Learning Bagging: An Ensemble Method for Machine Learning

Authored by

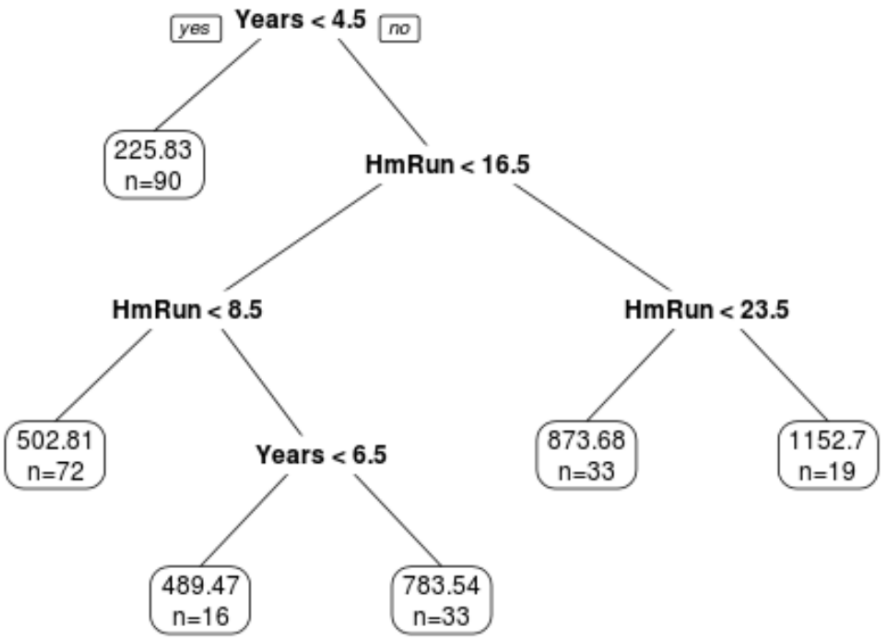
Mohammed looti

November 6, 2025

In the realm of machine learning, the goal is often to model the relationship between a set of predictor features and a response variable. When this underlying relationship exhibits a straightforward linear structure, established statistical methodologies like multiple linear regression prove highly effective and interpretable. These methods rely on well-understood assumptions about data distribution and error terms, providing robust predictions under ideal circumstances.

However, complexity is the rule, not the exception, in real-world datasets. Data often contains intricate, non-linear interactions and high-dimensional structures that simple linear models fail to capture adequately. This necessitates the deployment of more flexible, powerful algorithms capable of learning complex decision boundaries. The move toward non-linear modeling is essential for achieving high predictive performance in diverse fields ranging from computational biology to financial forecasting.

One of the most widely adopted non-linear techniques is the family of tree-based methods, specifically classification and regression trees (CART). CART models operate by recursively partitioning the feature space into distinct regions, forming a hierarchical structure known as a **decision tree**. Each node in the tree corresponds to a decision based on an input feature, ultimately leading to a prediction (a class label or a numerical value) at the leaf nodes.



Visualization of a regression tree structure. This example demonstrates how features like years of experience and average home runs are used to predict the salary of a professional athlete.

Despite the intuitive appeal and strong predictive potential of individual decision trees, they suffer from a significant statistical drawback: their inherent tendency toward high variance. A model with

high variance is overly sensitive to minor fluctuations in the training data. If we were to slightly modify the training set--perhaps by removing or adding a few samples--the resulting tree structure could change drastically, leading to highly divergent and unstable predictions. This instability makes single trees unreliable for robust deployment in production environments.

This high variance arises because decision trees, especially those grown deep without pruning, are designed to minimize bias by perfectly fitting the training data. However, this aggressive fitting often leads to overfitting, where the model captures noise rather than the underlying signal. To maintain the low bias afforded by deep trees while simultaneously mitigating the associated high variance, we turn to powerful ensemble methods. The most foundational of these techniques, designed specifically to counteract this instability and reduce variance, is **bagging**, which stands as an acronym for *bootstrap aggregating*.

What is Bagging (Bootstrap Aggregating)?

The core limitation of relying on any single machine learning model--be it a decision tree, neural network, or linear classifier--is that its performance is intrinsically tied to the specific training dataset it was exposed to. Bagging provides an elegant solution by shifting from a single model paradigm to an ensemble approach. By generating multiple, slightly different versions of the training data and building an independent model on each, bagging effectively diversifies the learning process, which stabilizes the final aggregate prediction.

While bagging is a general-purpose ensemble technique applicable across various algorithms, its impact is most pronounced when applied to base learners characterized by low bias and high variance, such as deep, unpruned decision trees. The inherent variance of the individual trees is the key target for reduction. By combining the predictions of many independently trained, high-variance models, the averaging process acts as a powerful dampener, leading to a significant reduction in the overall test error compared to any single constituent model.

To maximize the effectiveness of variance reduction through bagging, machine learning practitioners typically employ a specific strategy when growing the constituent decision trees. These individual trees are intentionally grown to their full depth, often without any pruning. This deliberate choice ensures that each base learner maintains high variance (fitting the data aggressively) and low bias. The effectiveness of the ensemble emerges not from the perfection of any single tree, but from the principle of aggregation, where the errors and variations of the individual high-variance predictions cancel one another out when averaged together.

The success of **bagging** relies fundamentally on the creation of independent training sets through the statistical technique known as bootstrapping. Bootstrapping is a resampling method where subsets of the data are drawn with replacement. This crucial step guarantees that each base model is trained on a unique perspective of the data, ensuring the necessary diversity among the

ensemble members.

The Bagging Procedure: Step-by-Step

The process of **bootstrap aggregating** (bagging) is systematic, requiring the generation of multiple synthetic training sets and the subsequent combination of the resulting models. This formalized approach ensures that the resulting ensemble minimizes variance effectively while preserving the low bias of the constituent learners.

The procedure requires the specification of *B*, the number of bootstrapped datasets and the corresponding number of base models (often decision trees) to be trained. A larger *B* generally leads to a more stable ensemble, although diminishing returns are eventually reached.

The steps for implementing bagging are as follows:

Bootstrap Sampling: Generate *B* independent bootstrapped samples, $D_1, D_2, dots, D_B$, from the original training dataset D.

Each bootstrapped sample is created by randomly drawing observations from the original dataset **with replacement**. This means that a specific observation might be selected multiple times within a single sample, while approximately one-third of the original observations are typically excluded entirely, forming the out-of-bag sample.

Parallel Model Training: Build an independent base learner, f_b, for each of the *B* bootstrapped samples. If using decision trees, these trees ($T_1, T_2, dots, T_B$) are typically grown deep and unpruned to ensure low bias.
Prediction Aggregation: Combine the predictions from all *B* base models to form a single, robust final prediction, $F(x)$.

For **regression problems**, the final prediction is calculated as the simple average of the predictions made by all *B* models: $F(x) = frac{1}{B} sum_{b=1}^{B} f_b(x)$.
For **classification problems**, the final prediction is determined by a majority vote across all *B* models. The class chosen most frequently by the ensemble members is designated as the final predicted class.

In practice, the number of trees (B) chosen is a hyperparameter often ranging from 50 to 500, though modern computation allows for fitting thousands of trees. Increasing *B* generally reduces the variance monotonically, meaning the ensemble performance stabilizes and rarely degrades once a sufficient number of models has been aggregated, making the selection of *B* less critical than other hyperparameters in traditional model tuning.

Out-of-Bag Error Estimation

One of the most valuable computational efficiencies offered by the bagging procedure is its inherent ability to provide an unbiased estimate of the generalization error--often referred to as the **out-of-bag (OOB) error**--without requiring external validation sets or computationally expensive methods like k-fold cross-validation. This feature significantly streamlines the model evaluation process.

As a statistical consequence of sampling with replacement, when a large training dataset of size N is bootstrapped, the probability that any specific observation is included in a given bootstrapped sample is $1 - (1 - 1/N)^N$. As N approaches infinity, this probability converges to $1 - 1/e$, which is approximately $63.2%$. Consequently, approximately one-third (about 36.8%) of the original observations are excluded from any specific bootstrapped sample. These excluded observations are defined as the **out-of-bag (OOB) observations** for that particular model.

The OOB observations act as a built-in test set for the trees that did not use them during training. To calculate the OOB prediction for any given observation x_i from the original dataset, we restrict the calculation to only those trees for which x_i was an OOB observation. We then average the predictions from this subset of trees. Since x_i was not used to train this subset of models, the resulting prediction is unbiased relative to the training process.

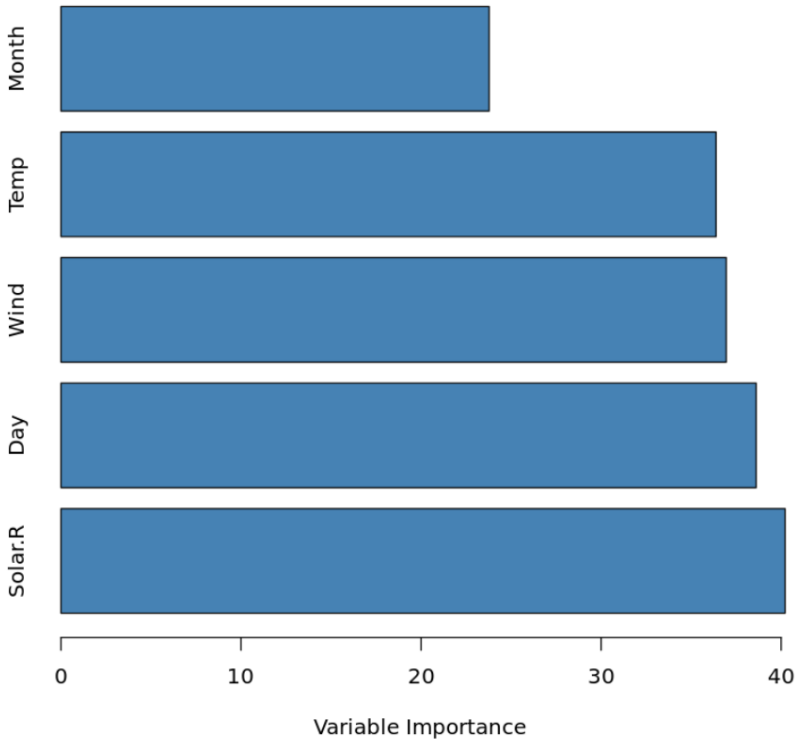
By repeating this process for all N observations in the original dataset, we can compute an overall error rate (e.g., Mean Squared Error for regression or misclassification rate for classification). This calculated OOB error provides a robust and reliable estimate of the true test error, offering a significant computational advantage, especially when dealing with massive datasets where repeated cross-validation folds would incur substantial overhead.

Understanding Predictor Importance

One compromise inherent in moving from a single, interpretable decision tree to a complex bagged ensemble is the loss of straightforward visualization and interpretability. A single tree offers clear insight into the decision path. In contrast, the final bagged model is an opaque average of potentially hundreds or thousands of distinct trees. The gain in predictive accuracy achieved through variance reduction often comes at the cost of model transparency.

Despite this loss of visual clarity, it remains critically important to understand which predictor variables contribute most significantly to the final prediction. Bagging allows us to derive a quantitative measure of variable importance by aggregating the performance metrics across all constituent trees. For regression tasks, the importance of a predictor is typically quantified by calculating the total decrease in the residual sum of squares (RSS) achieved by splits based on that specific feature.

This total RSS reduction is calculated for the feature across every tree in the ensemble and then averaged. A predictor that consistently results in a large average reduction in RSS is considered highly influential in determining the response variable. This averaged metric provides a standardized and reliable ranking of feature relevance, allowing analysts to understand the relative contribution of input variables even within the complex ensemble framework.



An example of a variable importance plot generated from a bagged model, showing the relative contribution of each input feature.

For classification models, the methodology is analogous, but the impurity metric changes. Importance is measured by calculating the total reduction in impurity, using metrics like the Gini Index or entropy, averaged across all *B* trees. A greater average reduction in impurity signifies that a predictor is essential for making accurate class distinctions. Therefore, while the decision path itself is obscured, bagging still yields a powerful, quantitative understanding of feature relevance.

Going Beyond Bagging: Introducing Random Forests

While bagging provides a significant and reliable improvement over the performance of a single decision tree, particularly in reducing variance, it is not the final word in tree-based ensemble methods. Standard bagging techniques suffer from a critical limitation that prevents them from achieving the absolute maximum potential for variance reduction in many datasets.

The primary drawback emerges when the dataset contains one or a few exceptionally strong, dominant predictor variables. In such cases, because bagging relies on resampling the observations (rows) but uses all predictors (columns), nearly every bootstrapped sample will contain this dominant feature. Consequently, almost all individual trees in the ensemble will select this same feature for the first few splits, leading to highly similar tree structures and, critically, highly correlated predictions.

When the predictions from the base models are highly correlated, averaging them yields only a modest reduction in variance. The strength of ensemble methods lies in averaging independent or weakly correlated predictors; strong correlation limits the ability of the errors to cancel each other out. To overcome this limitation and ensure greater independence among the base learners, the methodology is extended to random forests.

Random forests introduce an additional, crucial layer of randomness: at every split in the tree-building process, only a random subset of the available features is considered. This intentional process of feature subsetting ensures that the dominant predictor is not always selected, forcing different trees to rely on different features. This systematic decorrelation often results in a significantly lower test error rate compared to standard bagging, cementing random forests as one of the most powerful and widely used algorithms in applied machine learning.

Additional Resources for Machine Learning

To deepen your understanding of these powerful tree-based methodologies, consider exploring the following resources:

A Comprehensive Introduction to Classification and Regression Trees
Practical Guide: How to Perform Bagging in R (Step-by-Step Tutorial)
Detailed Guide to Implementing Random Forests

Understanding bagging provides a fundamental basis for comprehending more advanced ensemble techniques and is a prerequisite for mastering modern predictive modeling.