Decision trees

Course of Machine Learning

Master Degree in Computer Science

University of Rome “Tor Vergata”

Giorgio Gambosi

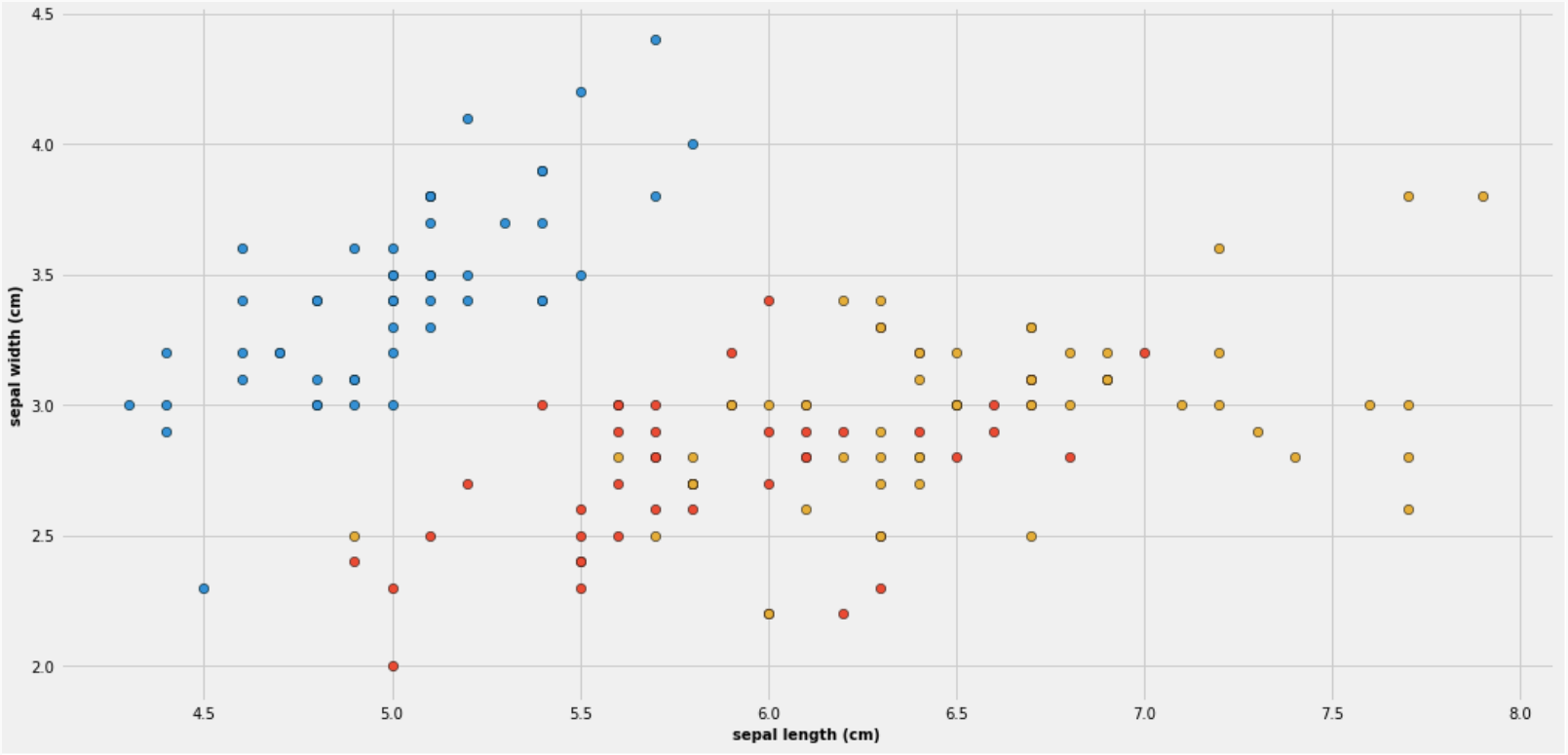
a.a. 2018-2019

Decision tree

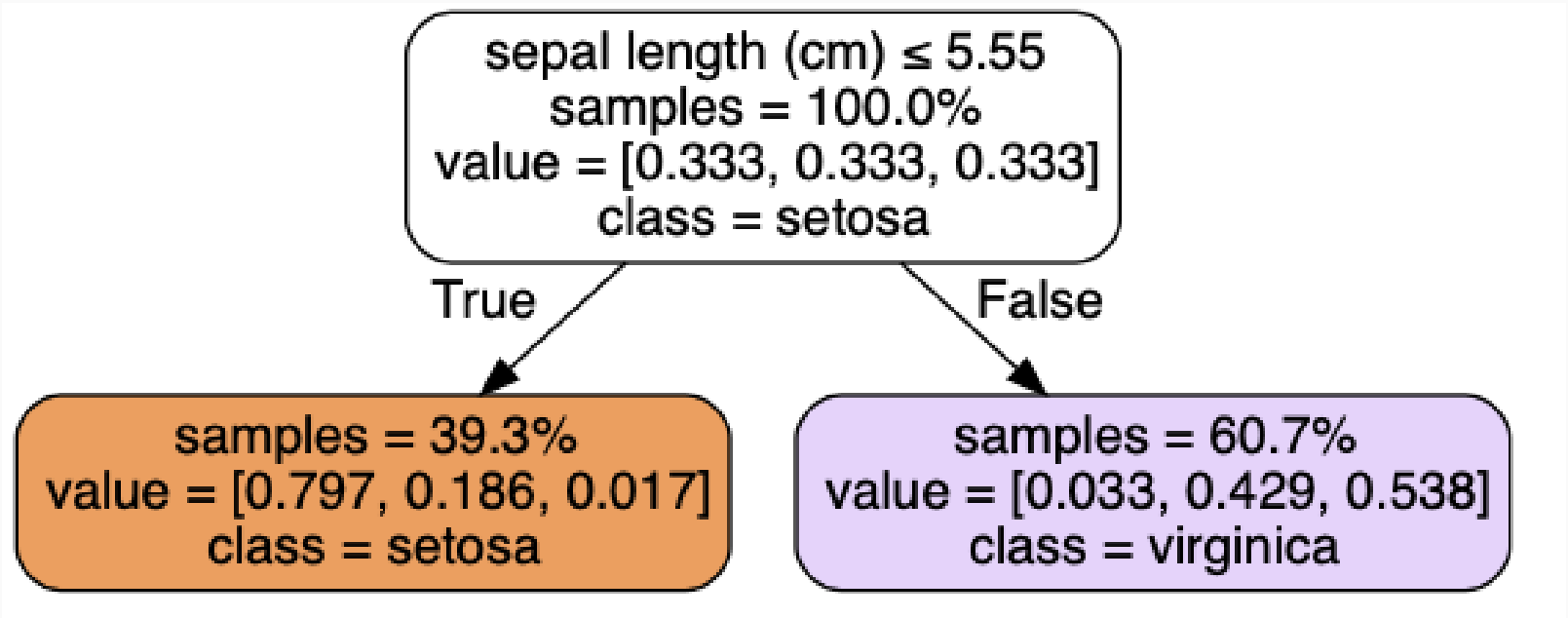
A decision tree is a classifier expressed as a recursive partition of the instance space.

- It consists of nodes that form a rooted tree
- Each internal node splits the instance space into two or more subspaces, according to a given discrete function of the attributes values
- Usually, each node is associated to a partition wrt a single attribute
- Each leaf is associated to a subspace and:
  - either a class, representing the most appropriate prediction for all points in the subspace
  - or a vector of class probabilities

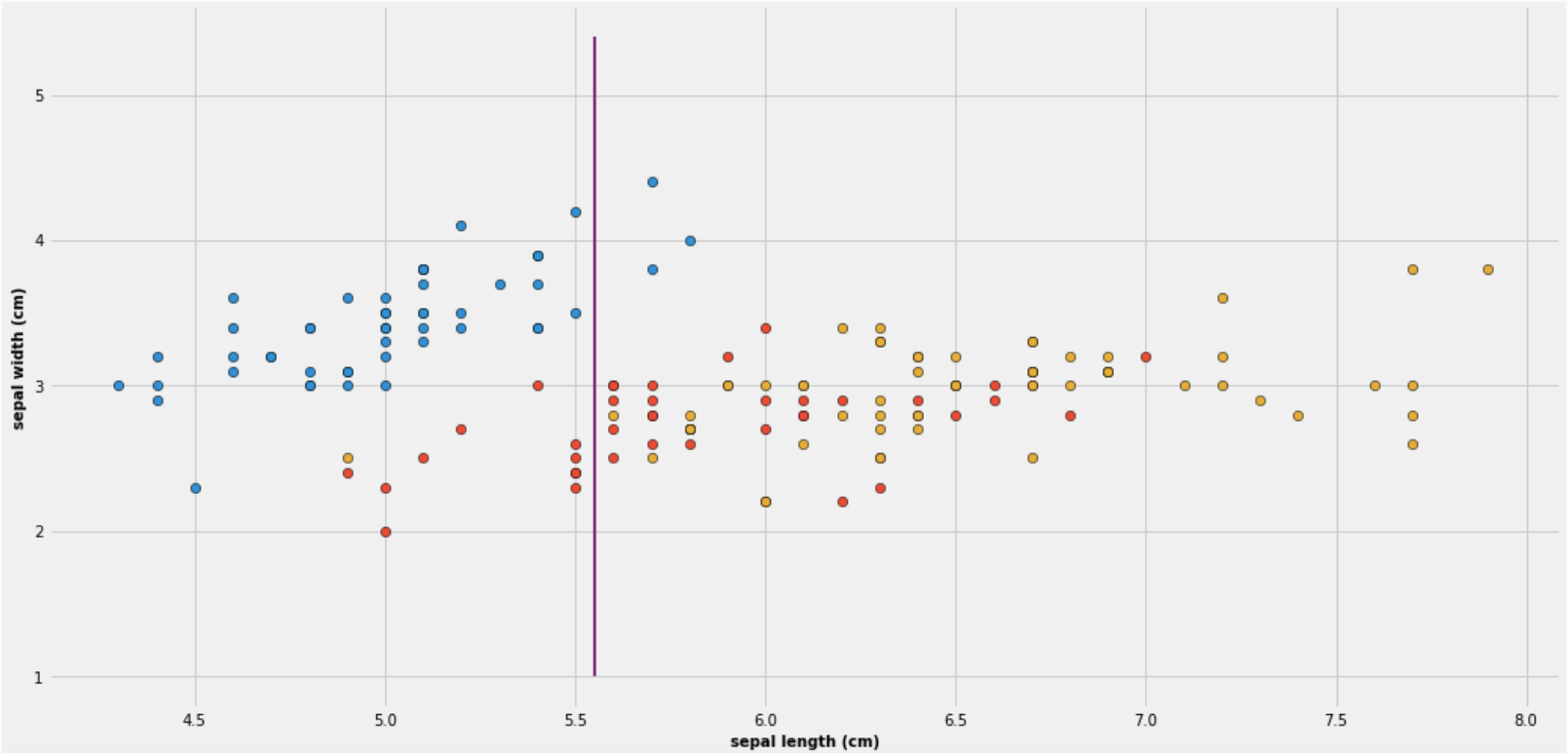
Decision tree



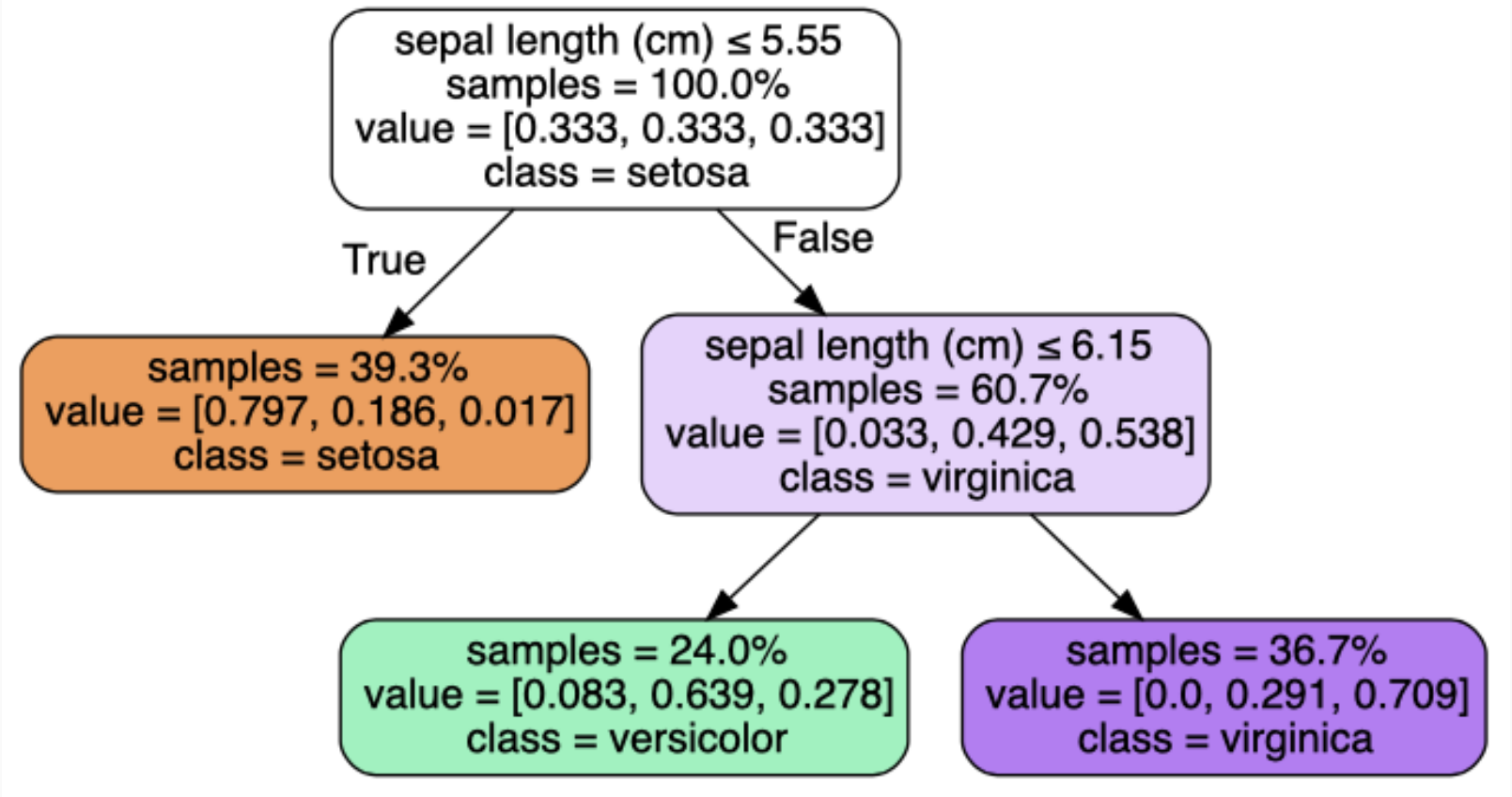
Decision tree



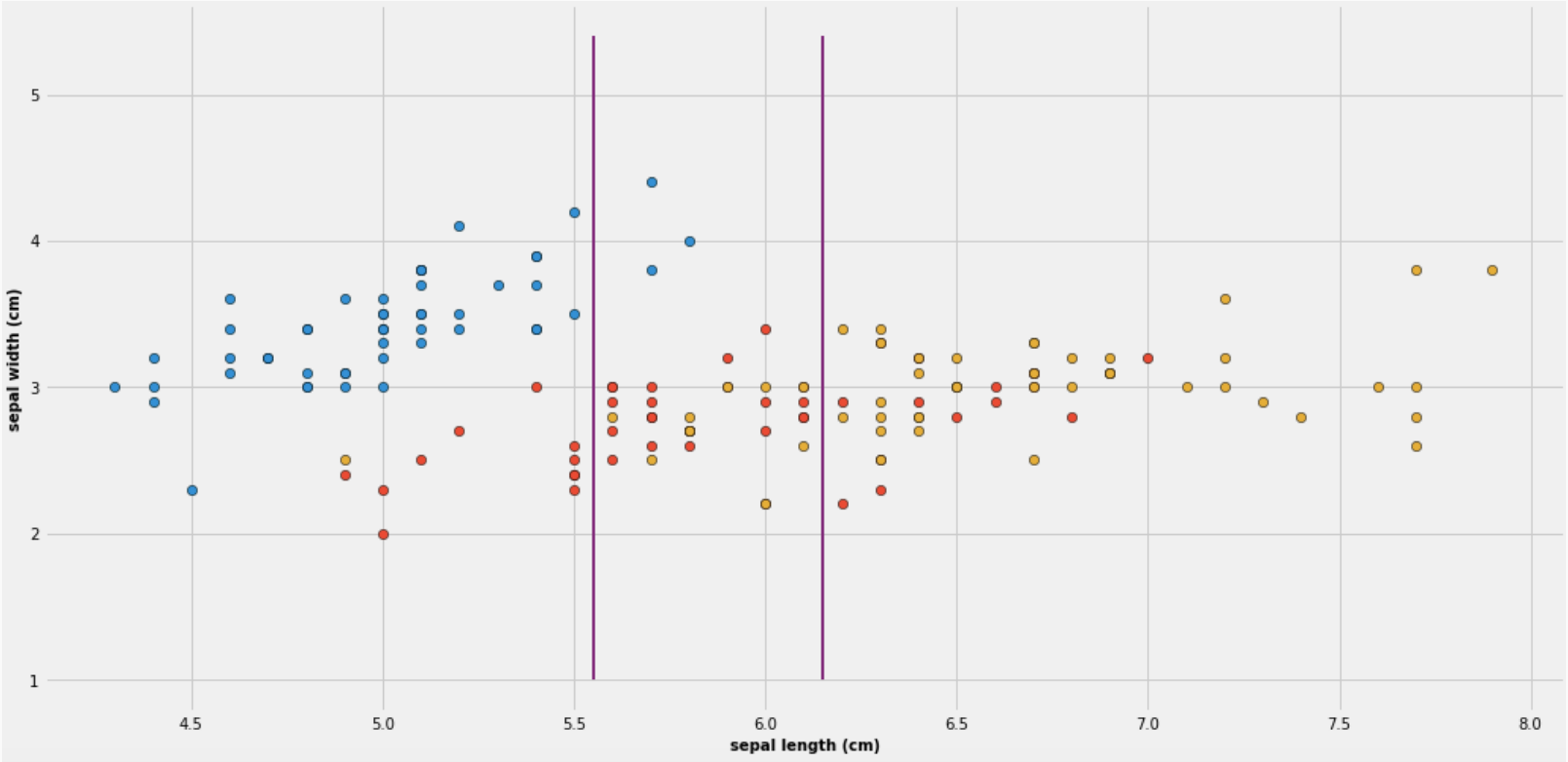
Decision tree



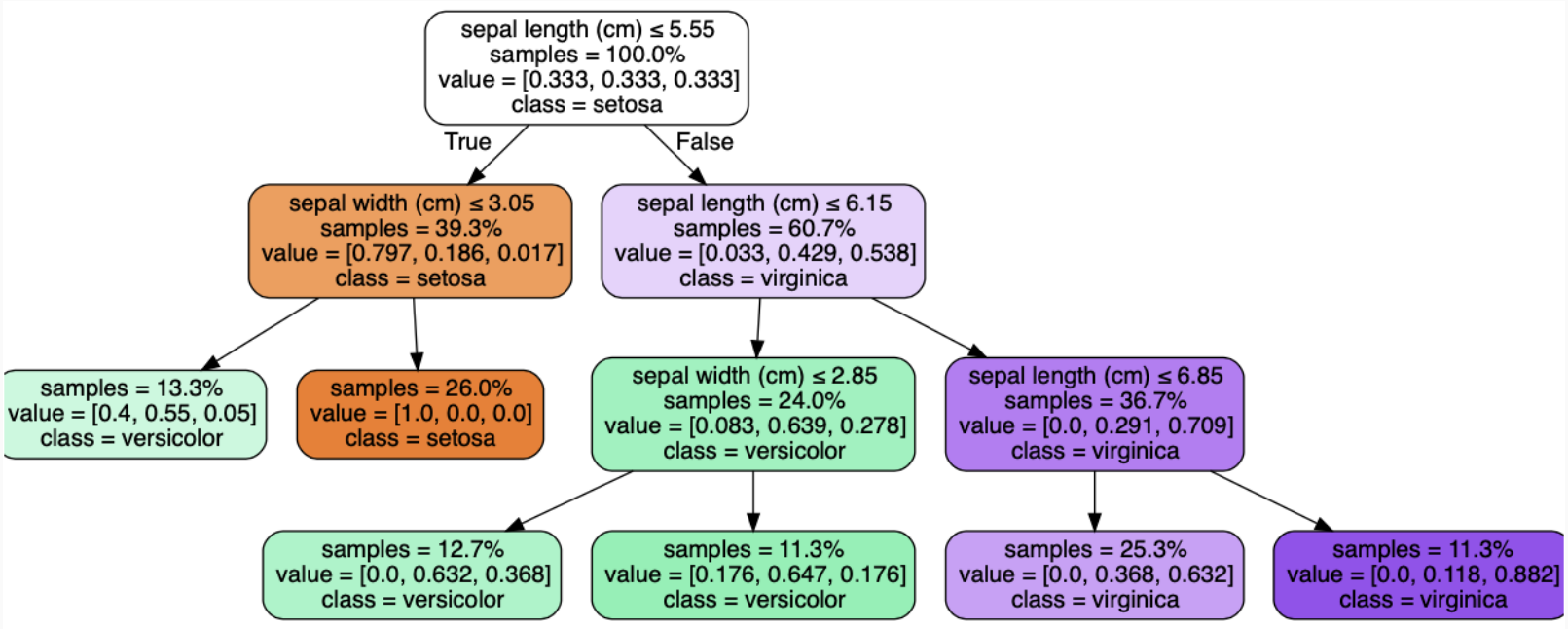
Decision tree



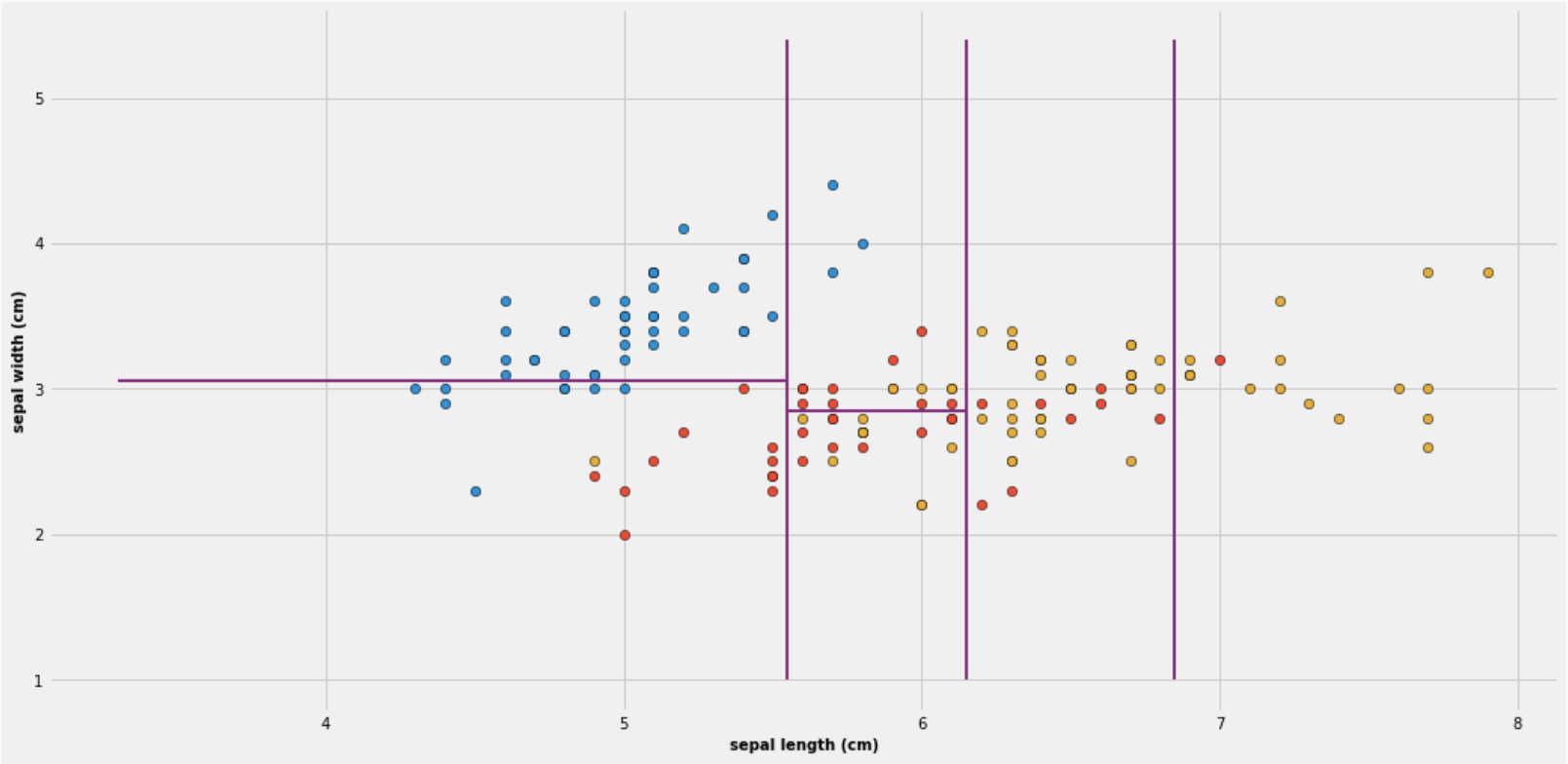
Decision tree



Decision tree



Decision tree



Decision tree: classification

- Given an item $\mathbf{z} = (z_1, \ldots, z_d)^T$, the decision tree is traversed starting from its root.
- At each node traversed, with associated feature x_i and function f_i, the value $f_i(z_i)$ is computed to decide which is the next node to be considered, among the set of children nodes. This is equivalent to considering smaller and smaller subregions of the space of data.
    - An important case is when a threshold θ_i is defined and a comparison between z_i and θ_i is performed to decide which is the next node to be considered, among two children nodes.
- The procedure halts when a leaf node is reached. The returned prediction is given by the corresponding class, or derived by the class probabilities vector.

Decision tree: construction

The space of data is recursively partitioned by constructing the decision tree from root to leaves.

At each node:

1. How to perform a partition of the corresponding region (choosing feature and function)?
2. When to stop partitioning? How to assign information to leaves?

Decision tree: partitioning at each node

Select the feature and function/threshold such that a given measure is maximized within the intersections of the training set with each subregion.

Measures of class impurity within a set. To be minimized.

Inpurity measure

Given a random variable with discrete domain $\{a_1, \ldots, a_k\}$ and corresponding probabilities $p = (p_1, \ldots, p_k)$, an impurity measure $\phi : p \mapsto \mathbb{R}$ has the following properties

- $\phi(p) \geq 0$ for all possible p
- $\phi(p)$ is minimum if there exists $i, 1 \leq i \leq k$ such that $p_i = 1$
- $\phi(p)$ is maximum if $p_i = 1/k$ for all i
- $\phi(p) = \phi(p')$ for all p' deriving from a permutation of p
- $\phi(p)$ is differentiable everywhere

Goodness of split

In our case, we consider the class of each item in S.

- For any set S of items, the probability vector associated to S can be defined as $p = \left(\frac{|S_1|}{|S|}, \ldots, \frac{|S_k|}{|S|}\right)$, where $x_i \in S_h$ iff $t_i = h$, that is if its class is a_h.
- Given a function $f : S \mapsto \{1, \ldots, r\}$, let $s_i = \{x \in S | f(x) = i\}$ (that is, $x \in s_i$ iff $f(x)=i). The goodness of split of S wrt f is given by

$$\Delta_\phi(S, f) = \phi(p_S) - \sum_{i=1}^{r} p_i \phi(p_{s_i})$$

that is, by the difference between the impurity of S and the sum of impurities of the subsets resulting from the application of f

Goodness of split

In practice, f is usually defined by considering a single feature and:

- if the feature is discrete, inducing a partition of its codomain in k subsets
  - as a special case, the partition is among items with the same value for the considered feature
- if it is continuous, inducing a partition of its codomain in a set of intervals, defined by thresholds
  - as a special case, a single threshold is given and f performs a binary partition on items in S

Entropy and information gain

- For any set S of items, let

$$H_S = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

  be the entropy of S.

- By using entropy as an impurity measure, the goodness of split is given by the information gain

- The information gain wrt to a partition function f is the decrease of entropy from S to the weighted sum of entropies of s_i

$$IG(S, f) = H_S - \sum_{j=1}^{r} \frac{|s_j|}{|S|} H_{s_j}$$

Gini index

Gini index is used in many cases as a tool to measure divergence from equality. It is defined as

$$G_S = 1 - \sum_{i=1}^{k} \left(\frac{|S_i|}{|S|} \right)$$

- The Gini gain wrt to a partition function f is the decrease of Gini index from S to the weighted sum of Gini indices of s_i

$$GG(S, f) = G_S - \sum_{j=1}^{r} \frac{|s_j|}{|S|} G_{s_j}$$

Decision tree: leaves

Often, conditions for deciding when partitioning has to stopped are predefined. For example:

1. All instances in the training set belong to a single value of y.
2. The maximum tree depth has been reached.
3. The number of items in the terminal node is less than the minimum number of cases for parent nodes.
4. The best splitting criteria is not greater than a certain threshold.

When a leaf is reached, the corresponding class can be defined as the majority class in the intersection of the subregion and the training set.

A pruning procedure can be also applied to a large tree: subtrees are merged into single nodes, thus reducing the tree size.

Decision tree: advantages

1. Self?explanatory and when compacted they are also easy to follow.
2. Can handle both nominal and numeric input attributes.
3. Nonparametric method: no assumptions made

Decision tree: disadvantages

Mainly, prone to overfitting.