Decision Trees

DSA 6000: Data Science and Analytics, Fall 2019

Wayne State University

Find your Whisky

Dr. David Wishart

A tasting note on a Scotch whisky reads: "Appetizing aroma of peat smoke, almost incense-like, heather honey with a fruity softness".

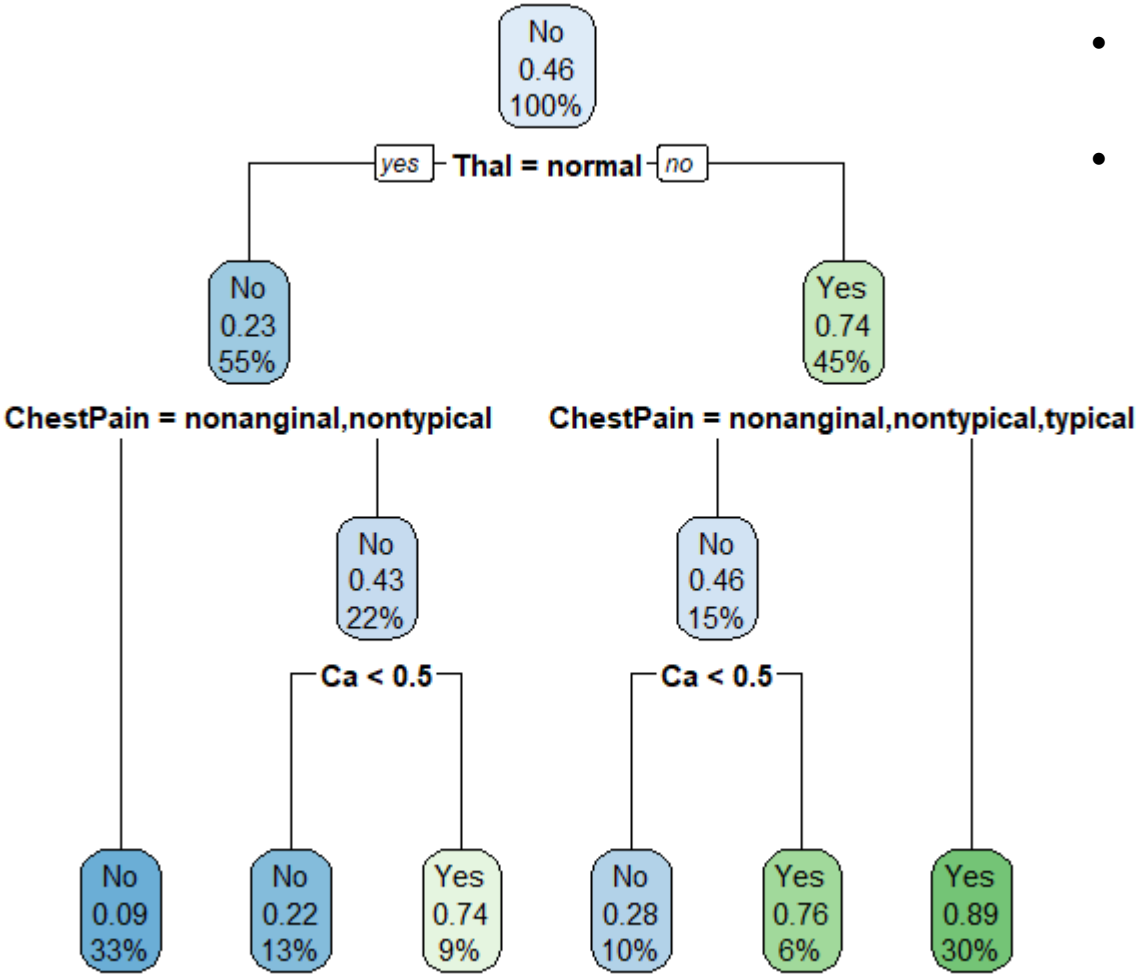
1. You like a particular malt whisky, and want to know what other brands tastes similar …
2. You want to diversity you collection, and want to choose a range of malts that taste different.

Data sets: adn.biol.umontreal.ca/~numericalecology/data/scotch.html
wishart.org/home.html

Decision Trees

- Decision Tree is a predictive model represented in a tree-like structure.
  - Regression Tree
  - Classification Tree
- Without making excessive assumptions about the data, a decision tree partitions the input space into rectilinear regions and ultimately gives a set of *If … Then …* rules to classify output or make predictions.

Example



- Classify AHD (heart disease), yes or no
- The root node contains 100% of all observations, 0.46 (or 46%) of observations in the node are yes, so the node is classified as a No.

Data set: archive.ics.uci.edu/ml/datasets/heart+disease

How is a decision tree constructed?

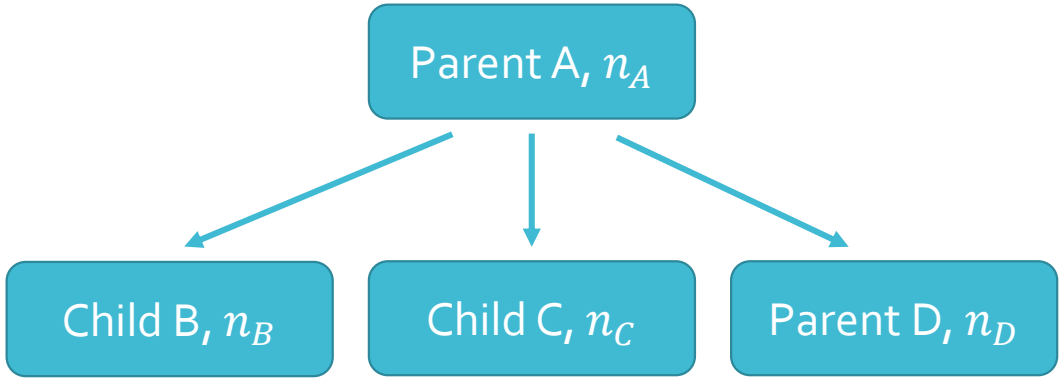
- Recursive Partitioning is the standard framework for fitting decision trees.
- Starting from the root node, each internal node is split into two or more child nodes based on the input values.
  - A splitting criterion is used to choose the split.
- The split stops when some terminal condition is met.
- The terminal nodes of the tree are called leaves, which represent the predicted target.
- Cases move down the tree along branches according to their input values and all cases reaching a particular leaf are assigned the same predicted value.

Node splitting process

- Determine a pool of candidate splits
  - Which variable to split on? (e.g., Age, Income or Balance)
  - What is the splitting rule? (e.g., Age > 35)
- Assess the quality of all candidate splits and choose the best one
  - What is a good split?
    - If the target variable distribution is the same in the child nodes as they are in the parent node, is it a good split?
    - If a split results in pure children (nodes each containing a single target class), is it a good split?
  - Most well-known splitting criteria:
    - Gini Index
    - Entropy
    - Chi-square test
- Typical two-step process:
  1. Select the best split on each input variable
  2. Select the best of the above

Splitting Criteria

- A splitting criterion measures the reduction in variability of the target distribution in the child node.
- The goal is to reduce variability, thus increase purity, in the child nodes.
- Let $i(\cdot)$ be some measure of within-node impurity, and n be the number of observations in a node.
- If a parent node A is split into three children nodes B, C and D, then the impurity reduction induced by the split is given by:
  - $\Delta i = i(A) - (\frac{n_B}{n_A} i(B) + \frac{n_C}{n_A} i(C) + \frac{n_D}{n_A} i(D))$

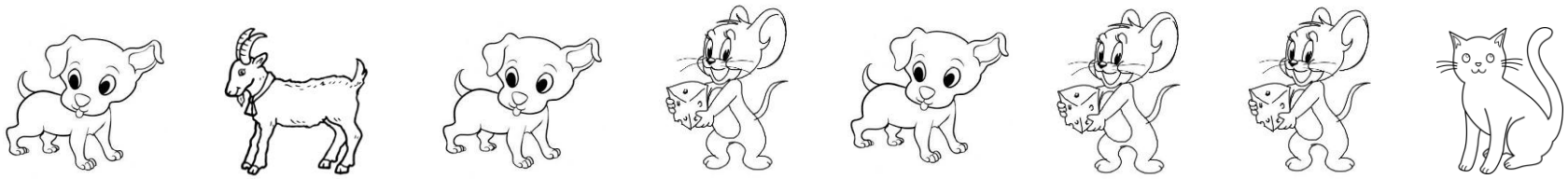


The split that induces the greatest Δi is the best split.

Gini Index

For a pure node, the Gini Index is 0.
As the number of evenly distributed classes increases, the Gini Index approaches 1.

- The Gini index is a measure of variability for categorical data (developed by Italian statistician Corrado Gini in 1912).
- Let $p_1, p_2, \ldots, p_r$ be the relative frequencies of each target class in a node, then Gini Index of the node is given by
  - $1 - \sum_{j=1}^{r} p_j^2$



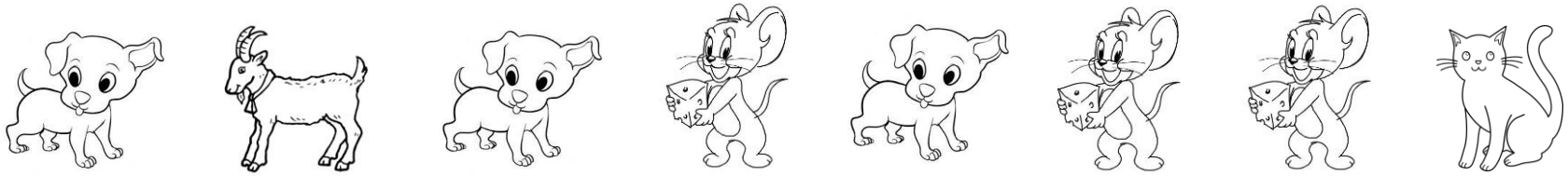
$$1 - 2\left(\frac{3}{8}\right)^2 - 2\left(\frac{1}{8}\right)^2 = 0.69$$

$$1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = 0.24$$

Entropy

For a pure node, the entropy is 0.
For a node containing the same number of observations for each class, the entropy is 1.

- Let $p_1, p_2, \ldots, p_r$ be the relative frequencies of each target class in a node, then the entropy of the node is given by:
  - $-\sum_{j=1}^{r} p_j \log_2(p_j)$



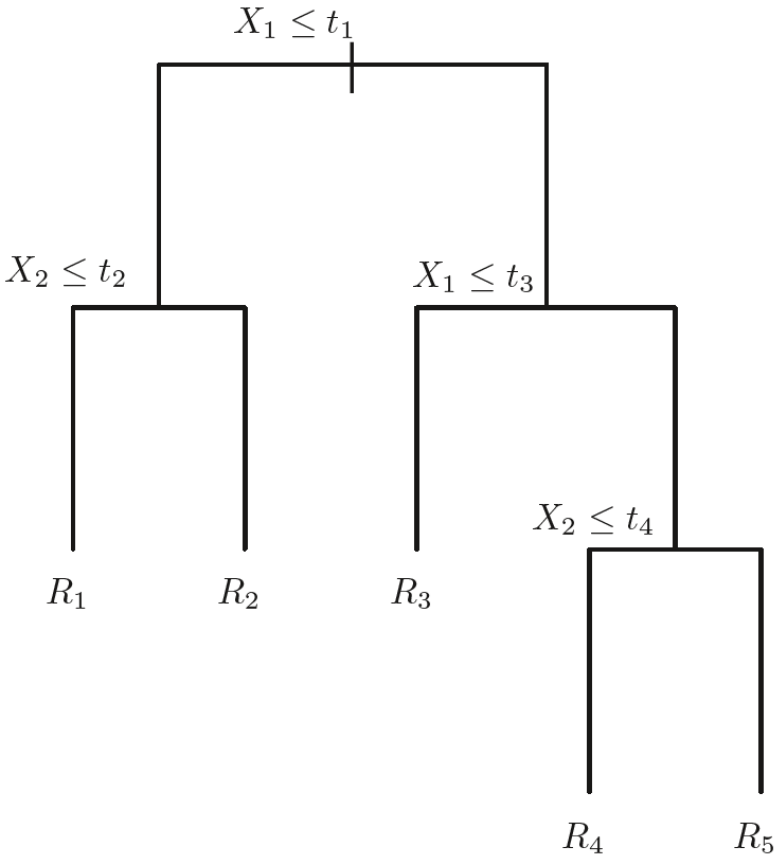
$$-2\left(\frac{3}{8}\right)\log_2\left(\frac{3}{8}\right) - 2\left(\frac{1}{8}\right)\log_2\left(\frac{1}{8}\right) = 1.81$$

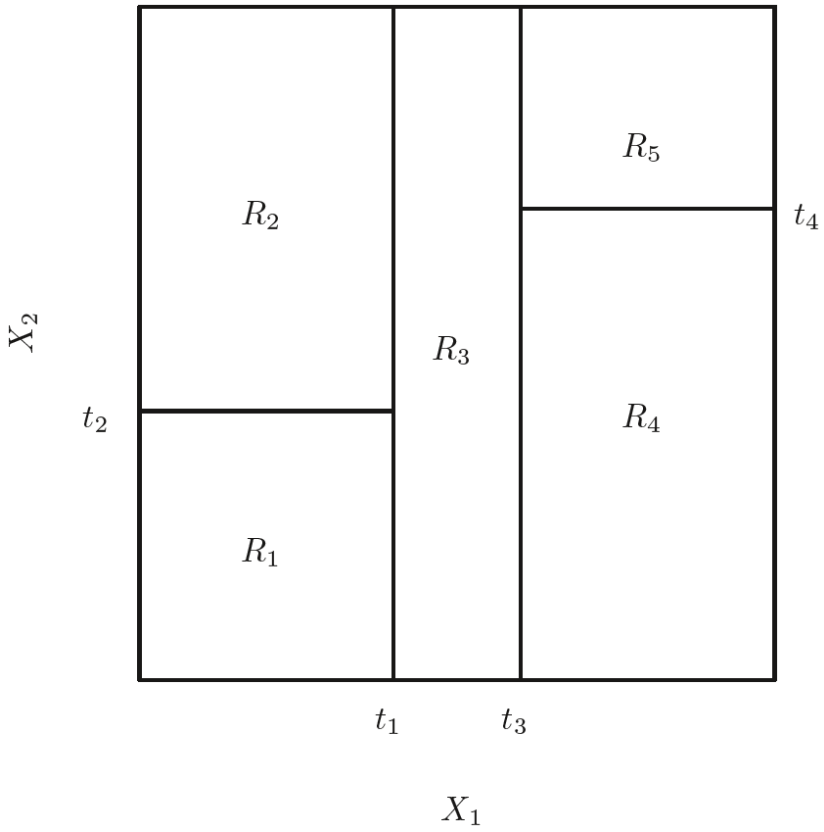
$$-\left(\frac{6}{7}\right)\log_2\left(\frac{6}{7}\right) - \left(\frac{1}{7}\right)\log_2\left(\frac{1}{7}\right) = 0.59$$

How a tree partitions the input space

Standard decision trees partition the input space using boundaries that are parallel to the input coordinates.

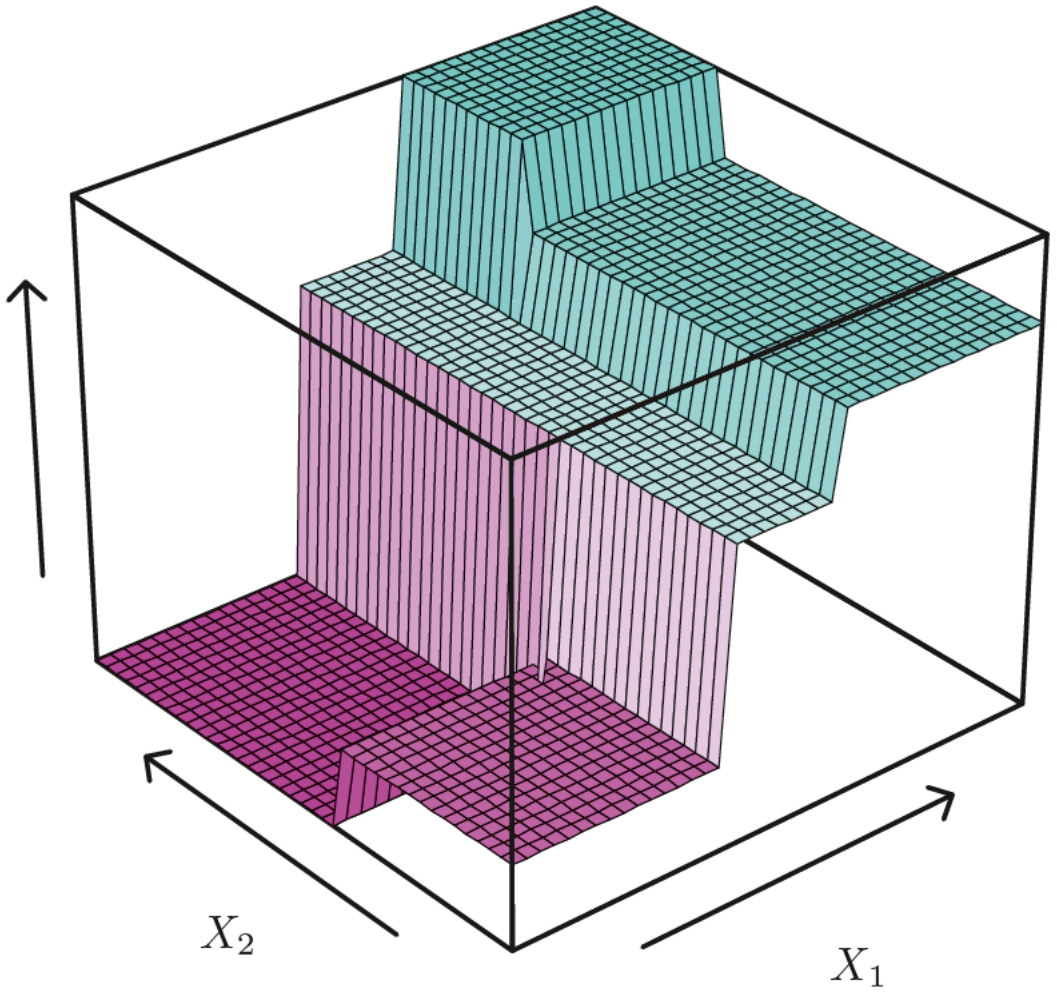
- Easy to interpret
- Resistance to the curse of dimensionality





Regression Tree

Observations that fall in the same leaf node has the same predicted value, which is the mean response of that node.

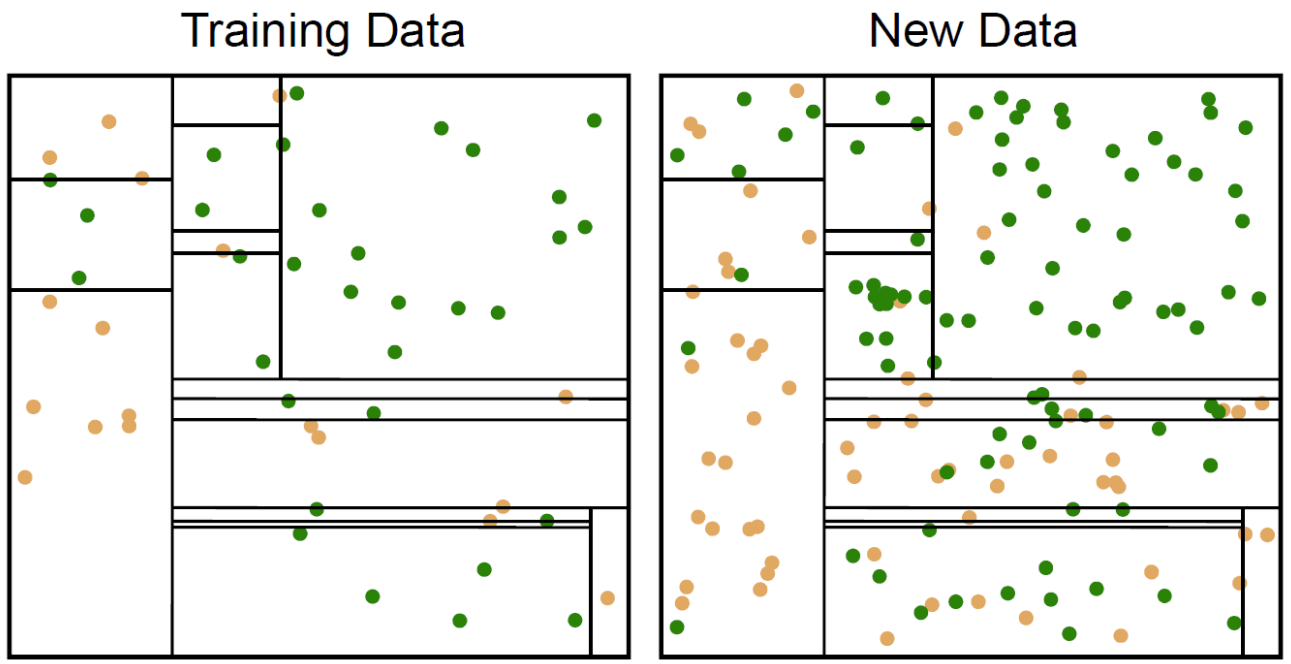


The tree is built by trying to minimize the RSS.

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

Tree Pruning

- A decision tree can be grown until every node is as pure as possible.
- The tree with the greatest possible purity on the training data is the maximal classification tree.
- Maximal tree is the result of overfitting.
- Use test data or cross-validation to prune overfitted branches.



Advantage and Disadvantages

- Advantages
  - Trees are very easy to explain to people, easier than linear regression.
  - Trees can be displayed graphically, and are easily interpreted by a non-expert.
  - Trees can easily handle categorical variables without the need to create dummy variables.
  - Trees can easily handle missing values (by treating missing value as a separate category).
  - Trees are immune to scaling difference among features
- Disadvantages
  - Prediction accuracy is less competitive than other methods
  - Trees can be very fragile (non-robust). A small change in data can cause a large change in the final estimated tree.

Bagging, Random Forest, Boosting

- Bagging
  - Construct many trees (without pruning) on bootstrapped training samples, then average their predictions
- Random Forest
  - Similar to bagging, but each split only considers a random subset of the predictors as candidates, e.g., only $m = \text{round}(\sqrt{p})$ predictors are considered at each split
  - The trees are less correlated than they are in bagging
- Boosting
  - Build many trees *sequentially*, each new tree is built to learn the residual error in the current prediction. The final prediction is a weighted sum of predictions from all trees.

Boosting Algorithm

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.

2. For $b = 1, 2, \ldots, B$, repeat:

   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).

   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$

   (c) Update the residuals,

   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$

Software packages

- Algorithms
  - CART, CHAID, C4.5, C5.0, ID3
  - CRUISE, QUEST, GUIDE
- R packages
  - DT: rpart, tree, party, C50, bsnsing
  - RF: randomForest
  - Boosting: gbm
- Integrated toolkit
  - rattle