

# Use of one-shot encoding of categorical features

In the example we've seen so far each of the features could take on only one of two possible values. The ear shape was either pointy or floppy, the face shape was either round or not round and whiskers were either present or absent. But whether if you have features that can take on more than two discrete values, in this video we'll look at how you can use one-hot encoding to address features like that.

Here's a new training set for our pet adoption center application where all the data is the same except for the ear shaped feature. Rather than ear shape only being pointy and floppy, it can now also take on an oval shape. And so the initial feature is still a categorical value feature but it can take on three possible values instead of just two possible values.

And this means that when you split on this feature you end up creating three subsets of the data and end up building three sub branches for this tree.

## Features with three possible values

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

3 possible values



But in this video I'd like to describe a different way of addressing features that can take on more than two values, which is to use the one-hot encoding. In particular rather than using an ear shaped feature, they can take on any of three possible values. We're instead going to create three new features where one feature is, does this animal have pointy ears, a second is does their floppy ears and the third is does it have oval ears.

And so for the first example whereas we previously had ear shape as pointy, we are now instead say that this animal has a value for the pointy ear feature of 1 and 0 for floppy and oval. whereas previously for the second example, we previously said it had oval ears now we'll say that it has a value of 0 for pointy ears because it doesn't have pointy ears.

It also doesn't have floppy ears but it does have oval ears which is why this value here is 1 and so on for the rest of the examples in the data set. And so instead of one feature taking on three possible values, we've now constructed three new features each of which can take on only one of two possible values, either 0 or 1.

## One hot encoding

Ear shape	Pointy ears	Floppy ears	Oval ears	Face shape	Whiskers	Cat
Pointy	1	0	0	Round	Present	1
Oval	0	0	1	Not round	Present	1
Oval	0	0	1	Round	Absent	0
Pointy	1	0	0	Not round	Present	0
Oval	0	0	1	Round	Present	1
Pointy	1	0	0	Round	Absent	1
Floppy	0	1	0	Not round	Absent	0
Oval	0	0	1	Round	Absent	1
Floppy	0	1	0	Round	Absent	0
Floppy	0	1	0	Round	Absent	0

In a little bit more detail, if a categorical feature can take on  $k$  possible values,  $k$  was three in our example, then we will replace it by creating  $k$  binary features that can only take on the values 0 or 1. And you notice that among all of these three features, if you look at any role here, exactly 1 of the values is equal to 1. And that's what gives this method of future construction the name one-hot encoding.

And because one of these features will always take on the value 1 that's the hot feature and hence the name one-hot encoding. And with this choice of features we're now back to the original setting of where each feature only takes on one of two possible values, and so the decision tree learning algorithm that we've seen previously will apply to this data with no further modifications.

Just an aside, even though this week's material has been focused on training decision tree models the idea of using one-hot encodings to encode categorical features also works for training neural networks. In particular if you were to take the face shape feature and replace round and not round with 1 and 0 where round gets matter 1, not round gets matter 0 and so on.

## One hot encoding

If a categorical feature can take on  $k$  values, create  $k$  binary features (0 or 1 valued).

And for whiskers similarly replace presence with 1 and absence with 0. They noticed that we took all the categorical features we had where we had three possible values for ear shape, two for face shape and one for whiskers and encoded as a list of these five features. Three from the one-hot encoding of ear shape, one from face shape and from whiskers and now this list of five features can also be fed to a new network or to logistic regression to try to train a cat classifier.

So one-hot encoding is a technique that works not just for decision tree learning but also lets you encode categorical features using ones and zeros, so that it can be fed as inputs to a neural network as well which expects numbers as inputs. So that's it, with a one-hot encoding you can get your decision tree to work on features that can take on more than two discrete values and you can also apply this to neural networks or linear regression or logistic regression training.

But how about features that are numbers that can take on any value, not just a small number of discrete values. In the next video let's look at how you can get the decision tree to handle continuous value features that can be any number.

## One hot encoding and neural networks

Pointy ears	Floppy ears	Rounded ears	Face shape	Whiskers	Cat
1	0	0	Round 1	Present 1	1
0	0	1	Not-round 0	Present 1	1
0	0	2	Around 1	Absent 0	0
1	0	0	Not-round 0	Present 1	0
0	0	1	Round 1	Present 1	1
1	0	0	Round 1	Absent 0	1
0	1	0	Not-round 0	Absent 0	1
0	0	2	Round 1	Absent 0	1
0	1	0	Round 1	Absent 0	1
0	1	0	Round 1	Absent 0	1