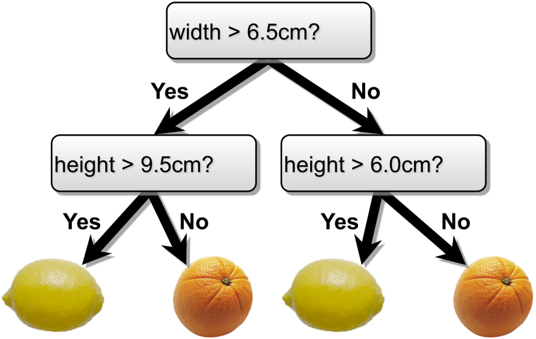
Table of Contents

Overview of Decision Trees

- Tree-based methods involve *stratifying* or *segmenting* the predictor space into a number of simple regions.
- The set of splitting rules used to segment the predictor space can be summarized in a tree.
- Tree-based methods are simple and useful for interpretation.
- However, they typically are not competitive with the best supervised learning approaches, in terms of prediction accuracy.
- Combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss in interpretation.

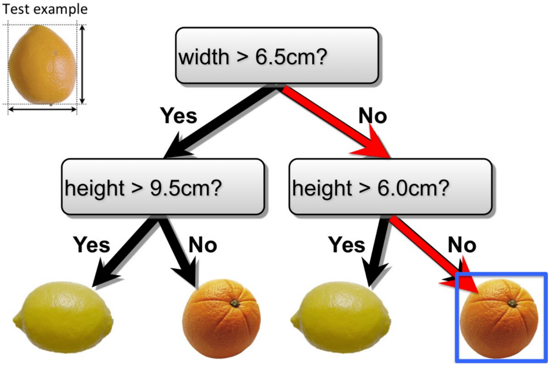
An Illustrative Example

- Internal nodes test attributes
- Branching is determined by attribute value
- Leaf nodes are outputs (predictions)

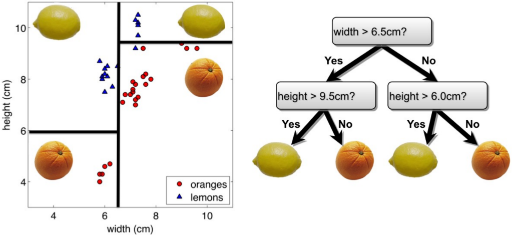


An Illustrative Example

- Decision trees make predictions by recursively splitting on different attributes according to a tree structure.



Building Decision Trees



- We divide the predictor space - that is, the set of possible values for $X_1, X_2, \cdots, X_p$ - into J distinct and non-overlapping regions, $R_1, R_2, \cdots, R_J$.
- Each path from root to a leaf defines a region R_j of input space.
- For every observation that falls into the region R_j, we make the same prediction.
- At each level, one must choose:
  1. Which variable to split.
  2. Possibly where to split it.

Building Decision Trees

- **Regression Tree**:
  - Continuous output
  - Leaf value y_j typically set to the **mean value** of the response values for the training observations in R_j.
- **Classification tree**:
  - Discrete output
  - Leaf value y_j typically set to the **most common value** of the response values for the training observations in R_j.

How do we construct the regions (Regression Tree)?

- In theory, the regions could have any shape.
- We choose to divide the space into high-dimensional rectangles, for simplicity and for ease of interpretation of the resulting predictive model.
- The goal is to find boxes $R_1, R_2, \cdots, R_J$ that minimize the RSS, given by

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

  where $\hat{y}_{R_j}$ is the mean response for the training observations within the jth box.
- It is computationally infeasible to consider every possible partition of the feature space into J boxes. So, what to do?

Recursive Binary Splitting

- Recursive, greedy approach to build a tree node-by-node.
- **Top-down**: Starts at the top of the tree and recursively splits the predictor space into branches; each split is indicated via two new branches further down on the tree.
- **Greedy**: Selects the best split at each step without considering future splits.