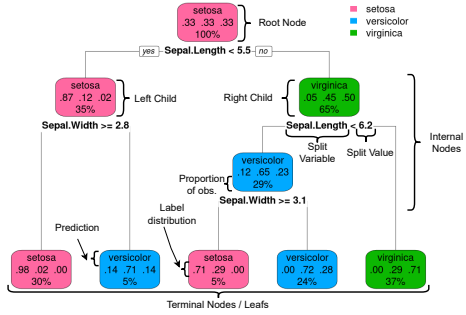
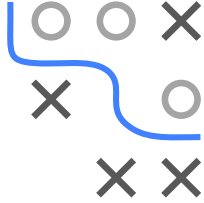
CLASSIFICATION TREES





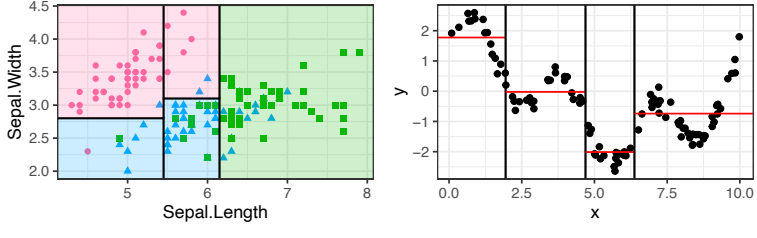
- Classification trees use the structure of a binary tree
- Binary splits are constructed top-down in a *data optimal* way
- Each split is a threshold decision for a single feature
- Each node contains the training points which follow its path
- Each leaf contains a constant prediction

TREE AS AN ADDITIVE MODEL

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

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

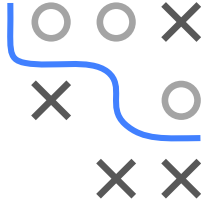
where a tree with M leaf nodes defines M "rectangles" Q_m.
c_m is the predicted numerical response, class label or class distribution in the respective leaf node.



CATEGORICAL FEATURES

- A split on a categorical feature partitions the feature levels:

$$x_j \in \{a, b, c\} \leftarrow \mathcal{N} \rightarrow x_j \in \{d, e\}$$



- For a feature with m levels, there are about 2^m different possible partitions of the m values into two groups ($2^{m-1} - 1$ because of symmetry and empty groups).
- Searching over all these becomes prohibitive for large values of m.
- For regression with L2 loss and for binary classification, we can define clever shortcuts.

For $0 - 1$ responses, in each node:

1. Calculate the proportion of 1-outcomes for each category of the feature in $\mathcal{N}$.
2. Sort the categories according to these proportions.
3. The feature can then be treated as if it was ordinal, so we only have to investigate at most $m - 1$ splits.

For continuous responses, in each node:

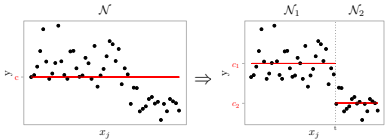
1. Calculate the mean of the outcome in each category
2. Sort the categories by increasing mean of the outcome

CART FINDING THE BEST SPLIT

Empirical risk

- Splitting **feature** x_j **at split point** t divides a parent node $\mathcal{N}$ into two child nodes:

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



- Compute empirical risks in child nodes and minimize their sum to find best split (impurity reduction):

$$\arg\min_{j,t} \mathcal{R}(\mathcal{N}, j, t) = \arg\min_{j,t} \mathcal{R}(\mathcal{N}_1) + \mathcal{R}(\mathcal{N}_2)$$

Note: If $\mathcal{R}$ is the average instead of the sum of loss functions, we need to reweight: $\frac{|\mathcal{N}_t|}{|\mathcal{N}|}\mathcal{R}(\mathcal{N}_t)$

-
  - g-way classification:

Brier score $\rightarrow$ **Gini** impurity	**Bernoulli** loss $\rightarrow$ **entropy** impurity
$\mathcal{R}(\mathcal{N}) = \sum_{(\mathbf{x},y)\in\mathcal{N}} \sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})}(1 - \hat{\pi}_k^{(\mathcal{N})})$	$\mathcal{R}(\mathcal{N}) = -\sum_{(\mathbf{x},y)\in\mathcal{N}} \sum_{k=1}^{g} \hat{\pi}_k^{(\mathcal{N})} \log \hat{\pi}_k^{(\mathcal{N})}$

  - Regression (**quadratic** loss): $\mathcal{R}(\mathcal{N}) = \sum_{(\mathbf{x},y)\in\mathcal{N}} (y - c)^2$ with $c = \frac{1}{|\mathcal{N}|} \sum_{(\mathbf{x},y)\in\mathcal{N}} y$

Optimization

- **Exhaustive** search over all split candidates, choice of risk-minimal split
- In practice: reduce number of split candidates (e.g., using quantiles instead of all observed values)