STA 529 2.0 Data Mining

Dr Thiyanga S. Talagala

Classification and Regression Trees

Lecture 2

Classification and Regression Trees (CART)

Classification and Regression Trees (CART)

- Decision trees
- Supervised learning method
- Data driven method

Model

$$Y = f(X_1, X_2, ...X_n) + \epsilon$$

Goal: What is f?

How do we estimate f?

Data-driven methods:

estimate f using observed data without making explicit assumptions about the functional form of f.

Parametric methods:

estimate f using observed data by making assumptions about the functional form of f.

Classification and Regression Trees

1. Classification tree - Outcome is categorical
2. Regression tree - Outcome is numeric

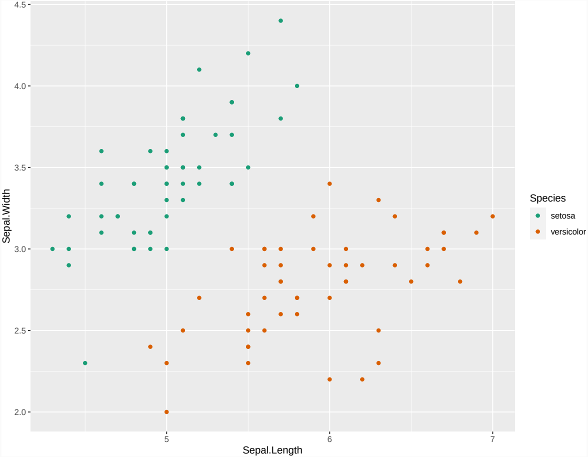
Classification and Regression Trees

- CART models work by partitioning the feature space into a number of simple rectangular regions, divided up by axis parallel **splits**.
- The **splits** are logical rules that split feature-space into two **non-overlapping** subregions.

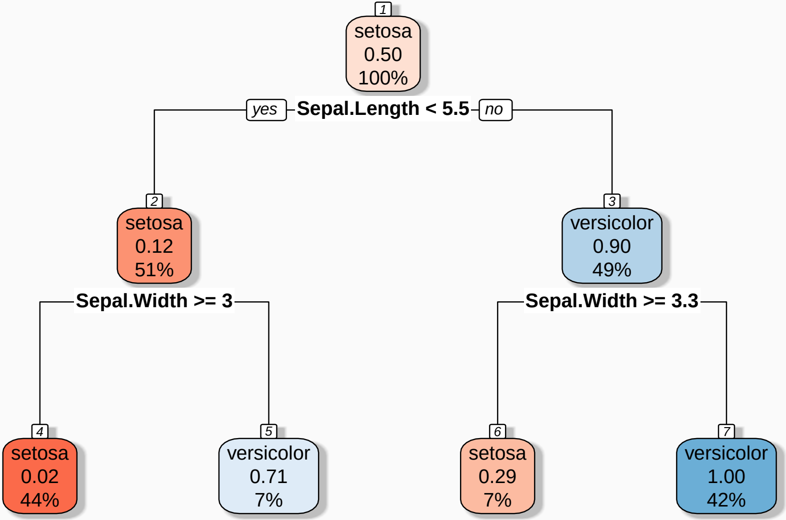
Example: Feature space

Features: Sepal Length, Sepal Width

Outcome: setosa/versicolor



Decision tree



Parts of a decision tree

- Root node
- Decision node
- Terminal node/ Leaf node (gives outputs/class assignments)
- Subtree

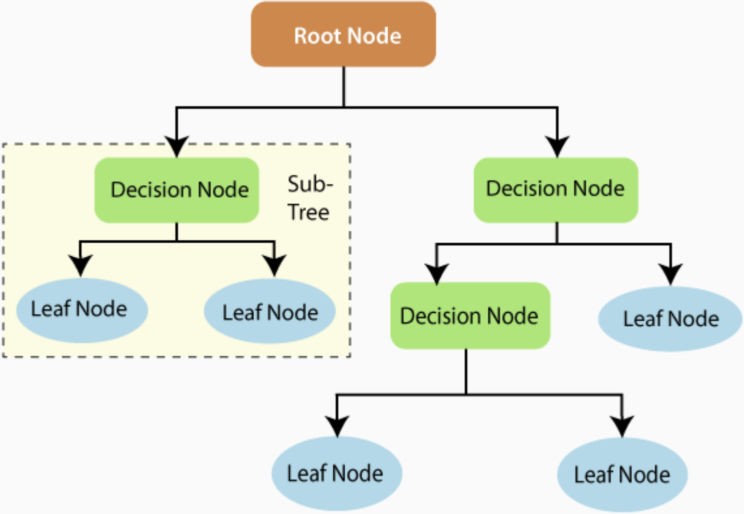
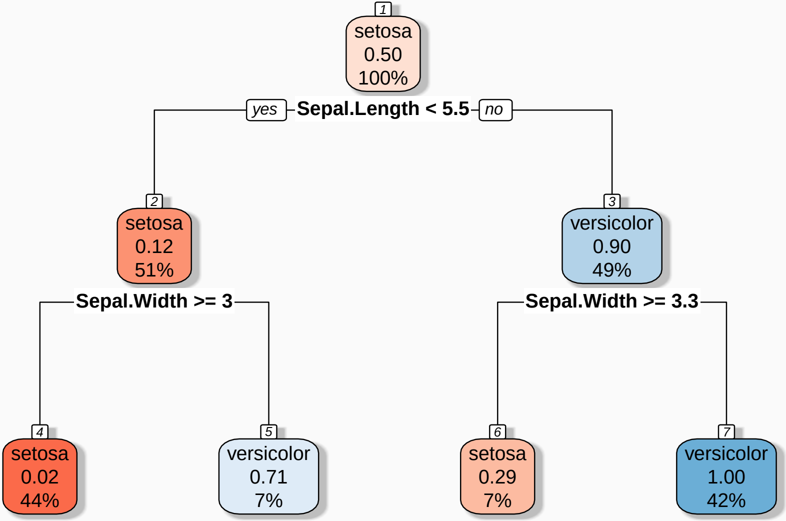
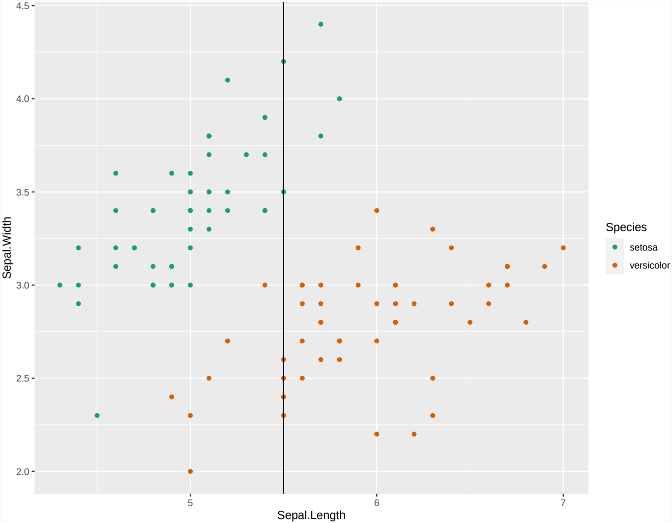


Image source:
`https://www.tutorialandexample.com/wp-content/uploads/2019/10/Decision-Trees-Root-Node.png`

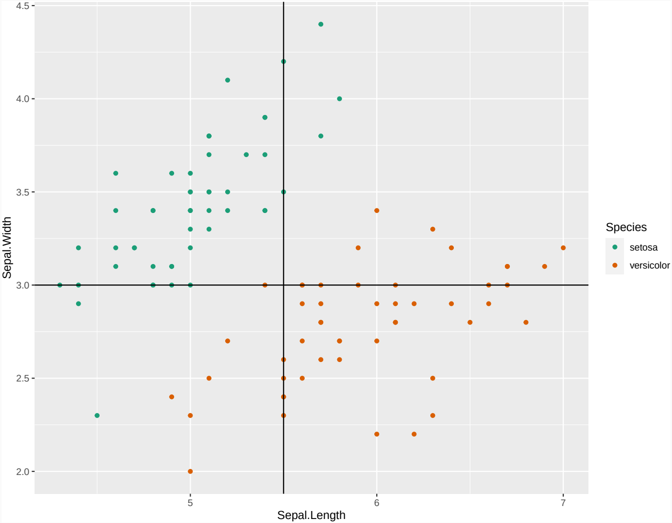
Decision tree



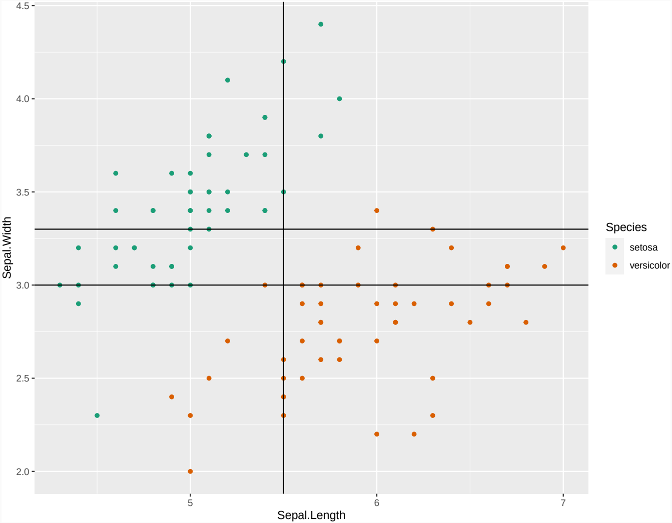
Root node split



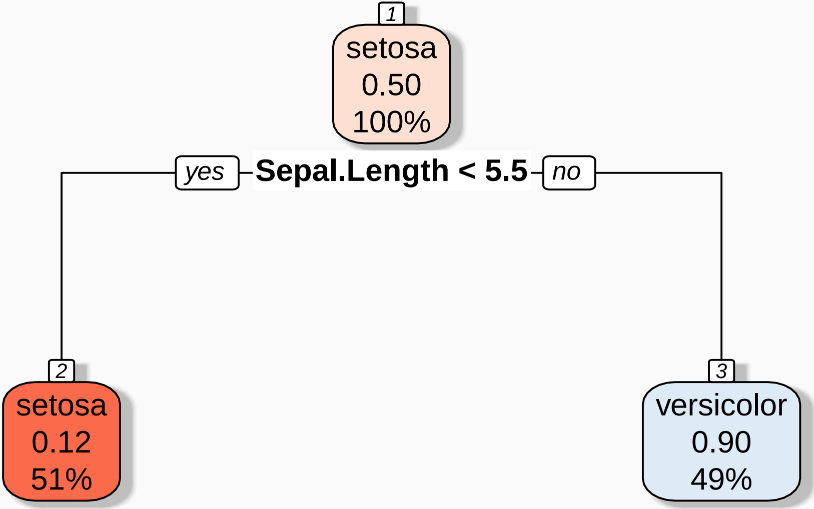
Root node split, Decision node split - right



Root node split, Decision node splits



Shallow decision tree



Two key ideas underlying trees

- Recursive partitioning (for constructing the tree)
- Pruning (for cutting the tree back)
- Pruning is a useful strategy for avoiding over fitting.
- There are some alternative methods to avoid over fitting as well.

Leo Breiman

Key references

Breiman, L., J. Friedman, R. Olshen, and C. Stone, 1984: Classification and regression trees. Wadsworth Books, 358.

Breiman, L., 1996: Bagging predictors. Machine learning, 24 (2), 123–140.

Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5–32. doi:10.1023/A: 1010933404324

Constructing Classification Trees

Recursive Partitioning

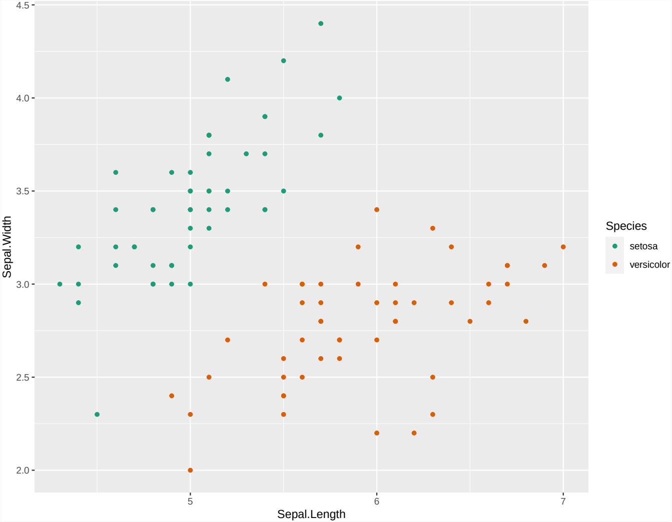
- Recursive partitioning splits P-dimensional feature space into nonoverlapping multidimensional rectangles.
- The division is accomplished recursively (i.e. operating on the results of prior division)

Main questions

- Splitting variable
  Which attribute/ feature should be placed at the root node?
  Which features will act as internal nodes?
- Splitting point
- Looking for a split that increases the homogeneity (or "pure" as possible) of the resulting subsets.

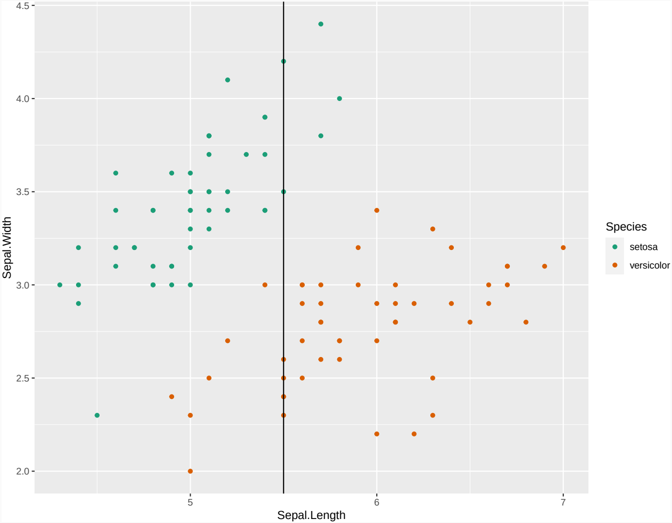
Example

split that increases the homogeneity



Example (cont.)

split that increases the homogeneity .



Key idea

1. Iteratively split variables into groups
2. Evaluate “homogeneity” within each group
3. Split again if necessary

How does a decision tree determine the best split?

Decision tree uses entropy and information gain to select a feature which gives the best split.

Measures of Impurity

- An impurity measure is a heuristic for selection of the splitting criterion that best separates a given feature space.
- The two most popular measures
    - Gini index
    - Entropy measure

Gini index

Gini index for rectangle A is defined by

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

p_k - proportion of records in rectangle A that belong to class k

- Gini index takes value 0 when all the records belong to the same class.

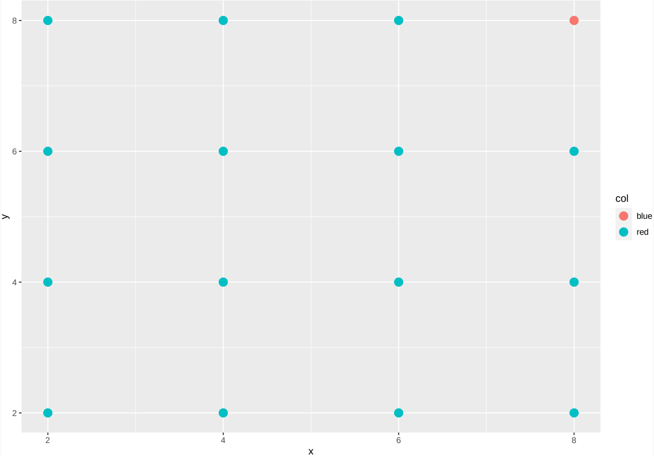
Gini index (cont)

In the two-class case Gini index is at peak when $p_k = 0.5$

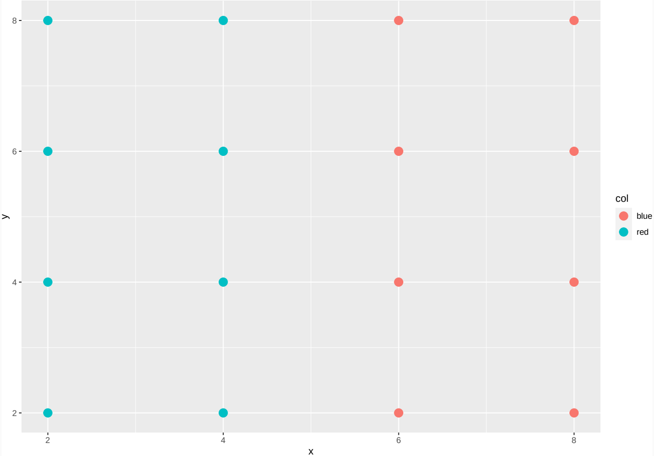
Entropy measure

$$entropy(A) = -\sum_{k=1}^{m} p_k log_2(p_k)$$

Example: Calculation (left)



Example: calculation (right) (cont.)



Finding the best threshold split?

In-class demonstration

Overfitting in decision trees

- Overfitting refers to the condition when the model completely fits the training data but fails to generalize the testing unseen data.
- If a decision tree is fully grown or when you increase the depth of the decision tree, it may lose some generalization capability.
- Pruning is a technique that is used to reduce overfitting. Pruning simplifies a decision tree by removing the weakest rules.

Stopping criteria

- Tree depth (number of splits)
- Minimum number of records in a terminal node
- Minimum reduction in impurity
- Complexity parameter (CP) - available in `rpart` package

Pre-pruning (early stopping)

- Stop the learning algorithm **before** the tree becomes too complex
- **Hyperparameters** of the decision tree algorithm that can be tuned to get a robust model

`max_depth`

`min_samples_leaf`

`min_samples_split`

Post pruning

Simplify the tree **after** the learning algorithm terminates

The idea here is to allow the decision tree to grow fully and observe the CP value

Simplify the tree after the learning algorithm terminates

- Complexity of tree is measured by number of leaves.

$L(T) =$ number of leaf nodes

- The more leaf nodes you have, the more complexity.
- We need a balance between complexity and predictive power

Total cost $=$ measure of fit $+$ measure of complexity

Total cost = measure of fit + measure of complexity

measure of fit: error

measure of complexity: number of leaf nodes ($L(T)$)

Total cost ($C(T)$) $= Error(T) + \lambda L(T)$

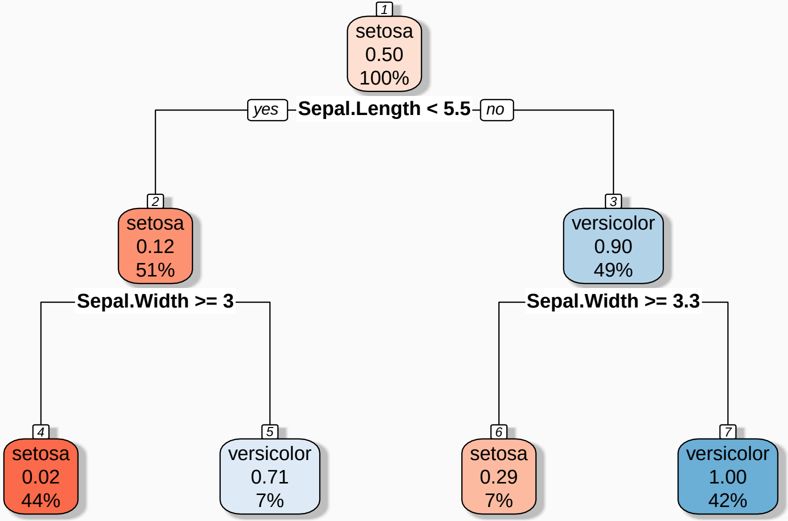
The parameter λ trade off between complexity and predictive power. The parameter λ is a penalty factor for tree size.

$\lambda = 0$: Fully grown decision tree

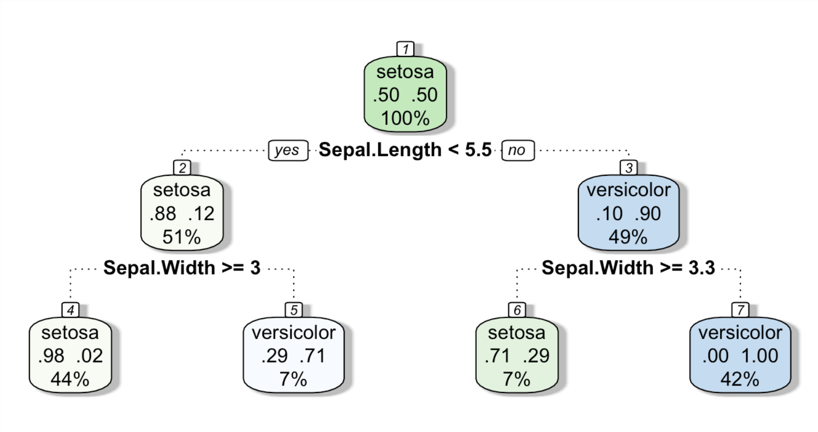
$\lambda = \infty$: Root node only

λ between 0 and ∞ balance predictive power and complexity.

Example: candidate for pruning (in-class)

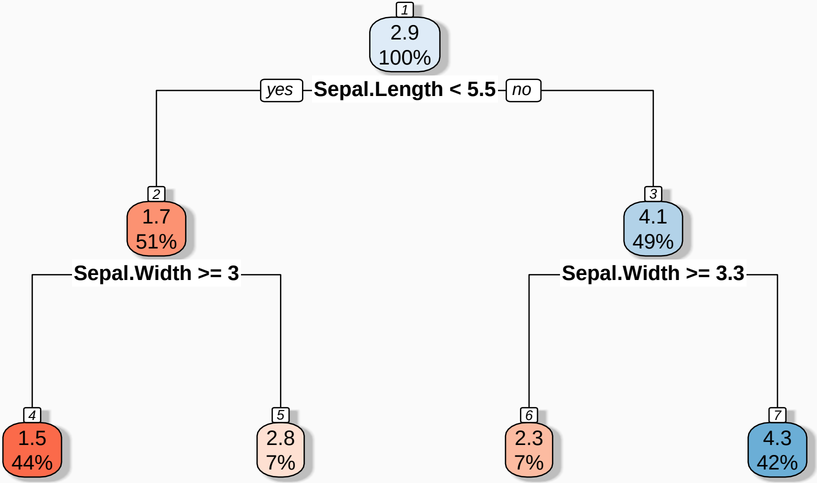


Classification trees - label of terminal node



labels are based on majority votes.

Regression Trees



Regression Trees

Value of the terminal node: average outcome value of the training records that were in that terminal node.

Your turn: Impurity measures for regression tree

Decision trees - advantages

- Easy to interpret
- Better performance in non-linear setting
- No feature scaling required

Decision trees - disadvantages

- Unstable: Adding a new data point or little bit of noise can lead to re-generation of the overall tree and all nodes need to be recalculated and recreated.
- Not suitable for large datasets