STATISTICAL HORIZONS

Advanced Machine Learning

Ross Jacobucci, Ph.D.

Upcoming Seminar:

October 7-9, 2021, Remote Seminar

Random Forests

Ross Jacobucci
University of Notre Dame

Downfalls to Decision Trees

We've previously discussed some of the drawbacks to decision trees. Mainly,

- Poor predictive performance
- Instability in structure
- Bias in variable selection (CART)

In this presentation, we are going to discuss one of the main algorithms proposed to overcome these problems – Random forests. Prior to discussing RF, we will go over bagging, a precursor.

Bagging

Bagging Overview

One of the first methods proposed to overcome the limitations of decision trees is bootstrap aggregating, or bagging (Breiman, 1996).

The algorithm proceeds in two steps. First, a bootstrap sample is taken. Then for each bootstrap sample, fit a tree model using all predictors. Given that each bootstrap sample is different, each tree will tend to have a unique structure as well, capitalizing in the instability of decision trees, manifested in either the variables used, splitting values, or tree size.

Bagging Predictions

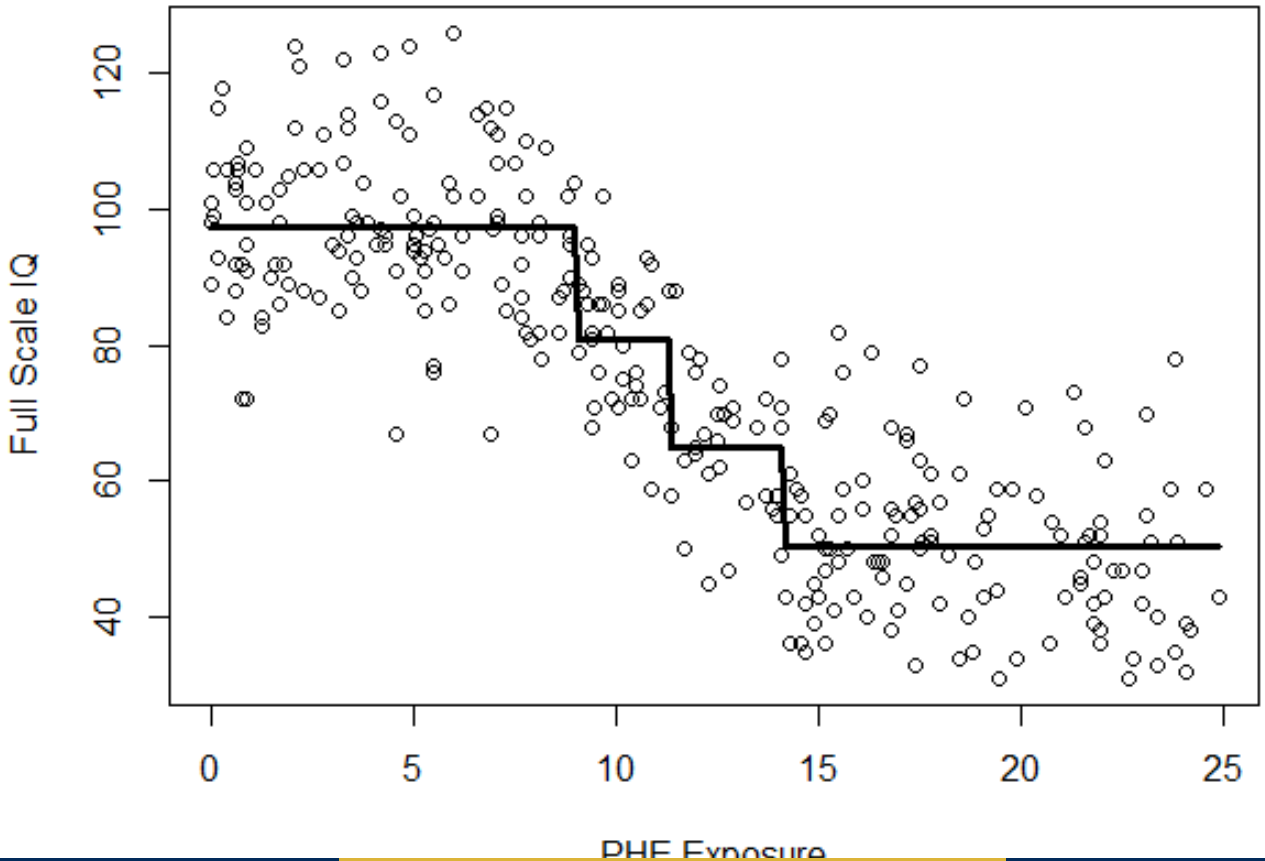
$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x) \tag{1}$$

where our predictions $\hat{f}_{bag}(x)$ are taken as the average predictions across each tree $\hat{f}^{*b}(x)$ for B number of trees created, with B being approximately 500. In the case of a categorical outcome, we can use majority vote across the trees.

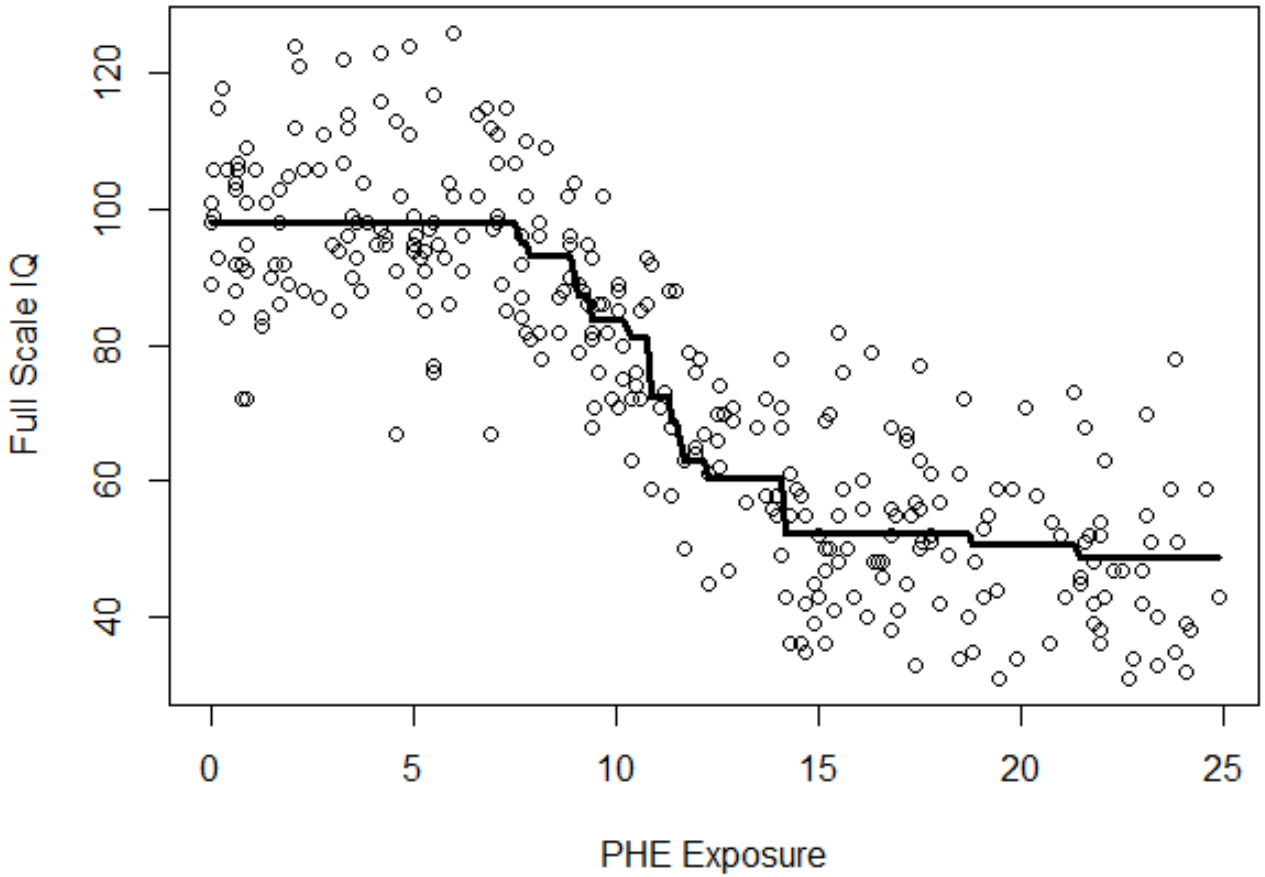
By averaging the prediction across 100's of trees, thereby reducing the variance.

Example

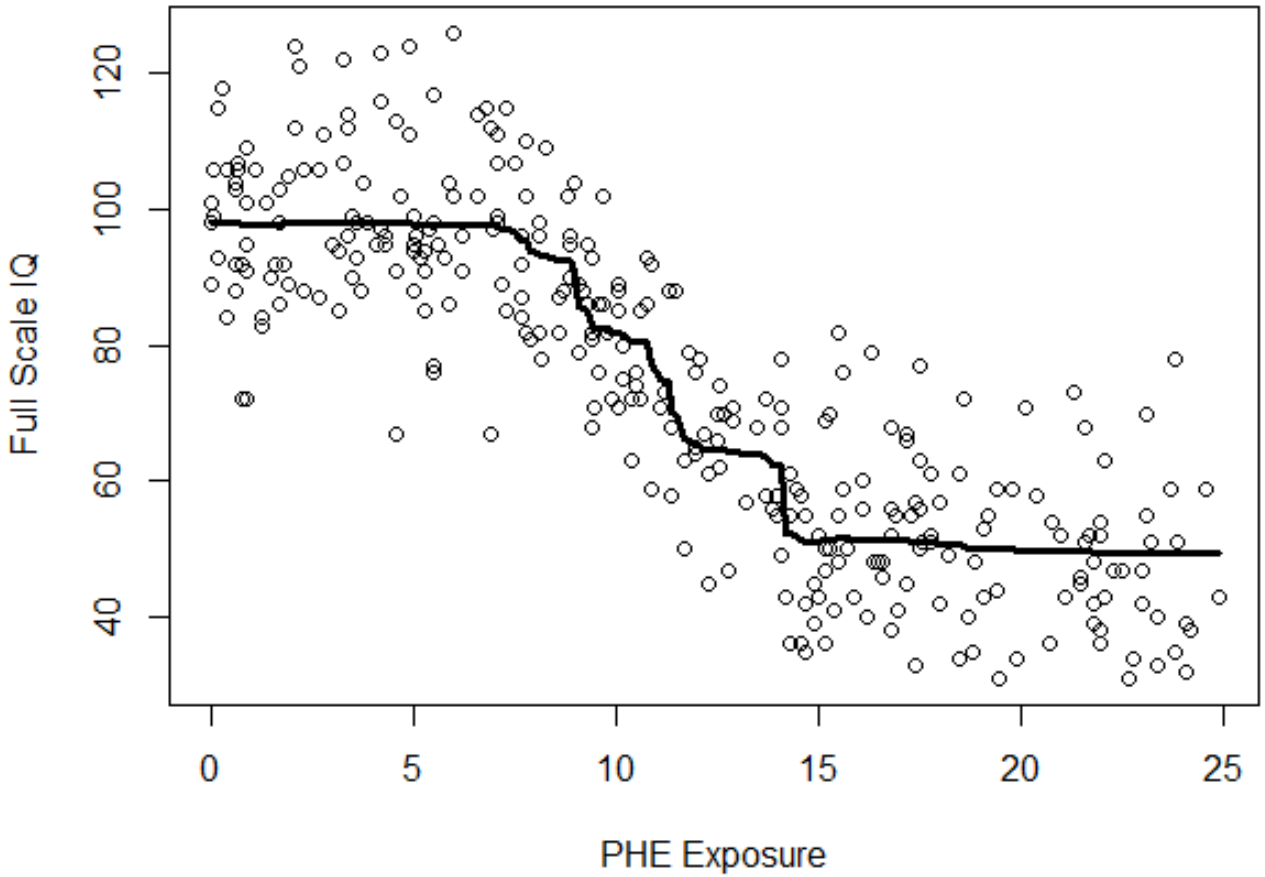
Using the PHE Exposure data, we start with a single tree.



5 Bagged Trees

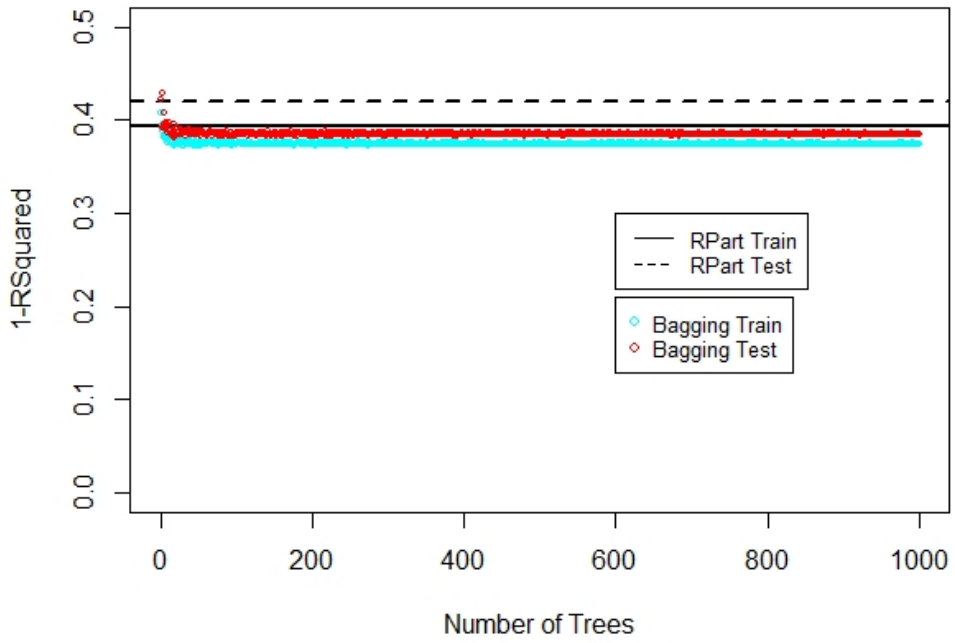


2000 Bagged Trees



Testing Different Numbers of Trees

Using the same example, we can see how our predictive performance changes across an increasing number of trees.



Drawbacks to Bagging

When using the same set of predictors to create each of the trees, many of the trees will be overly similar (or the same), resulting in a limited search of the prediction space. Bagging reduces the variance of individual trees, but the bias is approximately the same since the same trees are used. Put more concisely, the trees from bagging typically have high correlations.

Random forests was proposed to keep the low variance of bagging, but further reduce the bias by de-correlating the trees. This is achieved by randomly selecting a subset of the variables for the creation of each tree.

Random Forests

Algorithm

for b=1 to B **do**

1. Draw a bootstrap sample of size N from training data
2. Grow a decision tree T_b on the bootstrap sample using m variables

end

Result: Output the ensemble of trees $\{T_b\}_1^B$

To create predictions:

$$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{2}$$

Note that some versions of random forests use subsamples to create trees, not bootstrap samples.

mtry

`mtry`, or the number of randomly selected variables for creating each tree, is the main tuning parameter in random forests. You can vary the number of trees, but it typically doesn't matter for prediction (different for variable importance).

Default values in most programs:

- Classification is $\sqrt{p}$
- Regression is $p/3$

Out of Bag Samples

When using bootstrap samples, can use of the out of bag (OOB) samples to create samples. Then, average predictions for each observation just in those trees in which they were used (e.g., (1-.632)*B number of trees).

Using OOB prediction performance approximates the use of k-fold pretty well. However, it is more common to just use k-fold.

Compare to Bagging