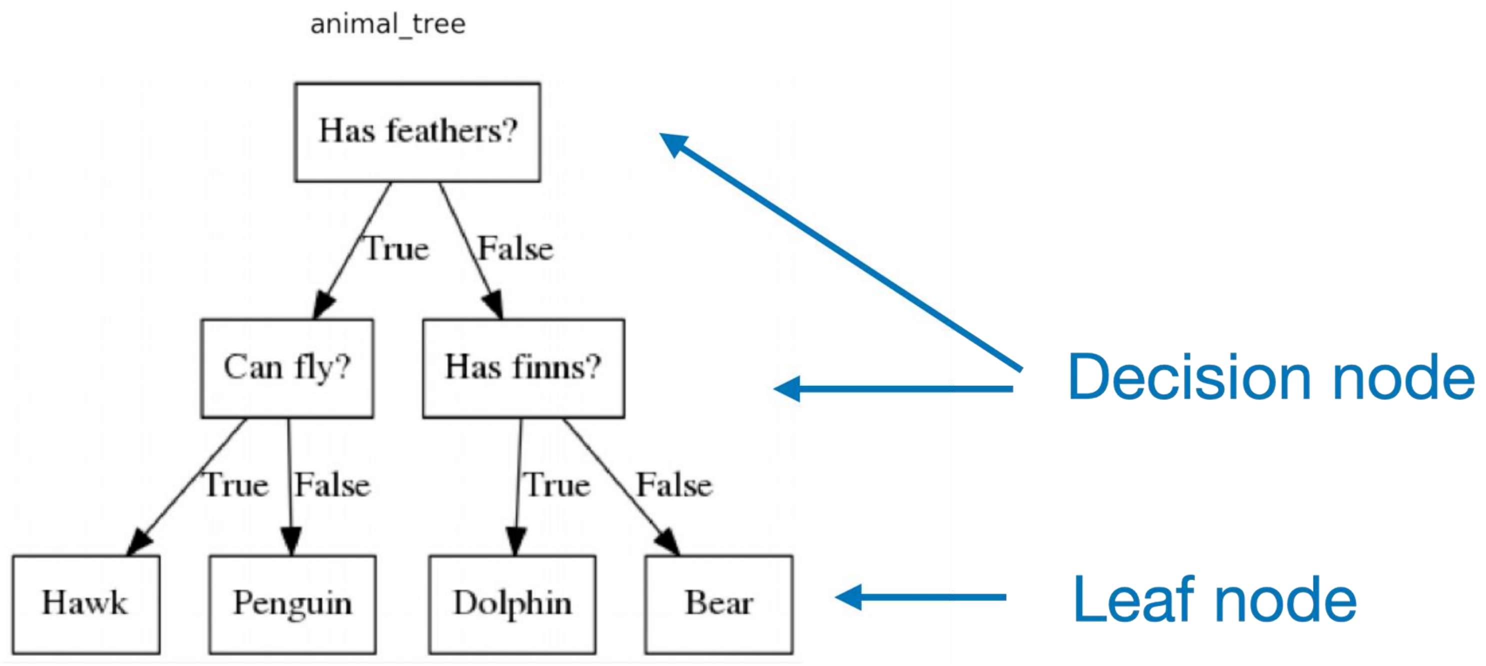


- They are popular for
  - their interpretability
  - ability to handle both categorical and numerical data*
  - suitability for both classification and regression tasks

* Unfortunately, this is not the case for decision trees in scikit-learn...

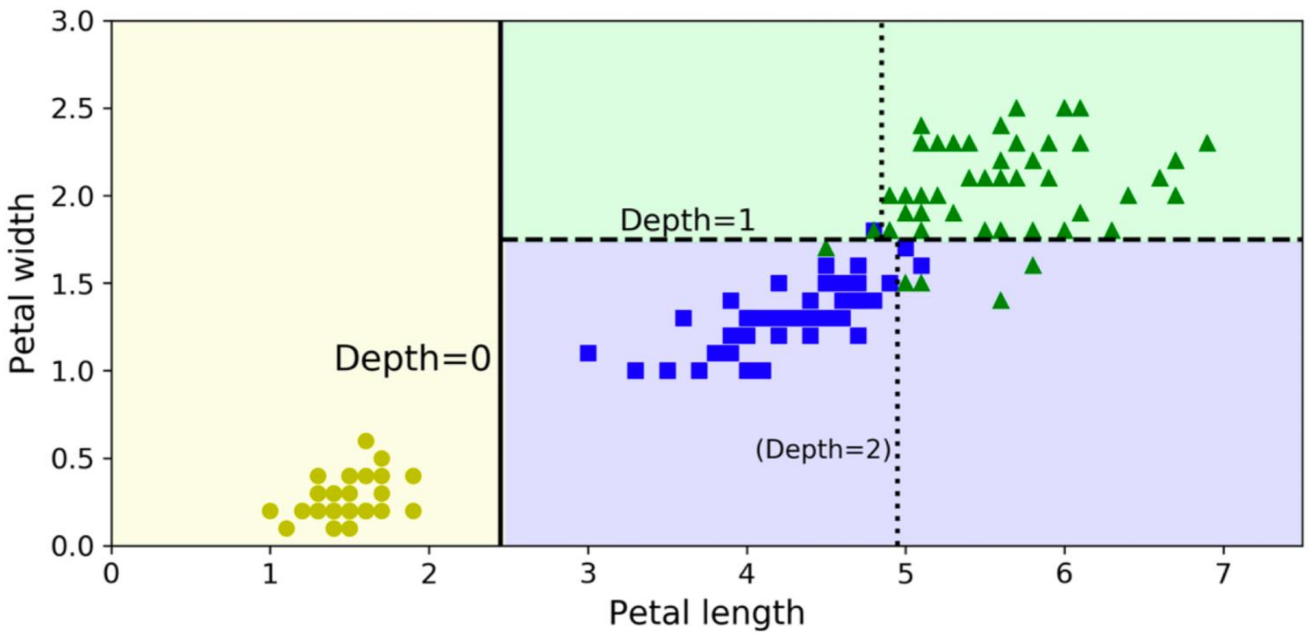
A tree of decisions

- Main idea: Algorithmically learn to construct a set of decision rules in the form of a tree.



Feature space view of decision trees

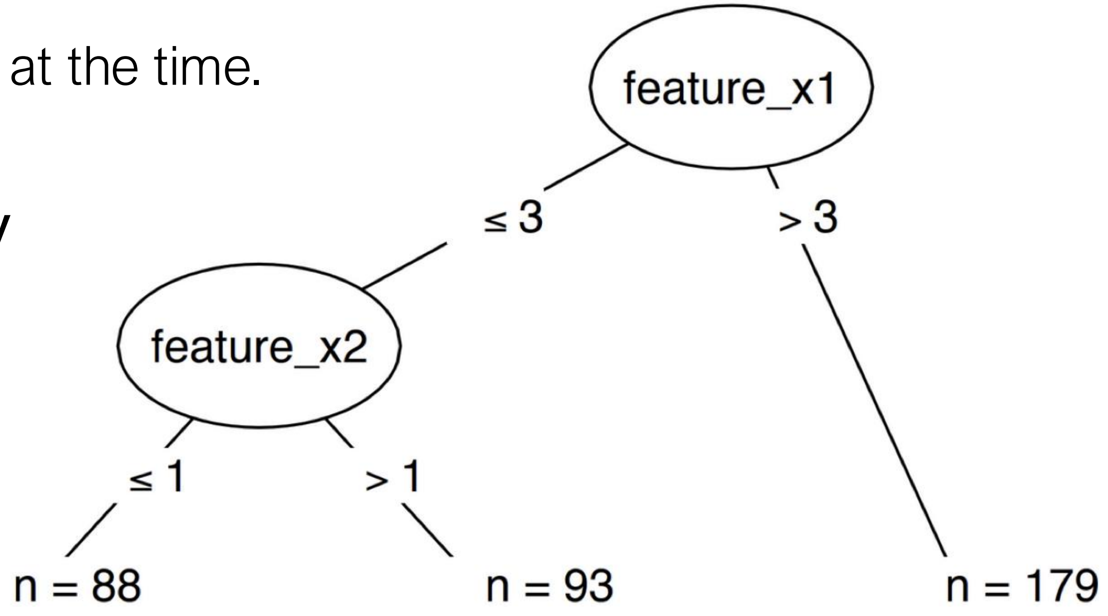
- A decision tree follows a set of if-else conditions to partition the feature space into different regions each assigned to a specific class.
- The class is determined by the majority value of the training data in the node.



Decision boundaries of a simple decision tree splitting the feature space

How the tree is built

- Consider a *set of samples* assigned to a given node (starting at the *root*)
- Decide (according to some rule) if and how to split...
  - If no: This node becomes a leaf node
  - If yes: Use a split rule to select the feature variable for the split and divide samples into two subsets (assign to the child nodes)
- What needs to selected per node?
  - The feature: only one feature can be tested per node at the time.
  - The split rule: how to split the dataset in the node?
- Apply the same splitting procedure recursively to the child nodes until all “active” nodes became leaves.



How the tree is built

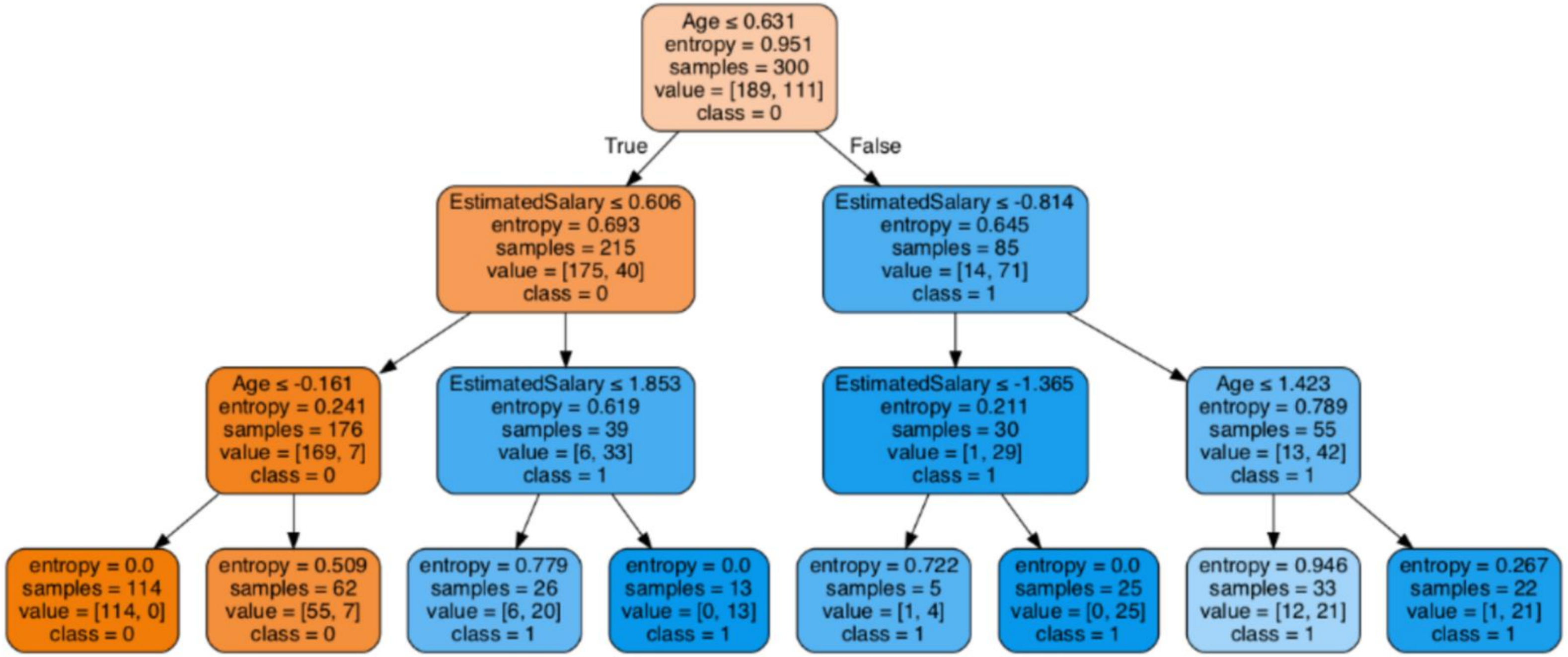
- But how to decide which feature to select, and how to split?
  - For each tree building step, the goal is to choose a *binary split* (yes, no) that makes the *two resulting subsets as different (or "pure") as possible* with respect to the target outcome
  - Different decision apply for different feature types:
    - For binary features, the dataset split follows naturally: yes, no
    - For multinomial features, different strategies may apply (one-versus-all, ordinal encoding, ...)
    - For numerical features, the best cut-off point is systematically searched for. (As a rule, the cut-oof value usually lies in-between two neighboring feature values.)
  - The best of all per-feature decisions determines the feature to be used.

- How to measure the "difference" (or purity) of two subsets with respect to the target?
  - Gini impurity: How impure is the distribution of classes in a node? (Minimize!)
  - Information gain: How much information is gained by the split? (Maximize!)
  - Variance reduction: How much do the y values vary in a node? (Maximize!)

- To understand the decision tree building algorithm: Link

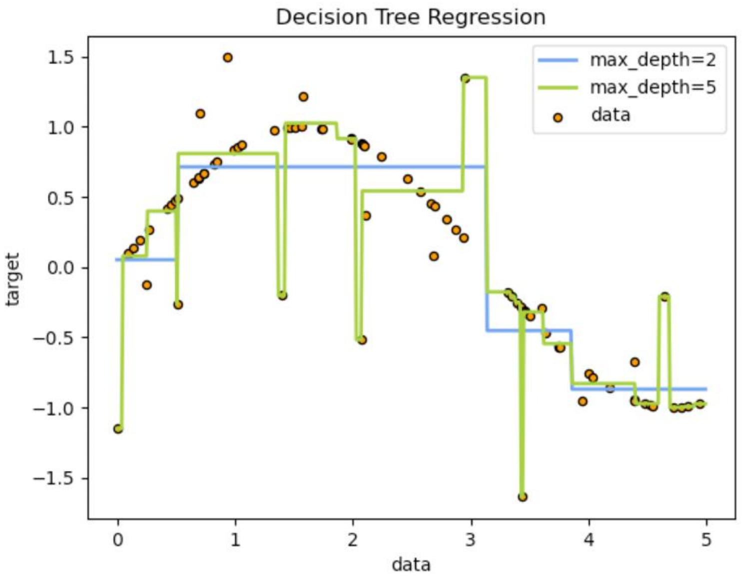
Visualization of decision trees

- One of the main advantage of decision trees is they can deliver clear explanations for their decisions.



Decision trees for regression (regression trees)

- The only difference here is that the target is a continuous variable.
- The output value at the leaves is determined (during training) by the average value of the target variable values within that leaf



Example of a decision trees trained to approximate a sine curve with a set of if-then decision rules. The deeper the tree, the more complex the decision rules, and the better the approximation.

Can you think of possible disadvantages of decision trees?

Pros and cons

Advantages

- Non-linear approach
- Simple to understand and interpret
- Requires little data preparation
- Non-parametric approach
- Fast training and inference
- Implicit feature selection (useless features will be ignored, the most important appear at the top)

Disadvantages

- Not robust to changes in training data, high model variance
- Prone to overfitting
- Predictions of decision trees are neither smooth nor continuous, but piecewise constant approximations. (Therefore, they are also bad at extrapolation.)
- Interpretability declines with depth

Outlook

Some of the above limitations can be mitigated by using **ensemble methods** like **Random Forests** or **Gradient Boosting**, which combine multiple decision trees for more stable and accurate predictions.