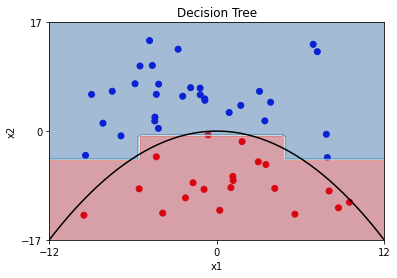
**Decision Trees Classification**

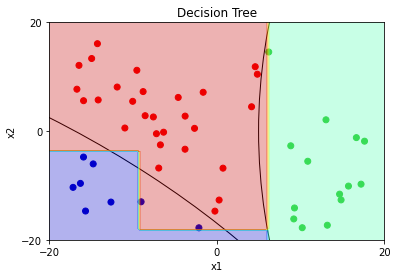
Now I’d like to look at the classifier more generally, and compare its performance to other classifiers so far, on the usual contrived random data set.

**Exploring the Model and Hyperparameters**

Here’s the decision tree with pure data (N = 50) on a linear boundary, and on a quadratic boundary.

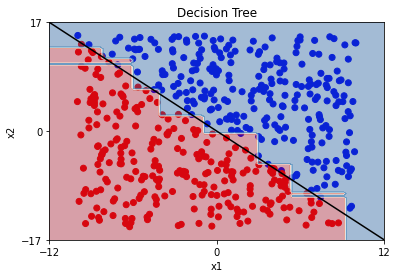
A diagram of a tree

Description automatically generated 

 A diagram of a tree

Description automatically generated

Can see the algorithm separates everything into grids: a<x1<b, c<x2<d. And here it is for N = 500.

 A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated A diagram of a tree

Description automatically generated

Now we’ll do 10% outliers with N = 50, and 15% outliers for triple class guys,

A diagram of a tree

Description automatically generated A diagram of a tree with red and blue dots

Description automatically generated

A diagram of a tree

Description automatically generated A diagram of a tree

Description automatically generated

and N = 500,

A diagram of a tree

Description automatically generated A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated A diagram of a decision tree

Description automatically generated

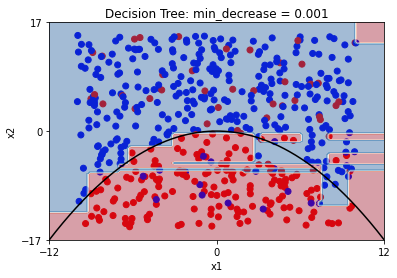
The yellow strips are overlapping colors I guess. As we can see, the decision tree, really likes to overfit. That’s why we have tuneable hyperparameters. I guess I’ll just discuss a few.

**Hyperparameter: loss**

So we can specify the loss function as *criterion = entropy, gini, or log\_loss*. I don’t imagine that would drastically change the performance here, apropos overfitting, and it doesn’t.

**Hyperparameter: min\_impurity\_decrease**

So when determining how to split a node, we look for the way which will maximize the (gini?) impurity decrease. If the gain isn’t substantial, then we may just forego doing so, as this might mean we’re following too closely the data. And so we can set a lower bound for this. Here’s the N = 500 guy with 10% outliers, and varying levels of impurity decrease minimums. And same for 15% outliers and triple-class circles guy,

 A diagram of a graph showing a red and blue dotted diagram

Description automatically generated with medium confidence A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated A diagram of a tree

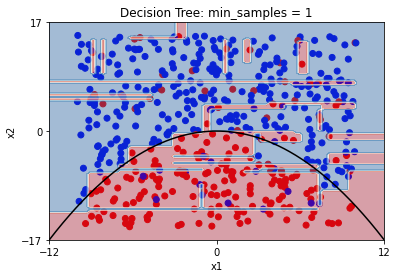
Description automatically generated A diagram of a tree

Description automatically generated

The higher the min is, the less overfitting we get. Can see that there is a Goldilocks zone.

**Hyperparameter: min\_samples\_leaf**

Another thing we can do is set a minimum on the number data points a leaf can contain. Here’s the N = 500 guy with 10% outliers, and varying levels of leaf population minimums, and same for 15% outliers triple class guy,

 A diagram of a tree

Description automatically generated A diagram of a tree

Description automatically generated

A diagram of a tree

Description automatically generated A diagram of a tree

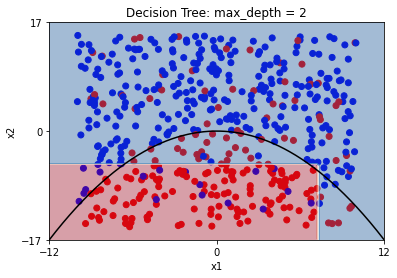
Description automatically generated A diagram of a tree

Description automatically generated

A similar parameter is **min\_samples\_split**, which prescribes how many samples a leaf must have before it can be split.

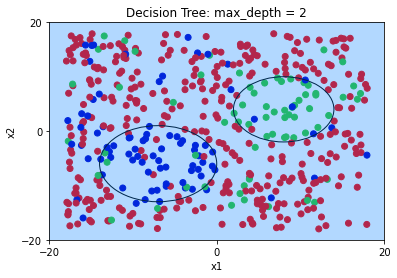
**Hyperparameter: max\_depth**

This guy sets an upper limit on the depth of the tree. Here’s the N = 500 guy with 10% outliers, and varying levels of depths. Same witth N = 500, 15% outliers triple guy,

 A diagram of a graph

Description automatically generated A diagram of a tree

Description automatically generated

 A diagram of a tree

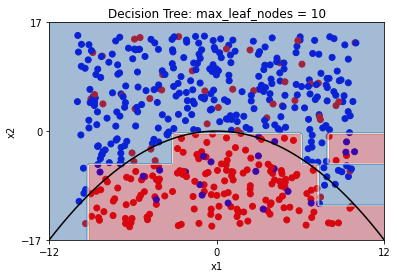
Description automatically generated A diagram of a tree

Description automatically generated

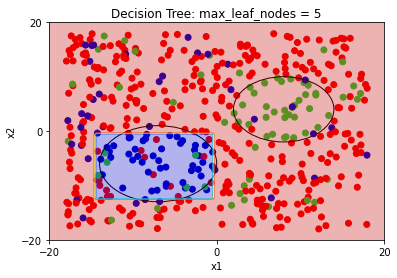
**Hyperparameter: max\_leaf\_nodes**

This guy sets an upper limit on the number of leaf (i.e. terminal) nodes. Here’s the N = 500 guy with 10% outliers, and varying levels of max\_nodes. Same with the tri-class, 15% outliers,

A diagram of a tree

Description automatically generated  A diagram of a tree

Description automatically generated

 A diagram of a tree

Description automatically generated A diagram of a tree

Description automatically generated

So anyway, yeah we can see that decision trees are apt to overfit. But there are ways to circumvent that. And they do often work well, like, again, on the Iris dataset. But they don’t seem to like this data.