

Decision tree model

Welcome to the final week of this course on Advanced Learning Algorithms. One of the learning algorithms that is very powerful, widely used in many applications, also used by many to win machine learning competitions is decision trees and tree ensembles.

Despite all the successes of decision trees, they somehow haven't received that much attention in academia, and so you may not hear about decision trees nearly that much, but it is a tool well worth having in your toolbox.

In this week, we'll learn about decision trees and you'll see how to get them to work for yourself. Let's dive in. To explain how decision trees work, I'm going to use as a running example this week a cat classification example. You are running a cat adoption center and given a few features, you want to train a classifier to quickly tell you if an animal is a cat or not.

I have here 10 training examples. Associated with each of these 10 examples, we're going to have features regarding the animal's ear shape, face shape, whether it has whiskers, and then the ground truth label that you want to predict this animal cat. The first example has pointy ears, round face, whiskers are present, and it is a cat.

The second example has floppy ears, the face shape is not round, whiskers are present, and yes, that is a cat, and so on for the rest of the examples. This dataset has five cats and five dogs in it. The input features X are these three columns, and the target output that you want to predict, Y, is this final column of, is this a cat or not? In this example, the features X take on categorical values.

Cat classification example

| Ear shape (x_1) | Face shape (x_2) | Whiskers (x_3) | Cat |
|---------------------|----------------------|--------------------|-----|
| Pointy ↗ | Round ↗ | Present ↗ | 1 |
| Floppy ↗ | Not round ↗ | Present | 1 |
| Floppy | Round | Absent ↗ | 0 |
| Pointy | Not round | Present | 0 |
| Pointy | Round | Present | 1 |
| Pointy | Round | Absent | 1 |
| Floppy | Not round | Absent | 0 |
| Pointy | Round | Absent | 1 |
| Floppy | Round | Absent | 0 |
| Floppy | Round | Absent | 0 |

Categorical (discrete values) X Y

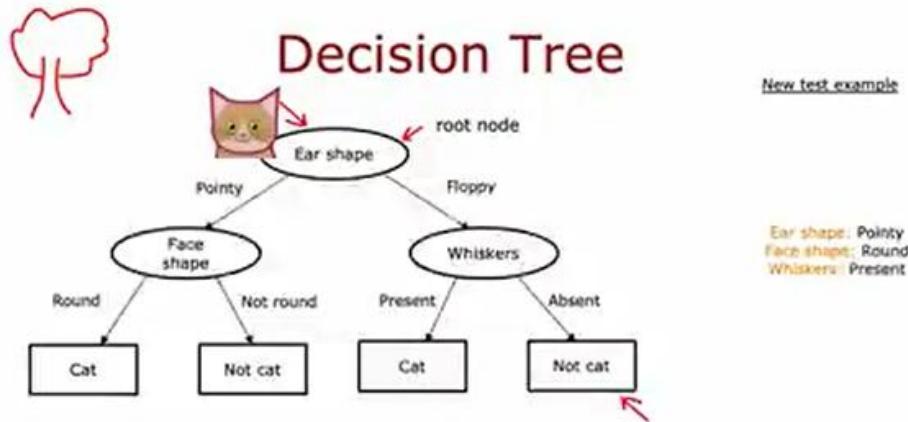
In other words, the features take on just a few discrete values. Your shapes are either pointy or floppy. The face shape is either round or not round and whiskers are either present or absent. This is a binary classification task because the labels are also one or zero.

For now, each of the features X_1 , X_2 , and X_3 take on only two possible values. We'll talk about features that can take on more than two possible values, as well as continuous-valued features later in this week. What is a decision tree?

Here's an example of a model that you might get after training a decision tree learning algorithm on the data set that you just saw. The model that is output by the learning algorithm looks like a tree, and a picture like this is what computer scientists call a tree. If it looks nothing like the biological trees that you see out there to you, it's okay, don't worry about it.

We'll go through an example to make sure that this computer science definition of a tree makes sense to you as well. Each one of these ovals or rectangles is called a node in the tree. The way this model works is if you have a new test example, she has a cat where the ear-shaped has pointy, face shape is round, and whiskers are present.

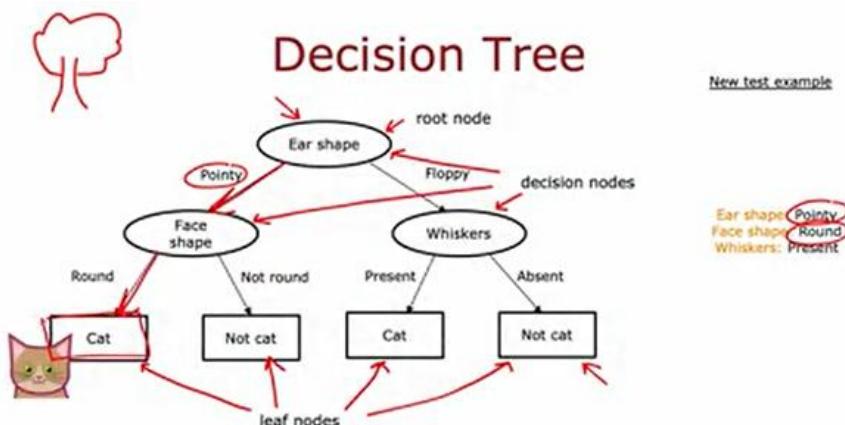
The way this model will look at this example and make a classification decision is will start with this example at this topmost node of the tree, this is called the root node of the tree, and we will look at the feature written inside, which is ear shape.



Based on the value of the ear shape of this example we'll either go left or go right. The value of the ear-shape with this example is pointy, and so we'll go down the left branch of the tree, like so, and end up at this oval node over here. We then look at the face shape of this example, which turns out to be round, and so we will follow this arrow down over here.

The algorithm will make an inference that it thinks this is a cat. You get to this node and the algorithm will make a prediction that this is a cat. What I've shown on this slide is one specific decision tree model. To introduce a bit more terminology, this top-most node in the tree is called the root node.

All of these nodes, that is, all of these oval shapes, but excluding the boxes at the bottom, all of these are called decision nodes. They're decision nodes because they look at a particular feature and then based on the value of the feature, cause you to decide whether to go left or right down the tree.



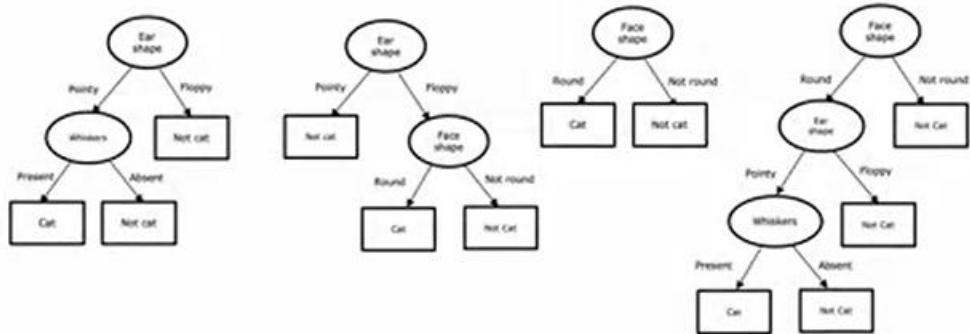
Finally, these nodes at the bottom, these rectangular boxes are called leaf nodes. They make a prediction. If you haven't seen computer scientists' definitions of trees before, it may seem non-

intuitive that the roots of the tree is at the top and the leaves of the tree are down at the bottom. Maybe one way to think about this is this is more akin to an indoor hanging plant, which is why the roots are up top, and then the leaves tend to fall down to the bottom of the tree. In this slide, I've shown just one example of a decision tree.

Here are a few others. This is a different decision tree for trying to classify cat versus not cat. In this tree, to make a classification decision, you would again start at this topmost root node. Depending on their ear shape of an example, you'd go either left or right. If the ear shape is pointy, then you look at the whiskers feature, and depending on whether whiskers are present or absent, you go left or right to gain and classify cat versus not cat.

Just for fun, here's a second example of a decision tree, here's a third one, and here's a fourth one. Among these different decision trees, some will do better and some will do worse on the training sets or on the cross-validation and test sets.

Decision Tree



The job of the decision tree learning algorithm is, out of all possible decision trees, to try to pick one that hopefully does well on the training set, and then also ideally generalizes well to new data such as your cross-validation and test sets as well. Seems like there are lots of different decision trees one could build for a given application. How do you get an algorithm to learn a specific decision tree based on a training set?