What is Bagging in Machine Learning

Authored by

stats writer

December 18, 2025

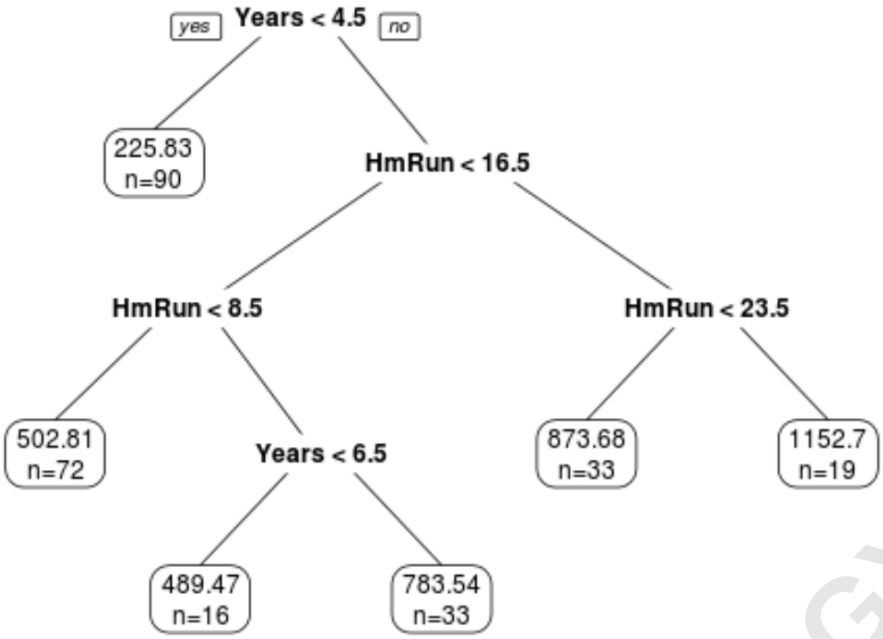
Ensemble learning is a powerful paradigm within **Machine Learning** that focuses on combining the predictions of multiple individual models, often referred to as base estimators, to achieve superior predictive performance compared to any single model alone. One of the foundational and most widely adopted ensemble methods is **Bagging**. This technique, short for Bootstrap Aggregating, is specifically designed to mitigate the inherent instability and high variance often associated with complex, non-linear models. By training diverse instances of the same base learning algorithm on various subsets of the training data, Bagging ensures that the combined aggregated output is both robust and significantly more accurate.

The fundamental goal of Bagging is to enhance the generalization capabilities of an unstable model--a model where minor changes in the training data lead to significant changes in the resulting structure or predictions. By generating many slightly different versions of the model and averaging their results, the noise introduced by the variance is effectively canceled out. This process creates a collective predictor that is far less prone to overfitting the training data, ultimately yielding a lower test error rate. While Bagging can be applied to nearly any machine learning algorithm, its utility is most pronounced when applied to high-variance models, such as deep Decision Trees.

When statistical modeling involves a clear relationship between a set of independent, or **predictor variables**, and a dependent **response variable**, straightforward methods like multiple linear regression often suffice. These linear methods provide interpretability and efficiency when the underlying data structure is simple and additive.

However, real-world data often exhibits highly complex, non-linear interactions and thresholds that cannot be adequately captured by linear models. In these scenarios, researchers and data scientists must turn to more flexible, non-linear techniques capable of modeling intricate relationships without making rigid assumptions about the data distribution.

One of the most popular and intuitive non-linear methods is the use of Classification and Regression Trees (CART). CART algorithms partition the predictor space into a set of distinct, non-overlapping regions. For any given input, the prediction is determined by the average or majority class within the region into which that input falls. The resulting structure, known as a **decision tree**, visually maps out the rules used to arrive at a prediction.



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

A significant drawback of using single, deep CART models is their inherent susceptibility to high variance. A decision tree built to full depth without pruning is highly sensitive to the specific observations in the training set. If the training data were slightly perturbed--for instance, by splitting the original dataset into two halves and training a tree on each--the resulting models and their terminal nodes could be drastically different. This instability means the model lacks reliability when generalizing to new, unseen data, often leading to poor performance despite achieving low training error.

This challenge of high variance is precisely where **Bagging** provides its primary benefit. By training multiple unstable but potentially highly accurate models and then aggregating their results, Bagging manages to smooth out the noisy predictions caused by localized fluctuations in the training data, thereby stabilizing the overall ensemble predictor.

The Core Mechanism of Bagging

When constructing a traditional machine learning model, such as a single decision tree, the entire training dataset is used to determine the optimal splits and structure. Bagging fundamentally deviates from this approach by introducing random sampling to ensure that the individual models within the ensemble are slightly different from one another. This difference, or decorrelation, among the base estimators is the key element that allows the aggregation step to reduce variance.

The entire process of **Bagging** is built upon two critical statistical concepts: bootstrapping and aggregation. Bootstrapping involves repeatedly drawing samples from a data set with replacement. Because we sample with replacement, each generated dataset (the bootstrap sample) has the same size as the original dataset but contains some duplicated observations and omits others. This ensures that each base learner is trained on a unique, though overlapping, subset of the data, guaranteeing diversity among the models.

The procedure results in the creation of B unique models, where B is the number of trees specified (typically 50 to 500, but potentially thousands). Because each model is trained independently on a bootstrapped dataset, the errors made by individual models are less correlated. When these uncorrelated errors are averaged together during the aggregation phase, they tend to cancel each other out, leading to a much lower overall ensemble error.

The Bootstrap Aggregating Procedure

The Bagging algorithm systematically executes a simple yet powerful series of steps to generate the ensemble model. This process ensures that a large collection of diverse, weak learners can be combined into a single, strong, and highly stable predictor.

The methodology relies on generating B independent training sets through bootstrapping, training a dedicated model on each set, and finally combining all resulting predictions.

The steps for implementing **Bagging** are as follows:

Generate Bootstrapped Samples: Take B independent bootstrapped samples from the original training dataset. A bootstrapped sample is created by randomly selecting observations from the original dataset with replacement, resulting in a dataset of the same size as the original.
Train Base Estimators: Build a high-variance base learning model (such as a fully grown decision tree) for each of the B bootstrapped samples. It is conventional when using Bagging with trees to allow them to grow deeply without pruning, ensuring individual models have low bias but high variance.
Aggregate Predictions: Combine the predictions from all B base estimators to produce the final aggregated prediction.

The aggregation step differs slightly depending on the nature of the task:

For **regression trees**, we take the simple arithmetic mean (average) of the predictions made by all B trees for a given input observation.
For **classification trees**, we use majority voting, where the final prediction is the class label that occurs most frequently among the B individual predictions.

This averaging or voting mechanism smooths out the highly variable decision boundaries of the

individual trees, yielding a much more stable and accurate overall prediction surface.

Addressing the Bias-Variance Tradeoff

The success of Bagging is best understood in the context of the **bias-variance tradeoff**. Any learning algorithm's total expected error can be decomposed into three main components: bias, variance, and irreducible error. Bias measures the systematic error due to overly simplistic assumptions in the model, while variance measures how much the model's prediction changes when trained on different subsets of the data.

Decision trees, particularly when grown deeply, are considered low-bias but high-variance models. They are highly flexible and capable of capturing complex relationships (low bias), but they are easily overfit to noise in the training data (high variance). When applying Bagging, we intentionally choose base models that exhibit high variance.

The beauty of Bagging lies in its ability to reduce variance dramatically without significantly increasing the bias. Since the base models (e.g., deep trees) already have low bias, the averaging process preserves this low bias while simultaneously dampening the high variability across the ensemble. The final bagged model inherits the low bias characteristic of the deep tree but achieves a significantly lower variance compared to any single constituent tree, leading directly to a lower overall test error.

Out-of-Bag (OOB) Error Estimation

One of the most valuable computational advantages of using Bagging is the inherent ability to estimate the generalization error internally, without needing to rely on computationally expensive methods like K-fold cross-validation or holding out a separate validation set. This unique feature stems directly from the bootstrapping procedure itself.

Statistical theory shows that when a bootstrap sample is drawn, approximately 63.2% of the original observations are included in the sample, meaning about 36.8% of the data points are left out. These excluded observations, which were not used to train a specific base estimator, are referred to as **Out-of-Bag (OOB) observations**. The OOB observations act as a natural, unbiased test set for the models they were not involved in training.

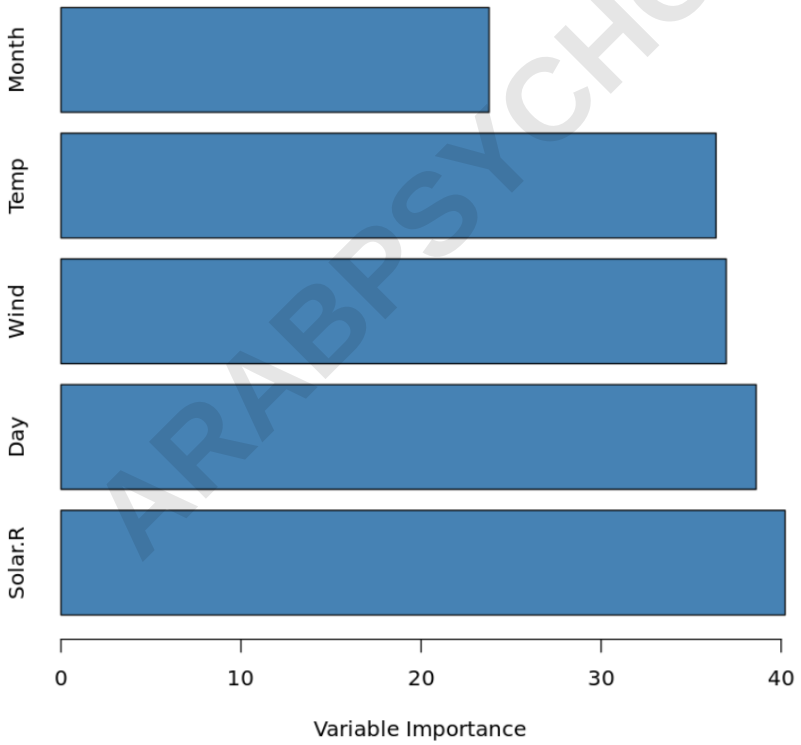
To calculate the **OOB error rate**, we predict the value for the i-th observation in the original dataset only using the subset of trees for which that i-th observation was OOB. We then average or majority-vote these predictions. By performing this for all n observations in the training set, we obtain an overall OOB prediction for the entire dataset. The error rate calculated using these OOB predictions provides a highly accurate and reliable estimate of the test error, often comparable to that obtained via cross-validation, but achieved with minimal additional computational cost.

Interpreting Bagged Models: Predictor Importance

A well-known advantage of a single decision tree is its excellent interpretability--we can easily visualize the structure and understand exactly how predictor variables lead to a classification or regression outcome. Unfortunately, when we move to **Bagging**, this interpretability is sacrificed. The final model is an average of hundreds of complex trees, making it impossible to visualize or easily trace a single path of logic. We gain predictive accuracy at the expense of model clarity.

However, this lack of global interpretability does not mean that all insights are lost. We can still quantify the relative contribution of each predictor variable to the final model performance. This is achieved by measuring how much each predictor contributes to the reduction of error across the entire ensemble.

For regression models, we calculate the total reduction in **Residual Sum of Squares (RSS)** due to splits based on a specific predictor, averaging this reduction across all B trees in the ensemble. Similarly, for classification models, we track the total reduction in a purity metric, often the **Gini Index** or cross-entropy, attributable to splits involving a specific predictor, again averaged across all trees. A larger averaged reduction value signifies a more important predictor.



Example of a variable importance plot.

While the internal mechanics of the ensemble remain opaque, the resulting variable importance

plot gives practitioners a clear, quantitative understanding of which features are most influential in driving the final prediction, allowing for informed feature engineering and domain insight extraction.

Limitations of Bagging and Introduction to Random Forests

The primary benefit of Bagging is its robust reduction in prediction variance, leading to a significant improvement in test error rates compared to using a single base estimator. However, Bagging is not without its limitations, particularly in datasets where one or a few predictor variables are overwhelmingly strong.

If a dataset contains an exceptionally powerful predictor, nearly every bootstrapped sample will retain this variable, and most of the resulting trees will choose this strong predictor for the first split near the root of the tree. This uniformity means that the collection of bagged trees will be highly similar to one another, resulting in **highly correlated predictions**. When predictions are highly correlated, the averaging process is less effective at reducing variance, as the noise components do not cancel out efficiently.

To overcome this challenge of correlation among base estimators, an evolution of Bagging was developed: the Random Forest algorithm. Random Forests employ the same bootstrapping methodology as Bagging, but introduce an additional randomization step: at each split in the construction of a decision tree, only a random subset of the total predictor variables is considered.

By restricting the feature set available at each split, Random Forests ensure that even strong predictors cannot dominate every tree. This forces the individual trees to be more diverse and less correlated, ultimately leading to a superior reduction in variance and often yielding even lower test error rates than standard Bagging, making it a powerful and frequently used technique in modern machine learning.

Further Reading and Resources

To delve deeper into related ensemble methods and modeling techniques, the following resources are recommended:

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