



Decision Trees - Introduction

 [_ \(https://github.com/learn-co-curriculum/dsc-decision-trees-section-intro\)](https://github.com/learn-co-curriculum/dsc-decision-trees-section-intro) 
(<https://github.com/learn-co-curriculum/dsc-decision-trees-section-intro/issues/new/choose>)

Introduction

In this section, we're going to introduce another kind of model for predicting values that can be used for both continuous and categorical predictions - decision trees. Decision trees are used to classify (or estimate continuous values) by partitioning the sample space as efficiently as possible into sets with similar data points until you get to (or close to) a homogenous set and can reasonably predict the value for new data points.

Despite the fact that they've been around for decades, they are still (in conjunction with ensemble methods that we'll learn about in the next section) one of the most powerful modeling tools available in the field of machine learning. They are also highly interpretable when compared to more complex models (they're simple to explain and it's easy to understand how they make their decisions).

Entropy and Information Gain

Due to the nature of decision trees, you can get very different predictions depending on what questions you ask and in what order. The question then is how to come up with the right questions to ask in the right order. In this section, we also introduce the idea of entropy and information gain as mechanisms for selecting the most promising questions to ask in a decision tree.

ID3 Classification Trees

We also talk about Ross Quinlan's ID3 (Iterative Dichotomiser 3) algorithm for generating a decision tree from a dataset.

Building Trees using Scikit-learn

Next up, we look at how to build a decision tree using the built-in functions available in scikit-learn, and how to test the accuracy of the predictions using a simple accuracy measure, AUC, and a confusion matrix. We also show how to use the `graph_viz` library to generate a visualization of the resulting decision tree.

Hyperparameter Tuning and Pruning

We then look at some of the hyperparameters available when optimizing a decision tree. For example, if you're not careful, generated decision trees can lead to overfitting of data (wherein a

model is a perfect match for training data, but horrible for test data). There are a number of hyperparameters you can use when generating a tree to minimize overfitting such as maximum depth or minimum leaf sample size. We look at these various "pruning" strategies to avoid overfitting of the data and to create a better model.

Regression with CART Trees

In addition to building decision tree classifiers, you will also build decision trees for regression problems.

Summary

Decision trees are highly effective and interpretable. This section will provide you with the skills to create both classifiers and to perform regression using decision trees and to use hyperparameter tuning to optimize your model.

How do you feel about this lesson?



Have specific feedback?

[Tell us here! \(https://github.com/learn-co-curriculum/dsc-decision-trees-section-intro/issues/new/choose\)](https://github.com/learn-co-curriculum/dsc-decision-trees-section-intro/issues/new/choose)