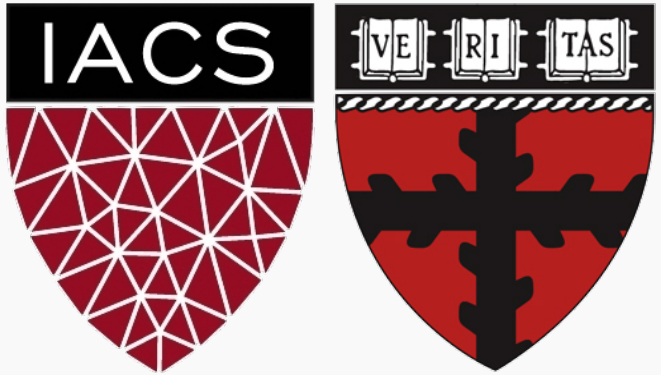
Decision Trees

CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai



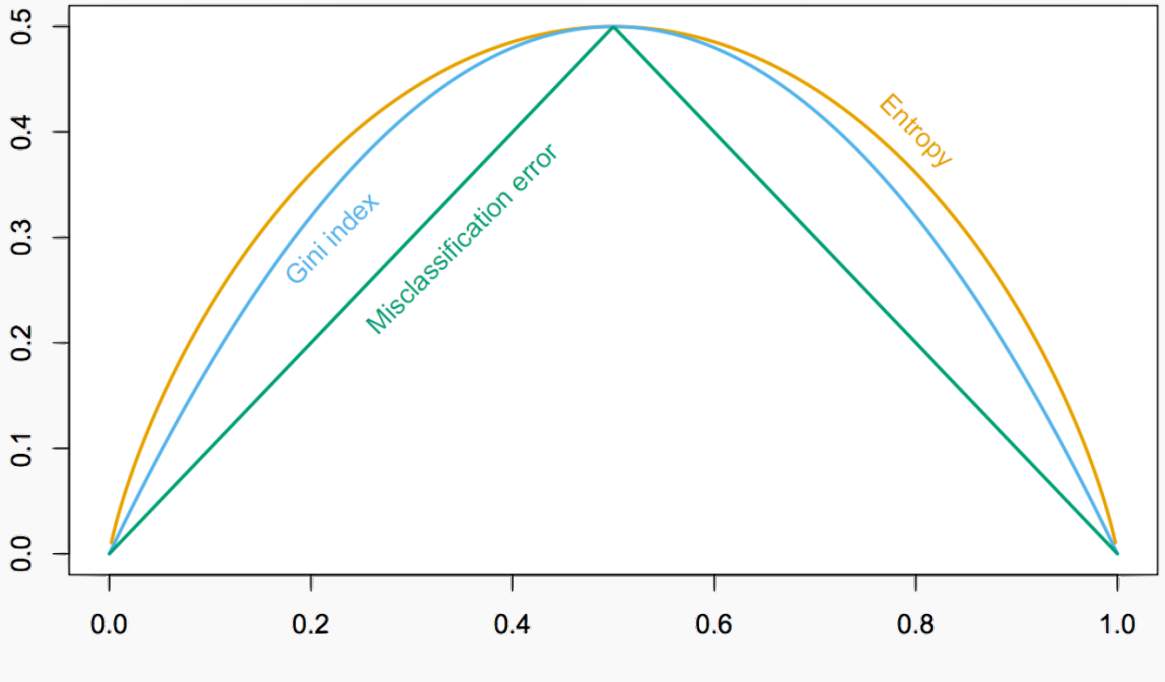
Outline

- Motivation
- Decision Trees
- Classification Trees
- Splitting Criteria
- **Stopping Conditions**
- Regression Trees
- Pruning

Comparison of Criteria

Recall our intuitive guidelines for splitting criteria, which of the three criteria fits our guideline the best?

We have the following comparison of the value of the three criteria at different levels of purity (from 0 to 1) in a single region (for binary outcomes).



Comparison of Criteria

Recall our intuitive guidelines for splitting criteria, which of the three criteria fits our guideline the best?

Note that entropy penalizes impurity the most is not to say that it is the best splitting criteria. For one, a model with purer leaf nodes on a training set may not perform better on the testing test.

Another factor to consider is the size of the tree (i.e. model complexity) each criteria tends to promote.

To compare different decision tree models, we need to first discuss ***stopping conditions***.

Stopping Conditions & Pruning

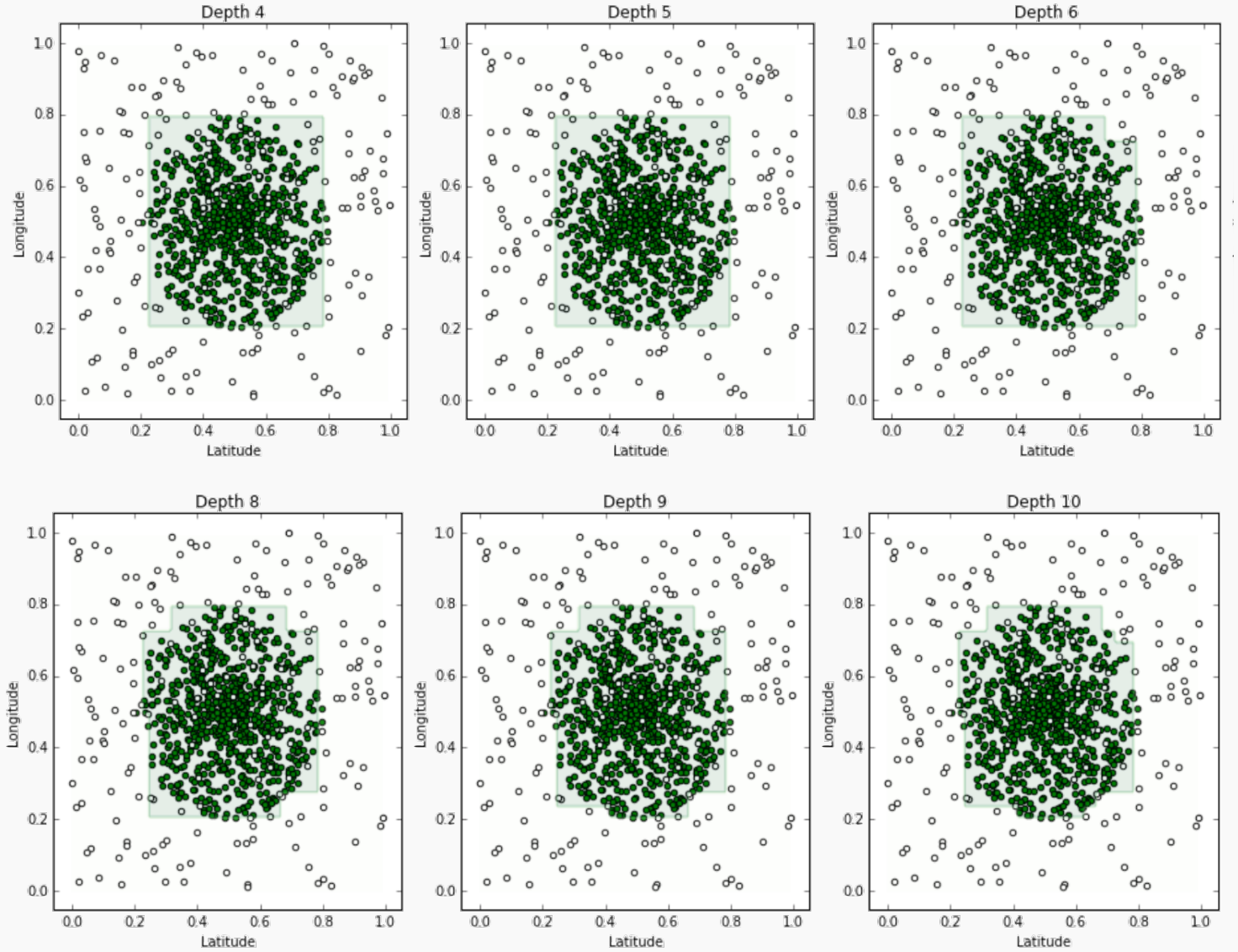
Variance vs Bias

If we don't terminate the decision tree learning algorithm manually, the tree will continue to grow until each region defined by the model possibly contains exactly one training point and the model attains 100% training accuracy (in the training set).

To prevent this from happening, we can simply stop the algorithm at a particular depth.

But how do we determine the appropriate depth?

Variance vs Bias



Variance vs Bias

We make some observations about our models:

- **(High Bias)** A tree of depth 4 is not a good fit for the training data - it's unable to capture the nonlinear boundary separating the two classes.
- **(Low Bias)** With an extremely high depth, we can obtain a model that correctly classifies all points on the boundary (by zig-zagging around each point).
- **(Low Variance)** The tree of depth 4 is robust to slight perturbations in the training data - the square carved out by the model is stable if you move the boundary points a bit.
- **(High Variance)** Trees of high depth are sensitive to perturbations in the training data, especially to changes in the boundary points.

Not surprisingly, complex trees have low bias (able to capture more complex geometry in the data) but high variance (can overfit). Complex trees are also harder to interpret and more computationally expensive to train.

Stopping Conditions

The most common simple stopping condition is to limit the maximum depth of the tree.

The appropriate `max_depth` can be determined by evaluating the model on a validation data set or, better yet, with

cross-validation

Stopping Conditions

Other common simple stopping conditions are.

- Don’t split a region if all instances in the region belong to the same class.
- Don’t split a region if the number of instances in the sub-region will fall below pre-defined threshold (**min_samples_leaf**).
- Don’t split a region if the total number of leaves in the tree will exceed pre-defined threshold.

To

cross-validation

Stopping Conditions

More restrictive stopping conditions:

- Don't split a region if the class distribution of the training points inside the region are independent of the predictors.
- Compute the gain in purity, information or reduction in entropy of splitting a region R into R_1 and R_2:

$$Gain(R) = \Delta(R) = m(R) - \frac{N_1}{N} m(R_1) - \frac{N_2}{N} m(R_2)$$

where m is a metric like the Gini Index or entropy. Don't split if the gain is less than some pre-defined threshold (**min_impurity_decrease**).

Numerical vs Categorical Attributes

Note that the 'compare and branch' method by which we defined classification tree works well for numerical features.

However, if a feature is categorical (with more than two possible values), comparisons like *feature* < *threshold* does not make sense.

Question: How can we handle this?

A simple solution is to encode the values of a categorical feature using numbers and treat this feature like a numerical variable. This is indeed what some computational libraries (e.g. `sklearn`) do, however, this method has drawbacks – as we know by now.

Numerical vs Categorical Attributes

In practice, the effect of our choice of naive encoding of categorical variables are often negligible - models resulting from different choices of encoding will perform comparably.

In cases where you might worry about encoding, there is a more sophisticated way to numerically encode the values of categorical variables so that one can optimize over all possible partitions of the values of the variable.

One-hot-encoding!

This more principled encoding scheme is computationally more expensive but is implemented in a number of computational libraries (e.g. `R's randomForest, H20, XGBoost`).

Outline

- Motivation
- Decision Trees
- Classification Trees
- Splitting Criteria
- Stopping Conditions
- **Regression Trees**
- Pruning

Regression Trees

How can this decision tree approach apply to a ***regression problem*** (quantitative outcome)?

Questions to consider:

- What would be a reasonable loss function?
- How would you determine any splitting criteria?
- How would you perform prediction in each leaf?

Regression Trees Prediction

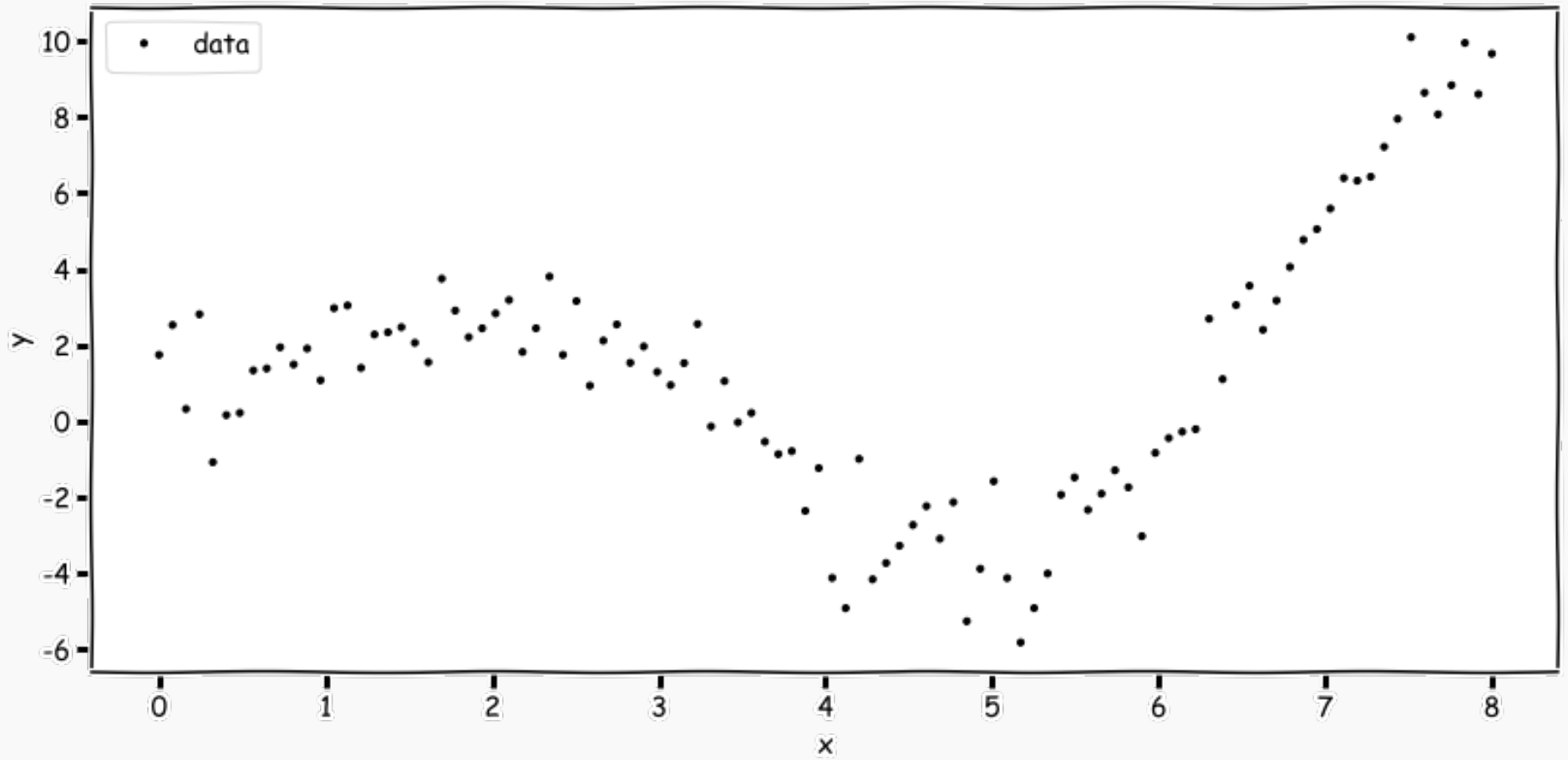
For any data point x_i

1. Traverse the tree until we reach a leaf node.

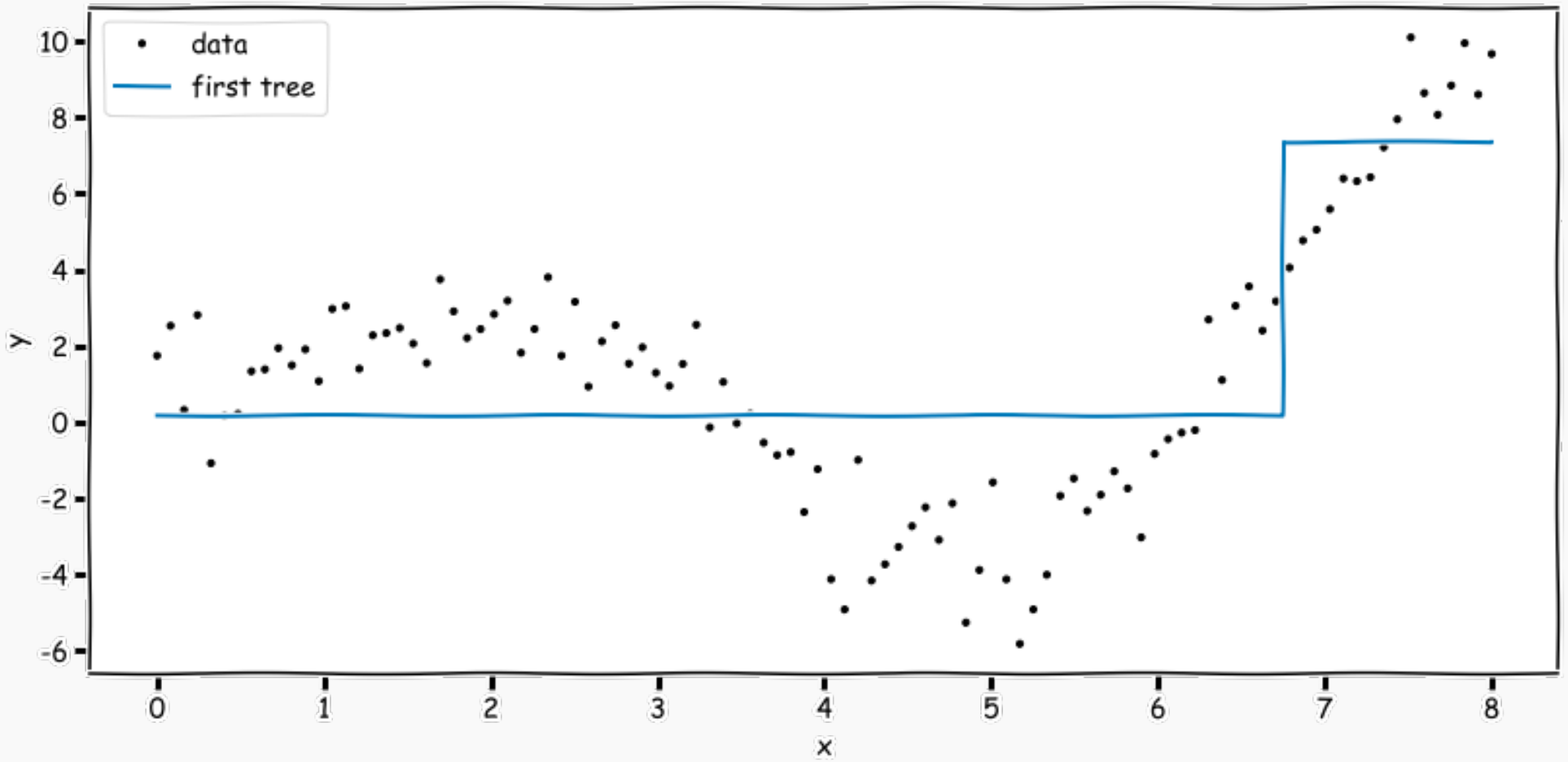
2. Averaged value of the response variable y's in the leaf (this is from the training set) is the $\widehat{y_i}$.

Regression Tree Example

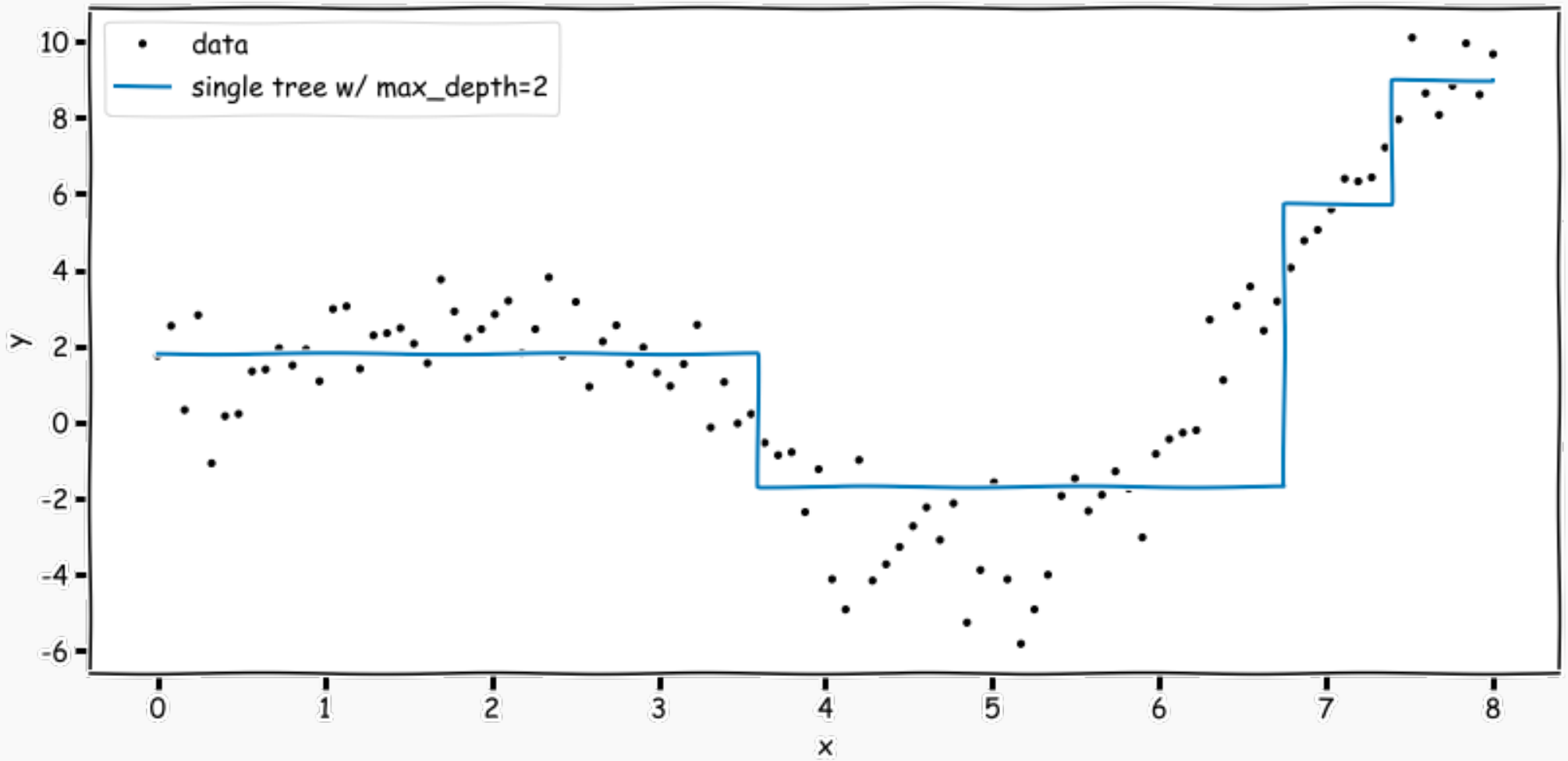
How do we decide a split here?



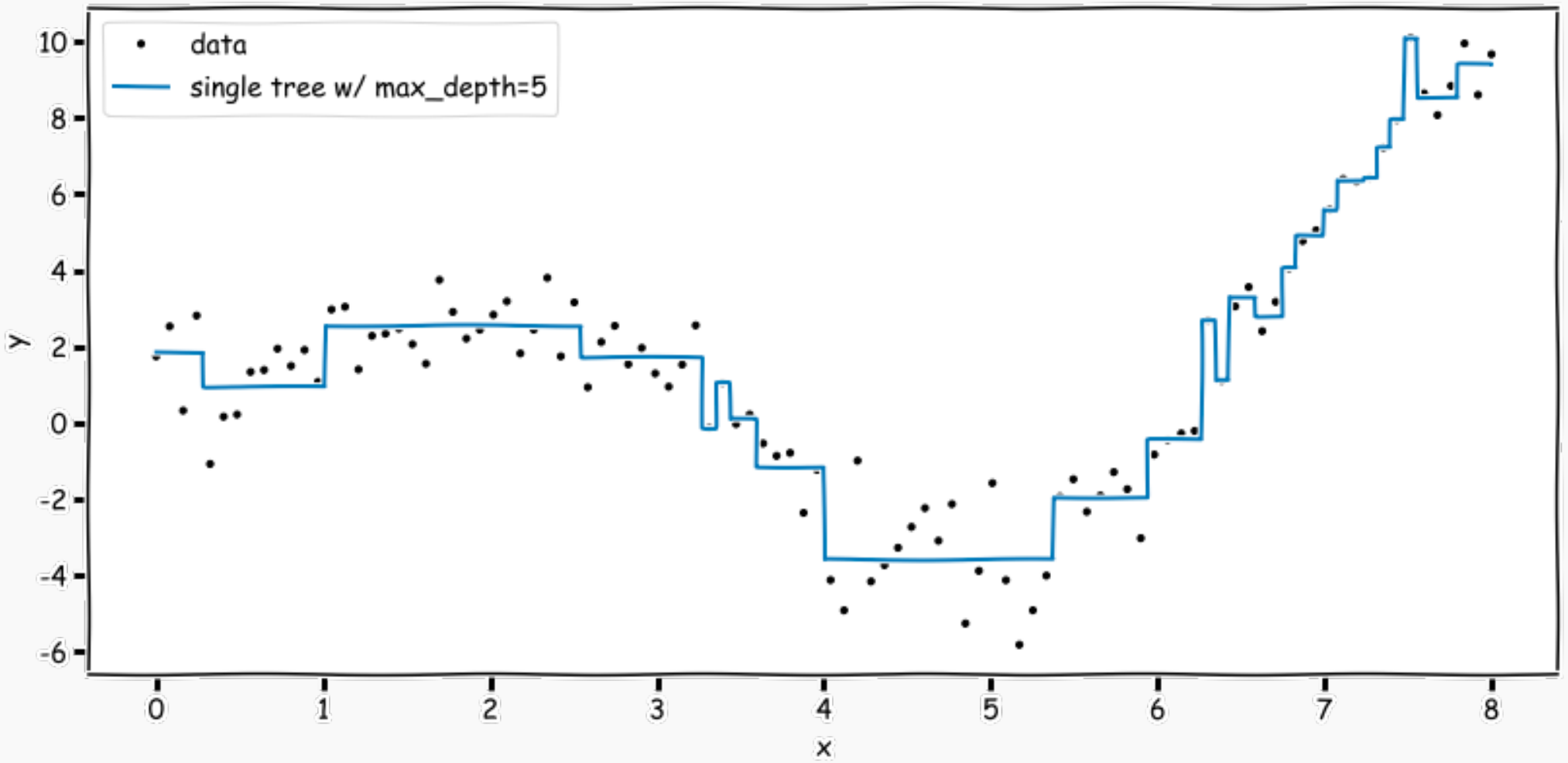
Regression Tree (max_depth = 1)



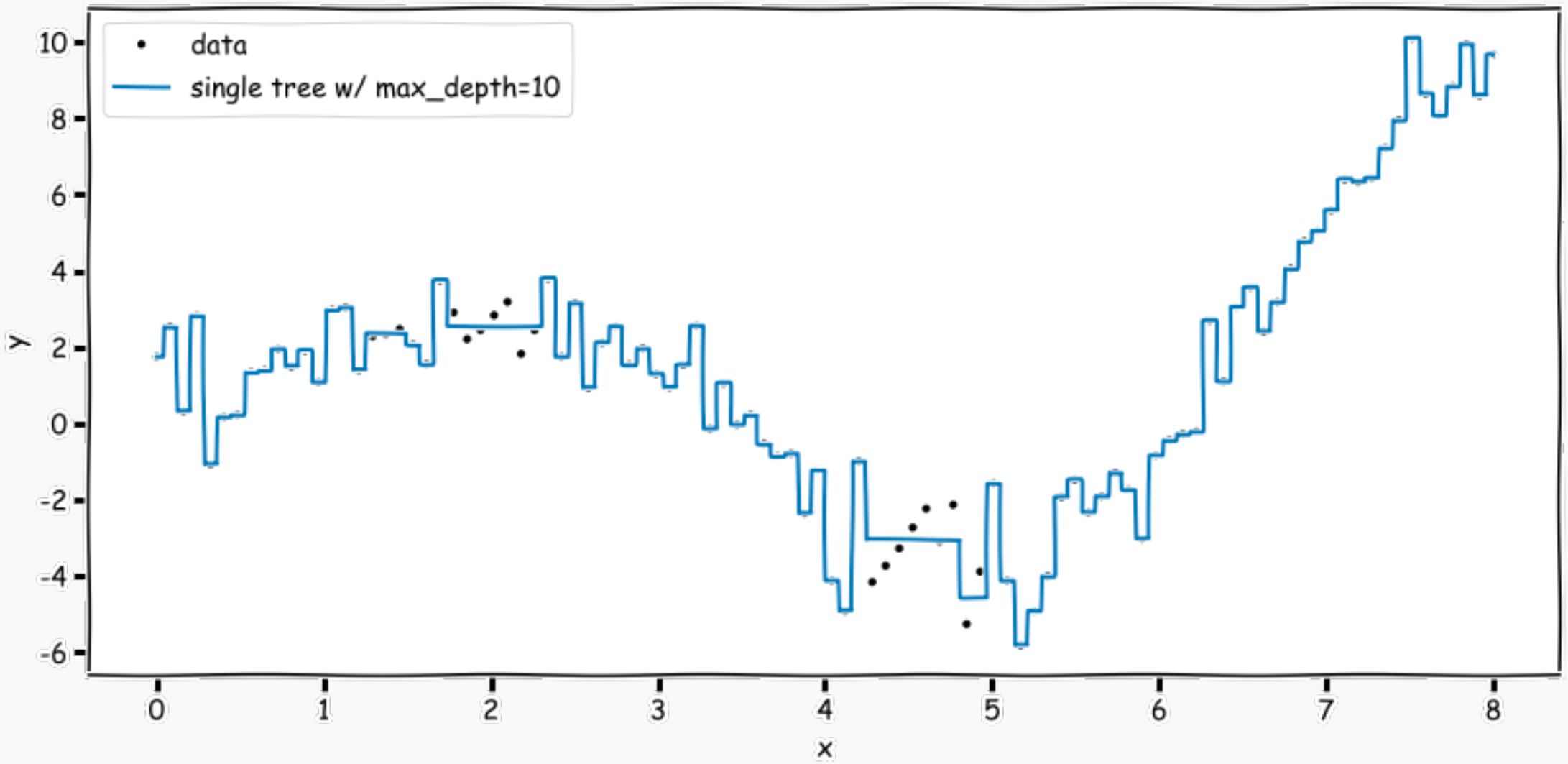
Regression Tree (max_depth = 2)



Regression Tree (max_depth = 5)



Regression Tree (max_depth = 10)



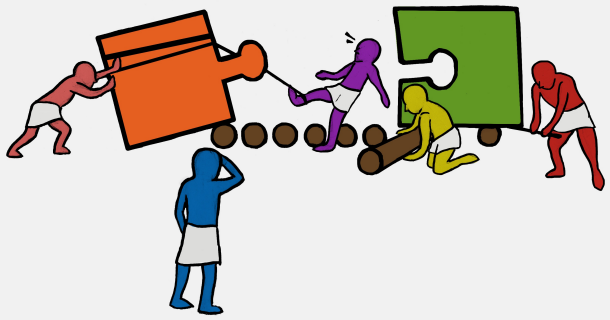
Stopping Conditions

Most of the stopping conditions, like maximum depth or minimum number of points in region, we saw for classification trees can still be applied.

In the place of purity gain, we can instead compute accuracy gain for splitting a region R

$$\text{Gain}(R) = \Delta(R) = MSE(R) - \frac{N_1}{N} MSE(R_1) - \frac{N_2}{N} MSE(R_2)$$

and stop the tree when the gain is less than some pre-defined threshold.



🏋️ Exercise - Classification using Decision Tree

The goal of this exercise is to get comfortable using Decision Trees for classification in `sklearn`. Eventually, you will produce a plot similar to the one given below: