17.8.1

## Overview of Ensemble Learning

As you and Jill discuss ways to improve a model's performance, she brings up ensemble learning. Ensemble learning builds on the idea that two is better than one. A single tree may be prone to errors, but many of them can be combined to form a stronger model. A random forest model, for example, combines many decision trees into a forest of trees.

The concept of ensemble learning is the process of combining multiple models, like decision tree algorithms, to help improve the accuracy and robustness, as well as decrease variance of the model, and therefore increase the overall performance of the model.

In the example below, there are multiple algorithms that are being used to learn and make their predictions based on the data. The final prediction is based on the accumulated predictions from each algorithm:
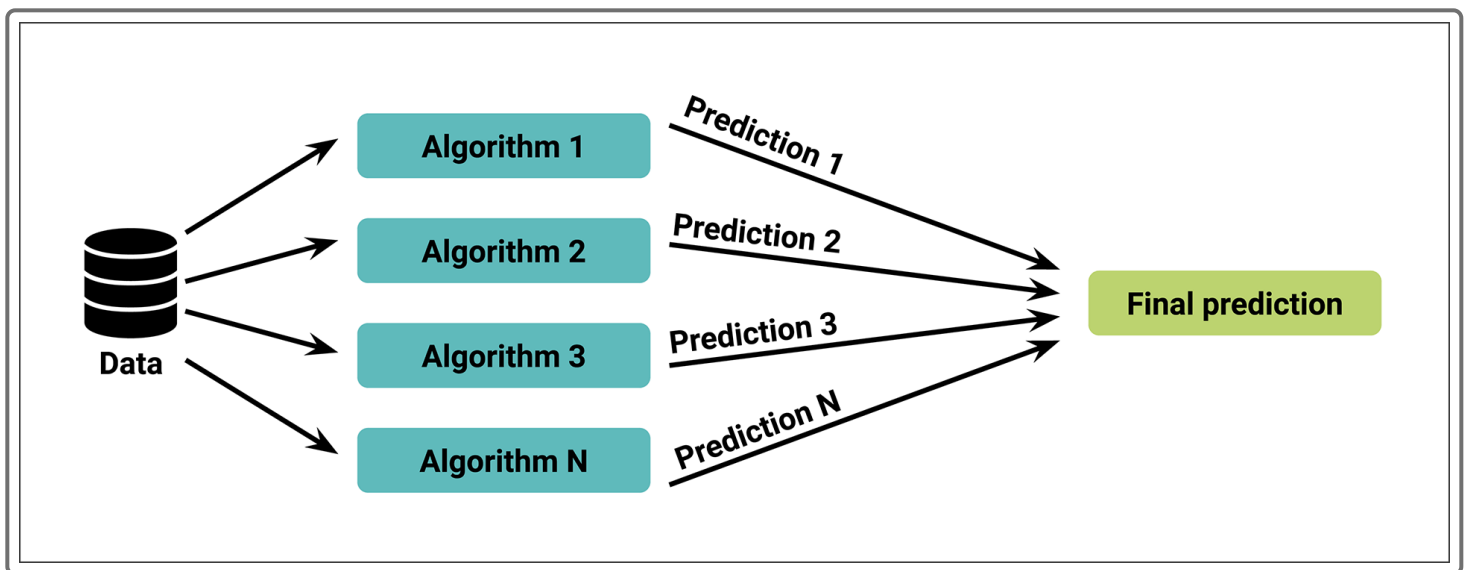
Weak Learners

Some algorithms are weak learners. Weak learners tend to have very few branches on the decision tree. A single weak learner will make inaccurate and imprecise predictions because they are poor at learning adequately as result of limited data, like too few features, or using data points that can't be classified.

Weak learners shouldn't be considered unworthy. Weak learners are valuable because there are models that can combine many weak learners to create a more accurate and robust prediction engine. When we combine weak learners, together they can perform just as well as any strong learner.

Our loan application prediction algorithm can be considered a moderate to weak learner because it was not good at classifying bad loan applications. However, if we combine this decision tree with more decision trees—all using different training and testing sets—then the prediction may be more accurate.

We can combine weak learners using a specific algorithm, like Random Forests, GradientBoostedTree, and XGBoost. We will learn about Random Forests next, and cover gradient boosting later in this module.

Random Forests

Instead of having a single, complex tree like the ones created by decision trees, a random forest algorithm will sample the data and build several smaller, simpler decision trees. Each tree is simpler because it is built from a random subset of features:
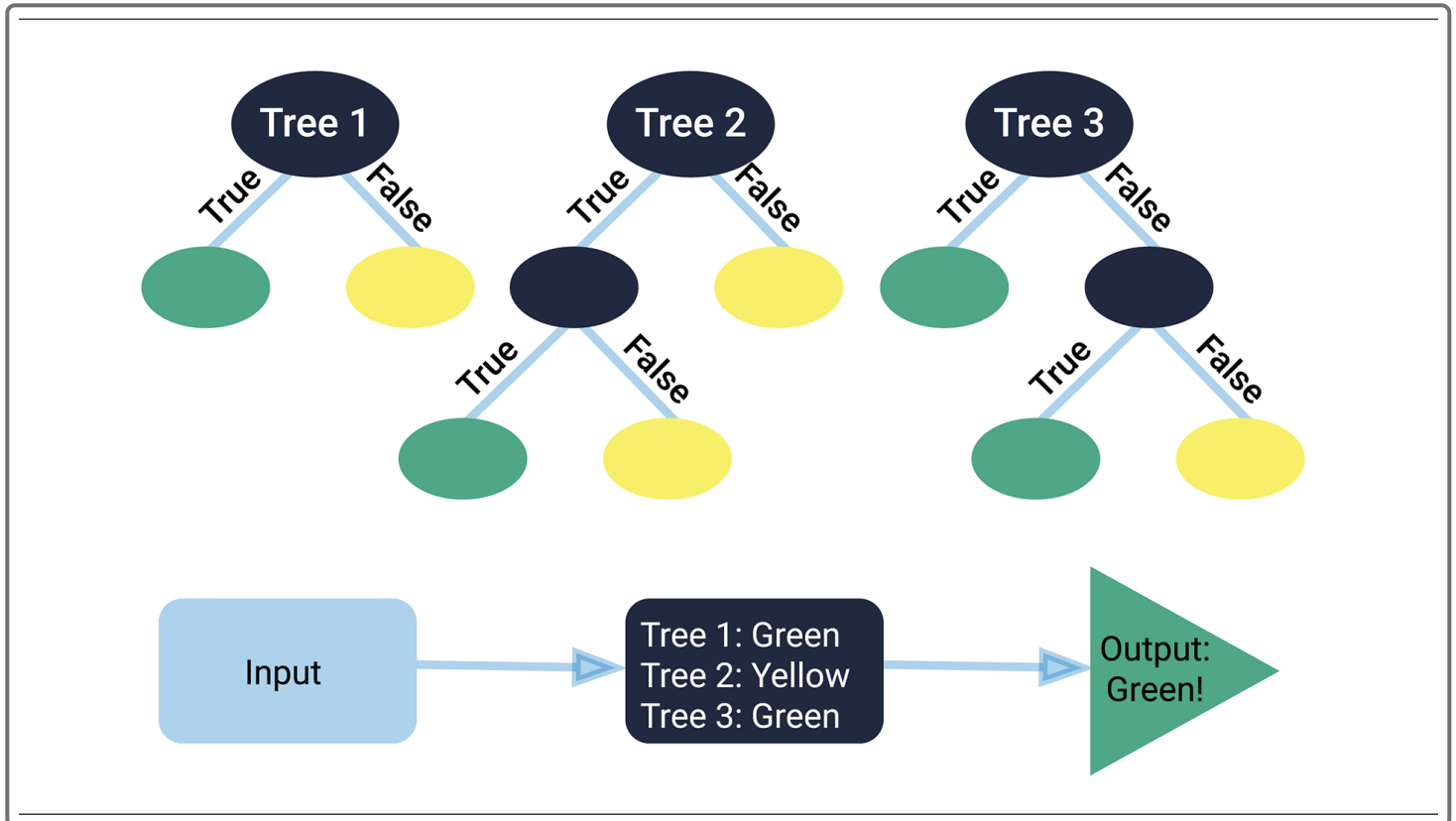
These simple trees are weak learners because they are created by randomly sampling the data and creating a decision tree for only that small portion of data. And since they are trained on a small piece of the original data, they are only slightly better than a random guess. However, many *slightly better than average* small decision trees can be combined to create a strong learner, which has much better decision-making power.

Random forest algorithms are beneficial because they:

- Are robust against overfitting as all of those weak learners are trained on different pieces of the data.

- Can be used to rank the importance of input variables in a natural way.

- Can handle thousands of input variables without variable deletion.

- Are robust to outliers and nonlinear data.

- Run efficiently on large datasets.

Since we determined that the decision tree in the previous example was not good at classifying bad loan applications, we're going to use the same loan applications' encoded dataset to predict bad loan applications using a random forest.