What is an Introduction to Bagging in Machine Learning?

Authored by

stats writer

April 22, 2024

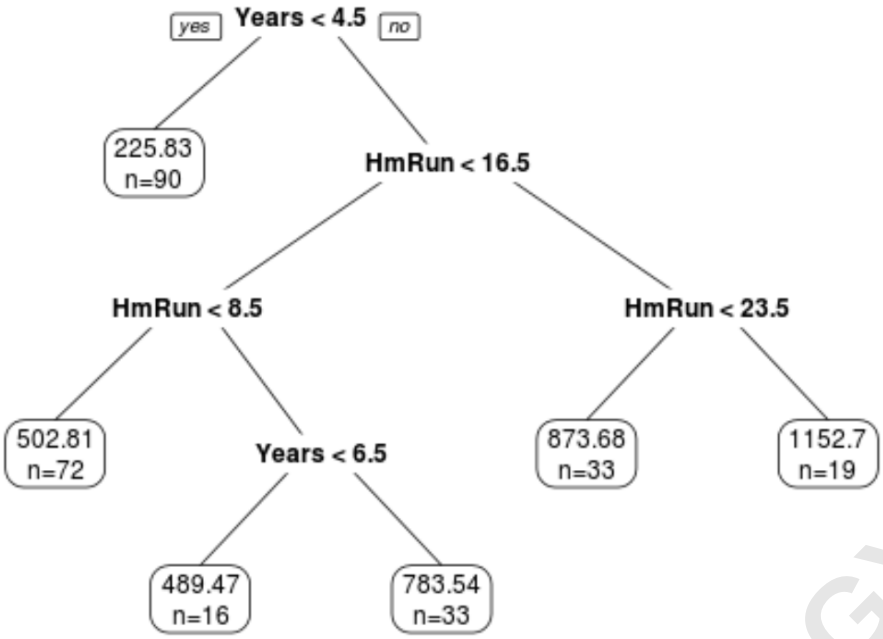
An Introduction to Bagging in Machine Learning is a method used to improve the accuracy and stability of predictive models. It involves creating multiple models using subsets of the training data and then combining them through a voting process. This approach helps to reduce the effects of overfitting and increases the generalizability of the model. Bagging can be applied to a variety of machine learning algorithms, making it a versatile and widely used technique in the field of data science.

An Introduction to Bagging in Machine Learning

When the relationship between a set of predictor variables and a response variable is linear, we can use methods like multiple linear regression to model the relationship between the variables.

However, when the relationship is more complex then we often need to rely on non-linear methods.

One such method is classification and regression trees (often abbreviated CART), which use a set of predictor variables to build *decision trees* that predict the value of a response variable.



Example of a regression tree that uses years of experience and average home runs to predict the salary of a professional baseball player.

However, the downside of CART models is that they tend to suffer from high variance. That is, if we split a dataset into two halves and apply a decision tree to both halves, the results could be quite different.

One method that we can use to reduce the variance of CART models is known as bagging, sometimes referred to as *bootstrap aggregating*.

What is Bagging?

When we create a single decision tree, we only use one training dataset to build the model.

However, bagging uses the following method:

1. Take *b* bootstrapped samples from the original dataset.

Recall that a *bootstrapped sample* is a sample of the original dataset in which the observations are taken with replacement.

2. Build a decision tree for each bootstrapped sample.

3. Average the predictions of each tree to come up with a final model.

For regression trees, we take the average of the prediction made by the *B* trees.For classification trees, we take the most commonly occurring prediction made by the *B* trees.

Bagging can be used with any machine learning algorithm, but it's particularly useful for decision trees because they inherently have high variance and bagging is able to dramatically reduce the variance,

which leads to lower test error.

To apply bagging to decision trees, we grow *B* individual trees deeply without pruning them. This results in individual trees that have high variance, but low bias. Then when we take the average predictions from these trees we're able to reduce the variance.

In practice, optimal performance typically occurs with 50 to 500 trees, but it's possible to fit thousands of trees to produce a final model.

Out-of-Bag Error Estimation

It turns out that we can calculate the test error of a bagged model without relying on **k-fold cross-validation**.

The reason is because it can be shown that each bootstrapped sample contains about 2/3 of the observations from the original dataset. The remaining 1/3 of the observations not used to fit the bagged tree are referred to as out-of-bag (OOB) observations.

We can predict the value for the ith observation in the original dataset by taking the average prediction from

each of the trees in which that observation was OOB.

We can use this approach to make a prediction for all *n* observations in the original dataset and thus calculate an error rate, which is a valid estimate of the test error.

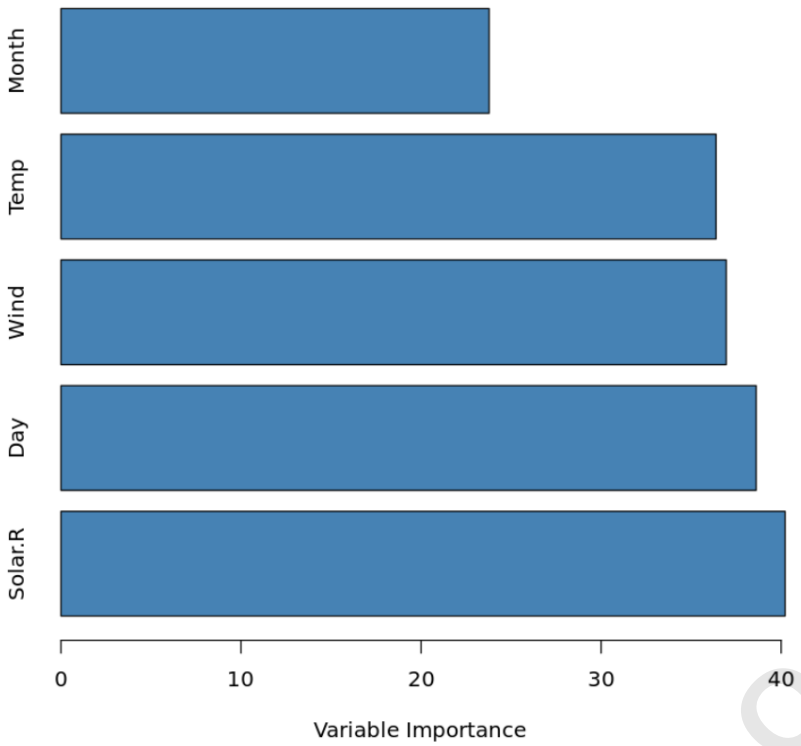
The benefit of using this approach to estimate the test error is that it's much quicker than k-fold cross-validation, especially when the dataset is large.

Understanding the Importance of Predictors

Recall that one of the benefits of decision trees is that they're easy to interpret and visualize.

When we instead use bagging, we're no longer able to interpret or visualize an individual tree since the final bagged model is the resulting of averaging many different trees. We gain prediction accuracy at the expense of interpretability.

However, we can still understand the importance of each predictor variable by calculating the total reduction in RSS (residual sum of squares) due to the split over a given predictor, averaged over all *B* trees. The larger the value, the more important the predictor.



Example of a variable importance plot.

Similarly, for classification models we can calculate the total reduction in the Gini Index due to the split over a given predictor, averaged over all *B* trees. The larger the value, the more important the predictor.

So, although we can't exactly interpret a final bagged model we can still get an idea of how important each predictor variable is when predicting the response.

Going Beyond Bagging

The benefit of bagging is that it typically offers an

improvement in test error rate compared to a single decision tree.

The downside is that the predictions from the collection of bagged trees can be highly correlated if there happens to be a very strong predictor in the dataset.

In this case, most or all of the bagged trees will use this predictor for the first split, which will result in trees that are similar to each other and have highly correlated predictions.

One way to get around this issue is to instead use random forests, which use a similar method as bagging but are able to produce decorrelated trees, which often leads to lower test error rates.

You can read a simple introduction to random forests **here**.

An Introduction to Classification and Regression Trees
How to Perform Bagging in R (Step-by-Step)