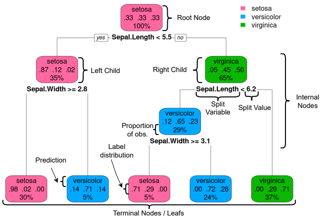
Introduction to Machine Learning

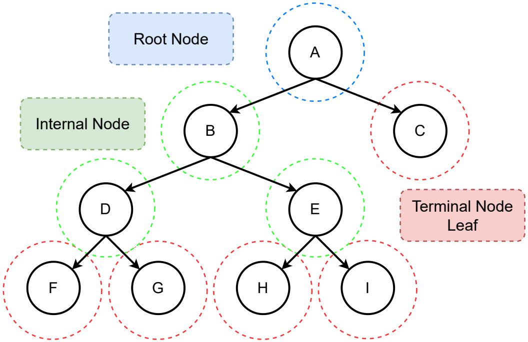
CART: Predictions with CART

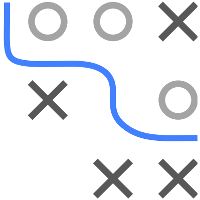


Learning goals

- Understand the basic structure of a tree model
- Understand that the basic idea of a tree model is the same for classification and regression
- Understand how the label of a new observation is predicted via CART
- Know hypothesis space of CART

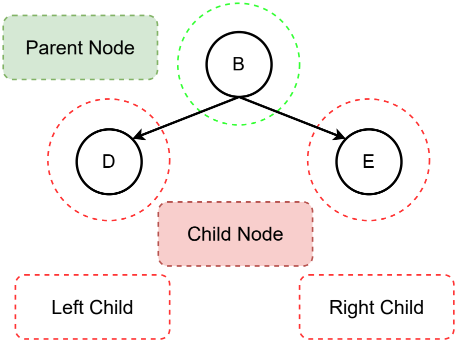
BINARY TREES

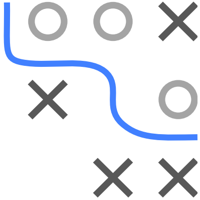




- Binary trees represent a top-down hierarchy with binary splits
- A tree is composed of different node types: a) the root node, b) internal nodes, and c) terminal nodes (also leaves).

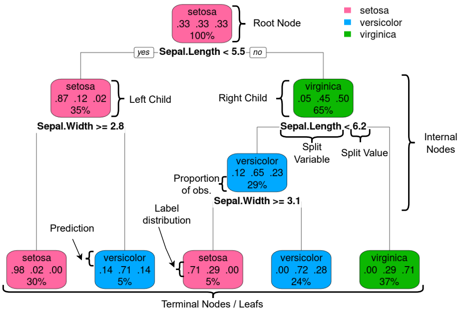
BINARY TREES

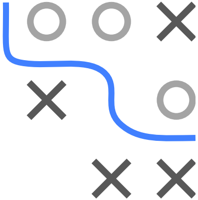




- Nodes have relative relationships, they can be:
  - Parent nodes
  - Child nodes
- Root nodes don't have parents – leaves don't have children

CLASSIFICATION TREES

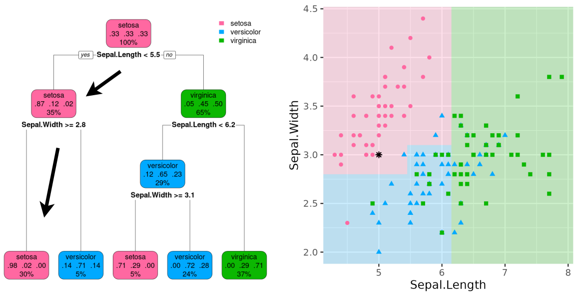




- Classification trees use the structure of a binary tree
- Binary splits are constructed top-down in a *data optimal* way
- Each split is a threshold decision for a single feature
- Each node contains the training points which follow its path
- Each leaf contains a constant prediction

CLASSIFICATION TREE MODEL AND PREDICTION

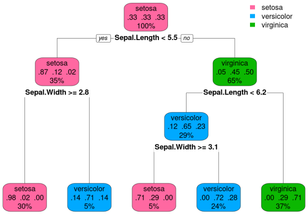
- When predicting new data (here*: $\text{Sepal.Length} = 5$, $\text{Sepal.Width} = 3$) we use the learned split points and pass an observation through the tree
- Each observation is assigned to exactly one leaf
- Classification trees can make hard-label predictions (here: setosa) or predict probabilities (here: 0.98, 0.02, 0.00)



CART AS A RULE BASED MODEL

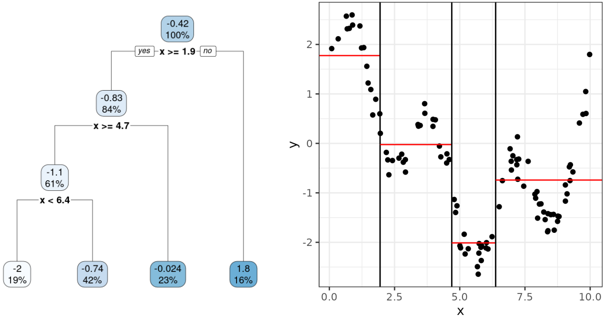
Leaf nodes can be expressed by a set of rules (left to right):

Hard label prediction	Label distribution	Sepal.Width	Sepal.Length
setosa	0.98, 0.02, 0.00	≥ 2.8	< 5.5
versicolor	0.14, 0.71, 0.14	< 2.8	< 5.5
setosa	0.71, 0.29, 0.00	≥ 3.1	≥ 5.5 & < 6.2
versicolor	0.00, 0.72, 0.28	< 3.1	≥ 5.5 & < 6.2
virginica	0.00, 0.29, 0.71	–	≥ 6.2

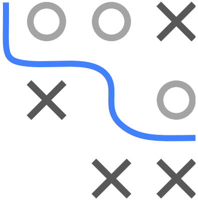


REGRESSION TREE MODEL AND PREDICTION

- Works the same way as for classification
- But predictions in leaf nodes are a numerical scalar



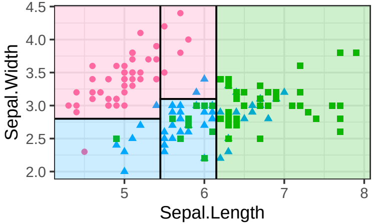
TREE AS AN ADDITIVE MODEL

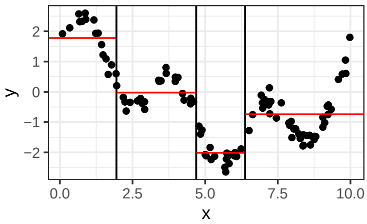


Trees divide the feature space $\mathcal{X}$ into **rectangular regions**:

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbb{I}(\mathbf{x} \in Q_m),$$

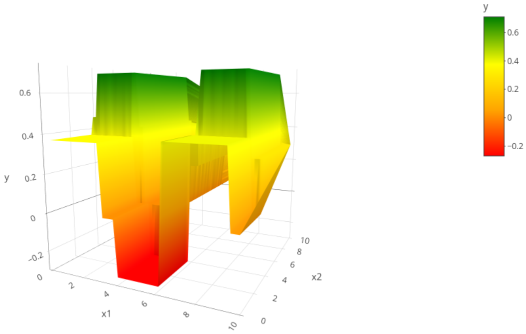
where a tree with M leaf nodes defines M "rectangles" Q_m.
c_m is the predicted numerical response, class label or class distribution in the respective leaf node.





TREE AS AN ADDITIVE MODEL

A 2D regression example:



(For binary classification with probabilities, 2D surface looks similar.)