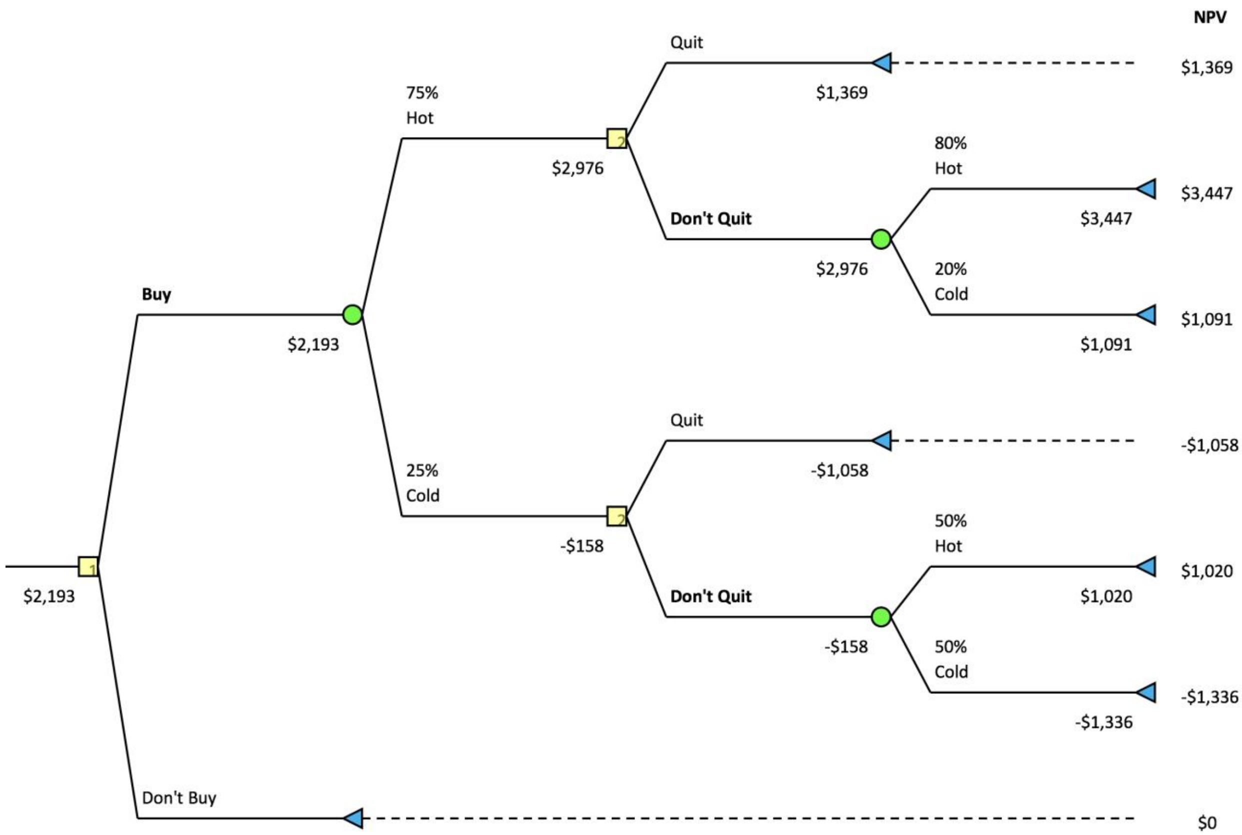
<u>A Decision Tree Primer</u>

Christopher Willmore, December 2018 (Updated June 2020)



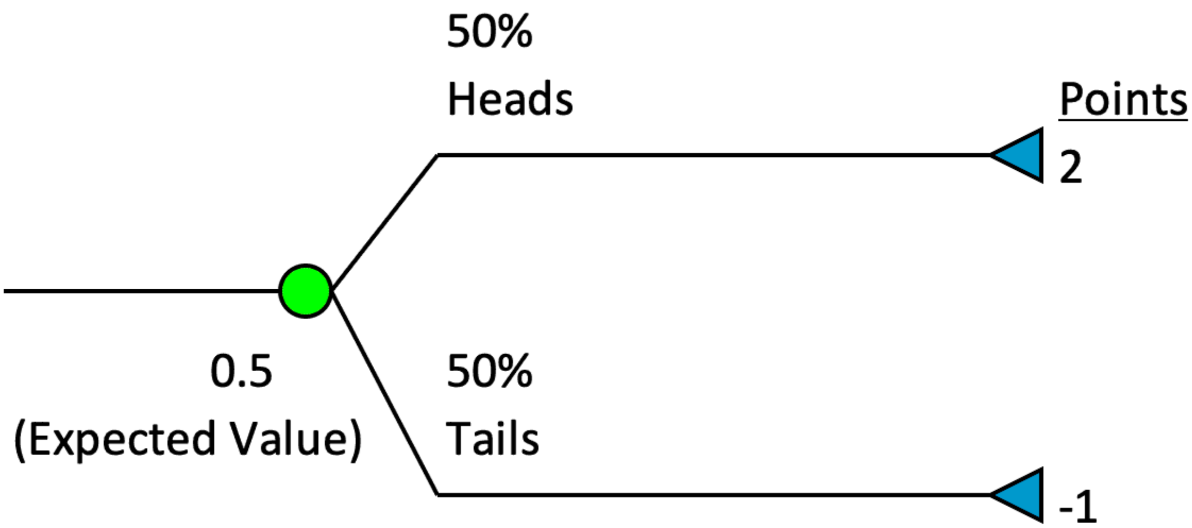
Above is a standard decision tree for a fictional popsicle business. A frequently asked question is, “What are those values on the right?” The answer to that question, for this particular tree, ‘the Net Present Value of the business given different choices and weather patterns. The answer for decision trees in general, is that the values on the right represent possible outcomes. At the very right, you should have everything that can happen. What the values are, depends on your context.

For example, your decision tree could be trying to figure out the number of episodes in shows that people watch on Netflix (maybe for a marketing algorithm). The numbers could include 1 (movie), 8 (miniseries), 10 (Short syndicated series), 13 (Netflix original) or 26 (full syndicated series).

You could be studying people's answers to a multiple-choice questionnaire about how much people are willing to pay for clean water, with possible values such as $0, $100, $1,000, etc.

You could be looking at the net present value of a lottery ticket purchase, in which case you'd list the present value of possible prizes, minus the cost of the lottery ticket. It all depends on the project you’re examining.

Let's look at an extremely simple decision tree.



You flip a coin. Heads, you win two points. Tails, you lose one point. Suppose this is a fair coin, so there's a 50% chance of heads, and a 50% chance of tails.

There are two possible outcomes: Heads, with a value of +2, and Tails, with a value of -1. (Since you gain two points with heads, and lose a point with tails.)

So, on the very right, we'll have two terminal (triangle) nodes - one with a value of +2, one with a value of -1.

That's getting ahead of ourselves, though. Let's go with the flow of time, and start our decision tree on the very left.

Well, it's a 'decision' tree, since there's really no decision to be made. You're flipping the coin, period, and the flip is stochastic (random) - it's a chance node.

So the first thing that happens in our simple tree is a random coin flip. We draw a circle, for a chance node.

What can happen if we flip the coin? It will either be heads, or be tails (ignoring the possibility of the coin landing on its side).

So we draw lines arrows leading out and to the right from the chance node: one for heads, one for tails. It's best to label these accordingly, for readability.

We should also label them with the probabilities of each thing happening: there's a 50% chance of Head, and a 50% chance of Tails.

The probabilities of all the lines leading out of a chance node, added up, have to add up to 100%. If they don't, you've either missed something (<100%) or double-counted (>100%).

What happens if we rolled heads? Then we're at our +2 point outcome. We draw a triangle at the end of the line leading out from the chance node, to represent a terminal node (outcome). Next to this outcome, we write '+2', indicating what the outcome's value is.

What happens if we rolled tails? Then we're at the -1 point outcome, and we work similarly to the above.

What's the expected value of our little tree?

Remember that all the lines leading out of a chance node have probabilities attached to them, and these must add up to exactly 100% (since you're describing everything that can happen).

This means it's really easy to calculate the (mathematical) expected value for a chance node. Just calculate the weighted average of outcomes leading out from it, using the probabilities as the weights. For each branch leading out of a particular chance node, multiply the probability times the value of the outcome. Then add those up, to get the expected value.

In our simple case, the expected value of our chance node (and therefore our entire tree) is:

50% x (+2) + 50% x (-1) = 1 - 0.5 = +0.5

Now let's consider a slightly different game. At the start, you have to decide whether to flip a coin, OR not flip a coin. If you flip a coin, the game proceeds as above. If you DON'T flip a coin, the game ends, and you get 0 points.

Now, at the start, you need to decide: flip a coin, or don't flip a coin?

We draw a square, representing a DECISION node.

There are two lines leading out from this decision node, corresponding to the two possible decisions: one for 'flip a coin', the other one for 'don't flip a coin'.

These lines DON'T have probabilities attached, because they don't represent something random; they represent the result of a conscious choice.

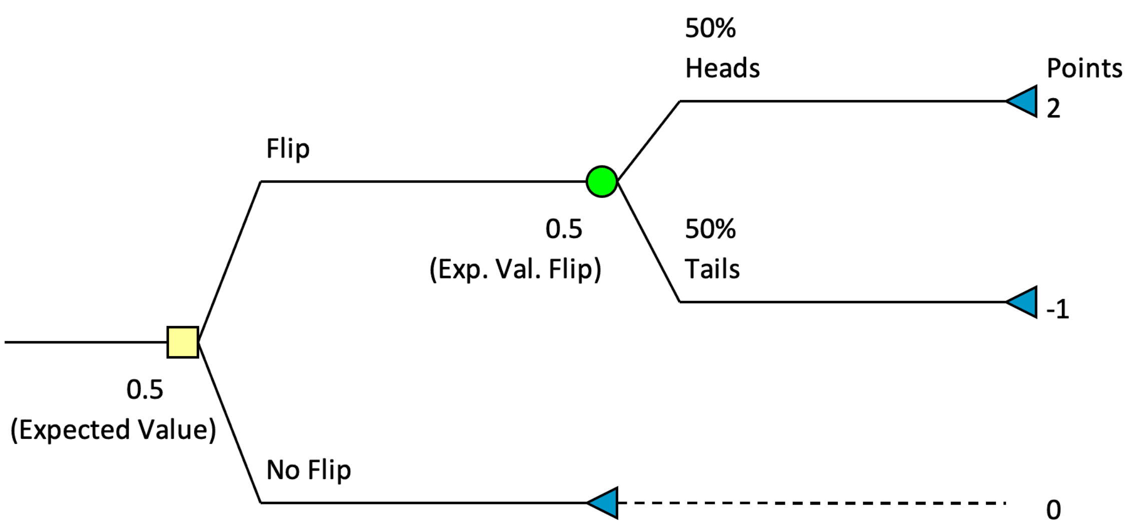
If we flip a coin, we know what happens. We've done that already. We can just attach our old tree to the end of that line. We even know the expected value of that already: +0.5.

If we DON'T flip a coin, the game ends and we get zero points. So we draw a terminal (triangle) node with a zero next to it.

So now, we have a decision node with two lines leading out of it. One has an expected value of +0.5, and the other has an expected value of 0.

Which do we choose?

It depends on the game rules. If the rules say more points are better than less points, then we'll choose to flip the coin, since +0.5 > 0, and the expected value of the whole tree will be +0.5 (since that's what we'll choose). That's what's shown on the tree drawn below.



This tree assumes that more points = better, so 0.5 points (in expectation) is better than 0 points.

If the rules are that less points are better than more points, then we'll choose to NOT flip the coin, since 0 < +0.5, and the expected value of the tree will be 0 (since that's what we'll choose).

I've only done one chance node and one decision node, but that's all you need to know to generalize - as you work backward, you're always just working one node at a time. Weighted average for chance nodes, whatever the 'correct' decision is for choice nodes.