Regression using Decision Trees

CS109OA Introduction to Data Science

Pavlos Protopapas, Kevin Rader, and Chris Gumb

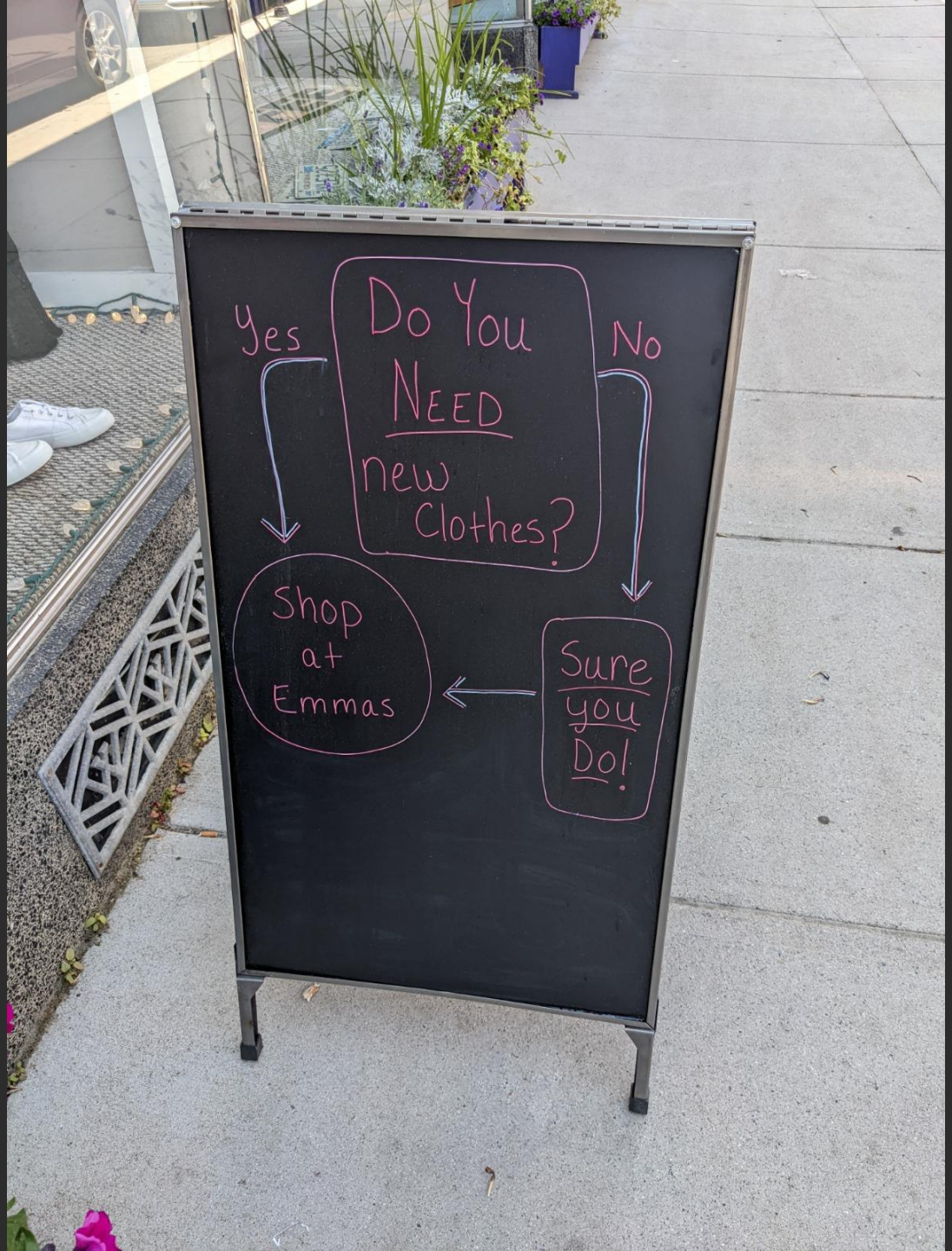
Photo: Eleonore Wen
Dolomite, Italy

Outline

- Decision Trees – Regression
- Numerical vs Categorical Attributes
- Pruning

Outline

- Decision Trees – Regression
- Numerical vs Categorical Attributes
- Pruning


Yes
Do You
NEED
new
Clothes?
No
Shop
at
Emmas
Sure
you
Do!

Regression Trees

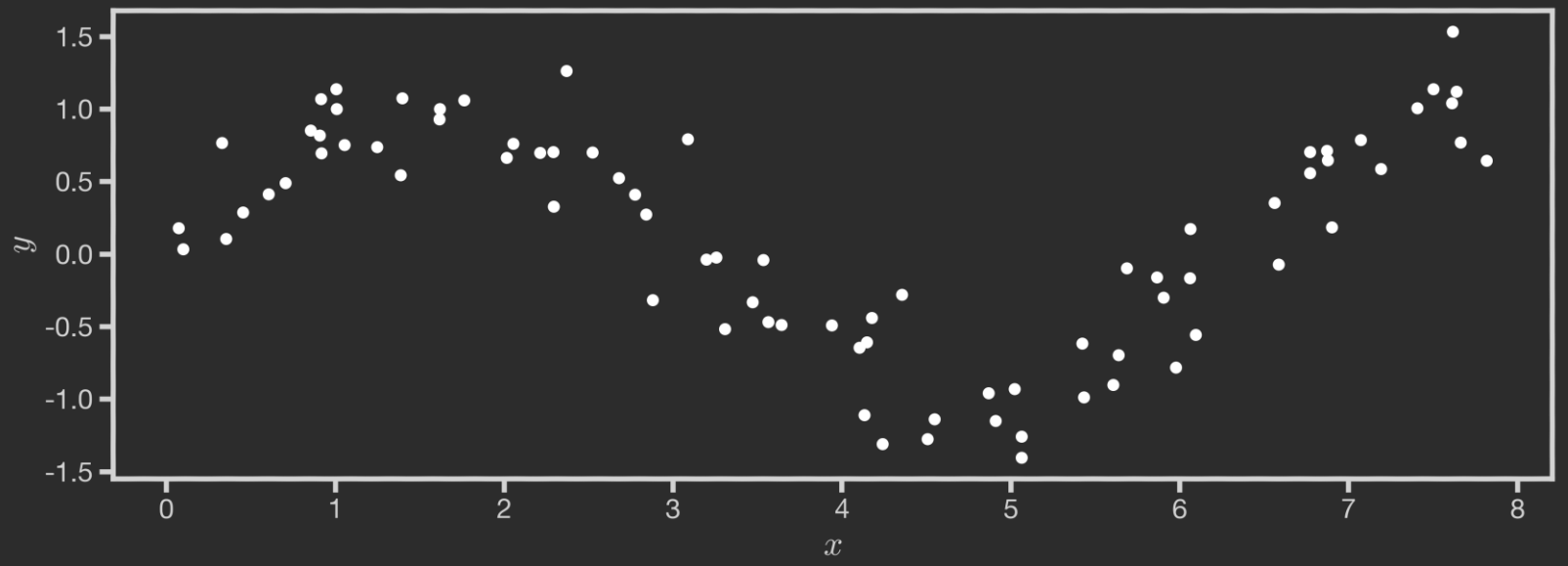
How can this decision tree approach apply to a ***regression problem*** (quantitative outcome)?

Questions to consider:

- How would you determine any **splitting criteria**?
- What would be a reasonable **objective function**?
- How would you perform **prediction at each leaf**?

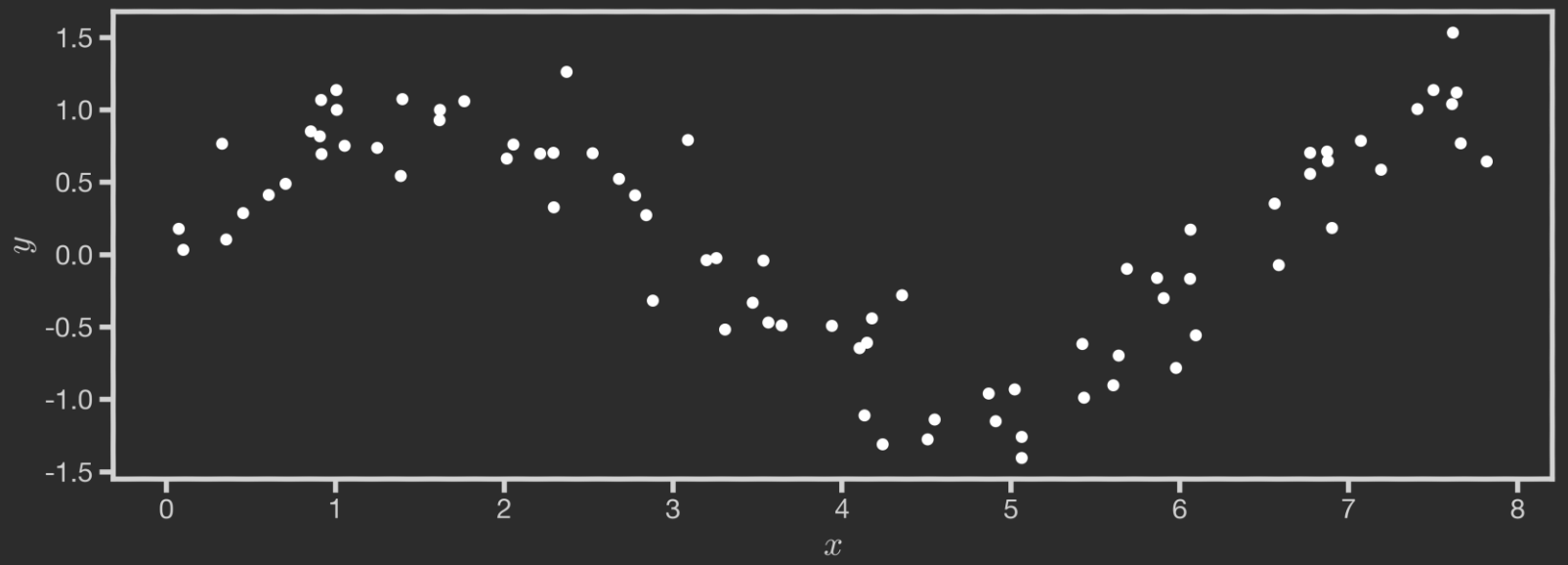
Splitting Criteria

The plot visualizes data points of $\{x, y\}$. By observing patterns, we can identify how and where to make optimal splits. Ideally, each split should lead to groups of data points with homogeneous values, allowing us to better predict based on values within that group.

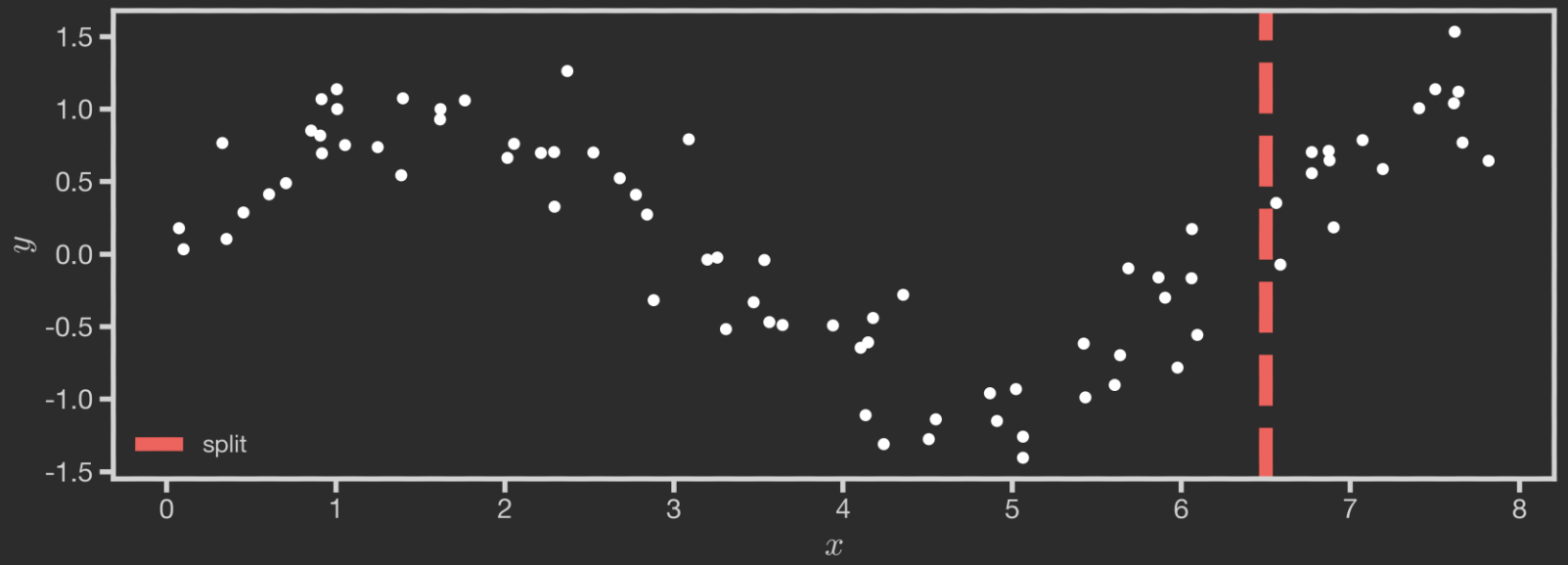


Splitting Criteria

Idea: Divide points into groups of homogeneous values. Predict for each group.

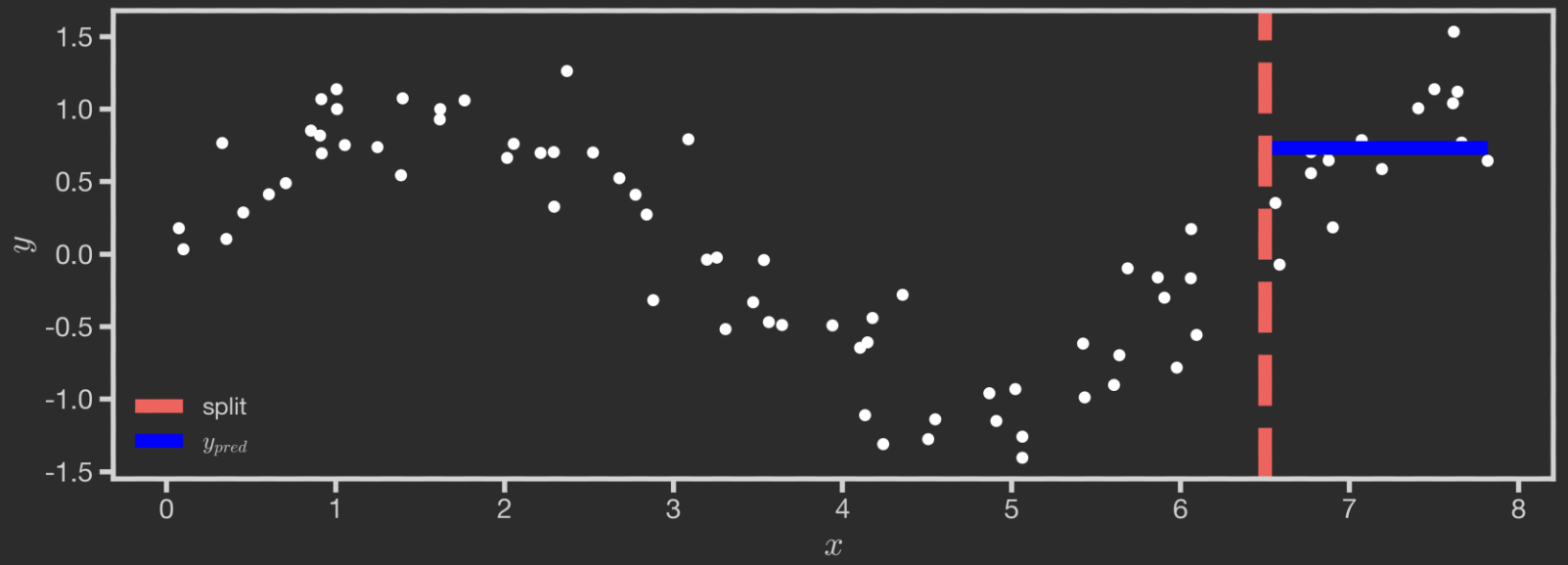


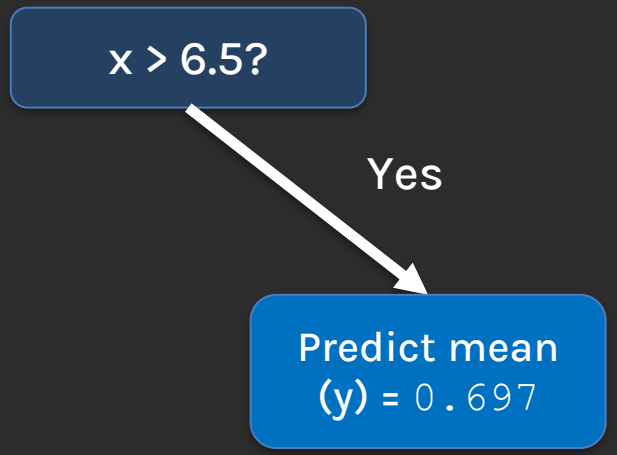
Splitting Criteria



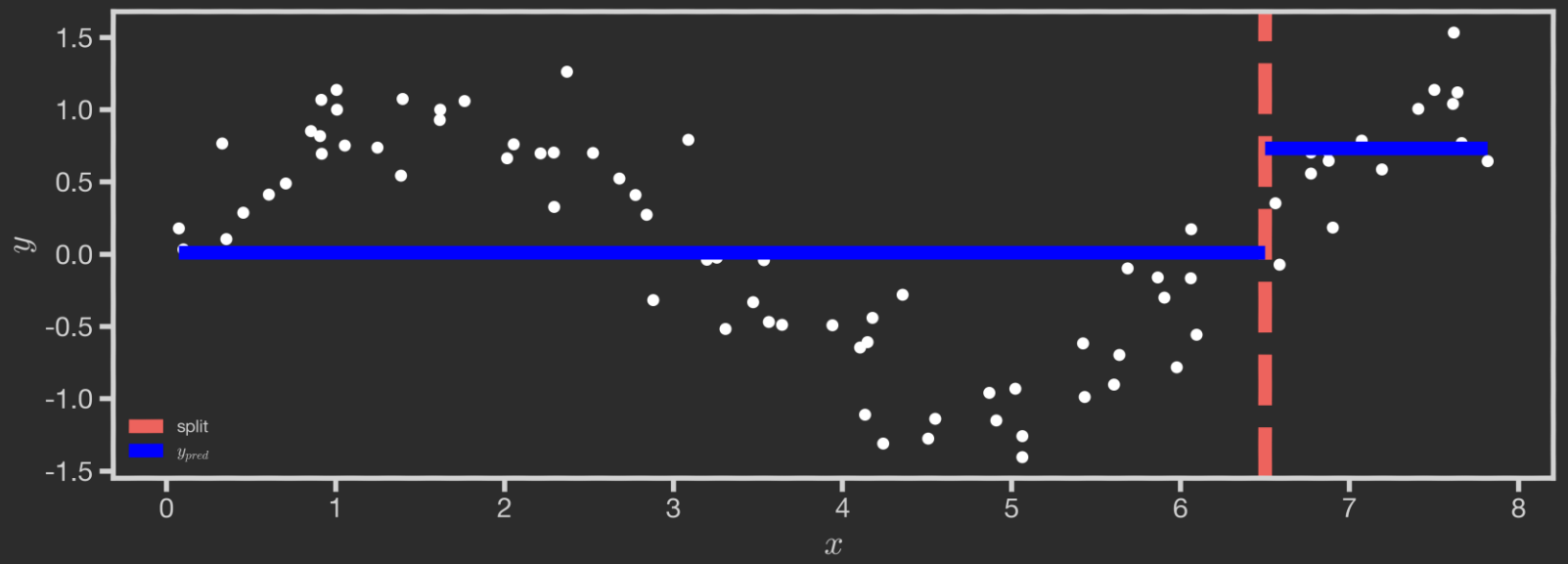
x > 6.5?

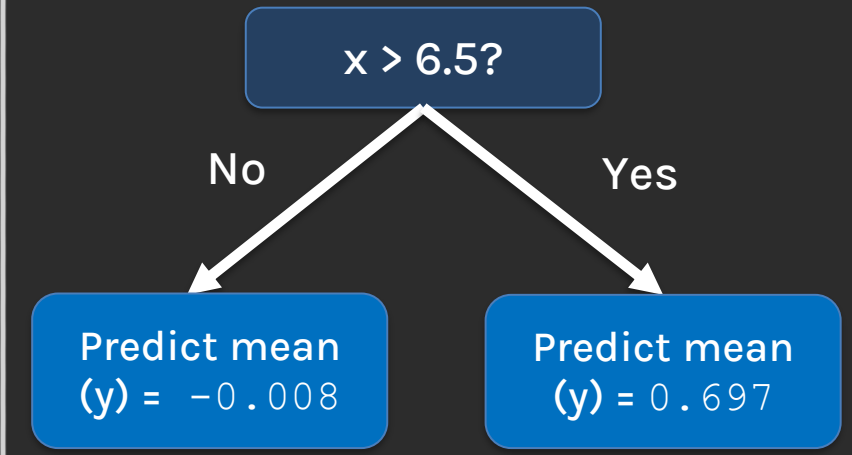
Splitting Criteria





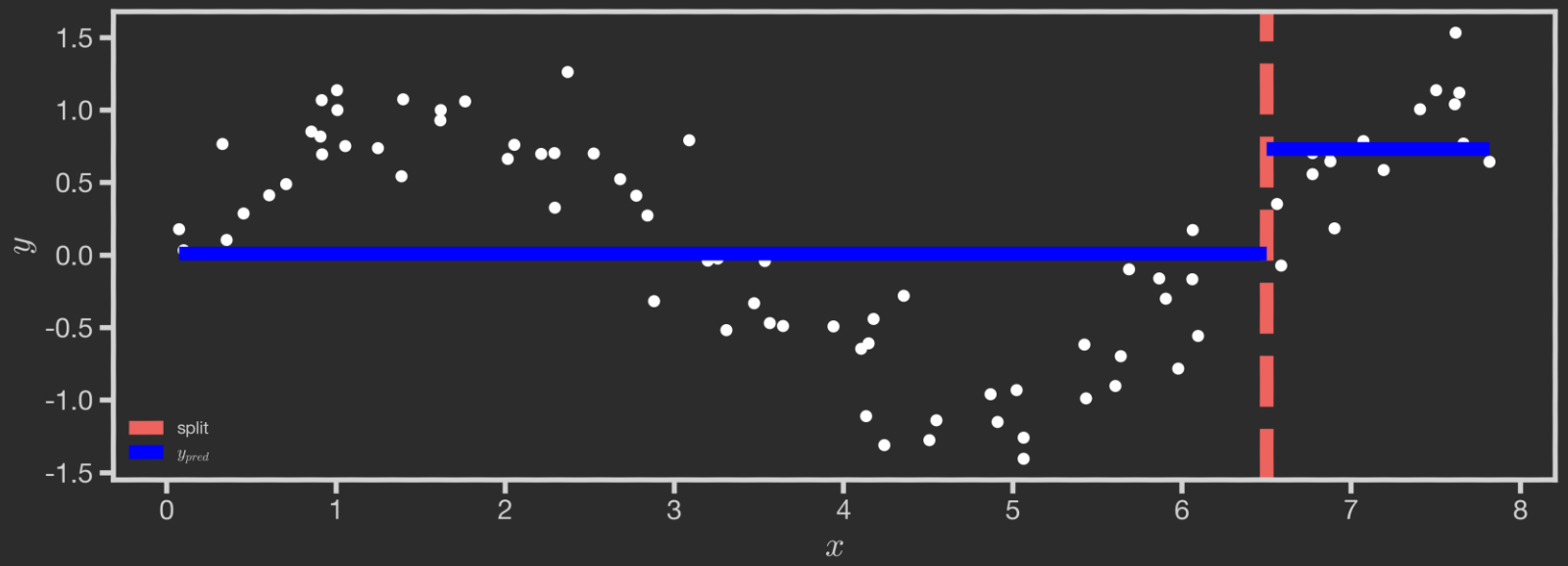
Splitting Criteria

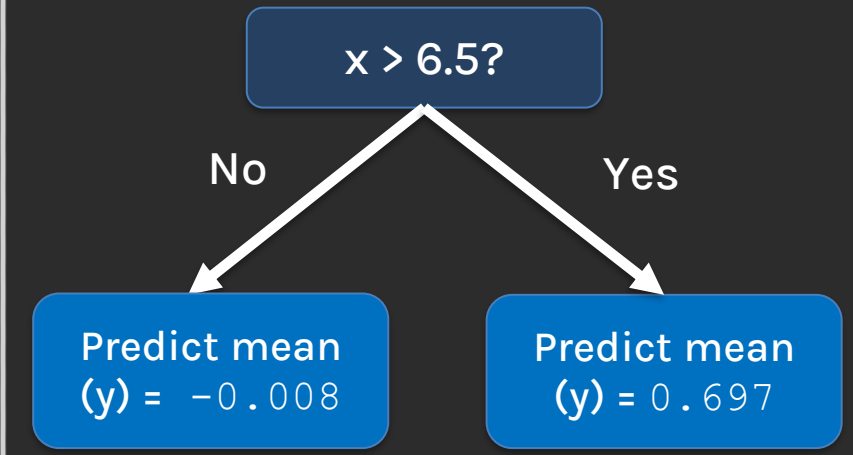




Splitting Criteria

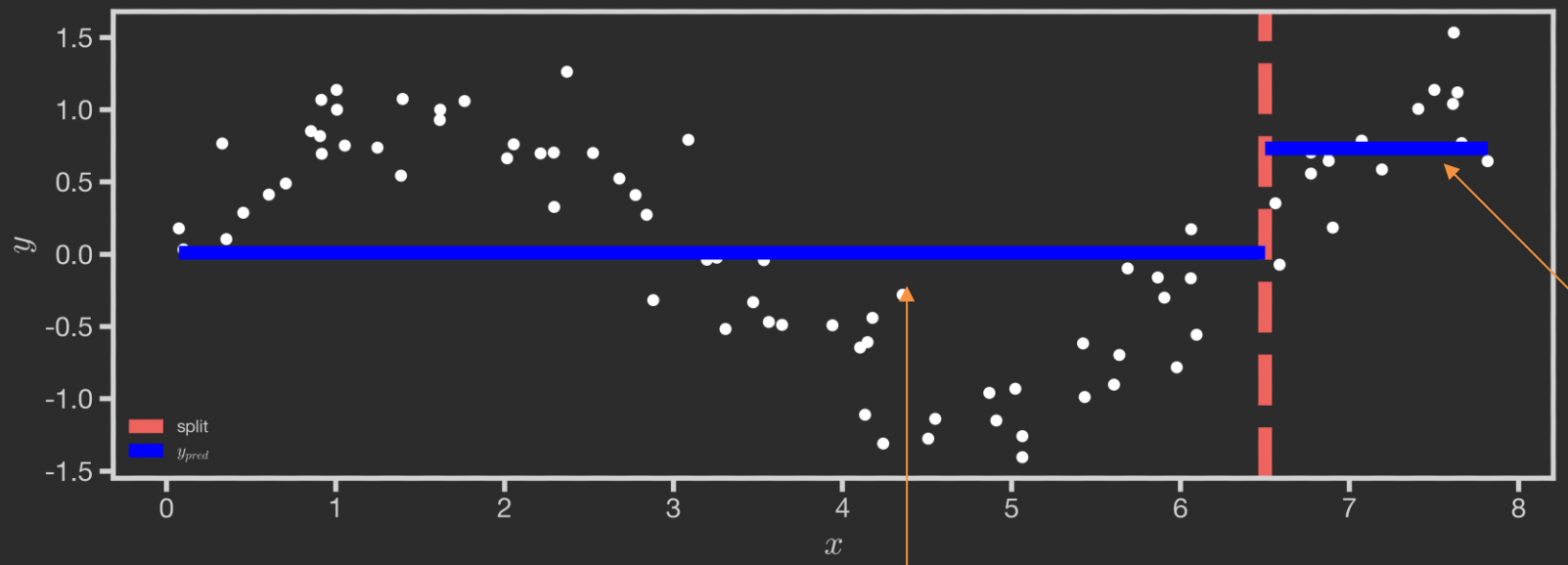
Question: How did we choose the splitting criteria for this regression tree?





Splitting Criteria

We can assess the quality of this split by calculating the mean squared error for each newly created region:

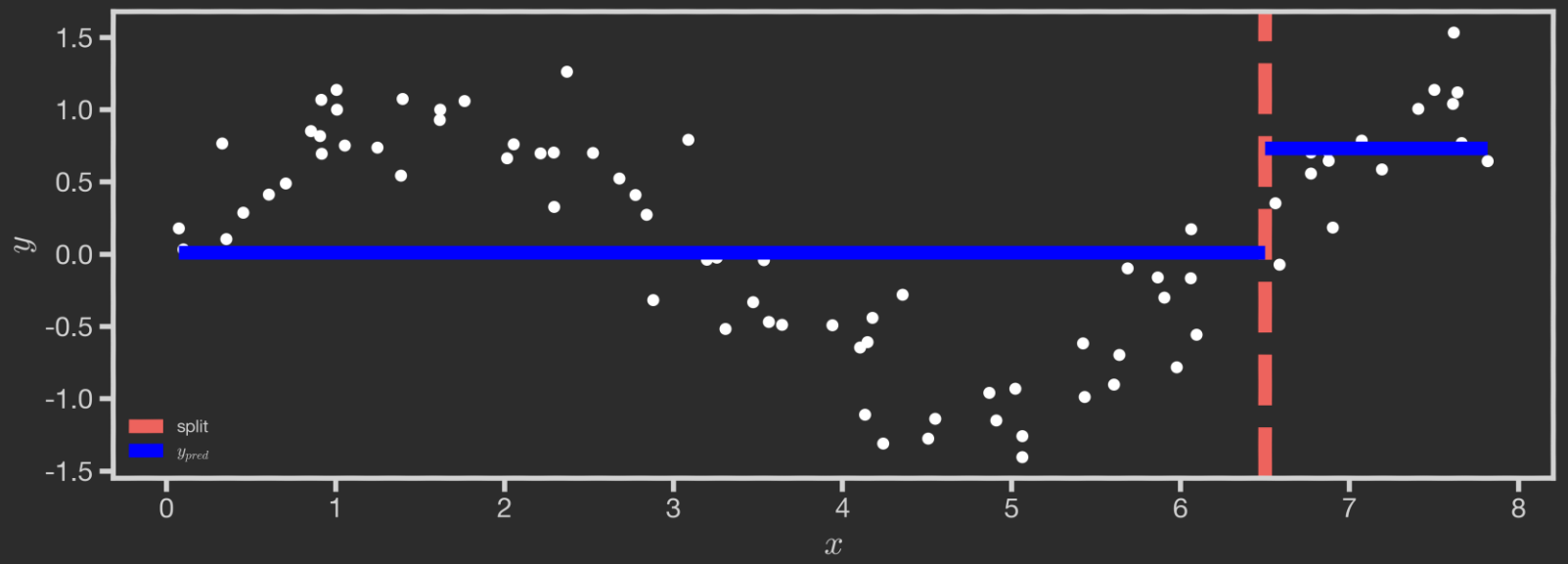


$$MSE(R_1) = \frac{1}{n_1} \sum_{i \in R_1} (y_i - \bar{y}_{R_1})^2$$

$$MSE(R_2) = \frac{1}{n_2} \sum_{i \in R_2} (y_i - \bar{y}_{R_2})^2$$

Splitting Criteria

We can assess the quality of this split by calculating the mean squared error for each newly created region:



This is the prediction

$$MSE(R_r) = \frac{1}{n_r} \sum_{i \in R_r} (y_i - \bar{y}_{R_r})^2$$

Note: This is the "same" as the variance within region R_r!

Splitting Criteria

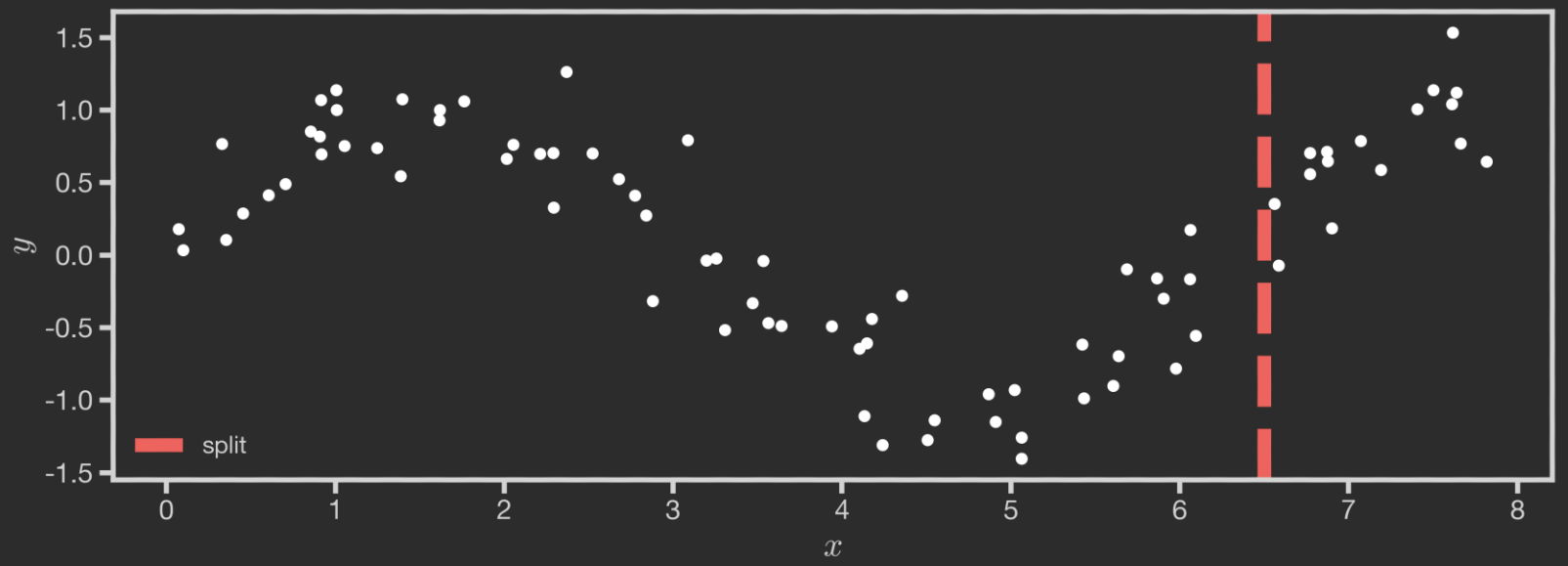
When calculating the MSE, we need to consider the number of points in each region. So, we take the **weighted** average over both regions.

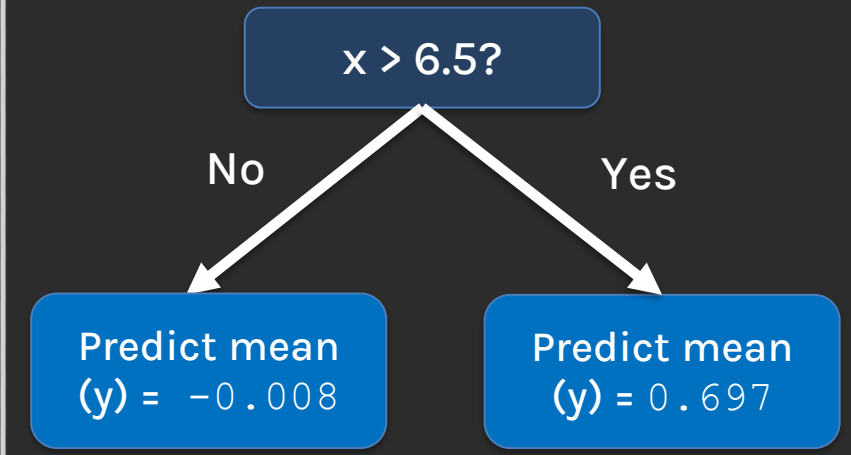
$$\min_{p,\, t_p} \left[\frac{N_1}{N} MSE(R_1) + \frac{N_2}{N} MSE(R_2) \right]$$

Reminder: p denotes the predictor, and t_p denotes the threshold.

Splitting Criteria

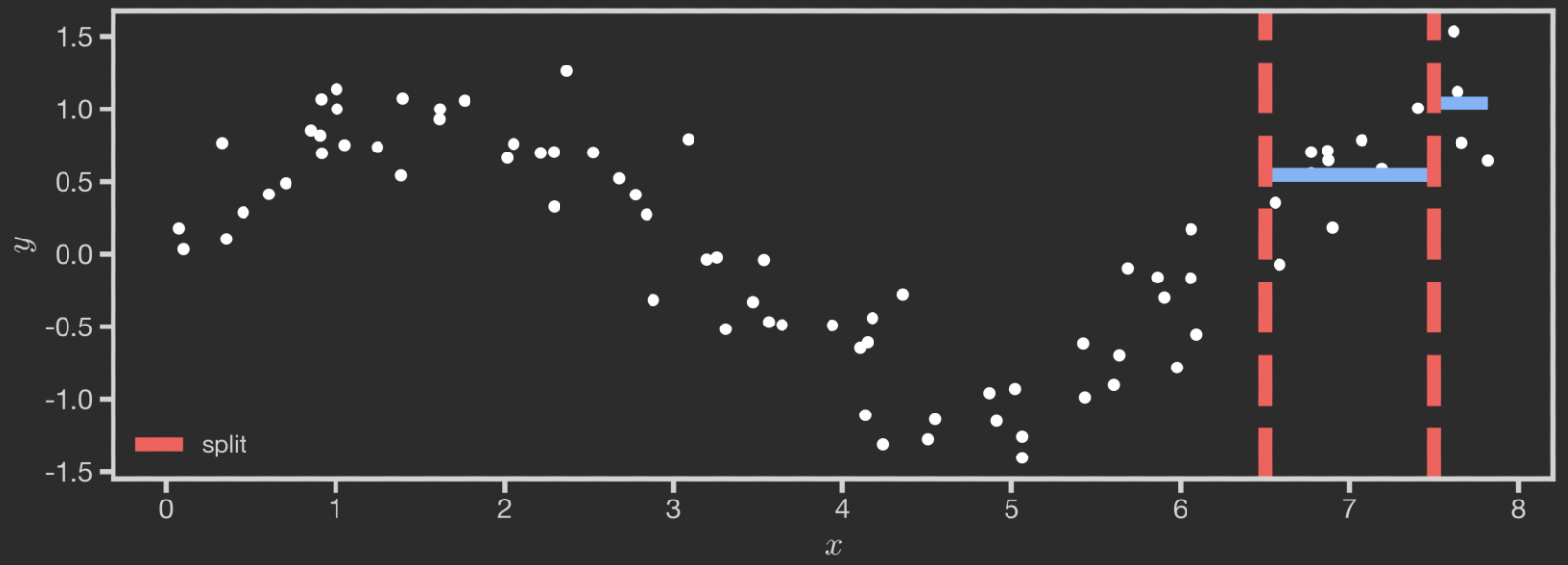
Question: How did we choose the splitting criteria for this regression tree?

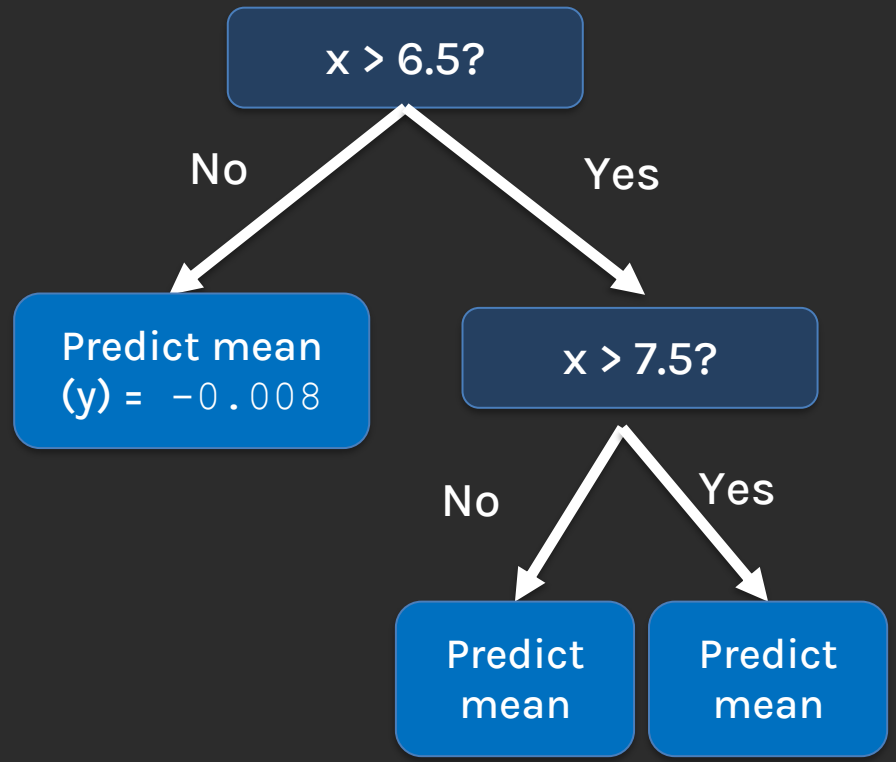




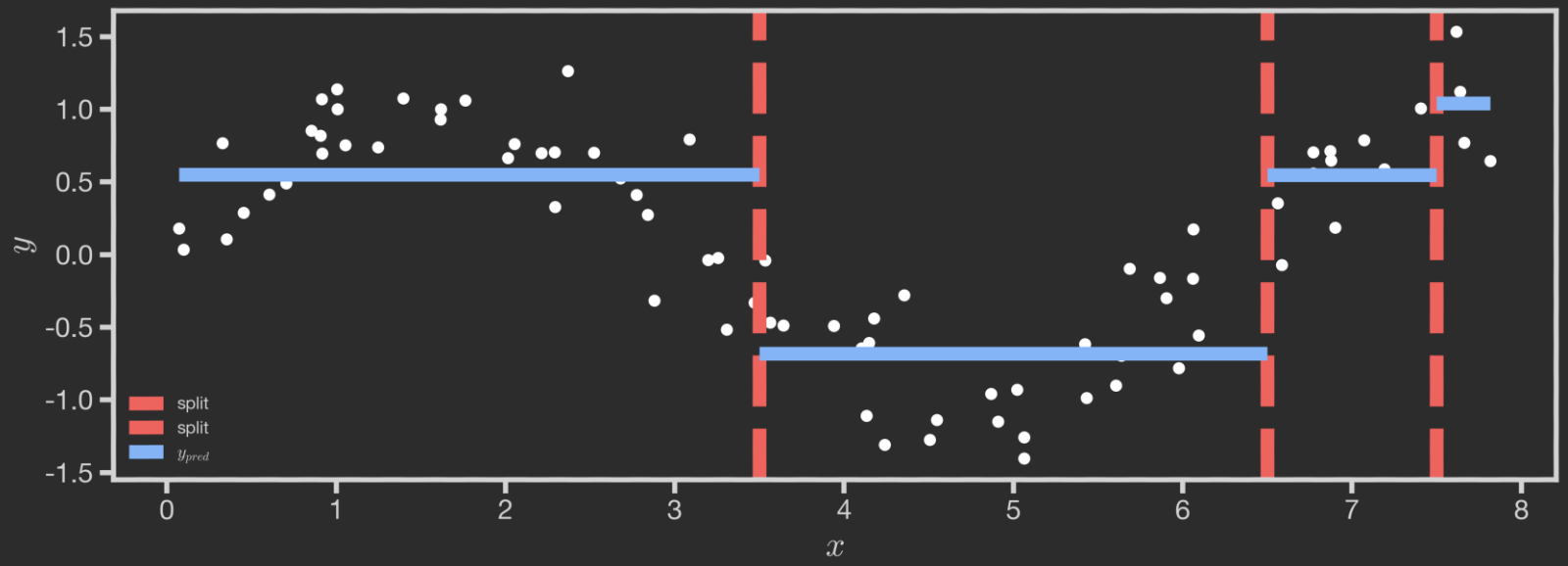
Splitting Criteria

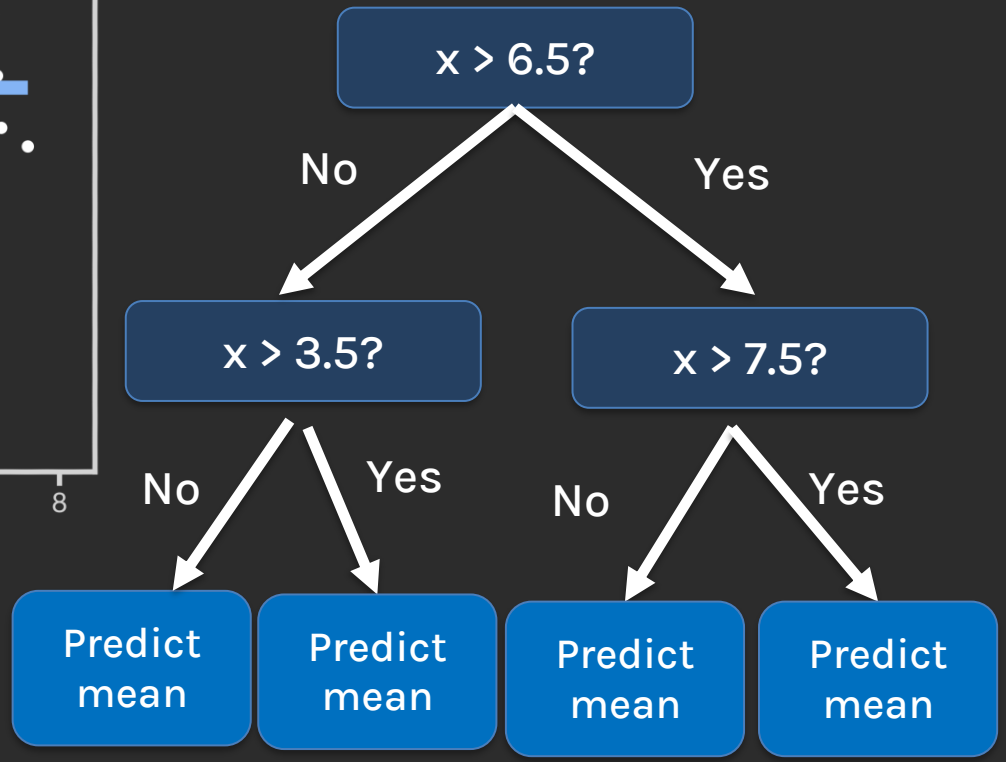
We continue splitting in the same way.



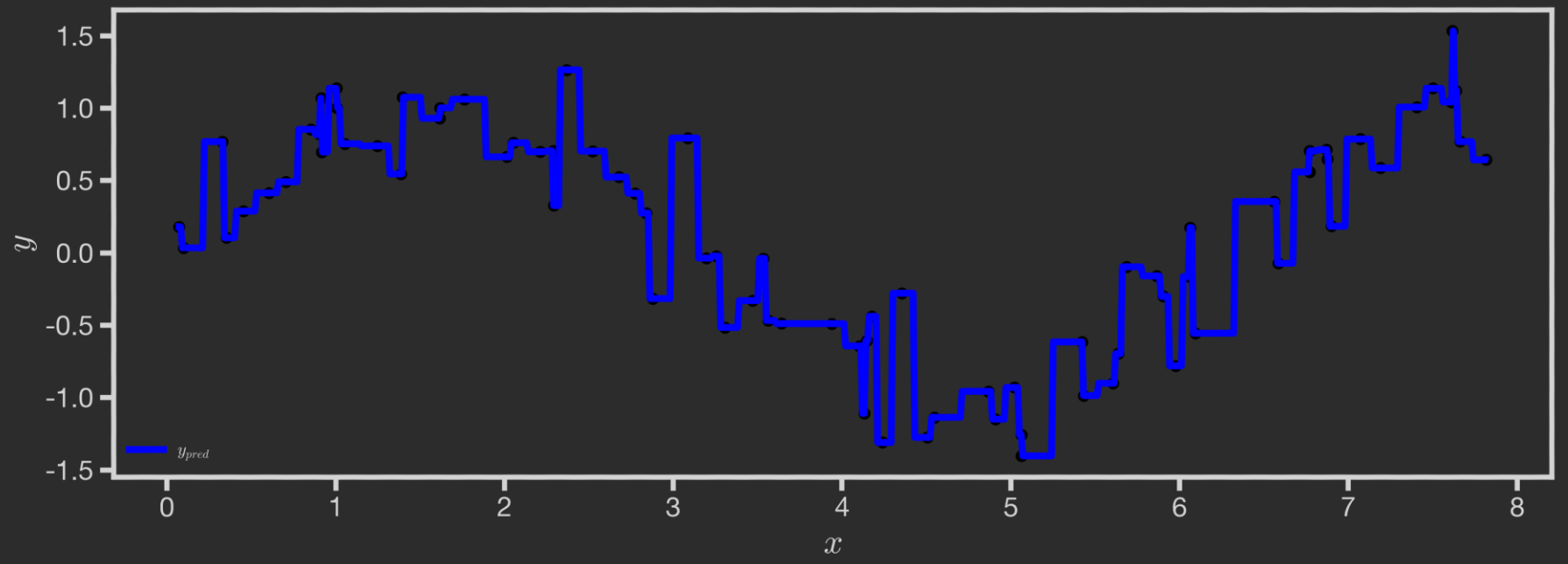


Regression Tree (max_depth = 2)

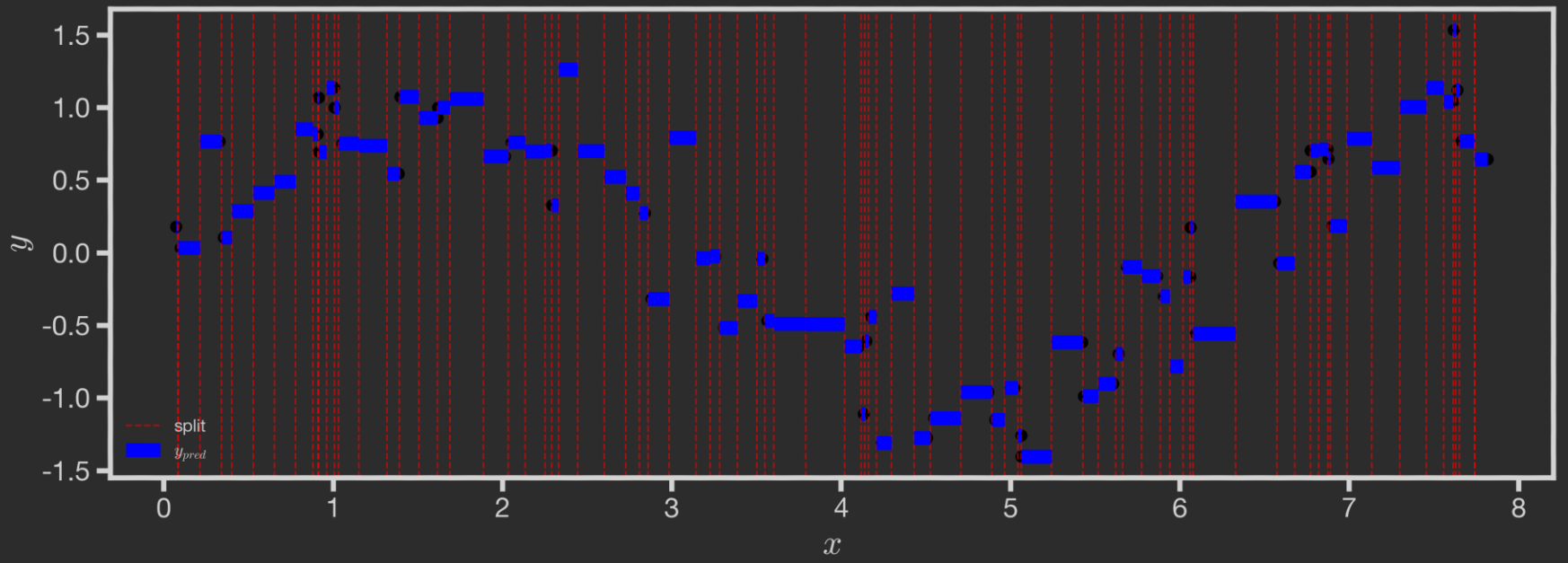




Regression Tree (max_depth = 5)



Regression Tree (max_depth = 10)



Stopping Conditions

Most of the stopping conditions we saw for classification trees, such as **maximum depth** or **minimum number** of points in a region, can also be applied to regression.

Instead of purity gain, we can compute **accuracy gain** **(in this case MSE reduction)** for splitting a region R and stop the tree when the gain is less than some pre-defined threshold.

$$Gain(R) = \Delta(R) = \underbrace{MSE(R)}_{\text{MSE without split}} - \underbrace{\left(\frac{N_1}{N} MSE(R_1) + \frac{N_2}{N} MSE(R_2)\right)}_{\text{MSE after split}}$$

Regression Trees Prediction

For any data point x_i

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

2. Predict $\hat{y}_i$ to be the **averaged value** of the response variable y's in the leaf (this is from the **training set**).

Outline

- Decision Trees – Regression
- Numerical vs Categorical Attributes
- Pruning

Numerical vs Categorical Attributes

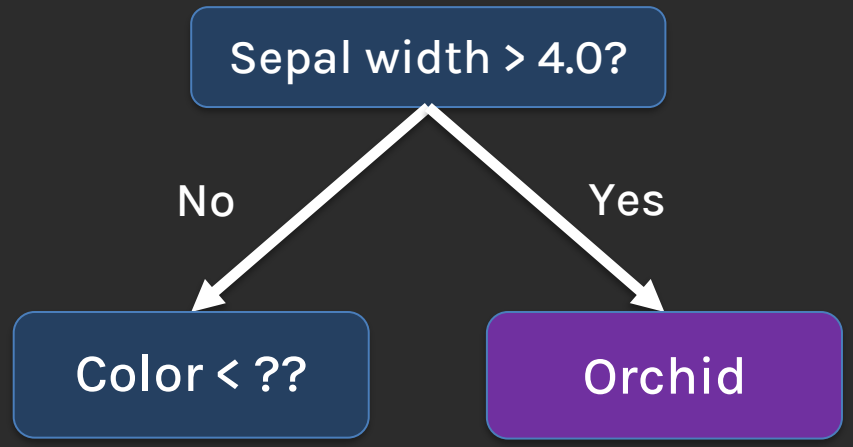
Consider the following data:

Sepal width	Color	Flower
3.0 mm	Yellow	Sunflower
3.5 mm	Red	Rose
3.7 mm	Purple	Tulip
4.5 mm	Purple	Orchid

Question: How do we construct a decision tree for this data?

Numerical vs Categorical Attributes

Sepal width	Color	Flower
3.0 mm	Yellow	Sunflower
3.5 mm	Red	Rose
3.7 mm	Purple	Tulip
4.5 mm	Purple	Orchid



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

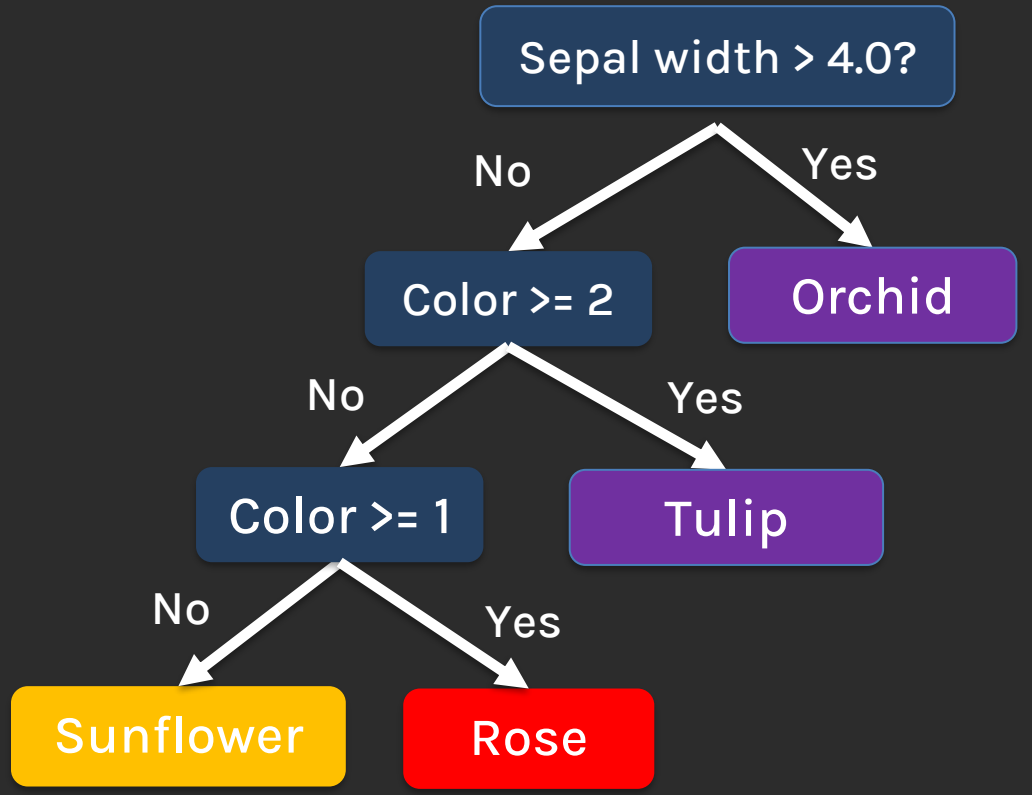
If a feature is categorical (with more than two possible values), a comparison like ***feature < threshold*** does not make sense.

Numerical vs Categorical Attributes

A simple solution is to encode the values of a categorical feature using numbers and treat this feature like a numerical variable.

If we encode **Yellow** = 0, **Red** = 1, **Purple** = 2, our decision tree can be:

Sepal width	Color	Flower
3.0 mm	0	Sunflower
3.5 mm	1	Rose
3.7 mm	2	Tulip
4.5 mm	2	Orchid



Numerical vs Categorical Attributes

In the example, we encoded:

Yellow = 0, **Red** = 1, **Purple** = 2

Then the possible non-trivial splits on **color** are:

{{**Yellow**}, {**Red**, **Purple**}} and {{**Yellow**, **Red**},{**Purple**}}

Color>0

Color>1

But if we encode the categories numerically as:

Yellow = 2, **Red** = 0, **Purple** = 1

The possible splits are:

{{**Red**}, {**Yellow**, **Purple**}} and {{**Red**, **Purple**}, {**Yellow**}}

Color>0

Color>1

Numerical vs Categorical Attributes

{{Yellow}, {Red, Purple}} and {{Yellow, Red},{Purple}}

{{Red}, {Yellow, Purple}} and {{Red, Purple}, {Yellow}}

Depending on the encoding, the splits we optimize over can be different!

Numerical vs Categorical Attributes

In the example, we used **ordinal encoding.** If your categorical data is not ordinal, this is not good. You'll end up with splits that do not make sense. How do we encode this?

One-hot-encoding or dummy encoding!

Numerical vs Categorical Attributes

Sepal width	Color	Flower
3.0 mm	Yellow	Sunflower
3.5 mm	Red	Rose
3.7 mm	Purple	Tulip
4.5 mm	Purple	Orchid

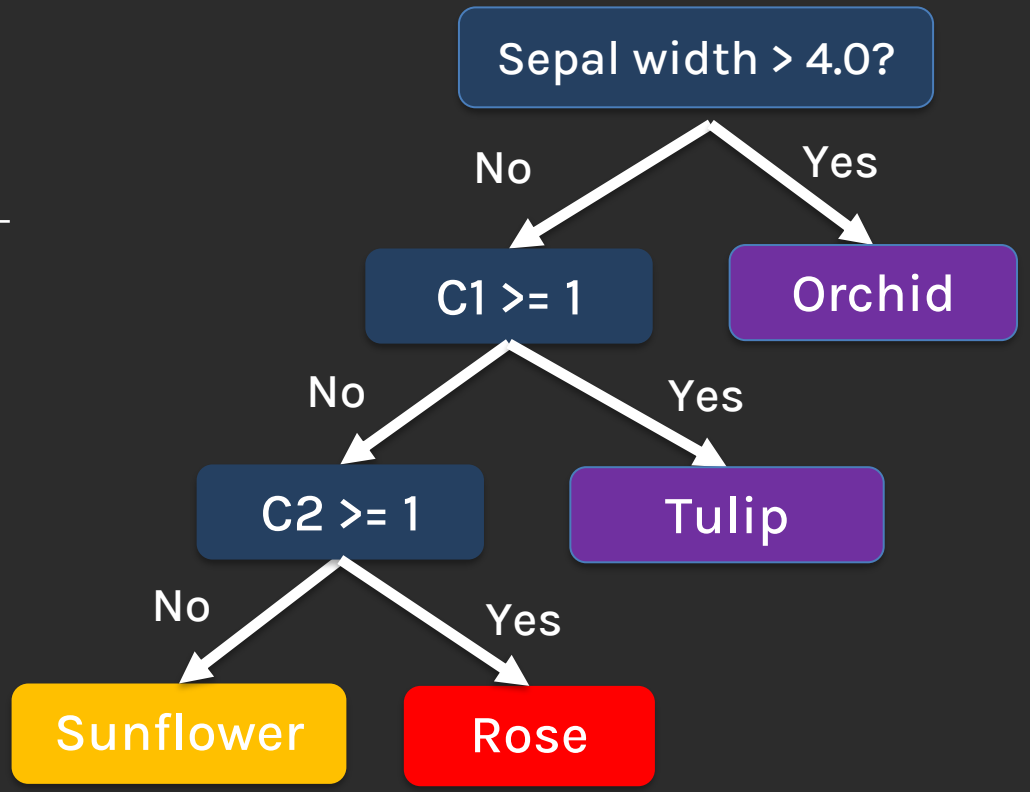
OHE →

Sepal width	C1	C2	C3	Flower
3.0 mm	0	0	1	Sunflower
3.5 mm	0	1	0	Rose
3.7 mm	1	0	0	Tulip
4.5 mm	1	0	0	Orchid

Numerical vs Categorical Attributes

We do not need this!

Sepal width	C1	C2	C3	Flower
3.0 mm	0	0	1	Sunflower
3.5 mm	0	1	0	Rose
3.7 mm	1	0	0	Tulip
4.5 mm	1	0	0	Orchid



Categorical Predictors

As it stands, sklearn decision trees do not handle categorical data.

From sklearn documentation:
scikit-learn uses an optimized version of the CART algorithm; however, scikit-learn implementation does not support categorical variables for now.

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

Summary

How does the prediction process differ between classification trees and regression trees?

Classification trees predict a class label for each leaf node, while regression trees predict the average value of the response variable in the leaf.

Explain how mean squared error (MSE) is used as a splitting criterion in regression trees.

MSE is calculated for each potential split, representing the average squared difference between predicted and actual values in each region. The split with the lowest weighted MSE is chosen.

Describe the concept of “accuracy gain” in the context of regression tree pruning.

Accuracy gain is the reduction in MSE from a split. If the gain is below a threshold, the split is skipped, effectively pruning the tree.

Summary

Why is a simple numerical encoding of categorical features often problematic for decision trees?

Numerical encoding can impose an artificial order on categories, leading to illogical splits, especially for non-ordinal data.

Compare and contrast ordinal encoding and one-hot encoding for categorical features in decision tree models.

Ordinal encoding assigns integers based on order, which can be useful for ordinal data. One-hot encoding creates binary features for each category, avoiding implied order but increasing dimensionality.

Explain why a comparison like “feature < threshold” is not applicable to categorical features.

Categorical features lack natural numerical ordering, so threshold comparisons don’t make sense for them.

Why is the greedy algorithm approach justified for decision tree construction despite not guaranteeing globally optimal solutions?

A feature can be used twice in the same decision tree at different levels of the tree.

When making a prediction for a new data point using a trained regression tree, what is the final step in the prediction process?

A regression tree produces four terminal regions:

Region # of Points MSE

R1 90 0.2

R2 5 1.2

R3 3 1.5

R4 2 1.8