W06D05 – Trees and Forests

Instructor: Brian Lynch

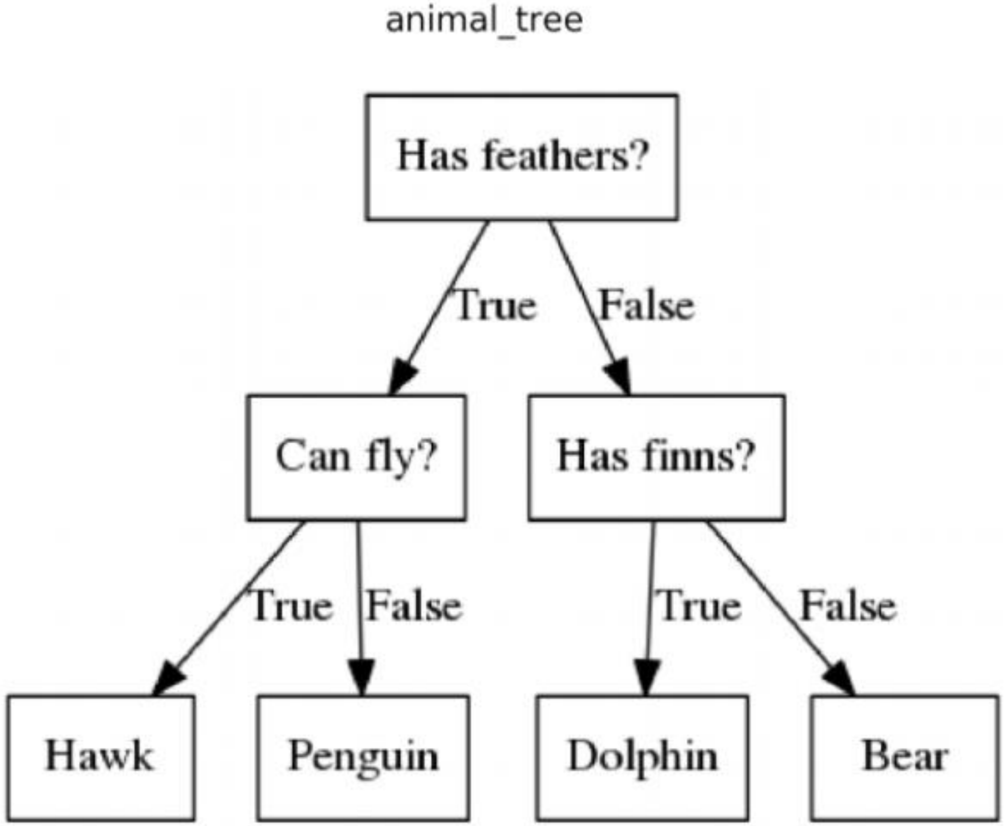
Credit: Zain Hasan, Jeremy Eng

Outline

- Decision Trees
    - Classification Trees
    - Regression Trees
- Random Forests
- Ensemble Methods
    - Bagging
    - Boosting
    - Stacking
- Demo

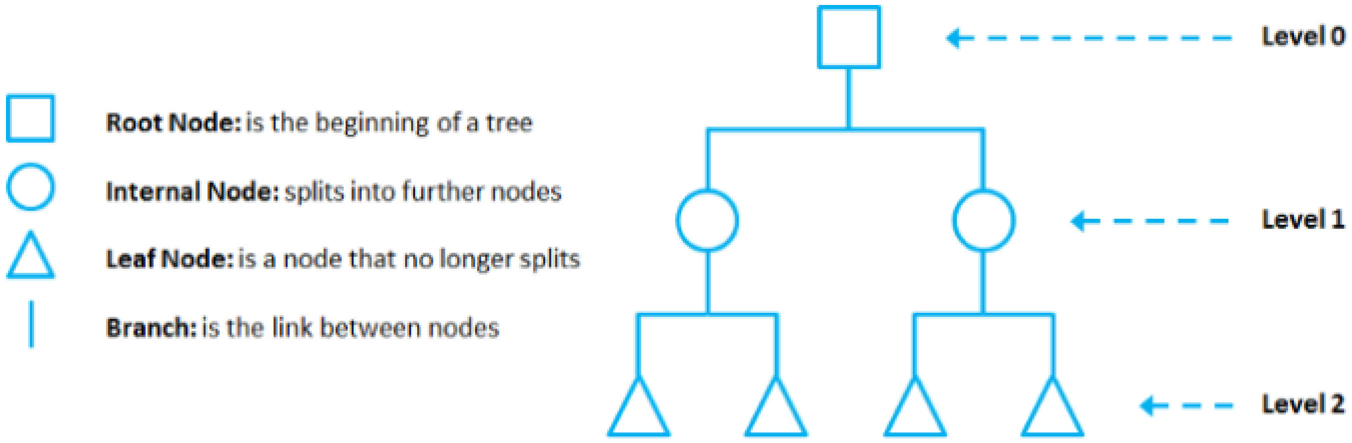
Decision Trees

- Flow chart based on features
- Can program using nested if-else statements.
- Tree models will create these automatically in an optimal fashion.
- Ex: Classification Tree



Decision Trees

- Flow chart based on features
- Can program using nested if-else statements.
- Tree models will create these automatically in an optimal fashion.
- Ex: Classification Tree
- Terminology:
  - Nodes and Branches
  - Root, internal, and leaf nodes.
  - Levels (depth)
- Let’s focus on classification trees first.



□ **Root Node:** is the beginning of a tree

○ **Internal Node:** splits into further nodes

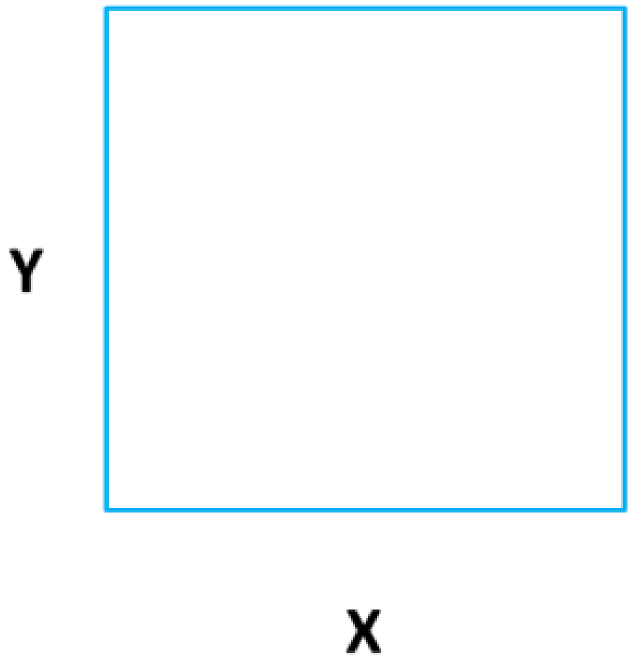
△ **Leaf Node:** is a node that no longer splits

| **Branch:** is the link between nodes

In this example, a DT of 2 levels.

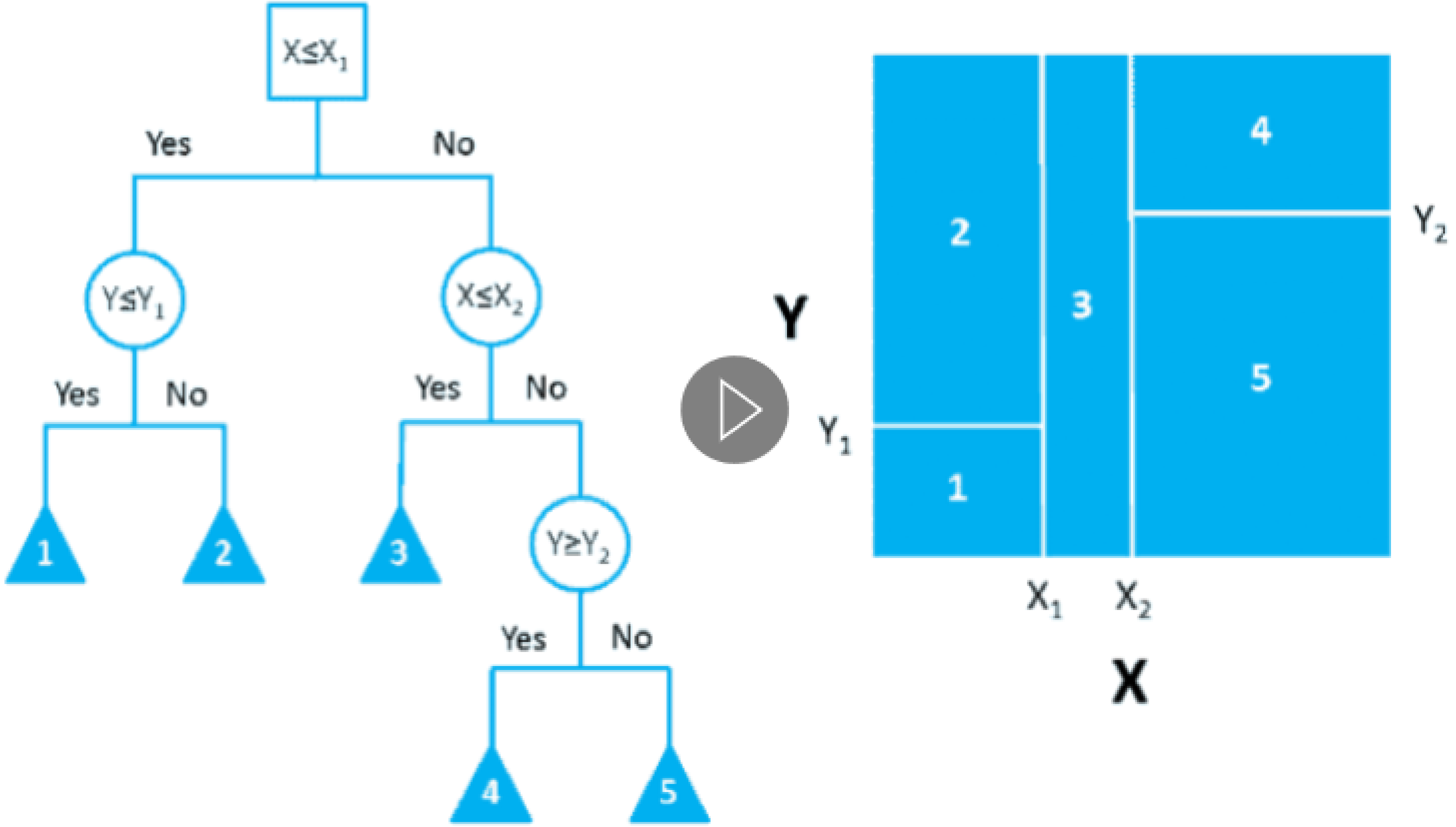
Classification Trees

- Classification trees create regions in the feature space.
  - Each boundary line represents a root/internal node
  - Each region represents a leaf node



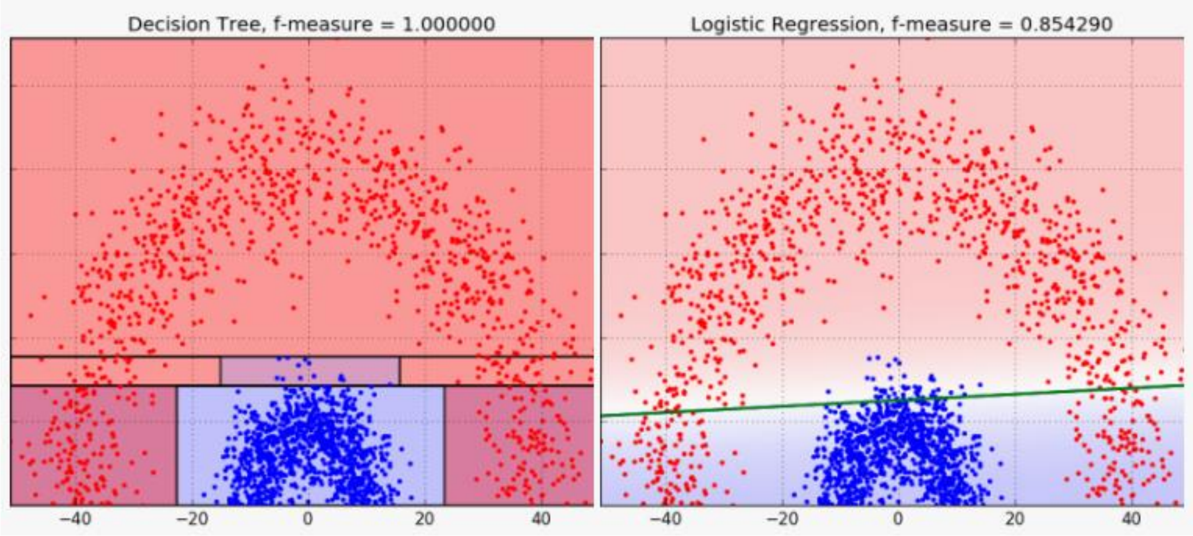
Classification Trees

- Classification trees create regions in the feature space.
  - Each boundary line represents a root/internal node
  - Each region represents a leaf node



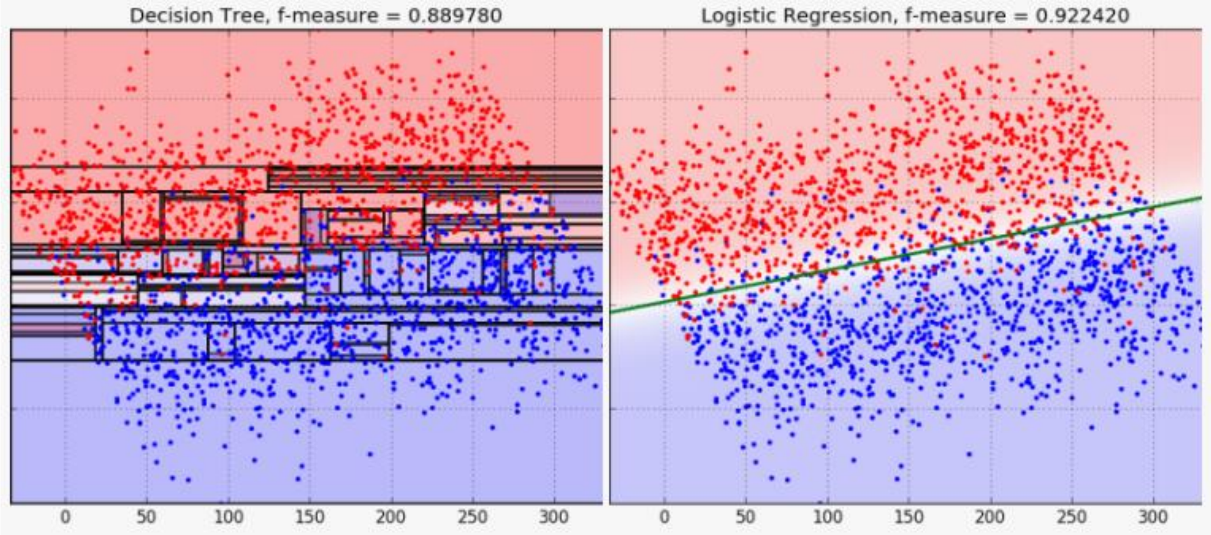
Classification Trees vs Logistic Regression

- Classification trees create regions in the feature space.
- Logistic regression creates a single decision boundary line.
- Example of over-fitting (2 features).
  - Main drawback with Trees



Decision Tree, f-measure = 1.000000

Logistic Regression, f-measure = 0.854290



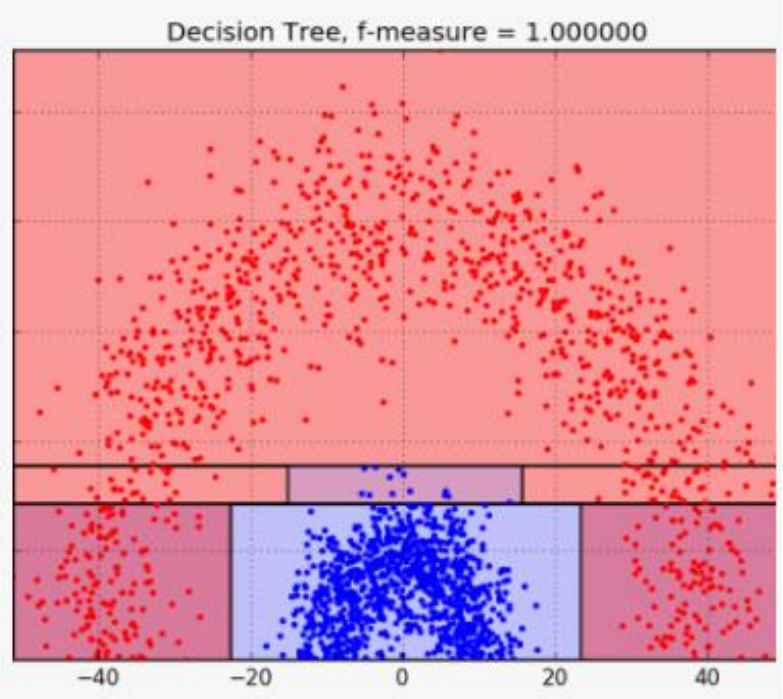
Decision Tree, f-measure = 0.889780

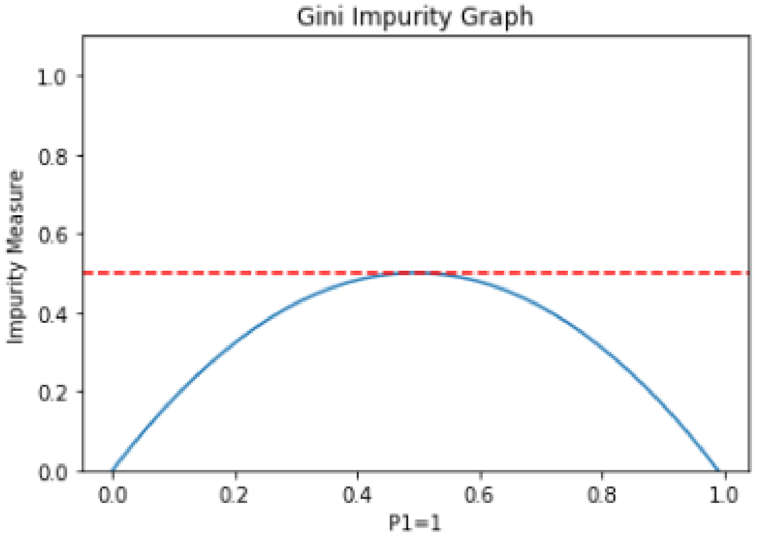
Logistic Regression, f-measure = 0.922420

Classification Trees

- How do we decide the split in an optimal way?
- Pick a criterion and minimize it across possible splits
  - Based on the proportions after split
  - Popular: Gini, entropy, misclassification (details)
- Ex: Gini impurity
  - *C*=number of classes
  - *p(i)*=proportion of class *i*

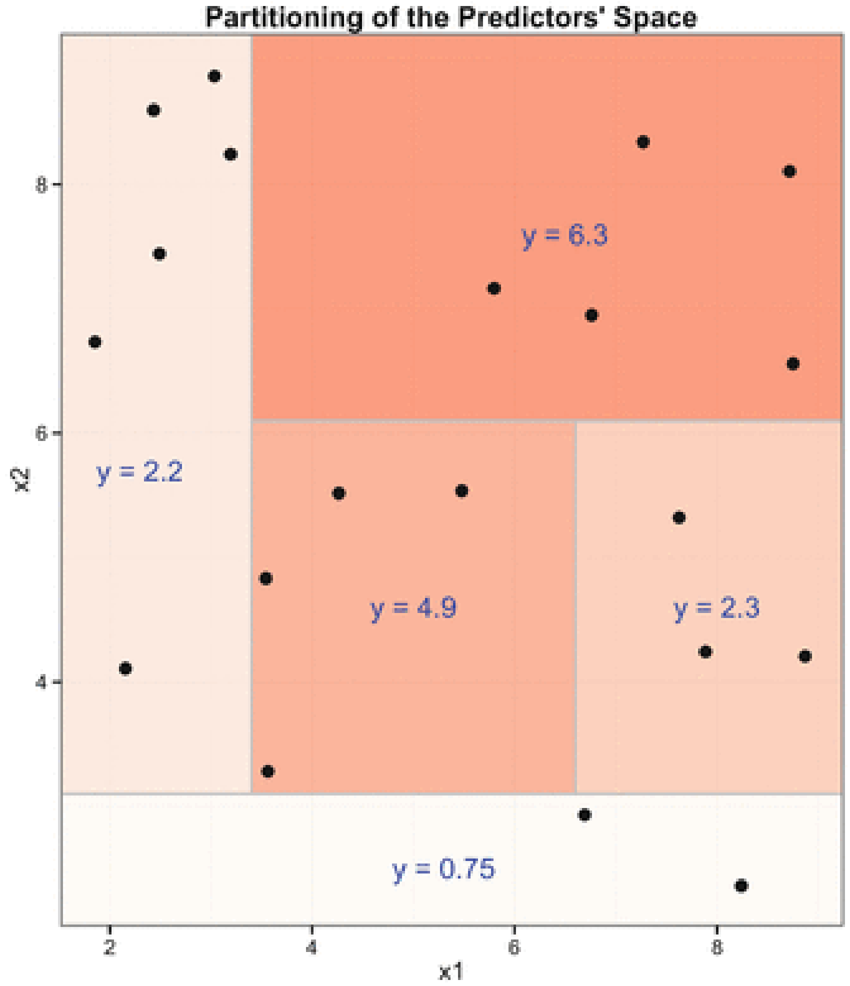
$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$



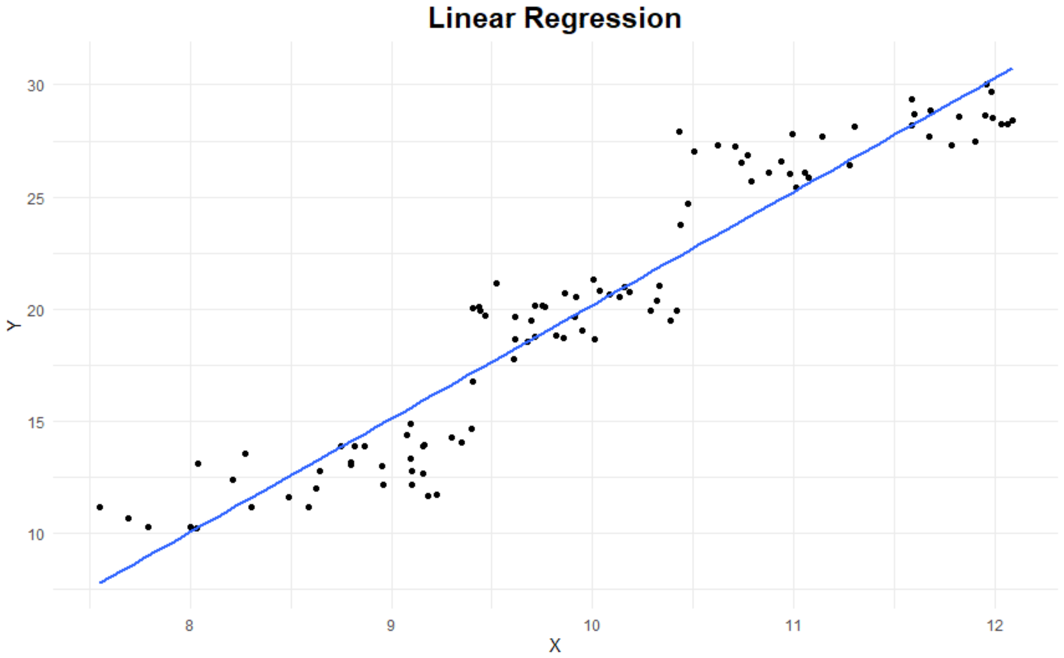


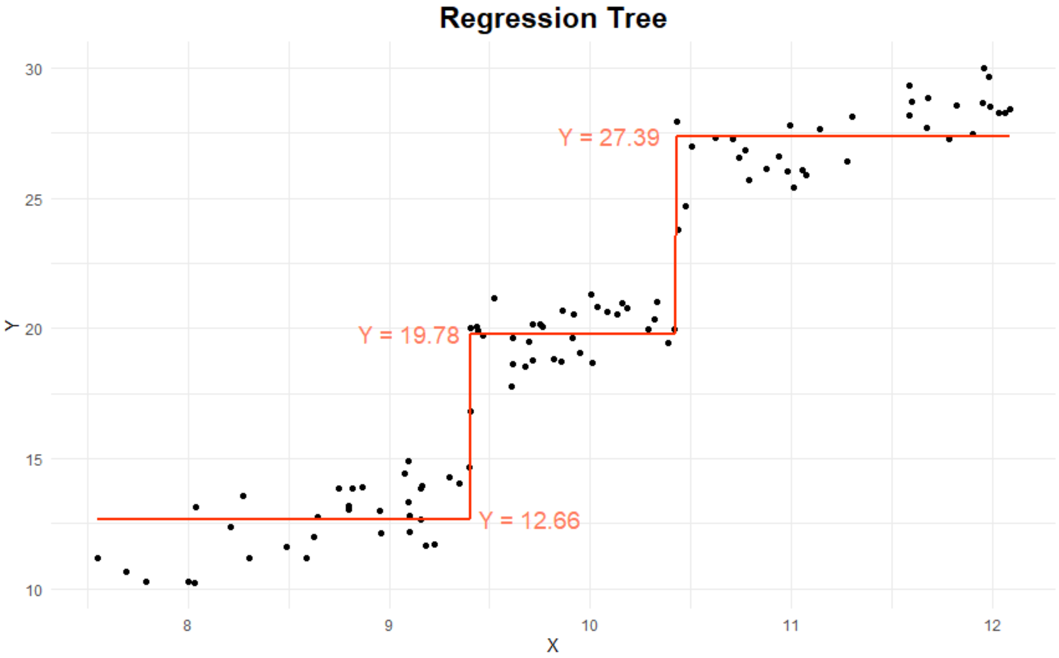
Regression Trees

- Leaf nodes will now give us a number, not a class.
- Regression tree criterion are calculated on the values in each region.
  - Popular: MSE, MAE, Half-Poisson deviance (details)
  - Minimized across all possible splits

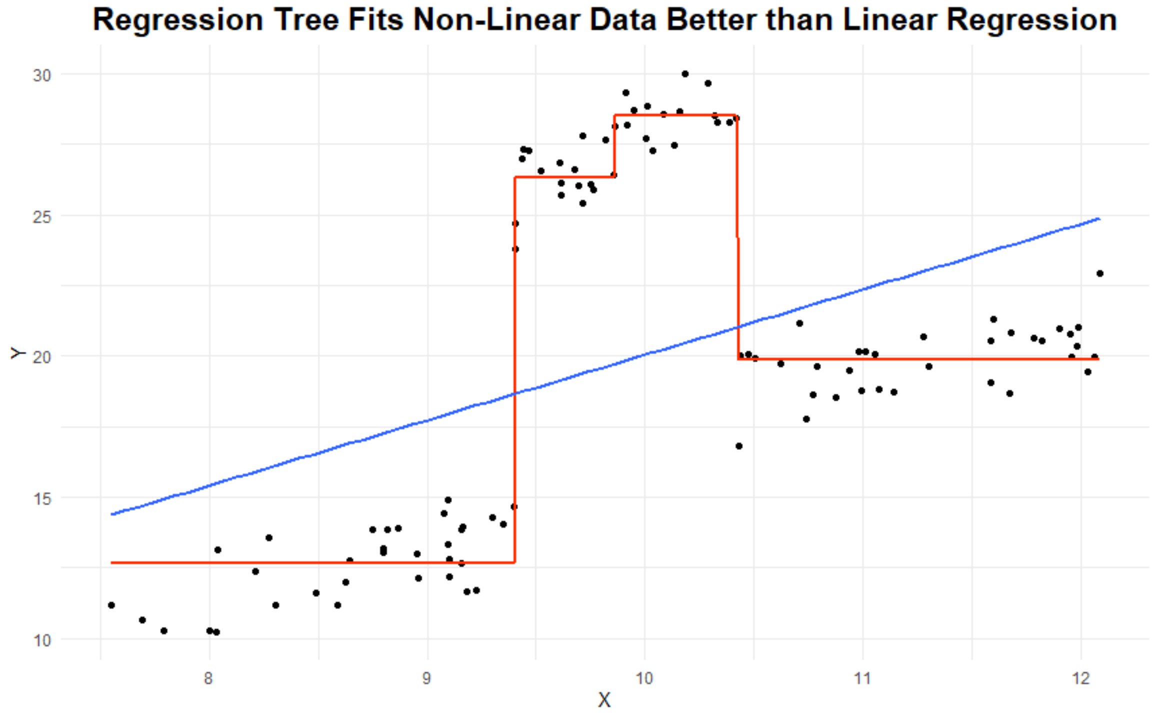


Regression Trees vs Linear Regression





Regression Trees vs Linear Regression

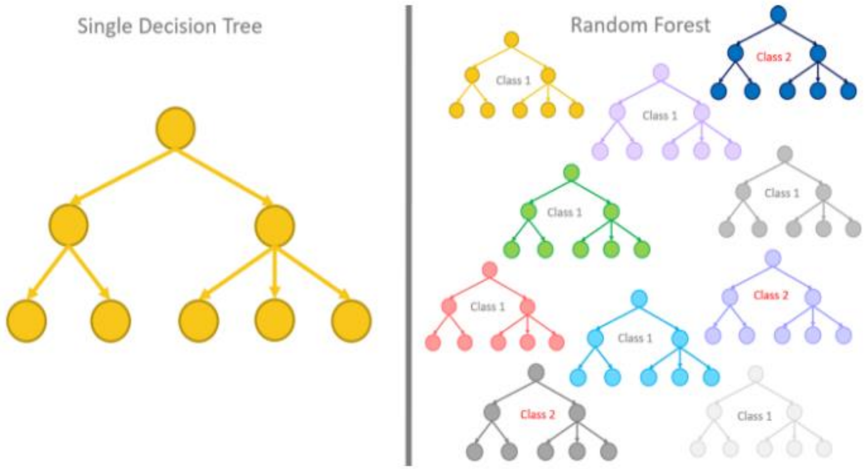


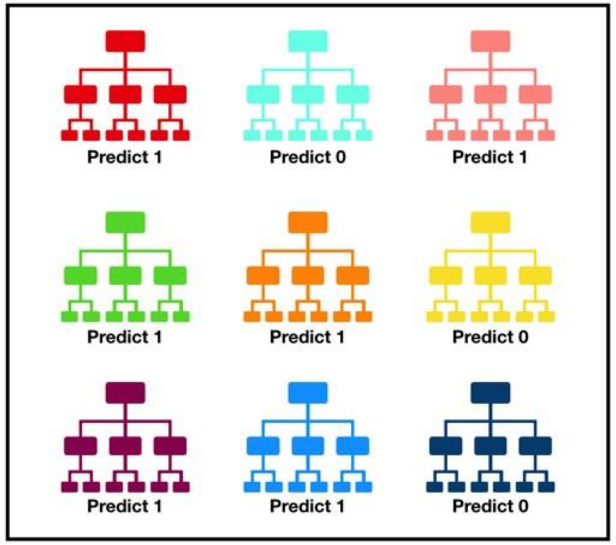
Decision Trees: Pros and Cons

- Pros:
  - Simple to understand and interpret.
  - Can be visualized
  - Requires little data preparation (doesn’t require normalization, can work with NaNs)
  - Can handle multi-class classification well.
- Cons:
  - Tendency to overfit (pruning techniques are needed)
  - Can be unstable (small change to data may result in a completely different tree)
  - Each node is locally optimized (not globally)

Random Forests

- Addresses the problem that decision trees are susceptible to over-fitting.
- General idea: fit a diverse set of trees by injecting “randomness”.
- Then use the most common (or average) of all the predictions as our single prediction.

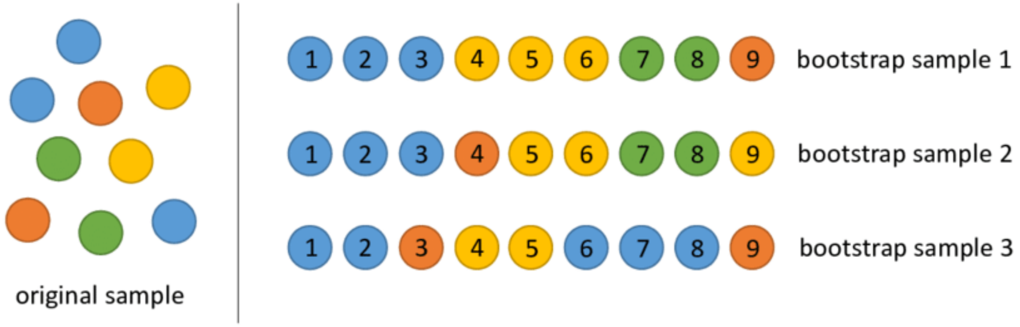


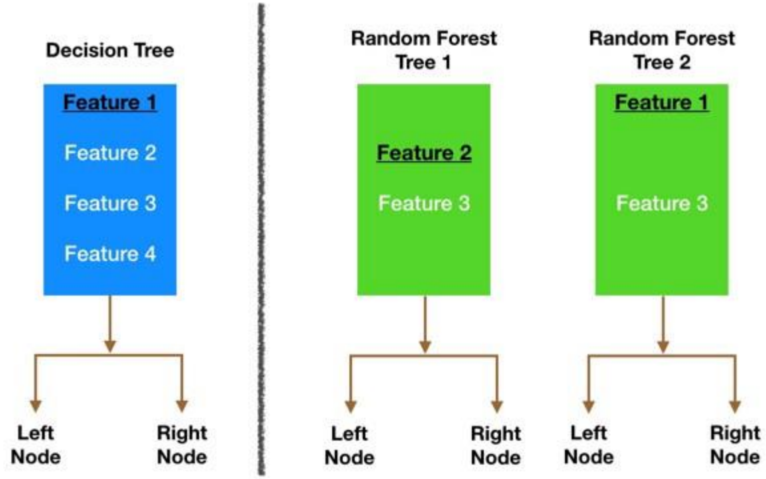


Tally: Six 1s and Three 0s
Prediction: 1

Random Forests

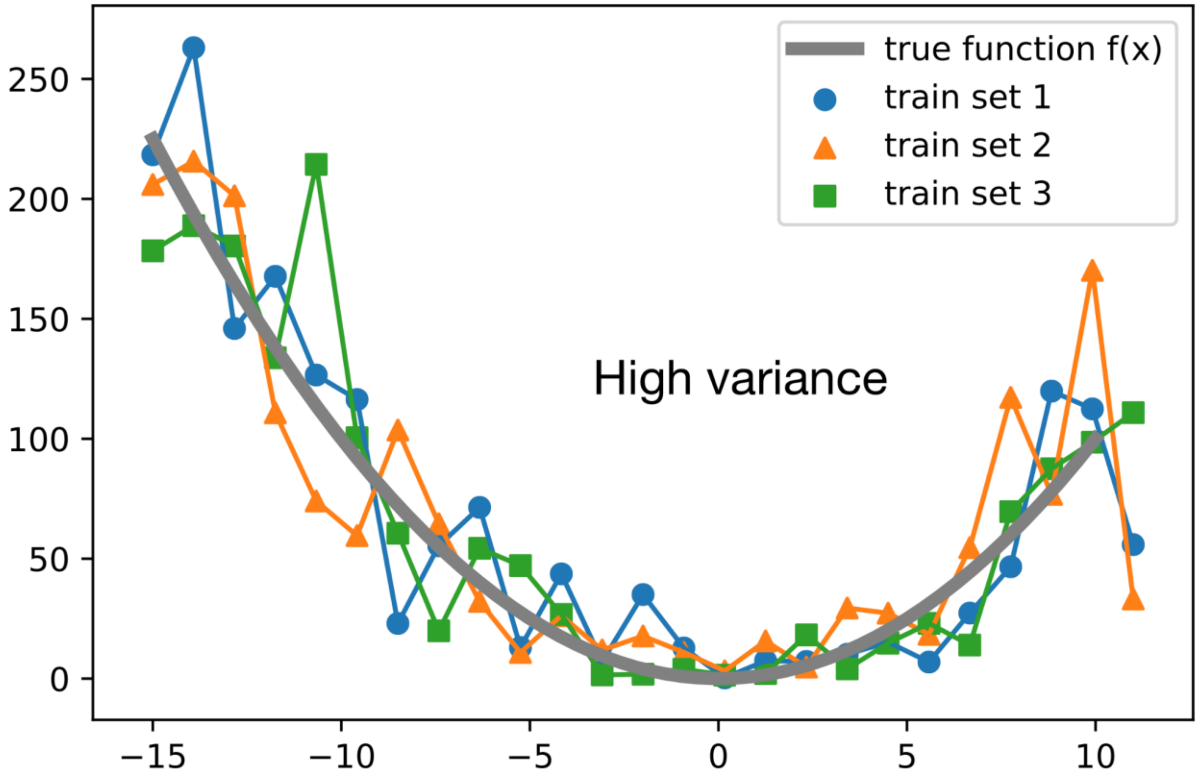
- Ways to inject randomness:
  1. Create “bootstrap samples”, and then build a tree for each bootstrap sample
  2. At each split, consider only a random subset of features.


original sample
1 2 3 4 5 6 7 8 9 bootstrap sample 1
1 2 3 4 5 6 7 8 9 bootstrap sample 2
1 2 3 4 5 6 7 8 9 bootstrap sample 3



Random Forests

- Averaging many over-fitted models reduces variance.



Random Forests

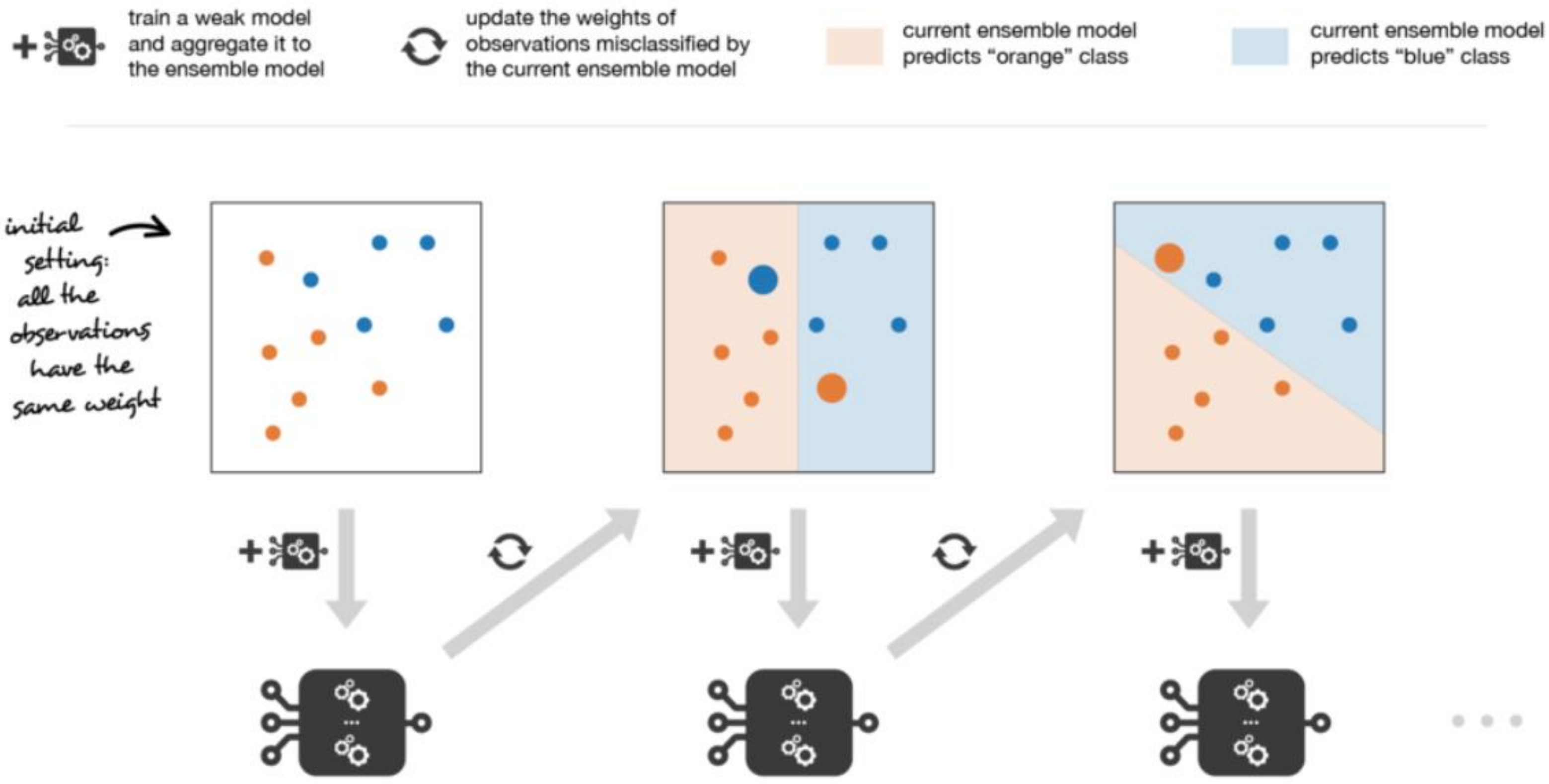
- Accuracy
  - Usually more accurate when compared to decision trees.
  - Usually one of the best performing off-the-shelf classifiers.
- Speed
  - Slower than decision trees because we are training multiple trees.
  - But can easily parallelize training because trees are independent.
- Overfitting
  - Addresses the over-fitting tendency of decision trees.
- Interpretability
  - Decision trees are more interpretable than random forests.

Ensemble Methods

- Random Forests are an example of an ensemble method.
- Ensemble methods are techniques that create multiple models (weak models) and then combine them to produce improved results.
- Bagging (bootstrap and aggregate, e.g. random forest)
    - Same type of weak model used, models learn in parallel, combined in a deterministic process.
    - Addresses over-fitting.
- Boosting
    - Add one model at a time that addresses the “shortcomings” of the current ensemble (iterative process).
    - Aggregation (averaging) is done during training, not after.
    - Addresses under-fitting.
- Stacking
    - Use a variety of weak models as input to a “meta-model”.
    - Similar to bagging, but can use different types of models.
- Source

Ensemble Methods: Boosting Example

- AdaBoost (Adaptive Boosting)



train a weak model and aggregate it to the ensemble model

update the weights of observations misclassified by the current ensemble model

current ensemble model predicts "orange" class

current ensemble model predicts "blue" class

initial setting: all the observations have the same weight

Adaboost updates weights of the observations at each iteration. Weights of well classified observations decrease relatively to weights of misclassified observations. Models that perform better have higher weights in the final ensemble model.

Jupyter Notebook Demo