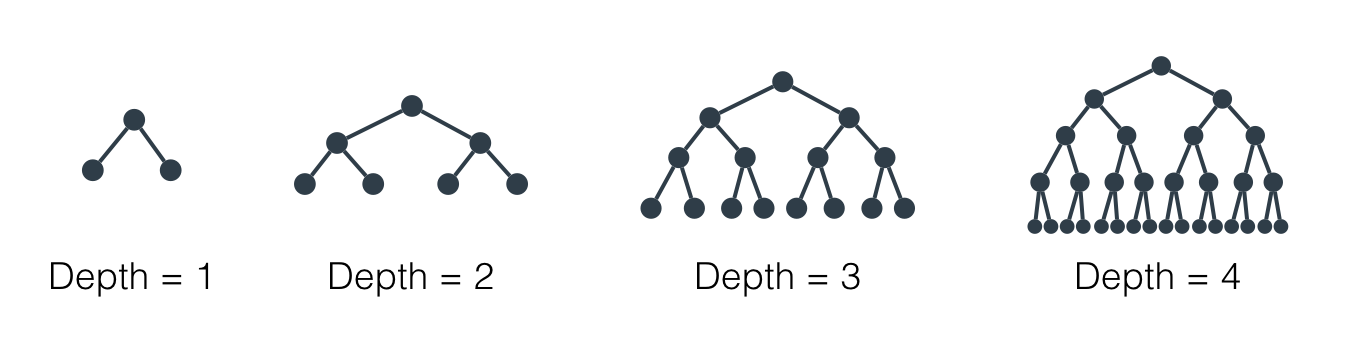
**Hyperparameters for Decision Trees**

In order to create decision trees that will generalize to new problems well, we can tune a number of different aspects about the trees. We call the different aspects of a decision tree "hyperparameters". These are some of the most important hyperparameters used in decision trees:

**Maximum Depth**

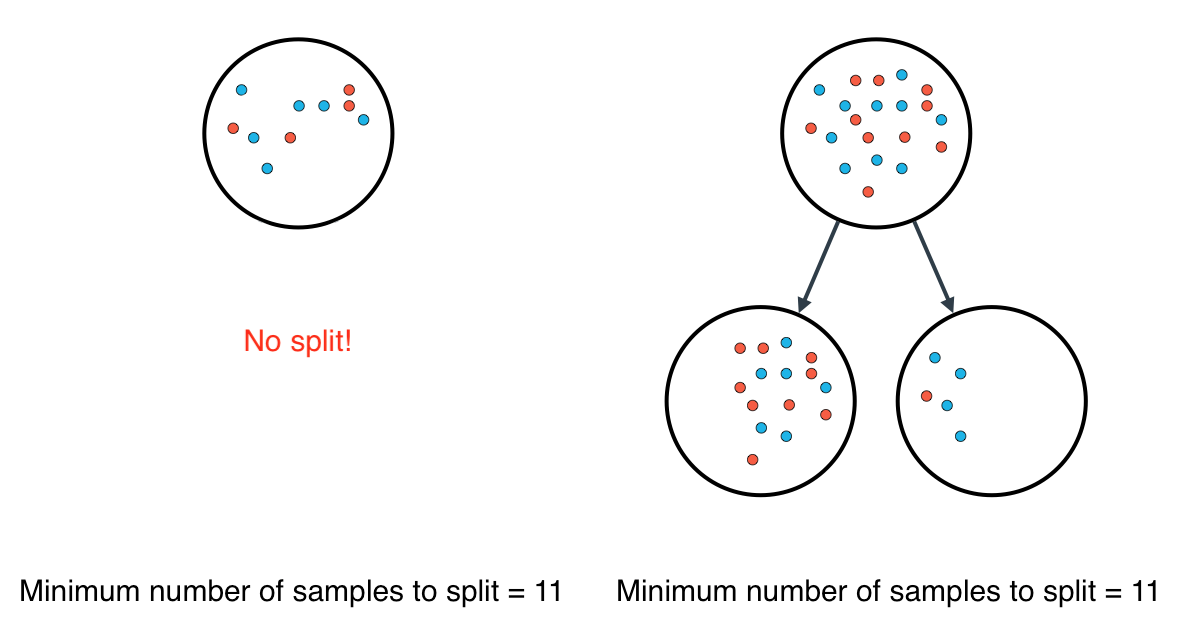
The maximum depth of a decision tree is simply the largest possible length between the root to a leaf. A tree of maximum length k*k* can have at most 2k2*k* leaves.

[[](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

[Maximum depth of a decision tree](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

**Minimum number of samples to split**

A node must have at least min\_samples\_split samples in order to be large enough to split. If a node has fewer samples than min\_samples\_split samples, it will not be split, and the splitting process stops.

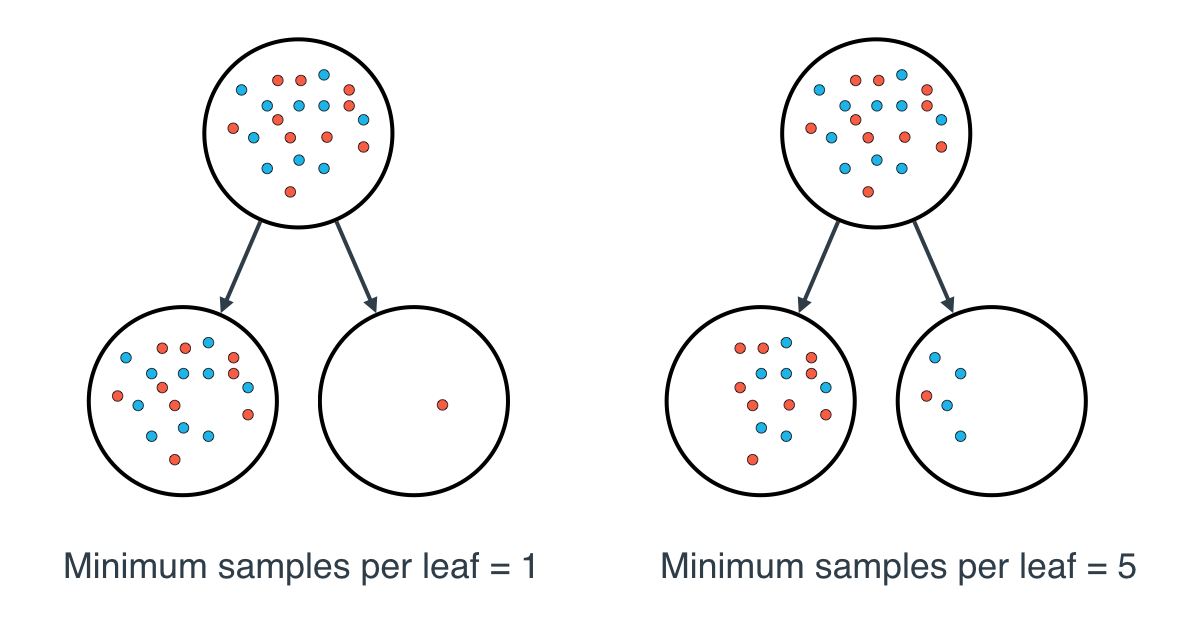
[](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

[Minimum number of samples to split](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

However, min\_samples\_split doesn't control the minimum size of leaves. As you can see in the example on the right, above, the parent node had 20 samples, greater than min\_samples\_split = 11, so the node was split. But when the node was split, a child node was created with that had 5 samples, less than min\_samples\_split = 11.

**Minimum number of samples per leaf**

When splitting a node, one could run into the problem of having 99 samples in one of them, and 1 on the other. This will not take us too far in our process, and would be a waste of resources and time. If we want to avoid this, we can set a minimum for the number of samples we allow on each leaf.

[](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

[Minimum number of samples per leaf](https://classroom.udacity.com/nanodegrees/nd025/parts/c6a7e66b-64f6-41eb-9b3e-e068e0ed962b/modules/aea22017-a8fe-4902-ba79-0d79f161a6bb/lessons/7bf3146d-1583-4e02-96ac-325b275892a7/concepts/e6359cb3-9797-490d-9b86-f9bfb321138d)

This number can be specified as an integer or as a float. If it's an integer, it's the minimum number of samples allowed in a leaf. If it's a float, it's the minimum percentage of samples allowed in a leaf. For example, 0.1, or 10%, implies that a particular split will not be allowed if one of the leaves that results contains less than 10% of the samples in the dataset.

If a threshold on a feature results in a leaf that has fewer samples than min\_samples\_leaf, the algorithm will not allow *that* split, but it may perform a split on the same feature at a *different threshold*, that *does* satisfy min\_samples\_leaf.

* **Large depth very often causes overfitting, since a tree that is too deep, can memorize the data. Small depth can result in a very simple model, which may cause underfitting.**
* **Small minimum samples per split may result in a complicated, highly branched tree, which can mean the model has memorized the data, or in other words, overfit. Large minimum samples may result in the tree not having enough flexibility to get built and may result in underfitting**.