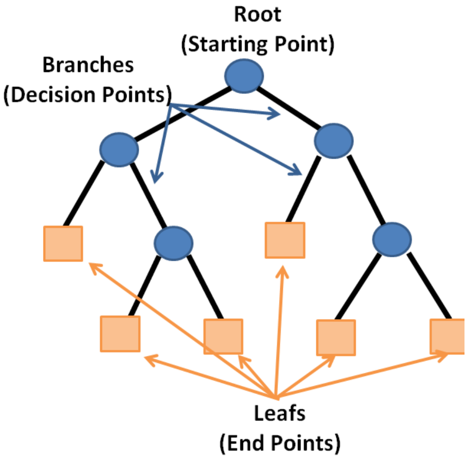
DECISION TREES

GENERAL INFO

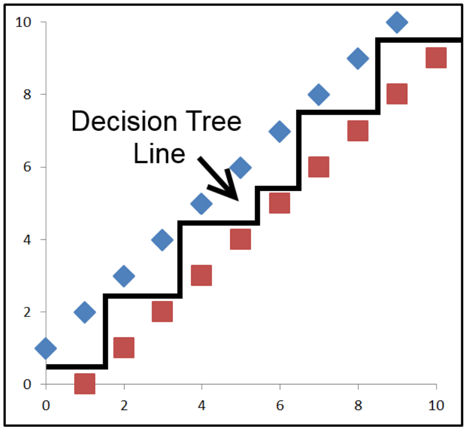
- Decision trees are a type of supervised machine learning
- They use known training data to create a process that predicts the results of that data. That process can then be used to predict the results for unknown data
- A decision tree processes data into groups based on the value of the data and the features it is provided
- At each decision gate, the data gets split into two separate branches, which might be split again
- The final grouping of the data is called Leaf Nodes



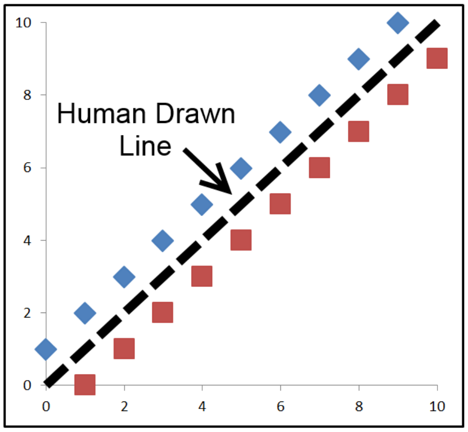
- Decision trees can be used for regression to get a real numeric value. Or they can be used for classification to split data into different distinct categories.

PICKING THE SPLITS

- Decision trees look through every possible split and pick the best split between two adjacent points in a given feature
- If the feature is in the real range, the split occurs halfway between the two points
- The split are always selected on a single feature only, not any interaction between multiple features
- That means if there is a relationship between multiple features, for instance, if density is the critical feature, but the data is provided in volume and mass, the decision tree will split the data as a series of splits on single features



- It will not do what a human might do, shown below



Time Complexity

- For a single decision tree, the most expensive operation is picking the best splits
- Picking the best split requires evaluating every possible split
- Finding every possible split requires sorting all the data on every feature
- That means for M features and N points, this takes M * N lg(N) time
- For machine learning algorithms that use multiple decision trees, those sorts can be done a single time and cached

Overfitting

- Over fitting, which means too closely matching the training data at the expense of the test data, is a key concern for decision trees
- Different stopping criteria should be evaluated with cross-validation to mitigate over fitting

STOPPING CRITERIA

- Additional splits can occur until a stopping criteria is reached
- Common stopping criteria are
- Maximum Depth: the maximum number of splits in a row has been reached

$$leafs = 2^{depth}$$

- Maximum Leafs: don't allow any more leafs
- Min Samples Split: Only split a node if it has at least this many samples
- Min Samples Leaf: Only split a node if both children resulting have at least this many samples
- Min Weight Fraction: Only split a node if it has at least this percentage of the total samples

RANDOMIZATION

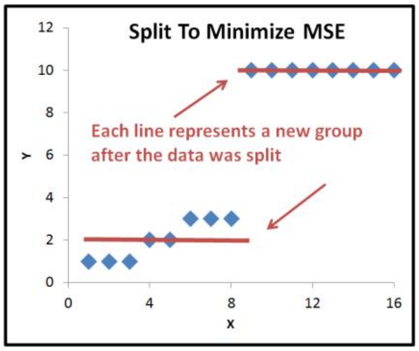
- When a single decision tree is run, it usually looks at every point in the available data.
- However some algorithms combine multiple decision trees to capture their benefits while mitigating over fitting
- Two widely used algorithms that use multiple decision trees are Random Forests and Gradient Boosted Trees
- Random Forests use a large number of slightly different decision trees run in parallel and averaged to get a final result.
- Gradient Boosted Trees use decision trees run in series, with later trees correcting errors on previous trees.
- If multiple trees are combined, it can be advantageous to have a random element in the creation of the trees, which can mitigate over fitting
- **Bagging** – (short for **B**ootstrap **agg**regat**ing**) is drawing samples from the data set with replacement.
- If you had 100 data points, you would randomly draw 100 points and on average get 63.2% unique points and 36.8% repeats

Visit www.FairlyNerdy.com for more statistics and machine learning information

- **Sub-Sampling -** is drawing a smaller set than your data set. For instance, if you have 100 points, randomly draw 60 of them (typically without replacement)
- Random Forests tend to use bagging, Gradient Boosted Trees tend to use sub-sampling.
- Additionally the features the tree splits on can be randomized
- Instead of evaluating every feature for the optimum split, a subset of the features can be evaluated (frequenlty the square root of the number of features are evaluated)

REGRESSION DECISION TREES

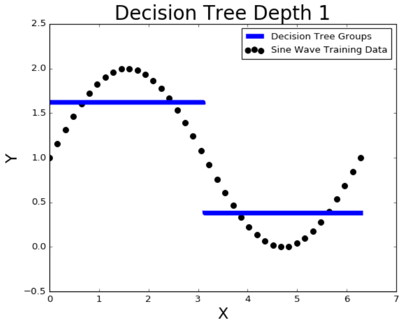
- Regression trees default to splitting data based on mean squared error (MSE)
- To calculate MSE: take the average value of all points in each group, and for each point in that group subtract the average value from the true value, square the error, sum all of the squares and take the average.
- The chart below shows data that was split in order to generate two new groups that had the lowest possible MSE



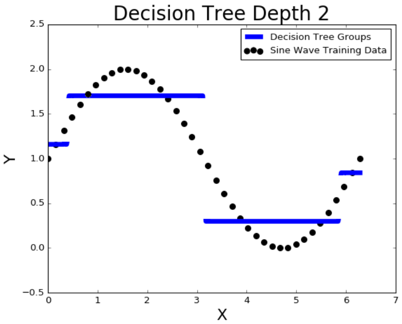
- MSE by default tends to focus on points with the highest error and split them into their own groups, since their error has a squared effect.
- Mean Average Error (MAE) is also an option for picking splits instead of MSE.
- To calculate MAE: take the median (not mean) of all points in the group, and for each point subtract the median value from the true value, take the absolute value of that error, sum the results and take the average

EXAMPLE REGRESSION

- This is an example of a regression tree attempting to match a sine wave using a single split



- Below is an example of a depth 2 regression tree, which gets 3 splits (i.e. 4 final groups)



INFORMATION GAIN

- Decision trees pick their splits in order to maximize "Information Gain"
- "Information" is a metric which tells you how much error you probably have in your decision tree
- Information gain is how much less error you have after the split than before the split

Regression Tree Information Gain

- Regression trees use either MSE or MAE as their metric
- The information gain is the MSE / MAE before the split minus the MSE / MAE after the split (weighted by the number of data points that split operated on)
- The best MSE / MAE is zero

Classification Information Gain

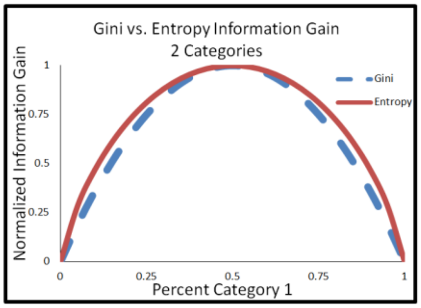
- Classification trees measure their information gain either by "Entropy" or "Gini Impurity"
- The best result for those values is also zero, and the worst result occurs when every category has an equal likelihood of being at any given node
- Gini equation

$$Gini = 1 - \sum_j p_j^2$$

- Where p is the probability of having a given data class in your data set.
- Entropy Equation

$$Entropy = \sum_j -p_j * log_2(p_j)$$

- Those results end up being fairly similar



FEATURE IMPORTANCES

- Decision trees lend themselves to identifying which features were the most important
- This is by calculating which feature resulted in the most information gain (weighted by number of data points), across all of the splits
- The information gain can be MSE / MAE for regression trees or Entropy / Gini for classification trees