## **[Classification](https://youtu.be/7VeUPuFGJHk?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&t=837)** [Decision Trees Part 1, Basics](https://youtu.be/7VeUPuFGJHk?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&t=837)

[Ranger documentation](https://www.rdocumentation.org/packages/ranger/versions/0.12.1/topics/ranger)

[Hyperparameter explanation](https://bradleyboehmke.github.io/HOML/random-forest.html#hyperparameters)



1. Classification Trees: Predictor -- Is the fruit a citrus fruit?
2. Terms
   1. Root Note
   2. Internal Nodes
   3. Leaves (Leaf Nodes)
3. How do you determine the rank/order of the nodes from top to bottom?
   1. Determine how ell each feature predict sthe outcome
      1. How well does the shape (categorical) of the fruit determine if it’s a citrus fruit?
      2. How well does the weight (numeric) of the fruit determine if it’s a citrus fruit?
   2. Put in ML lingo, determine how impure each feature is!
      1. The less impure (more pure) a feature is, the better it is at predicting the outcome.
      2. [*Gini Impurity*](https://youtu.be/7VeUPuFGJHk?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&t=438)
         1. For categorical vars
            1. count unique combinations of the predictor and outcome being yes or no to arrive at the probabilities

|  |  |  |
| --- | --- | --- |
|  | Outcome TRUE | Outcome FALSE |
| Predictor TRUE | # count | # count |
| Predictor FALSE | # count | # count |

* + - * 1. applying weighing mechanism to arrive at Gini Impurity for the feature’s 2 nodes
      1. [For numeric vars](https://youtu.be/7VeUPuFGJHk?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&t=836)
         1. Sorting
         2. Averaging
         3. Create #counts using inequalities/cutoffs
         4. Find the cutoff where the GI is the lowest

1. When do you stop splitting?
   1. When the Gini Impurity of the next node down would be greater than the current bottommost node

## **[Classification](https://youtu.be/wpNl-JwwplA?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF)** [Decision Trees Part 2, Feature Selection and Missing Data](https://youtu.be/wpNl-JwwplA?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF)

1. Feature Selection
   1. Built in Automatic Feature Selection
      1. If a feature is unable to reduce Gini impurity, it won’t appear in the tree
   2. Manual Feature Selection
      1. Setting thresholds for marginal decrease in Gini impurity at each split
      2. Helps prevent overfitting
2. Missing Data for Feature X
   1. Imputing with
      1. the most common value
      2. the value suggested by a closely correlated feature Y

## [**Regression** Decision Trees](https://youtu.be/g9c66TUylZ4?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF)

1. Regression Trees: Predictor – How heavy in oz is the fruit?
2. How are Predictions Made For One Predictor?
   1. Keep splitting downwards at thresholds of decreasing SSRs
      1. How do you know when to stop splitting?
      2. Use a heuristic to determine **min\_n**,the number of minimal observations needed for leaf nodes
   2. The average of the observations within the threshold of the leaf node is the predicted value at that threshold
3. For multiple predictors
   1. For each predictor, split once, at the best threshold that minimizes SSRs
   2. The best predictor, the one with the lowest SSR, becomes the root node on top
      1. Its predicted values is the average outcome value at its threshold
   3. The next best predictor, … 2nd lowest SSR becomes the 2nd node down
      1. Its predicted values is the average outcome value at its threshold
   4. …
4. How do you determine the rank/order of the feature nodes from top to bottom?
   1. Features with the smallest SSRs should be on top
5. [ELI5 Summary](https://youtu.be/g9c66TUylZ4?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&t=1256)

## [Random Forests](https://youtu.be/J4Wdy0Wc_xQ?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF)

1. Bootstrap Data
   1. Sample data with replacement
   2. Typically about 1/3 of the original dataset doesn’t end up being used
      1. OOB out of bag (boot) samples
2. Create Decision Tree #1
   1. Use a random subset of preditors/vars at each step (at each horizontal level of nodes)
      1. The number is set using **mtrees**
      2. Once a candidate is used at a higher level, it is not used again at a lower level
3. The Same Process is repeated until we have a collection of Decision Trees, a Random Forest
4. Using a single observation, we bag to determine the outcome
   1. Bagging = bootstrapping + using the aggregate to make a decision
5. Accuracy
   1. Determined by bagging OOB and comparing predicted vs actual outcomes
   2. Proportion that was incorrectly classified = OOB Error