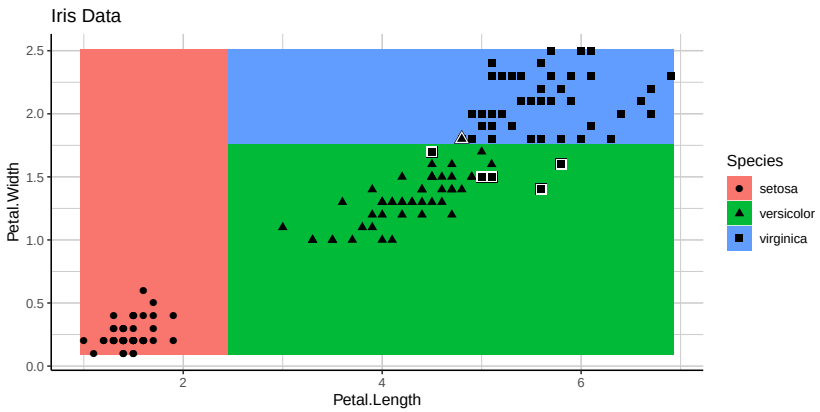
Introduction to Machine Learning

CART: Splitting Criteria

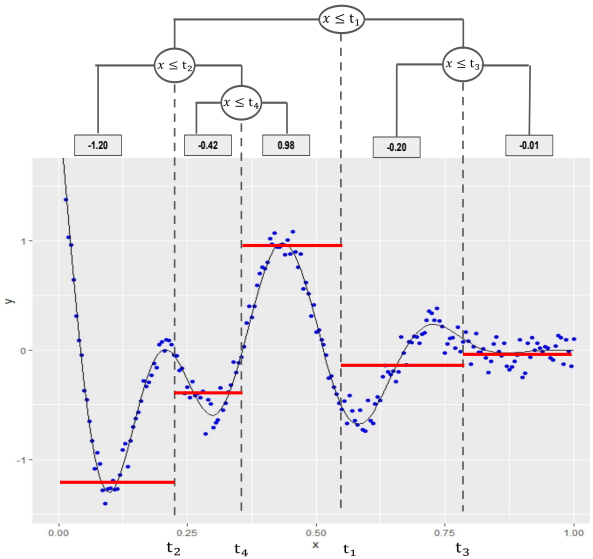
compstat-lmu.github.io/lecture_i2ml

TREES

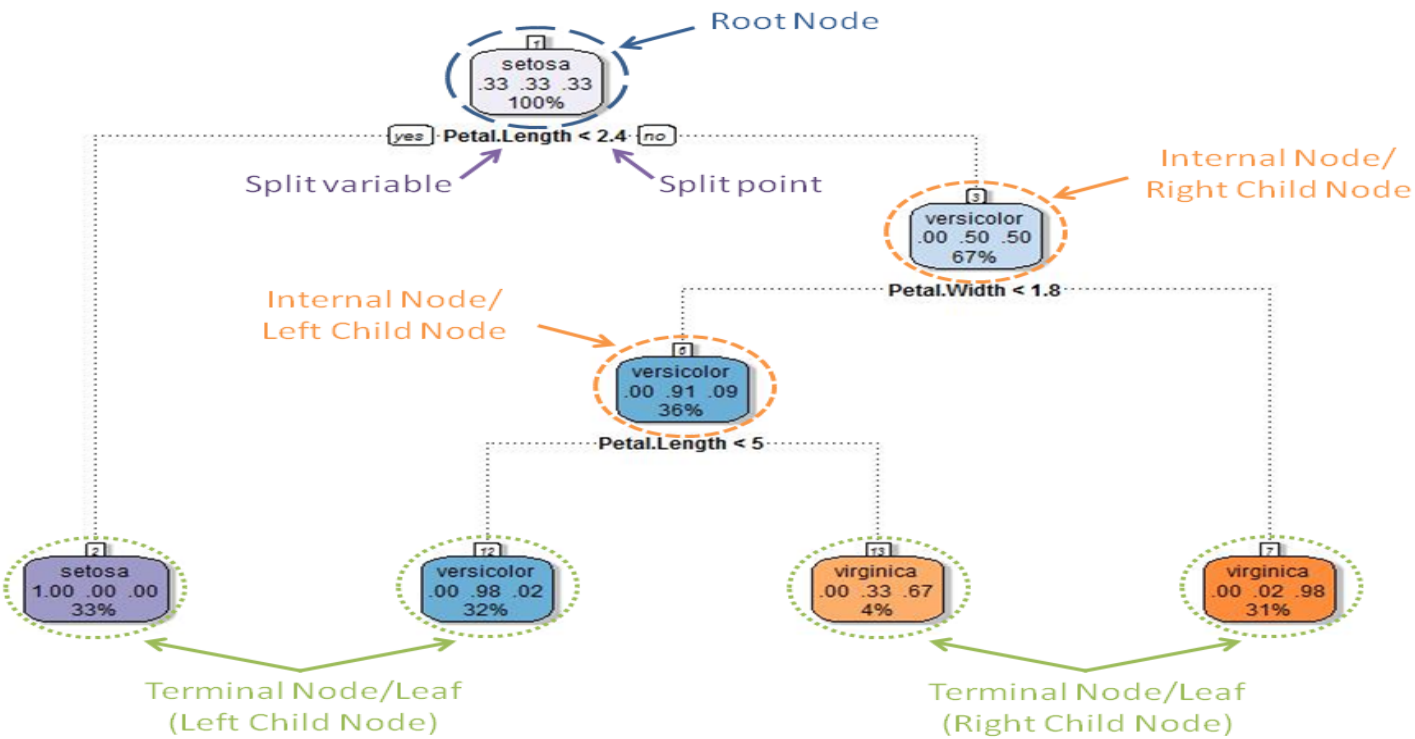
Classification Tree:



Regression Tree:



SPLITTING CRITERIA



How to find good splitting rules to define the tree?

$\Longrightarrow$ **empirical risk minimization**

SPLITTING CRITERIA: FORMALIZATION

- Let $\mathcal{N} \subseteq \mathcal{D}$ be the data that is assigned to a terminal node $\mathcal{N}$ of a tree.
- Let c be the predicted constant value for the data assigned to $\mathcal{N}$: $\hat{y} \equiv c$ for all $(x, y) \in \mathcal{N}$.
- Then the risk $\mathcal{R}(\mathcal{N})$ for a leaf is simply the average loss for the data assigned to that leaf under a given loss function L:

$$\mathcal{R}(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{(x,y) \in \mathcal{N}} L(y, c)$$

- The prediction is given by the optimal constant $c = \arg\min_c \mathcal{R}(\mathcal{N})$

SPLITTING CRITERIA: FORMALIZATION

- A split w.r.t. **feature $\mathbf{x}_j$ at split point** t divides a parent node $\mathcal{N}$ into

$$\mathcal{N}_1 = \{(x, y) \in \mathcal{N} : \mathbf{x}_j \leq t\} \text{ and } \mathcal{N}_2 = \{(x, y) \in \mathcal{N} : \mathbf{x}_j > t\}.$$

- In order to evaluate how good a split is, we compute the empirical risks in both child nodes and sum it up

$$\begin{aligned} \mathcal{R}(\mathcal{N}, j, t) &= \frac{|\mathcal{N}_1|}{|\mathcal{N}|} \mathcal{R}(\mathcal{N}_1) + \frac{|\mathcal{N}_2|}{|\mathcal{N}|} \mathcal{R}(\mathcal{N}_2) \\ &= \frac{1}{|\mathcal{N}|} \left(\sum_{(x,y) \in \mathcal{N}_1} L(y, c_1) + \sum_{(x,y) \in \mathcal{N}_2} L(y, c_2) \right) \end{aligned}$$

- finding the best way to split $\mathcal{N}$ into $\mathcal{N}_1, \mathcal{N}_2$ means solving

$$\arg\min_{j,t} \mathcal{R}(\mathcal{N}, j, t)$$

SPLITTING CRITERIA: REGRESSION

- For regression trees, we usually use L_2 loss:

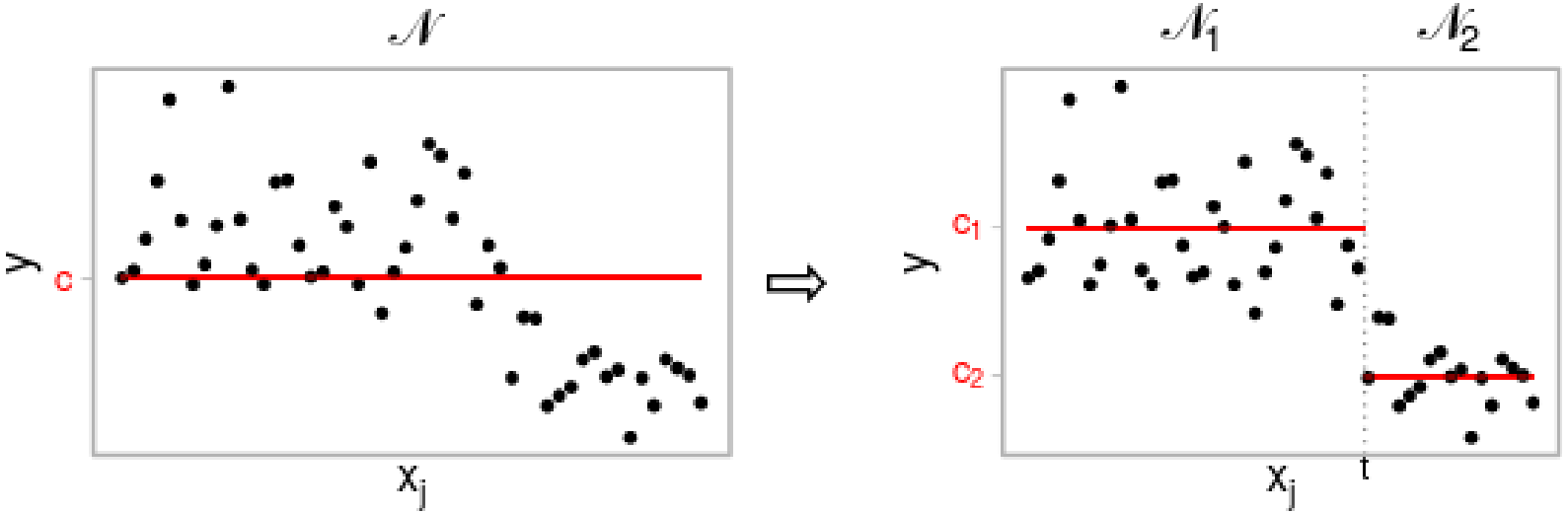
$$\mathcal{R}(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{(x,y) \in \mathcal{N}} (y - c)^2$$

- The best constant prediction under L_2 is the mean

$$c = \bar{y}_{\mathcal{N}} = \frac{1}{|\mathcal{N}|} \sum_{(x,y) \in \mathcal{N}} y$$

SPLITTING CRITERIA: REGRESSION

- This means the best split is the one that minimizes the (pooled) variance of the target distribution in the child nodes $\mathcal{N}_1$ and $\mathcal{N}_2$:



We can also interpret this as a way of measuring the impurity of the target distribution, i.e., how much it diverges from a constant in each of the child nodes.

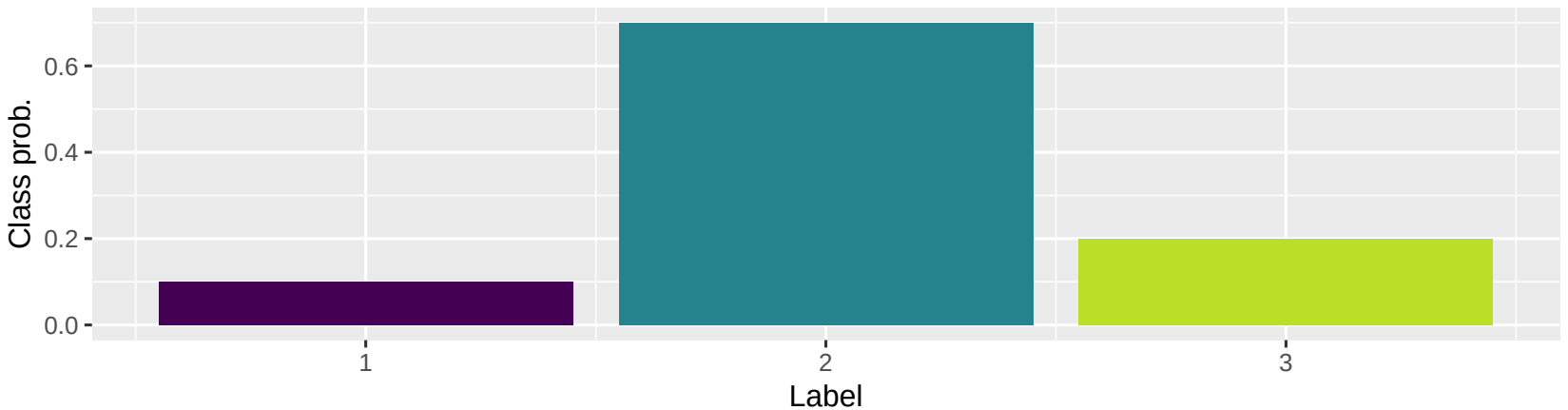
- For L_1 loss, c is the median of $y \in \mathcal{N}$.

SPLITTING CRITERIA: CLASSIFICATION

- Typically uses either Brier score (so: L_2 loss on probabilities) or Bernoulli loss (as in logistic regression) as loss functions
- Predicted probabilities in node $\mathcal{N}$ are simply the class proportions in the node:

$$\hat{\pi}_k^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|} \sum_{(x,y) \in \mathcal{N}} \mathbb{I}(y = k)$$

This is the optimal prediction under both the logistic / Bernoulli loss and the Brier loss.



SPLITTING CRITERIA: COMMENTS

- Splitting criteria for trees are usually defined in terms of "impurity reduction". Instead of minimizing empirical risk in the child nodes over all possible splits, a measure of “impurity” of the distribution of the target y in the child nodes is minimized.
- For regression trees, the “impurity” of a node is usuallly defined as the variance of the $y^{(i)}$ in the node. Minimizing this “variance impurity” is equivalent to minimizing the squared error loss for a predicted constant in the nodes.

SPLITTING CRITERIA: COMMENTS

- Minimizing the Brier score is equivalent to minimizing the Gini impurity

$$I(\mathcal{N}) = \sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})}(1 - \hat{\pi}_k^{(\mathcal{N})})$$

- Minimizing the Bernoulli loss is equivalent to minimizing entropy impurity

$$I(\mathcal{N}) = -\sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})} \log \hat{\pi}_k^{(\mathcal{N})}$$

- The approach based on loss functions instead of impurity measures is simpler and more straightforward, mathematically equivalent and shows that growing a tree can be understood in terms of empirical risk minimization.

SPLITTING WITH MISCLASSIFICATION LOSS

- Why don't we use the misclassification loss for classification trees? I.e., always predict the majority class in each child node and count how many errors we make.
- In many other cases, we are interested in minimizing this kind of error, but have to approximate it by some other criterion instead since the misclassification loss does not have derivatives that we can use for optimization.
  We don't need derivatives when we optimize the tree, so we could go for it!
- This is possible, but Brier score and Bernoulli loss are more sensitive to changes in the node probabilities, and therefore often preferred

SPLITTING WITH MISCLASSIFICATION LOSS

Example: two-class problem with 400 obs in each class and two possible splits:

Split 1:

	class 0	class 1
$\mathcal{N}_1$	300	100
$\mathcal{N}_2$	100	300

Split 2:

	class 0	class 1
$\mathcal{N}_1$	400	200
$\mathcal{N}_2$	0	200

- Both splits are equivalent in terms of misclassification error, they each misclassify 200 observations.
- But: Split 2 produces one pure node and is probably preferable.
- Brier loss (Gini impurity) and Bernoulli loss (entropy impurity) prefer the second split

SPLITTING WITH MISCLASSIFICATION LOSS

- Calculation for Gini:

$$\text{Split 1}: \frac{|\mathcal{N}_1|}{|\mathcal{N}|} \cdot 2 \cdot \hat{\pi}_0^{(\mathcal{N}_1)} \hat{\pi}_1^{(\mathcal{N}_1)} + \frac{|\mathcal{N}_2|}{|\mathcal{N}|} \cdot 2 \cdot \hat{\pi}_0^{(\mathcal{N}_2)} \hat{\pi}_1^{(\mathcal{N}_2)} =$$

$$\frac{3}{4} \cdot \frac{1}{4} + \frac{1}{4} \cdot \frac{3}{4} = \frac{3}{16}$$

$$\text{Split 2}: \quad \frac{3}{4} \cdot 2 \cdot \frac{2}{3} \cdot \frac{1}{3} + \frac{1}{4} \cdot 0 \cdot 1 = \frac{1}{3}$$