We create a new categorical variable, y, which represents two or more ranges of value that Gic is in. We first use 5 as the threshold and have y=”High” whenever Gic>=5 and y=”Low” otherwise. The problem becomes a classification problem of Gic versus 5 DoE parameters.

**1. Decision tree (classification)**

Decision tree learning uses a decision tree to go from observations about an item to conclusions about the item's target value. Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Gini impurity. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. To compute Gini impurity for a set of items with J classes, suppose , and let be the fraction of of items labeled with class in the set:

Tree advantages and limitations. A decision tree is simple to understand and interpret, especially graphically, and it mirrors human decision making more closely than other approaches. It requires little data preparation. It is a parametric method and the explanation for the condition is easily explained by Boolean logic. It also performs well with large datasets. Large amounts of data can be analyzed using standard computing resources in reasonable time. On the other hand, trees has its limitations. Trees can be very non-robust. A small change in the training data can result in a big change in the tree, and thus a big change in final predictions. Decision-tree learners can create over-complex trees that do not generalize well from the training data.

**2. Random forest**

Trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.

This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x1, ..., xn with responses Y = y1, ..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B:

1. Sample, with replacement, B training examples from X, Y; call these Xb, Yb.
2. Train a decision or regression tree fb on Xb, Yb.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

or by taking the majority vote in the case of decision trees.

The number of samples/trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees B can be found using cross-validation, or by observing the out-of-bag error: the mean prediction error on each training sample xᵢ, using only the trees that did not have xᵢ in their bootstrap sample.

Variable importance. Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way. To measure the importance of the j-th feature after training, the values of the j-th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set. The importance score for the j-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences.