This is your first free member story since this month. Sign up for Medium and get extra one.

Photo by Marvin Meyer on Unsplash

# Decision Trees: A step-by-step approach to building DTs

Nick Kunal · Jun 23, 2020 · 10 min read

## Introduction

**Decision Trees (DTs)** are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Decision trees are commonly used in operation research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

## Context

In this article, we will be discussing the following topics

1. What are decision trees in general
2. Types of decision trees.
3. Algorithm used to build decision trees.
4. The step by step process of building a Decision tree.

## What are Decision trees?

Fig 1 Decision tree classifier to predict a person's preference

The above picture is a simple decision tree. If a person is non-vegetarian, then he/she eats chicken (most probably), otherwise, he/she doesn't eat chicken. The decision tree, in general, asks a question and classifies the person based on the answer. This decision tree is based on a yes/no question. It is just as simple to build a decision tree on numeric data.

Fig 2 Decision tree based on numeric data

If a person is driving above 80kmph, we can consider it as over-speeding, else not.

Fig 3 Decision tree on a certain data

Here is one more simple decision tree. This decision tree is based on ranked data, where 1 means the speed is very high, 2 corresponds to a much lower speed. If a person is speeding above rank 1 then he/she is highly over speeding. If the person is above speed rank 2 but below speed rank 1 then he/she is over-speeding but not that much. If the person is below speed rank 2 then he/she is driving well within speed limits.

The classification in a decision tree can be both categoric or numeric.

Fig 4 Complex DT

Here's a more complicated decision tree. It combines numeric data with yes/no data. For the most part Decision trees are pretty simple to work with. You start at the top and work your way down till you get to a point where you can go further. That's how a sample is classified.

The very top of the tree is called the **root node** or just the root. The nodes in between are called **internal nodes**. Internal nodes have arrows pointing to them and arrows pointing away from them. The end nodes are called the **leaf nodes** or just **leaves**. Leaf nodes have arrows pointing to them but no arrows pointing away from them.

In the above diagram, root nodes are represented by rectangles, internal nodes by circles, and leaf nodes by inverted triangles.

## Building a Decision tree

There are several algorithms to build a decision tree.

1. CART-Classification and Regression Trees
2. ID3-Iterative Dichotomiser 3
3. C4.5
4. CHAID-Chi-squared Automatic Interaction Detection

We will be discussing only CART and ID3 algorithms as they are the ones majorly used.

## CART

CART is a DT algorithm that produces **binary** Classification or Regression Trees, depending on whether the dependent (or target) variable is categorical or numeric respectively. It chooses a single feature in each case (no proprietary models) and use the same variable/feature/feature cross interact different parts of the same DT, which may uncover complex interdependencies between sets of variables.

Fig 5. Sample dataset

Now we are going to discuss how to build a decision tree from a raw table of data. In the example given above, we will be building a decision tree that predicts chest pain, good blood circulation and blocked arteries based on patients' medical records.

Fig 6 Chest pain as the root node

There are two leaf nodes, one each for the two outcomes of chest pain. Each of the leaves contains the no. of patients having heart disease and not having heart disease for the corresponding entry of chest pain. Now we do the same thing for good blood circulation and blocked arteries.

Fig 7 Good blood circulation as the root node

Fig 8 Blocked arteries as the root node

We can see that neither of the 3 features separates the patients having heart disease from the patients not having heart disease perfectly. It is to be noted that the leaf nodes of a particular feature have been different in all other cases. This is done to simulate the mixing values phrase in real world datasets.

Because none of the leaf nodes is either 100% 'yes heart disease' or 100% 'no heart disease', they are all considered **impure**. To decide on which separation is the best, we need a method to measure and compare **impurity**.

The metric used in the CART algorithm to measure impurity is the **Gini impurity score**. Calculating Gini impurity is very easy. Let's start by calculating the Gini impurity for chest pain.

Fig 9. Chest pain separation

For the left leaf,

```
Gini impurity = 1 - (probability of 'yes')² - (probability of 'no')²
Gini impurity = 1 - (105/(105 + 39))² - (39/(105+39))²
Gini impurity = 0.395
```

Similarly, calculate the Gini impurity for the right leaf node.

```
Gini impurity = 1 - (probability of 'yes')² - (probability of 'no')²
Gini impurity = 1 - (34/(34 + 125))² - (125/(34+125))²
Gini impurity = 0.336
```

Now that we have measured the Gini impurity for both leaf nodes, we can calculate the total Gini impurity by using chest pain to separate patients with and without heart disease.

The leaf nodes do not represent the same no. of patients so the left leaf represents 144 patients and the right leaf represents 159 patients. Thus the total Gini impurity will be the weighted average of the leaf node Gini impurities.

```
Gini impurity = (144/(144+159))*0.395 + (159/(144+159))*0.336
              = 0.361
```

Similarly the total Gini impurity for 'good blood circulation' and 'blocked arteries' is calculated as

```
Gini impurity for 'good blood circulation' = 0.360
Gini impurity for 'blocked arteries' = 0.381
```

'Good blood circulation' has the lowest impurity score among the tree which symbolizes that it best separates the patients having and not having heart disease, so we will use it as the root node.

Fig 10 Good blood circulation as the root node

Now we need to figure out how well 'chest pain' and 'blocked arteries' separate the 164 patients in the left node(247 with heart disease and 127 without heart disease).

Just like we did before we will separate these patients with 'chest pain' and calculate the Gini impurity value.

Fig 11. Chest pain separation

The Gini impurity was found to be 0.3. Then we do the same thing for 'blocked arteries'.



Fig 12 Blocked arteries separation

The Gini impurity was found to be 0.29. Since 'blocked arteries' has the lowest Gini impurity, we will use it at the left node in Fig.10 for further separating the patients.

Fig 13 Blocked arteries separation

All we have left is 'chest pain', so we will see how well it separates the 89 patients in the left node(24 with heart disease and 25 without heart disease).

Fig 14 Chest pain separation in the left node

We can see that chest pain does a good job separating the patients.



Fig 15 Final chest pain separation

So these are the final leaf nodes of the left side of this branch of the tree. Now let's see what happens when we try to separate the node having 13/102 patients using 'chest pain'. Note that almost 90% of the people in this node are not having heart disease.

Fig 16 Chest pain separation in the left node

The Gini impurity of this separation is 0.29. But the Gini impurity for the

parent-node before using chest-pain to separate the patients is

```
Gini impurity = [ - (probability of yes)² - (probability of no)²
Gini impurity = 0.2 [1/13]²+[12/13]²] - [8]/13]²]
```

The impurity is lower if we don't separate patients using 'chest pain'. So we will make it a leaf node.

Fig 17: Left side completed

At this point, we have worked out the entire left side of the tree. The same steps are to be followed to work out the right side of the tree.

1. Calculate the Gini impurity scores.
2. If the node itself has the lowest score, then there is no point in separating the patients anymore and it becomes a leaf node.
3. If separating the data results in improvement then pick the separation with the lowest impurity value.

Fig 18: Complete Decision tree

## ID3

The process of building a decision tree using the ID3 algorithm is almost similar to using the CART algorithm except for the method used for measuring purity/impurity. The metric used in the ID3 algorithm to measure purity is called ***Entropy***.



Entropy is a way to measure the uncertainty of a class in a subset of examples. Assume item belongs to subset 5 having two classes positive and negative. Entropy is defined as the no. of bits needed to say whether x is positive or negative.

Entropy always gives a number between 0 and 1. So if a subset formed after separation using an attribute is pure, then we will be needing zero bits to tell if is positive or negative. If the subset formed is having equal no. of positive and negative items then the no. of bits needed would be 1.



Fig 19:Entropy vs p(+)

The above plot shows the relation between entropy and i.e., the probability of positive class. As we can see, the entropy reaches 1 which is the maximum value when which is there are equal chances for an item to be either positive or negative. The entropy is at its minimum when p(+) tends to zero(symbolizing x is negative) or 1(symbolizing x is positive).

Entropy tells us how pure or impure each subset is after the split. What we need to do is aggregate these scores to check whether the split is feasible or not. This is done by **Information gain**.

Fig 20:Building an ID3 tree

Consider this part of the problem we discussed above for the CART algorithm. We need to decide which attribute to use from `chest pain` and `blocked arteries` for separating the left node containing 164 patients(127 having heart disease and 127 not having heart disease). We can calculate the entropy before splitting as



Let's see how well `chest pain` separates the patients

Fig 21: Chest pain separation

The entropy for the left node can be calculated



Similarly the entropy for the right node



The total gain in entropy after splitting using `chest pain`

This implies that if in the current situation if we were to pick `chest pain` for splitting the patients, we would gain 0.098 bits in certainty on the patient having or not having heart disease. Doing the same for `blocked arteries`, the gain obtained was 0.117. Since splitting with `blocked arteries` gives us more certainty, it would be picked. We can repeat the same procedure for all the nodes to build a DT based on the ID3 algorithm.

Note: The decision of whether to split a node into 2 or to declare it as a leaf node can be made by imposing a minimum threshold on the gain value required. If the acquired gain is above the threshold value, we can split the node, otherwise, leave it as a leaf node.

## Summary

The following are the take-aways from this article

1. The general concept behind decision trees.
2. The basic types of decision trees.
3. Different algorithms to build a Decision tree.
4. Building a Decision tree using CART algorithm.
5. Building a Decision tree using ID3 algorithm.

## References

1. Refer to this playlist on youtube for more details on building Decision trees using CART algorithm.

2. Refer to this playlist on youtube for more details on building Decision trees using ID3 algorithm.

PS:- I will be posting another article regarding Regression trees and Random Forests soon. *Stay tuned* :)

Do check out my other articles regarding Data Science and Machine Learning here. Feel free to reach out for a more in-depth discussion in the comments and on *LinkedIn*.


Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. Take a look.

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

98

Data Science | Machine Learning | Decision Tree | Mathematics

More from Towards Data Science

Follow

A Medium publication sharing concepts, ideas, and codes.

Read more from Towards Data Science

More From Medium

9 Distance Measures in Data Science
Maurilio Gorodestsky in Towards Data Science

Are You Still Using Pandas to Process Big Data in 2025?
Roman Orac in Towards Data Science

10 Git Commands I Learned During My First Year as a Software Developer
Nikhil Adithyan in Towards Data Science

Stylize and Automate Your Excel Files with Python
Nathan Pradhan in Towards Data Science

8 Fundamental Statistical Concepts for Data Science
Rebecca Vickery in Towards Data Science

Creating Automated Python Dashboards Using Jupyter Notebooks and GitHub Actions
Hatda Ibsen Simsek in Towards Data Science

You Should Master Data Analytics First Before Becoming a Data Scientist
Matt Przybyla in Towards Data Science

6 Web Scraping Tools That Make Collecting Data A Breeze
Zara A. Metwalli in Towards Data Science

Medium

About | Help | Legal