Bagging: How it works and why to use it

As you know, ensembles of base learners can combine to become powerful predictors. You learned about bagging, and that it's one of the more commonly used modeling strategies. In this reading, you'll learn not only what this technique is and how it works, but also why it can be beneficial.

A review of bagging

Bagging stands for **bootstrap aggregating**, but knowing this doesn't exactly clarify much, does it? Let's review by unpacking these terms.

Bootstrapping

Recall that **bootstrapping** refers to sampling with replacement. In ensemble modeling architectures, this means that for each base learner, the same observation can and will be sampled multiple times. Suppose you have a dataset of 1,000 observations, and you bootstrap sample it to generate a new dataset of 1,000 observations, on average, you should find about 632 of those observations in your sampled dataset (~63.2%).

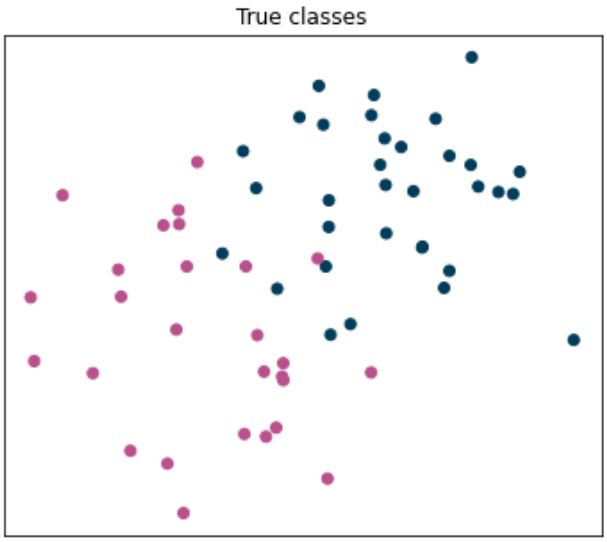
Aggregating

Building a single model with bootstrapped data probably wouldn't be very useful. To use the example above, if you start with 1,000 unique observations and use bootstrapping to create a sampled dataset of 1,000 observations, you'd only expect to get an average of 632 unique observations in that new dataset. This means that you'd lose whatever information was contained in the 368 observations that didn't make it into the new sampled dataset.

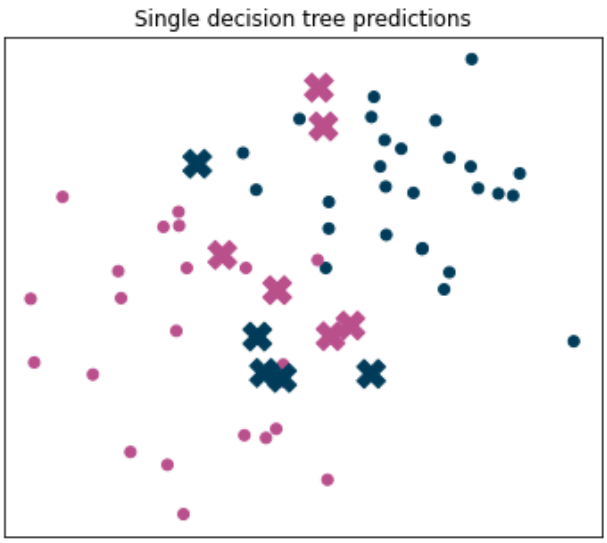
This is when ensemble learning—or ensembling—comes to the rescue. **Ensemble learning** refers to building multiple models and aggregating their predictions. Sure, those 368 observations might not make it into that particular sampled dataset, but if you keep repeating the bootstrapping process —once for each base learner—eventually your overall ensemble of base learners will see all of the observations.

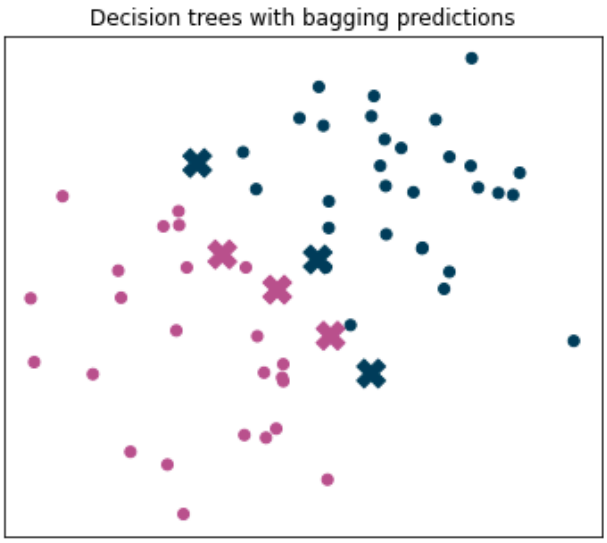
Example: bagging vs. single decision tree

Here is some test data taken from a dataset containing two classes:



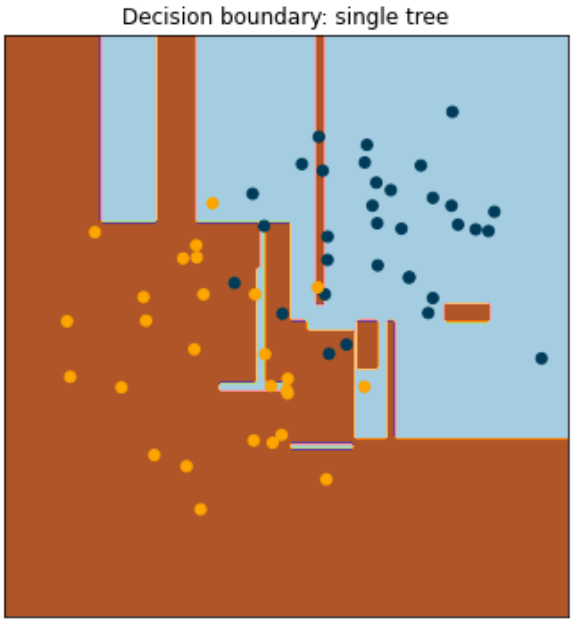
And here is a comparison of the predictions on this test data made by a single decision tree versus the predictions made by an ensemble of 50 decision trees using bagging:

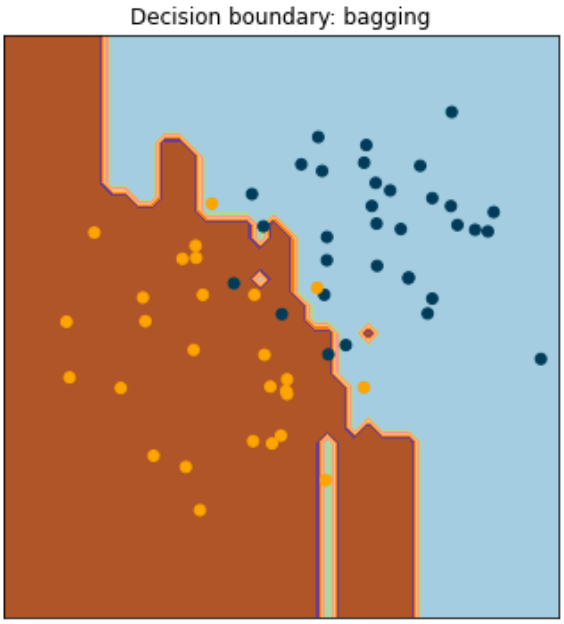




The Xs indicate incorrect predictions. Notice that the single decision tree got 11 predictions wrong out of 60—an accuracy score of 81.7%. Meanwhile, the ensemble of decision trees with bagging only got 6 wrong. Bagging resulted in a 10% improvement in accuracy!

Another way to examine the results of these models is to plot their decision boundaries:





You may notice that the decision boundary for a tree-based model isn't linear like those of logistic regression or Naive Bayes models. This is illustrative of decision trees' tendency to fit—and overfit—the data. The single decision tree has many more "decision islands," or areas where one class is surrounded by the other class. Even slightly different data would likely result in a very different decision boundary plot, which is indicative of greater variance. Because

bagging aggregates the predictions of many different trees, its resulting decision boundary is more stable, because the model has lower variance.

Why to use it

- **Reduces variance:** Standalone models can result in high variance. Aggregating base models' predictions in an ensemble help reduce it.
- **Fast:** Training can happen in parallel across CPU cores and even across different servers.
- **Good for big data:** Bagging doesn't require an entire training dataset to be stored in memory during model training. You can set the sample size for each bootstrap to a fraction of the overall data, train a base learner, and string these base learners together without ever reading in the entire dataset all at once.