

Learning process

The process of building a decision tree given a training set has a few steps. In this video, let's take a look at the overall process of what you need to do to build a decision tree. Given a training set of 10 examples of cats and dogs like you saw in the last video. The first step of decision tree learning is, we have to decide what feature to use at the root node.

That is the first node at the very top of the decision tree. Via an algorithm that we'll talk about in the next few videos. Let's say that we decided to choose as the feature and the root node, the ear shape feature.

Decision Tree Learning



What that means is we will decide to look at all of our training examples, all ten examples shown here, I split them according to the value of the ear shape feature. In particular, let's pick out the five examples with pointy ears and move them over down to the left. Let's pick the five examples with floppy ears and move them down to the right.

The second step is focusing just on the left part or sometimes called the left branch of the decision tree to decide what nodes to put over there. In particular, what feature that we want to split on or what feature do we want to use next. Via an algorithm that again, we'll talk about later this week. Let's say you decide to use the face shape feature there.

Decision Tree Learning



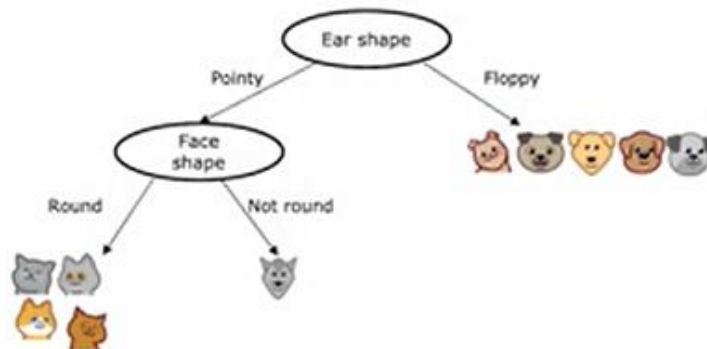
What we'll do now is take these five examples and split these five examples into two subsets based on their value of the face shape. We'll take the four examples out of these five with a round face shape and move them down to the left. The one example with a not round face shape and move it down to the right. Finally, we notice that these four examples are all cats four of them are cats.

Rather than splitting further, we created a leaf node that makes a prediction that things that get down to that node of cats. Over here we notice that none of the examples zero of the one examples are cats or alternative 100 percent of the examples here are dogs. We can create a leaf node here that makes a prediction of not cat.

Having done this on the left part to the left branch of this decision tree, we now repeat a similar process on the right part or the right branch of this decision tree. Focus attention on just these five

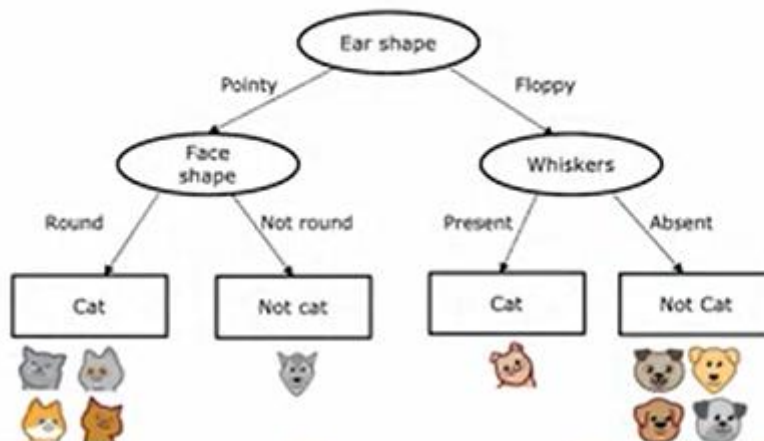
examples, which contains one cat and four dogs. We would have to pick some feature over here to use the split these five examples further, if we end up choosing the whiskers feature, we would then split these five examples based on where the whiskers are present or absent, like so.

Decision Tree Learning



You notice that one out of one examples on the left for cats and zeros out of four are cats. Each of these nodes is completely pure, meaning that is, all cats or not cats and there's no longer a mix of cats and dogs. We can create these leaf nodes, making a cat prediction on the left and a nightcap prediction here on the right. This is a process of building a decision tree.

Decision Tree Learning



Through this process, there were a couple of key decisions that we had to make at various steps during the algorithm. Let's talk through what those key decisions were and we'll keep on session of the details of how to make these decisions in the next few videos. The first key decision was, how do you choose what features to use to split on at each node?

At the root node, as well as on the left branch and the right branch of the decision tree, we had to decide if there were a few examples at that node including a mix of cats and dogs. Do you want to split on the ear-shaped feature or the facial feature or the whiskers feature? We'll see in the next

video, that decision trees will choose what feature to split on in order to try to maximize purity. By purity, I mean, you want to get to what subsets, which are as close as possible to all cats or all dogs.

Decision Tree Learning

Decision 1: How to choose what feature to split on at each node?

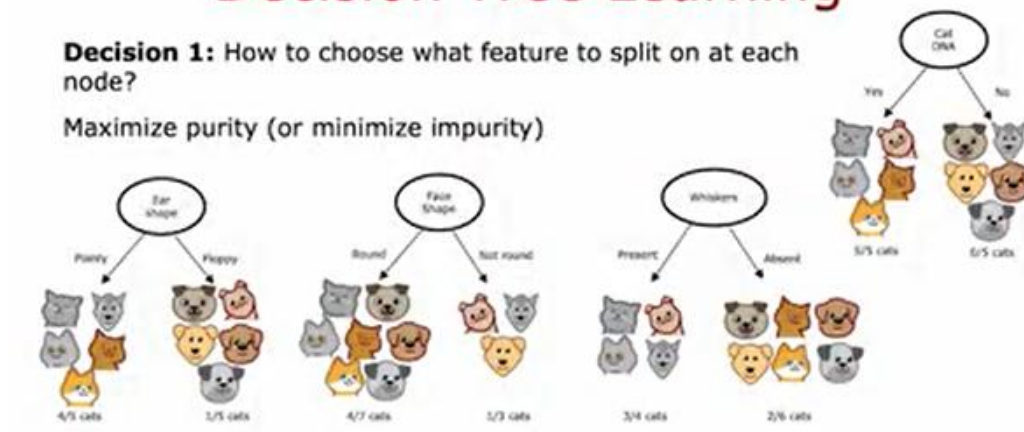


For example, if we had a feature that said, does this animal have cat DNA, we don't actually have this feature. But if we did, we could have split on this feature at the root node, which would have resulted in five out of five cats in the left branch and zero of the five cats in the right branch. Both these left and right subsets of the data are completely pure, meaning that there's only one class, either cats only or not cats only in both of these left and right sub-branches, which is why the cat DNA feature if we had this feature, would have been a great feature to use.

But with the features that we actually have, we had to decide, what is the split on year shape, which result in four out of five examples on the left being cats, and one of the five examples on the right being cats or face shape where it resulted in the four of the seven on the left and one of the three on the right, or whiskers, which resulted in three out four examples being cast on the left and two out of six being not cats on the right. The decision tree learning algorithm has to choose between ear-shaped, face shape, and whiskers.

Which of these features results in the greatest purity of the labels on the left and right sub branches? Because it is if you can get to a highly pure subsets of examples, then you can either predict cat or predict not cat and get it mostly right.

Decision Tree Learning



The next video on entropy, we'll talk about how to estimate impurity and how to minimize impurity. The first decision we have to make when learning a decision tree is how to choose which feature to split on on each node. The second key decision you need to make when building a decision tree is to decide when do you stop splitting.

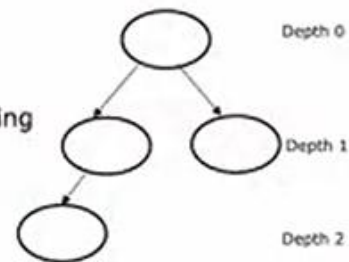
The criteria that we use just now was until I know there's either 100 percent, all cats or a 100 percent of dogs and not cats. Because at that point it seems natural to build a leaf node that just makes a classification prediction. Alternatively, you might also decide to stop splitting when splitting and no further results in the tree exceeding the maximum depth.

Where the maximum depth that you allow the tree to go to, is a parameter that you could just say.

Decision Tree Learning

Decision 2: When do you stop splitting?

- When a node is 100% one class
- When splitting a node will result in the tree exceeding a maximum depth



In decision tree, the depth of a node is defined as the number of hops that it takes to get from the root node that is denoted the very top to that particular node. So the root node takes zero hops, to get to itself and is at Depth 0. The nodes below it are at depth one and in those below it would be at Depth 2.

If you had decided that the maximum depth of the decision tree is say two, then you would decide not to split any nodes below this level so that the tree never gets to Depth 3. One reason you might want to limit the depth of the decision tree is to make sure for us to tree doesn't get too big and unwieldy and second, by keeping the tree small, it makes it less prone to overfitting.

Another criteria you might use to decide to stop splitting might be if the improvements in the purity score, which you see in a later video of below a certain threshold. If splitting a node results in minimum improvements to purity or you see later is actually decreases in impurity.

But if the gains are too small, they might not bother. Again, both to keep the trees smaller and to reduce the risk of overfitting. Finally, if the number of examples that a node is below a certain threshold, then you might also decide to stop splitting.

For example, if at the root node we have split on the face shape feature, then the right branch will have had just three training examples with one cat and two dogs and rather than splitting this into even smaller subsets, if you decided not to split further set of examples with just three of your examples, then you will just create a decision node and because there are mainly dogs, 2 out three adults here, this would be a node and this makes a prediction of not cat.

Decision Tree Learning

Decision 2: When do you stop splitting?

- When a node is 100% one class
- When splitting a node will result in the tree exceeding a maximum depth
- When improvements in purity score are below a threshold
- When number of examples in a node is below a threshold



Again, one reason you might decide this is not worth splitting on is to keep the tree smaller and to avoid overfitting. When I look at decision tree learning algorithms myself, sometimes I feel like, boy, there's a lot of different pieces and lots of different things going on in this algorithm. Part of the reason it might feel like this is in the evolution of decision trees.

There was one researcher that proposed a basic version of decision trees and then a different researcher said, oh, we can modify this thing this way, such as his new criteria for splitting. Then a different researcher comes up with a different thing like, oh, maybe we should stop splitting when it reaches a certain maximum depth. Over the years, different researchers came up with different refinements to the algorithm.

As a result of that, it does work really well but we look at all the details of how to implement a decision tree. It feels a lot of different pieces such as why there's so many different ways to decide when to stop splitting. If it feels like a somewhat complicated, messy algorithm to you, it does to me too.

But these different pieces, they do fit together into a very effective learning algorithm and what you learn in this course is the key, most important ideas for how to make it work well, and then at the end of this week, I'll also share with you some guidance, some suggestions for how to use open source packages so that you don't have to have too complicated the procedure for making all these decisions.

Like how do I decide to stop splitting? You really get these algorithms to work well for yourself. But I want to reassure you that if this algorithm seems complicated and messy, it frankly does to me too, but it does work well. Now, the next key decision that I want to dive more deeply into is how do you decide how to split a node. In the next video, let's take a look at this definition of entropy, which would be a way for us to measure purity, or more precisely, impurity in a node.