What is the difference between Decision Trees and Random Forests?

Authored by

stats writer

May 5, 2024

Decision Trees and Random Forests are two popular machine learning algorithms used for classification and regression tasks. The main difference between these two methods lies in their approach to building predictive models.

Decision Trees are a simple and intuitive algorithm that works by recursively partitioning the data into smaller and smaller subsets based on the most significant attributes. This results in a tree-like structure where each node represents a test on a particular attribute. The final leaf nodes of the tree contain the predicted outcome.

On the other hand, Random Forests are an ensemble learning method that combines multiple decision trees to make more accurate predictions. In this method, a large number of decision trees are built on different subsets of the data, and the final prediction is made by taking the average or majority vote of all the trees.

One of the main advantages of Random Forests over Decision Trees is that it reduces the risk of overfitting, which occurs when a model performs well on training data but poorly on new data. This is because each decision tree in a Random Forest is built on a subset of the data, reducing the chances of a single tree being overly biased towards the training data.

In summary, while Decision Trees are simple and easy to interpret, Random Forests provide more accurate predictions by combining multiple decision trees and reducing the risk of overfitting. Therefore, the choice between these two algorithms depends on the specific needs and goals of the predictive modeling task at hand.

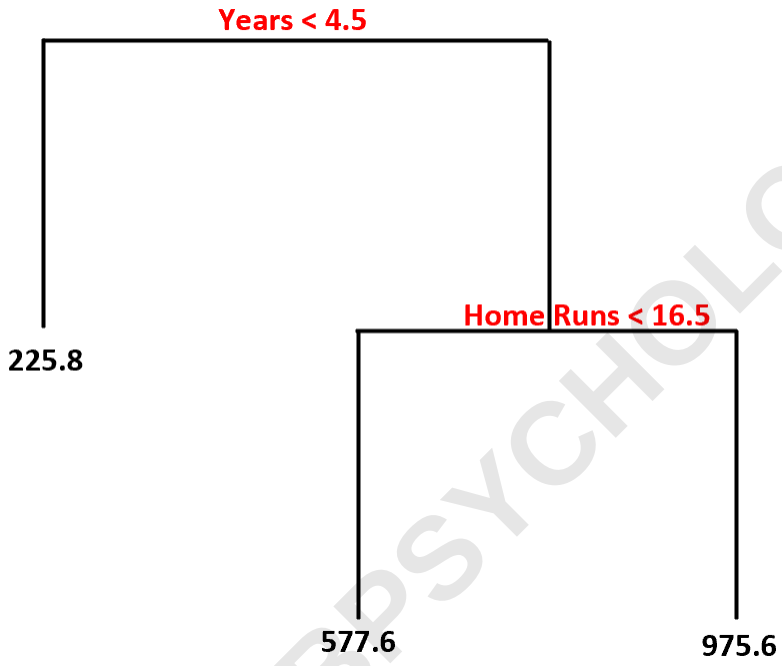
Decision Tree vs. Random Forests: What's the Difference?

A decision tree is a type of machine learning model that is used when the relationship between a set of predictor variables and a response variable is non-linear.

The basic idea behind a decision tree is to build a "tree" using a set of predictor variables that predicts the value of some response variable using decision rules.

For example, we might use the predictor variables "years played" and "average home runs" to predict the annual salary of professional baseball players.

Using this dataset , here's what the decision tree model might look like:



Here's how we would interpret this decision tree:

Players with less than 4.5 years played have a predicted salary of $225.8k.Players with greater than or equal to 4.5 years played and less than 16.5 average home runs have a predicted salary of $577.6k.Players with greater than or equal to 4.5 years played and greater than or

equal to 16.5 average home runs have a predicted salary of $975.6k.

The main advantage of a decision tree is that it can be fit to a dataset quickly and the final model can be neatly visualized and interpreted using a "tree" diagram like the one above.

The main disadvantage is that a decision tree is prone to a training dataset, which means it's likely to perform poorly on unseen data. It can also be heavily influenced by outliers in the dataset.

An extension of the decision tree is a model known as a random forest, which is essentially a collection of decision trees.

Here are the steps we use to build a random forest model:

1. Take bootstrapped samples from the original dataset.

2. For each bootstrapped sample, build a decision tree using a random subset of the predictor variables.

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

a final model.

The benefit of random forests is that they tend to perform much better than decision trees on unseen data and they're less prone to outliers.

The downside of random forests is that there's no way to visualize the final model and they can take a long time to build if you don't have enough computational power or if the dataset you're working with is extremely large.

Pros & Cons: Decision Trees vs. Random Forests

	Decision Tree	Random Forest
Interpretability	Easy to interpret	Hard to interpret
Accuracy	Accuracy can vary	Highly accurate
Overfitting	Likely to overfit data	Unlikely to overfit data
Outliers	Can be highly affected by outliers	Robust against outliers
Computation	Quick to build	Slow to build (computationally intensive)

Here's a brief explanation of each row in the table:

1. Interpretability

Decision trees are easy to interpret because we can

create a tree diagram to visualize and understand the final model.

Conversely, we can't visualize a random forest and it can often be difficulty to understand how the final random forest model makes decisions.

2. Accuracy

Since decision trees are likely to overfit a training dataset, they tend to perform less than stellar on unseen datasets.

Conversely, random forests tend to be highly accurate on unseen datasets because they avoid overfitting training datasets.

3. Overfitting

As mentioned earlier, decision trees often overfit training data - this means they're likely to fit the "noise" in a dataset as opposed to the true underlaying pattern.

Conversely, because random forests only use some predictor variables to build each individual decision tree, the final trees tend to be decorrelated which

means random forest models are unlikely to overfit datasets.

4. Outliers

Decision trees are highly prone to being affected by outliers.

Conversely, since a random forest model builds many individual decision trees and then takes the average of those trees predictions, it's much less likely to be affected by outliers.

5. Computation

Decision trees can be fit to datasets quickly.

Conversely, random forests are much more computationally intensive and can take a long time to build depending on the size of the dataset.

When to Use Decision Trees vs. Random Forests

As a rule of thumb:

You should use a decision tree if you want to build a non-linear model quickly and you want to be able to

easily interpret how the model is making decisions.

However, you should a random forest if you have plenty of computational ability and you want to build a model that is likely to be highly accurate without worrying about how to interpret the model.

In the real-world, machine learning engineers and data scientists often use random forests because they're highly accurate and modern-day computers and systems can often handle large datasets that couldn't previously be handled in the past.

The following tutorials provide an introduction to both decision trees and random forest models:

The following tutorials explain how to fit decision trees and random forests in R: