Example: Predicting the Apartment Price

- Add intervals of the features:

$$a(x) = w_0 + w_1 * [30 < area < 50] + w_2 * [50 < area < 80]$$
$$+ w_{20} * [2 < floor < 5] + \cdots$$
$$+ w_{100} * [30 < area < 50][2 < floor < 5] + \cdots$$

- It is easier to interpret features:

$$[30 < area < 50][2 < floor < 5]$$

- But we will have even more features!

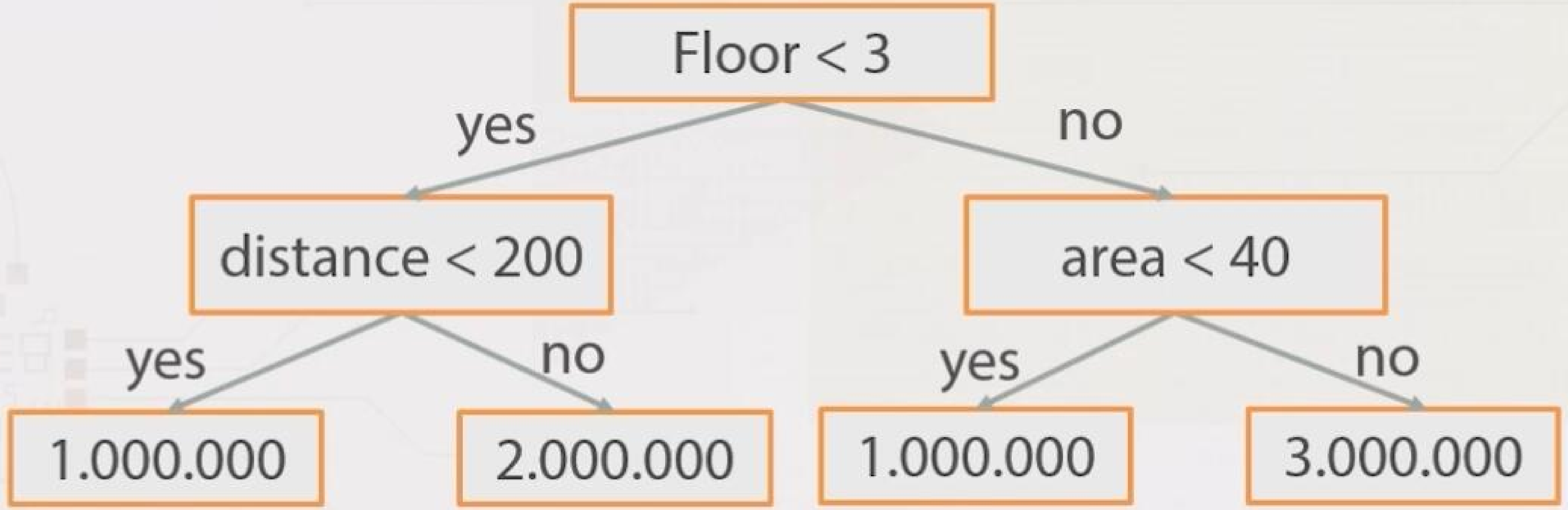
Logical Rule

$$[30 < area < 50][2 < floor < 5][500 < dist. < 1000]$$

- Easy to explain
- Find non-linear dependencies

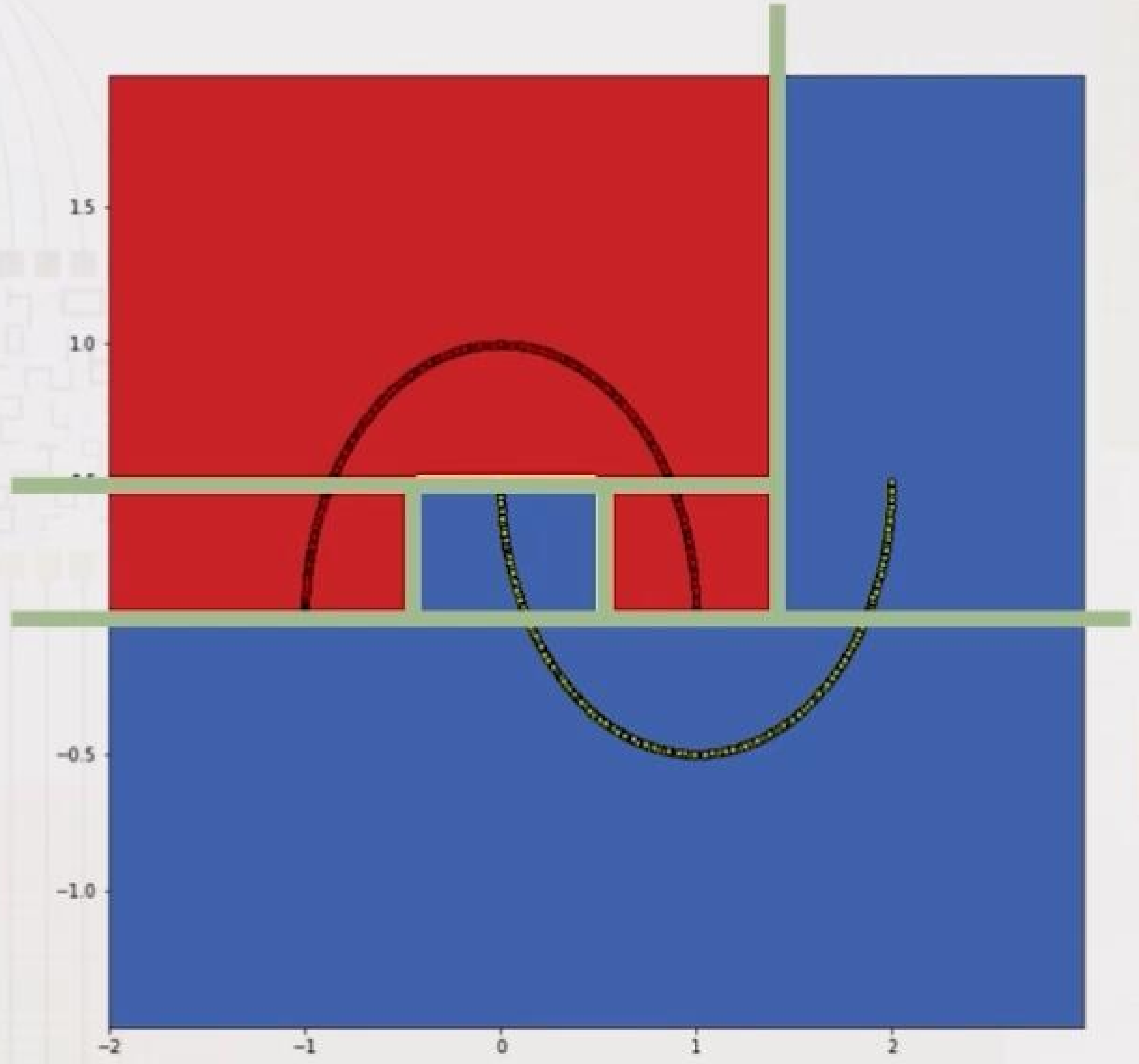
- We need to find good logical rules
- We need to build models out of them

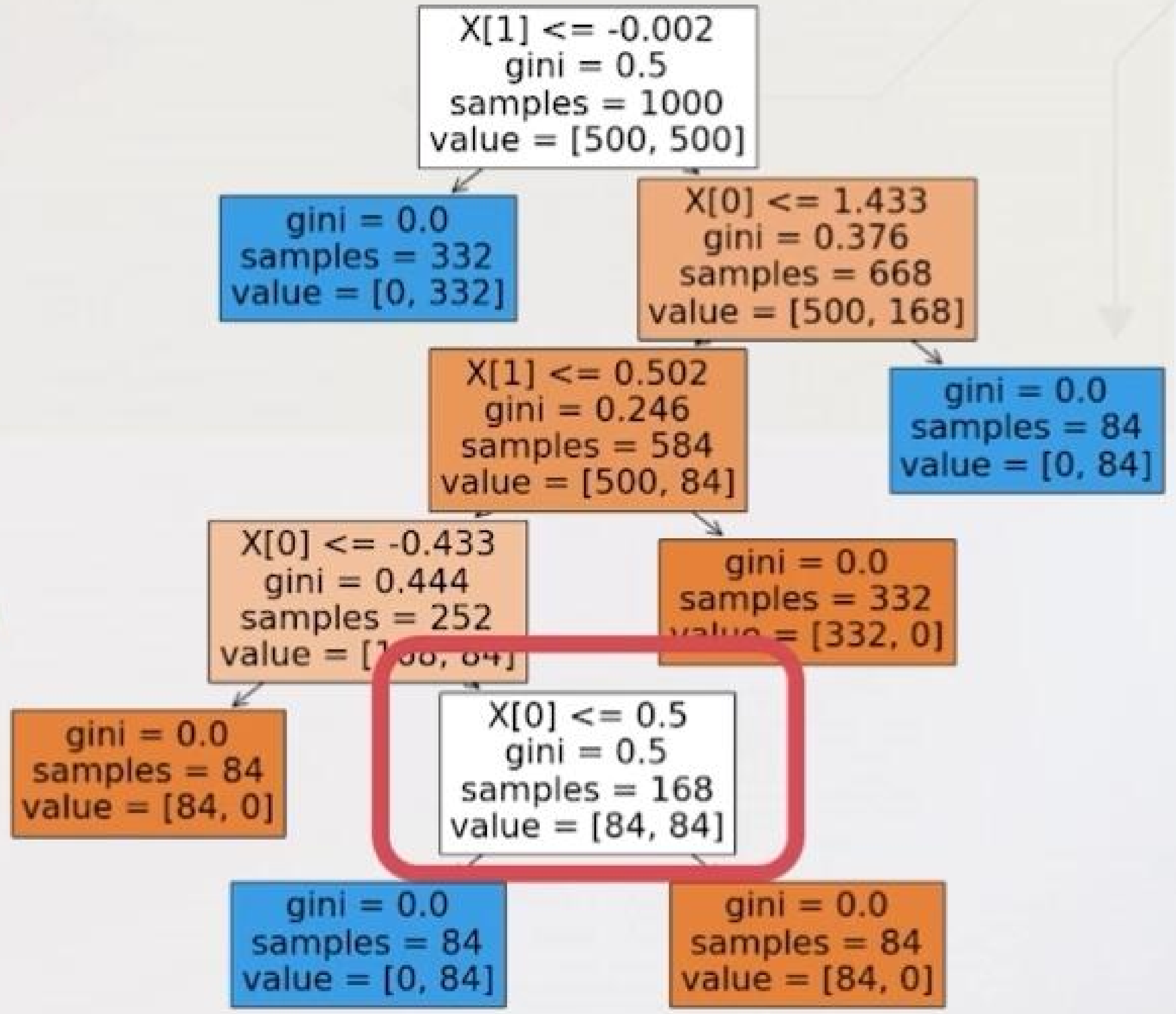
Decision Trees



- **Internal Nodes:** splitting criterion $[x_j < t]$
- **Leaves:** predictions $c \in \mathbb{Y}$

Decision Trees

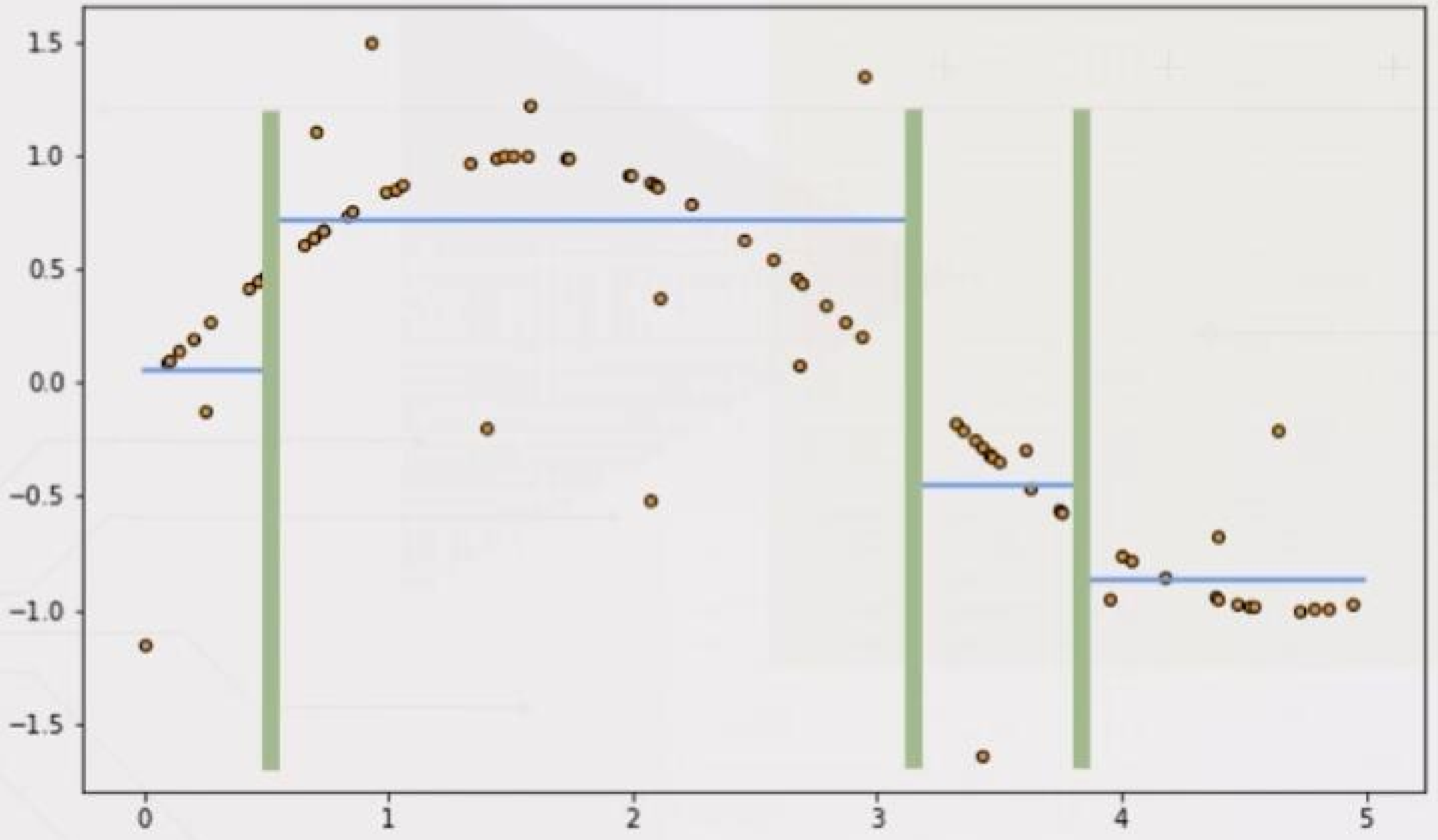


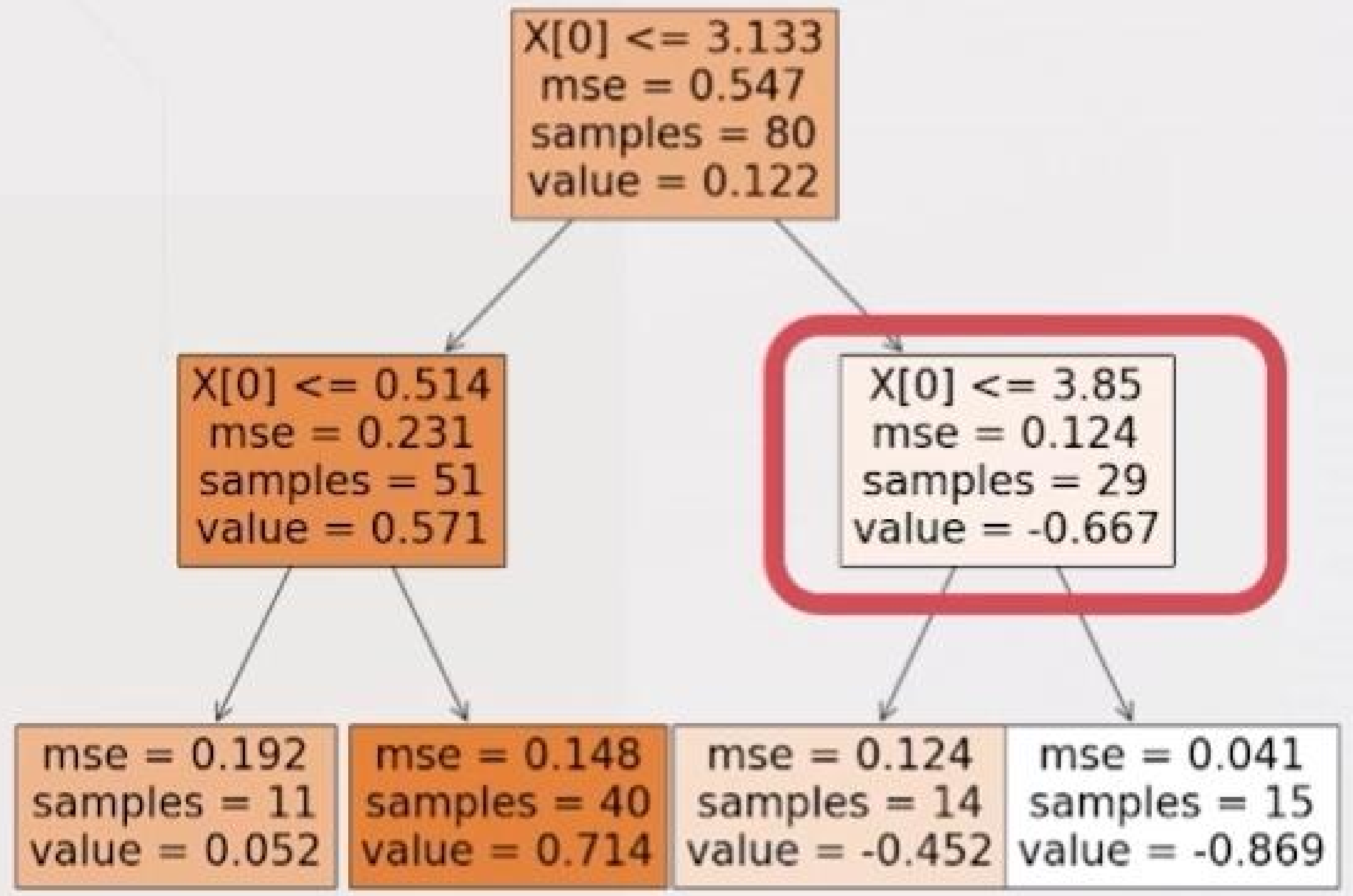


Complexity of Decision Trees

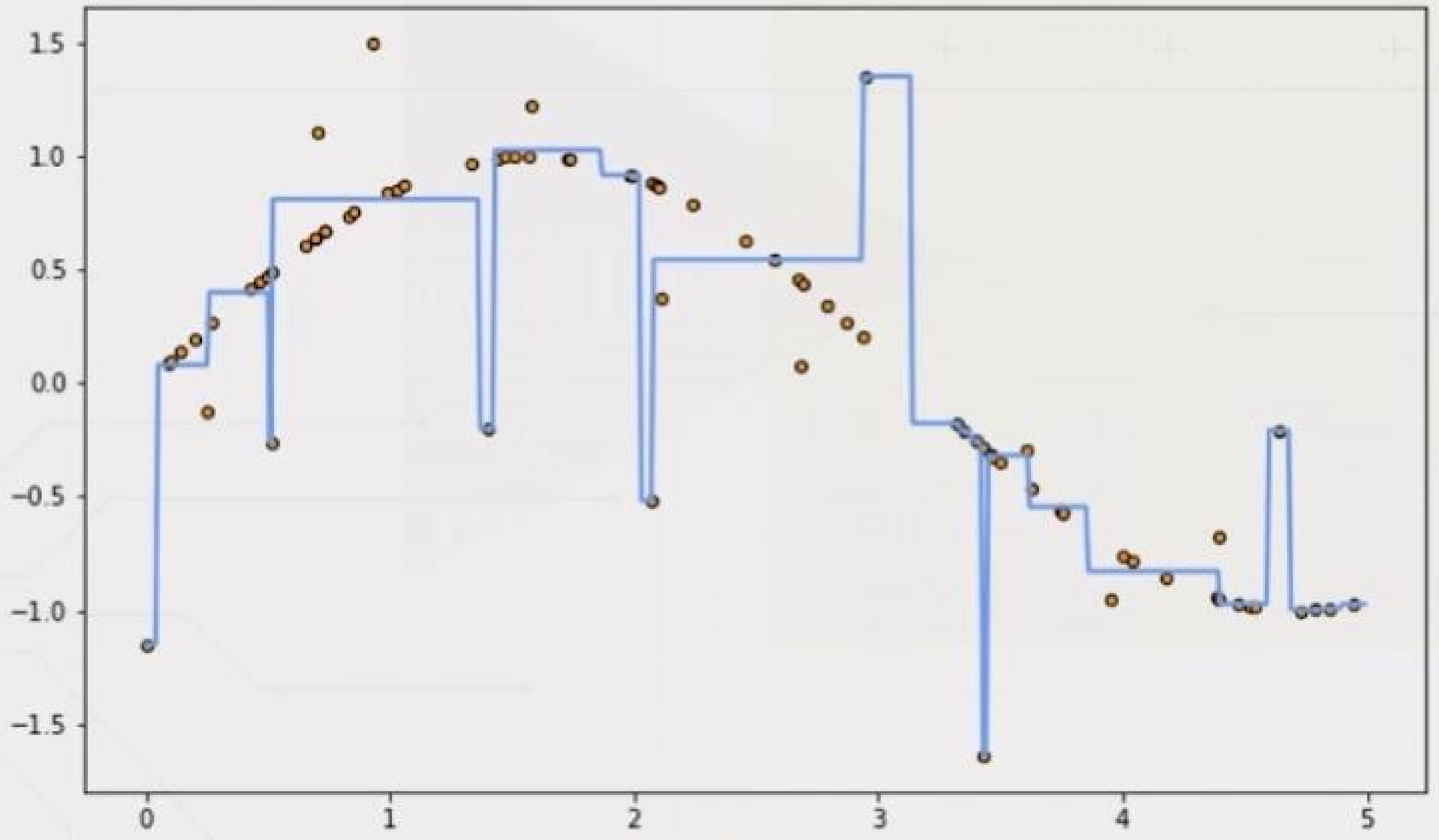
- We can keep splitting until we have only one object in each leaf
- We can ideally fit **any** training data
- Unless there are several objects with the same features and different target values

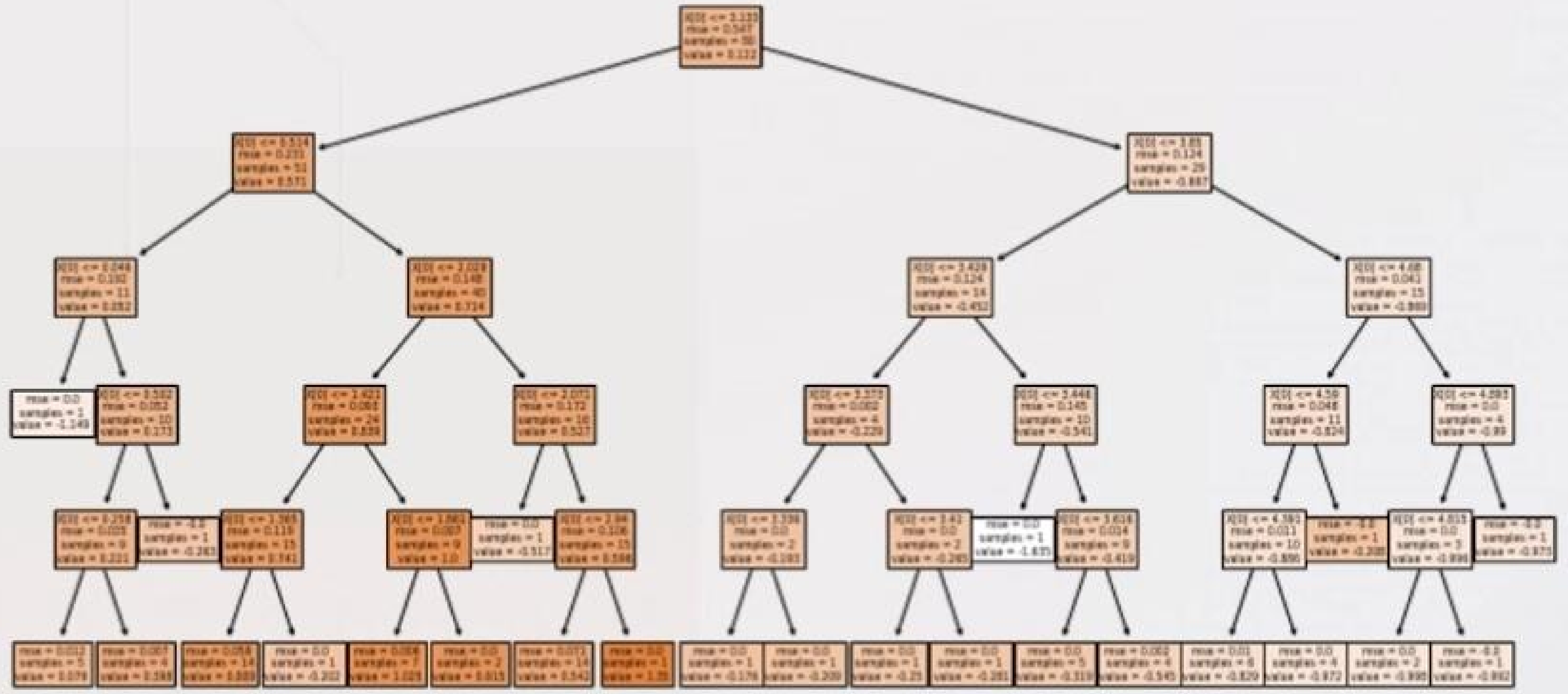
Decision Trees for Regression Task





Decision Trees for Regression and Overfitting





Summary

- Decision trees — combination of simple logic rules
- Decision Trees split feature space into several areas with constant prediction in each of them
- It is quite easy to overfit, when you use decision trees

The course is taught under the online Master's programme "Master of Data Science"

Splitting Criterion

- Step function: $[x_j < t]$ — not the only option
- Use a linear model: $[\langle w, x\rangle < t]$
- A specific metric: $[\rho(x, x_0) < t]$
- …

- But we can build arbitrary complex models even with the most simple predicates

Predictions in Leaves: Regression

- We will use constant predictions $c_v \in \mathbb{Y}$
- Average value:

$$c_v = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} y_i$$

Predictions in Leaves: Classification

- We will use constant predictions $c_v \in \mathbb{Y}$
- The most common class:

$$c_v = \arg\max_{k \in \mathbb{Y}} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

- Class probabilities

$$c_{vk} = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

Predictions in Leaves

- We could use more complex prediction functions in leaves
- E.g. linear regression:

$$c_v(x) = \langle w_v, x \rangle$$

Decision Tree: Interpretation

- Tree splits feature space on disjoint sub-spaces $R_1, \dots, R_J$
- Each sub-space R_j corresponds to the leaf
- At each sub-space R_j prediction c_j is constant

$$a(x) = \sum_{j=1}^{J} c_j \left[x \in R_j\right]$$

Decision Tree: Interpretation

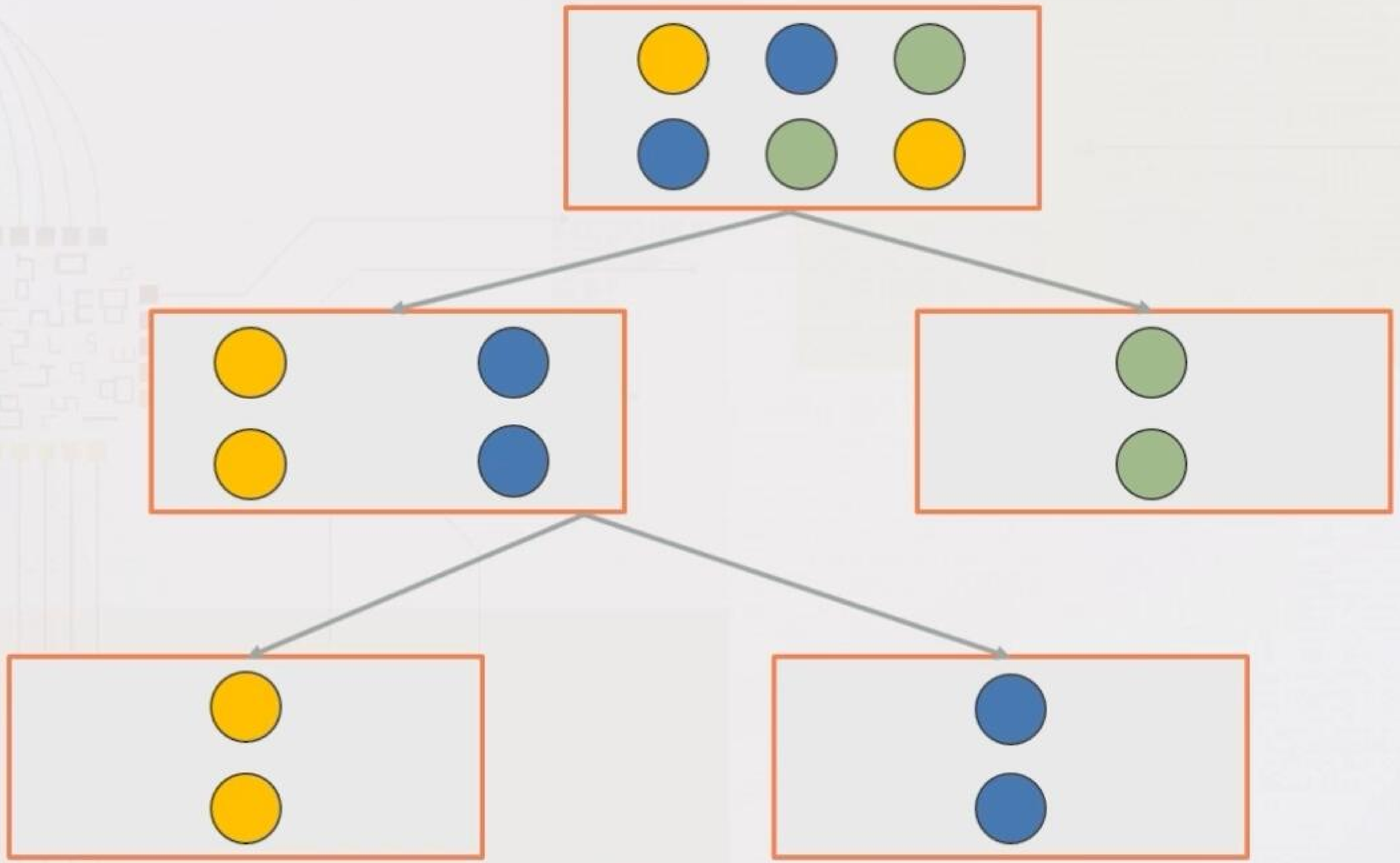
$$a(x) = \sum_{j=1}^{J} c_j \left[x \in R_j\right]$$

- Decision tree constructs new powerful features
- Therefore, the predictions is a linear combination of new features

Summary

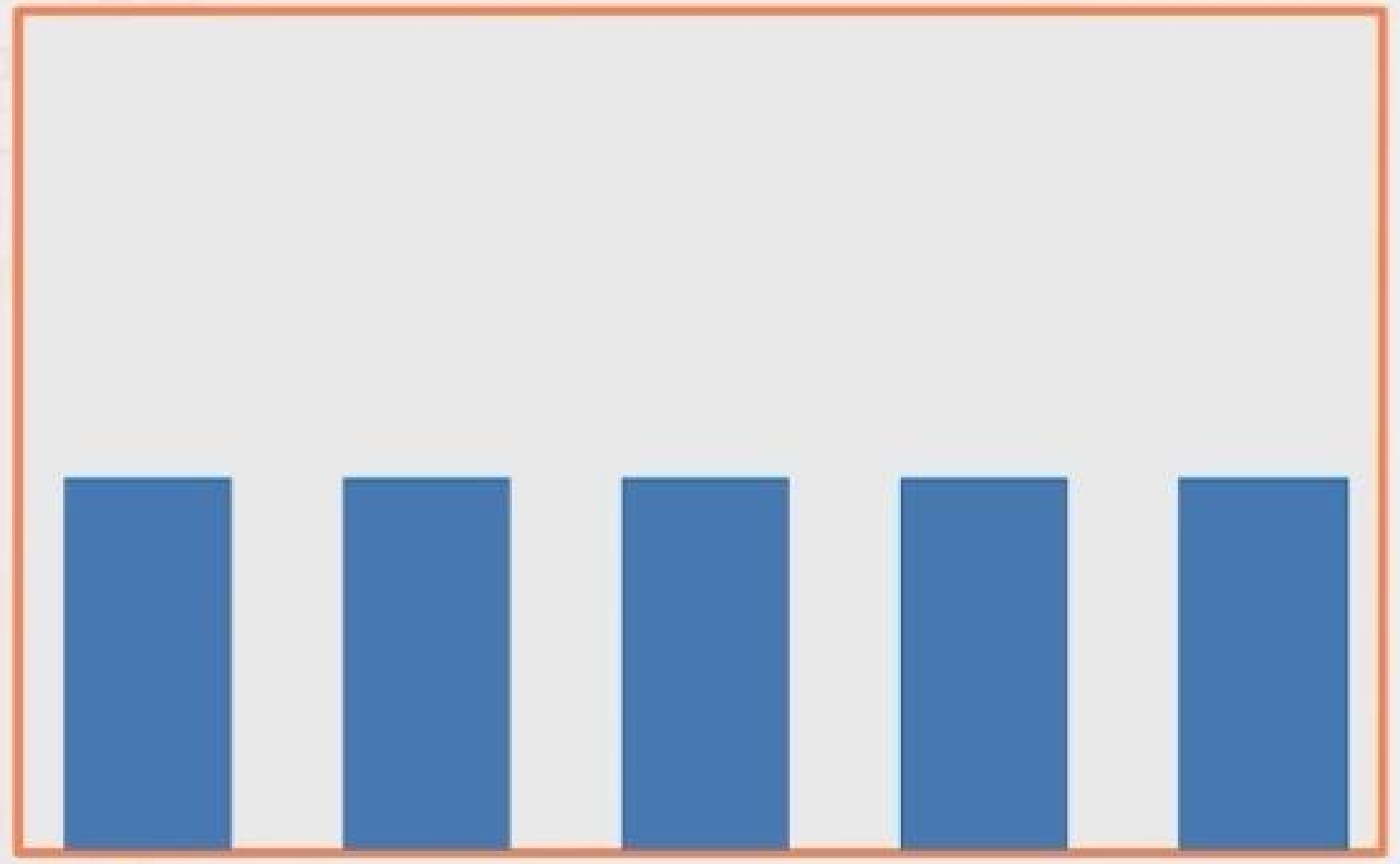
- One could use different approaches in splitting and making predictions in leaves. Usually the simplest one is good enough.
- One could think about decision tree as linear model over new features

Greedy Tree Construction

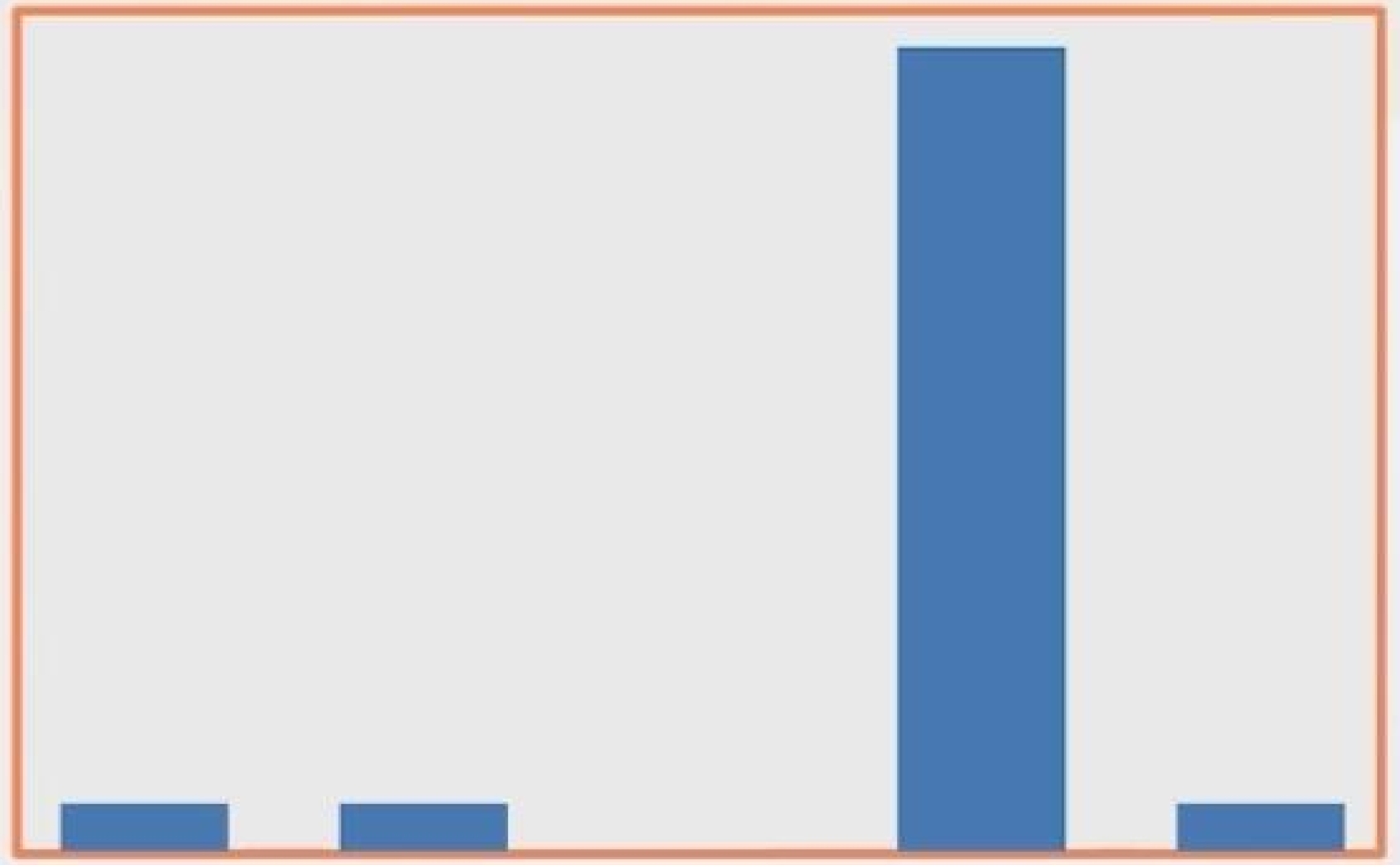


Entropy

We consider entropy as a way to measure the uncertainty of an experiment's outcome



High entropy



Low entropy

Entropy

- Assume we are given discrete distribution with n possible outcomes
- Probability of outcomes: $p_1, p_2, \dots, p_n$
- Entropy of distribution:

$$H(p_1, \dots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Entropy

$$H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- $p = (0.2, 0.2, 0.2, 0.2, 0.2)$
  - $H = 2.3219$
- $p = (0.9, 0.05, 0.05, 0, 0)$
  - $H = 0.5689$
- $p = (0, 0, 0, 1, 0)$
  - $H = 0$

Entropy

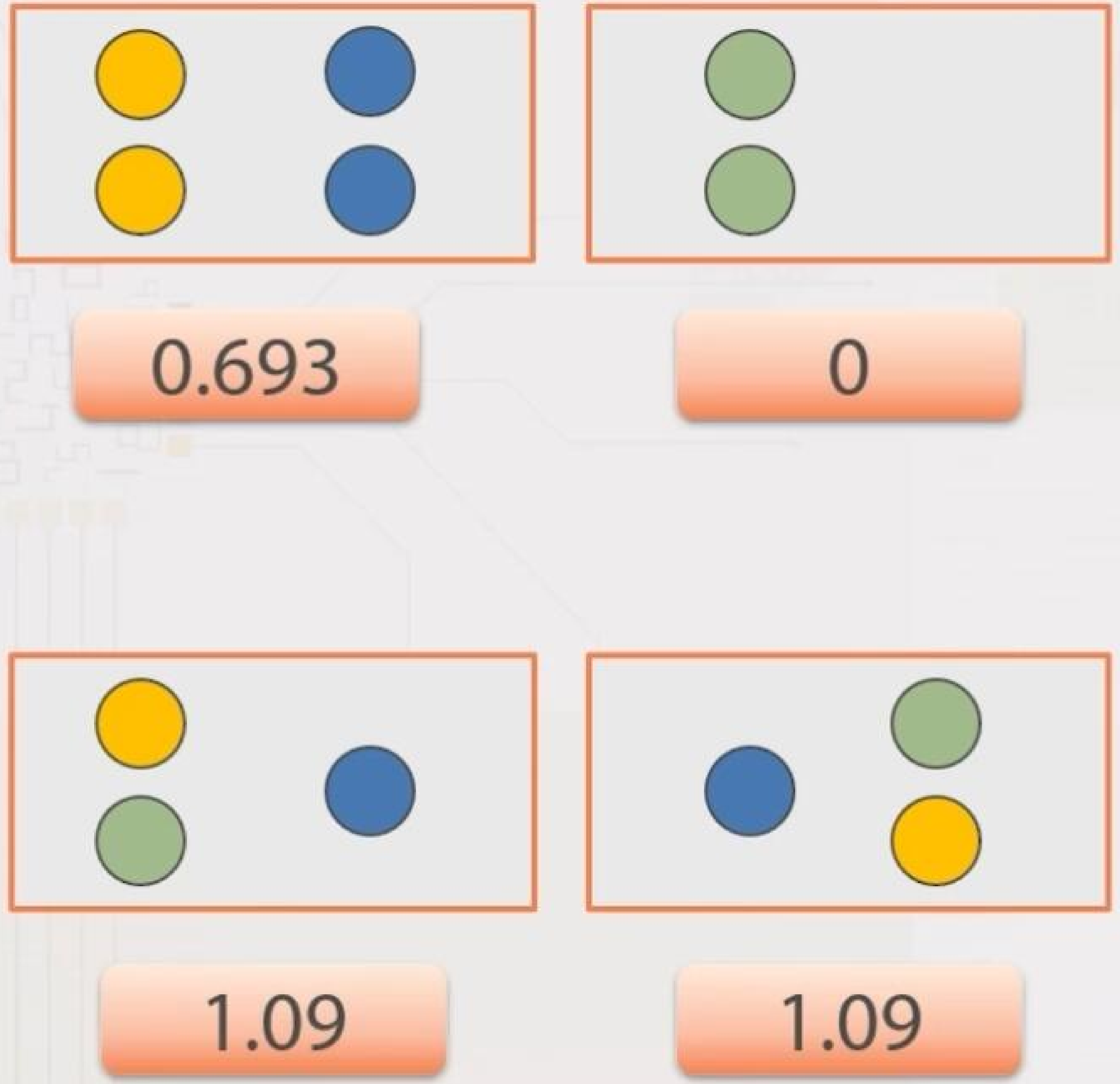
- In classification tasks the number of possible outcomes is the number of classes K

- Probability to be at the class k — fraction of objects of class k

$$p_k = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i = k]$$

- Zero Entropy — there are **only** objects form **one class** at the leaf
- Max. Entropy — there are **equal proportion** of objects from **each class**

How to Select Between Two Splits?



- $(0.5, 0.5, 0)$ and $(0, 0, 1)$
- $0.693 + 0 = 0.693$

- $(0.33, 0.33, 0.33)$ and $(0.33, 0.33, 0.33)$
- $1.09 + 1.09 = 2.18$

Summary

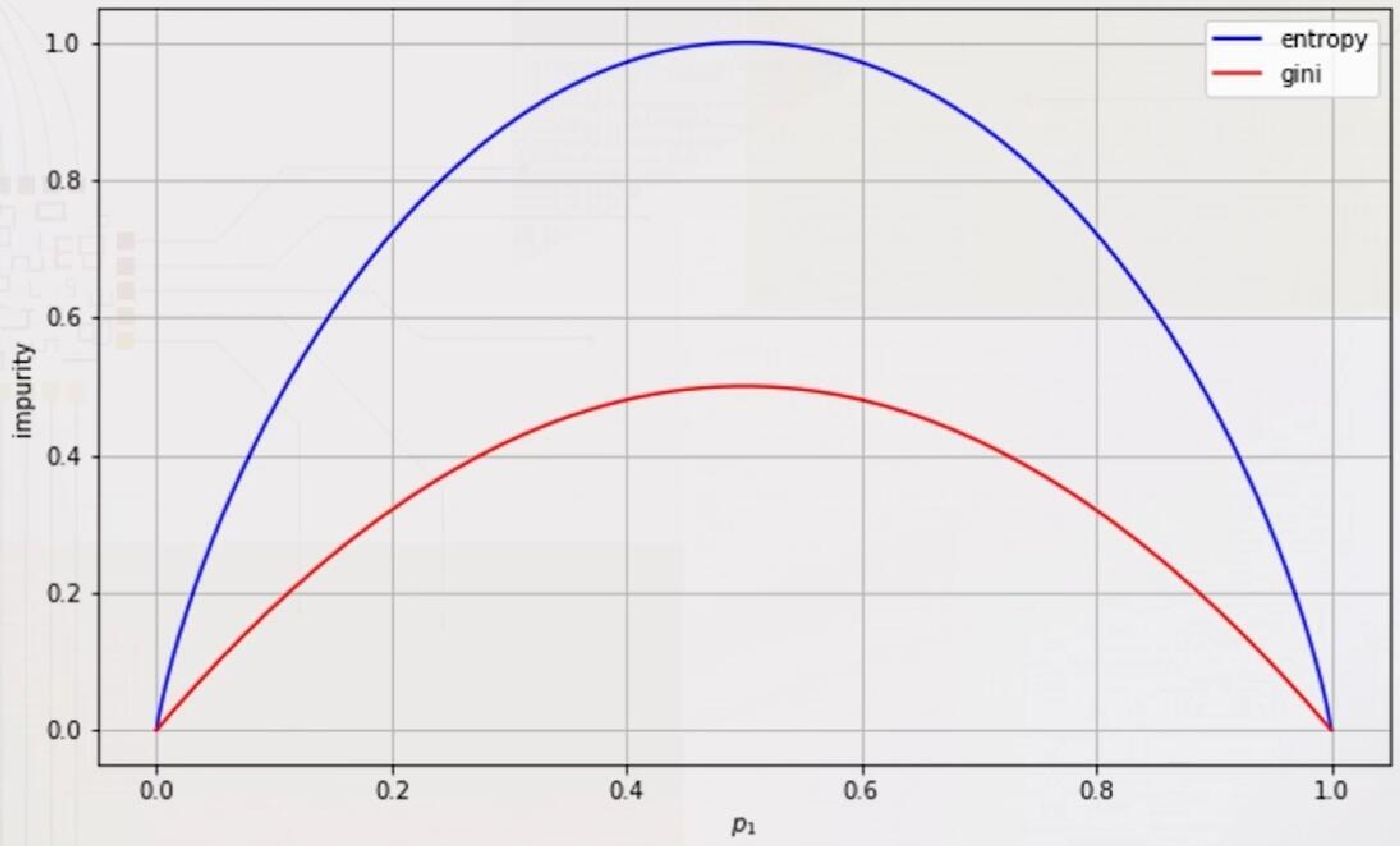
- Decision tree could be constructed in a greedy manner from the root node to the leaves
- For classification task we could choose a split, so that it minimizes class diversity at resulting groups

Gini Index

$$H(p_1, \ldots, p_K) = \sum_{i=1}^{K} p_i \, (1 - p_i)$$

- Consider a classifier, which outputs class k with probability p_k
- Gini index is a probability that the object will be classified incorrectly if the class is assigned with probabilities $p_1, \ldots, p_k$

Gini Index vs Entropy



Impurity Criterions

- How to decide which split is better?
- Compare the impurity before the split (in the initial node R) and in the two nodes after the split (R_ℓ and R_r)

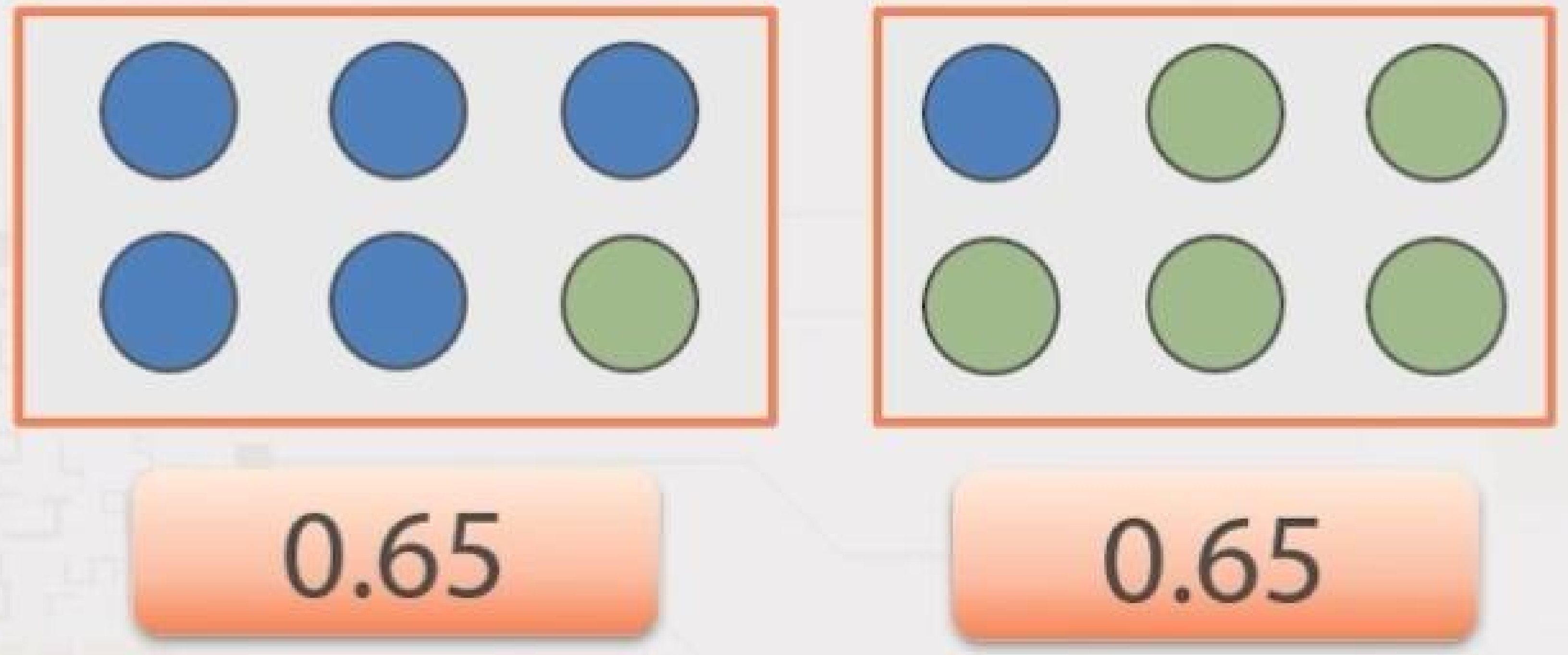
Impurity Criterions

- How to decide which split is better?
- Compare the impurity before the split (in the initial node R) and in the two nodes after the split (R_ℓ and R_r)

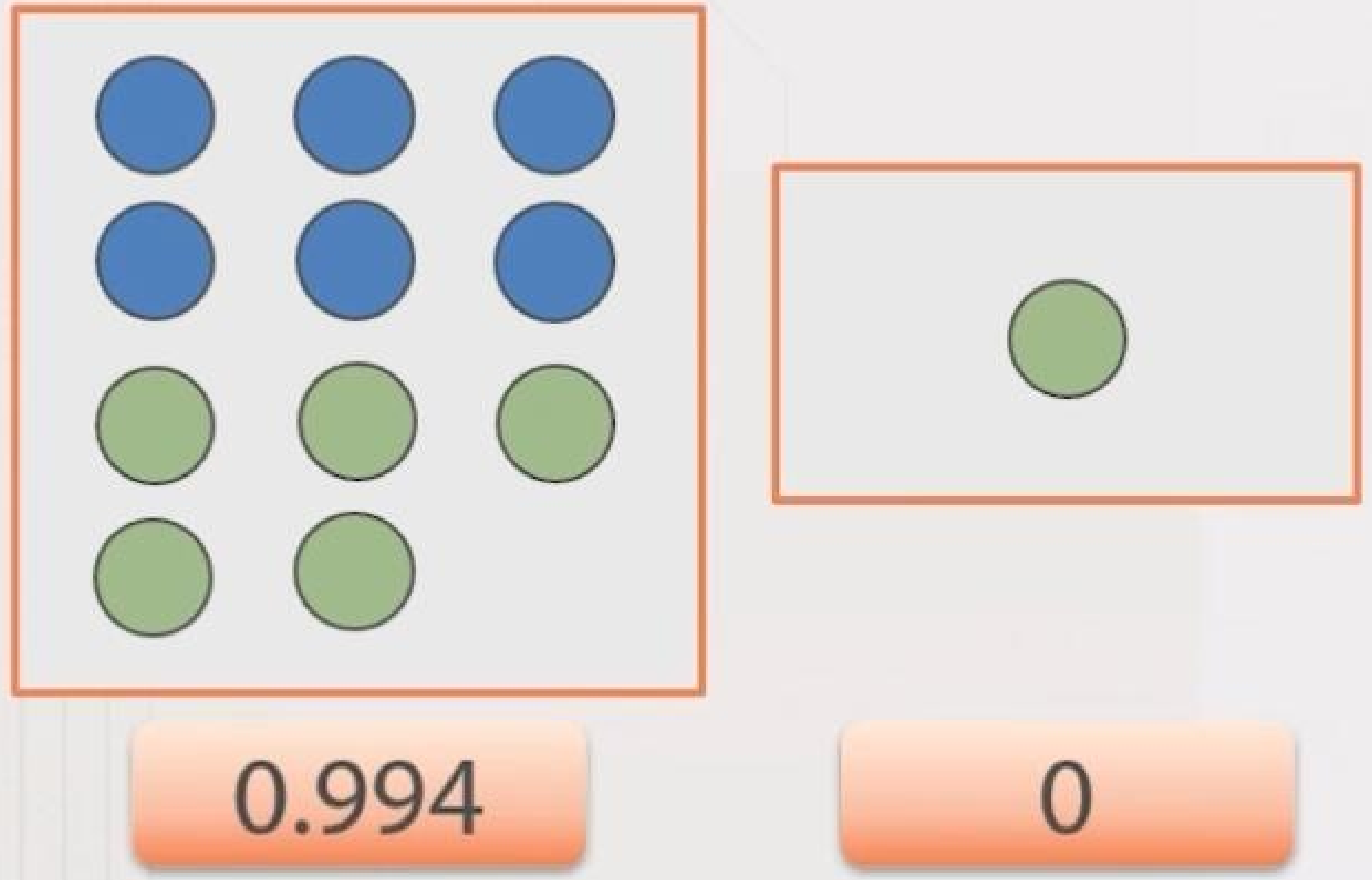
$$Q(R, j, t) = H(R) - \frac{|R_\ell|}{|R|} H(R_\ell) - \frac{|R_r|}{|R|} H(R_r) \to \max_{j,t}$$

How to Compare Two Splits?

$H(R) = 1$

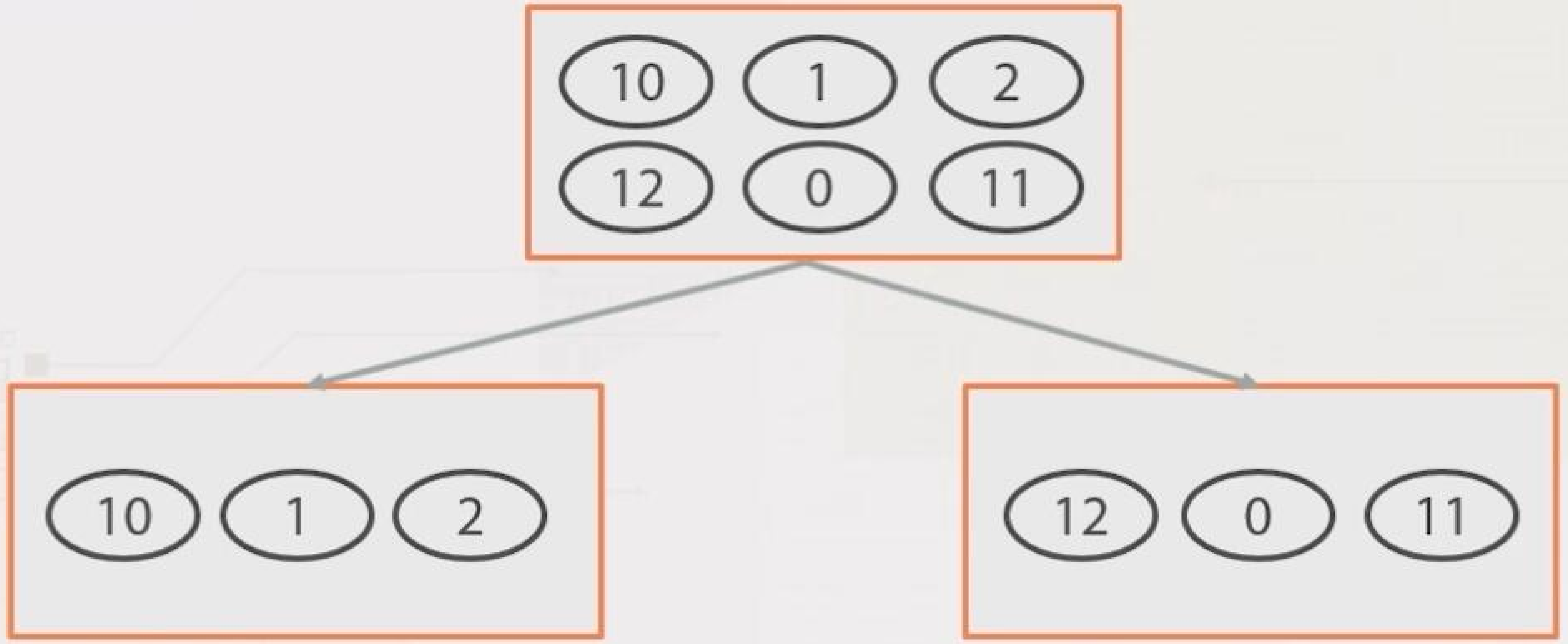


- $Q(R) = 1 - \frac{1}{2}0.65 - \frac{1}{2}0.65$
- $Q(R) = 0.35$



- $Q(R) = 1 - \frac{11}{12}0.994 - \frac{1}{12}0$
- $Q(R) = 0.088$

Greedy Construction: Regression



Regression Task

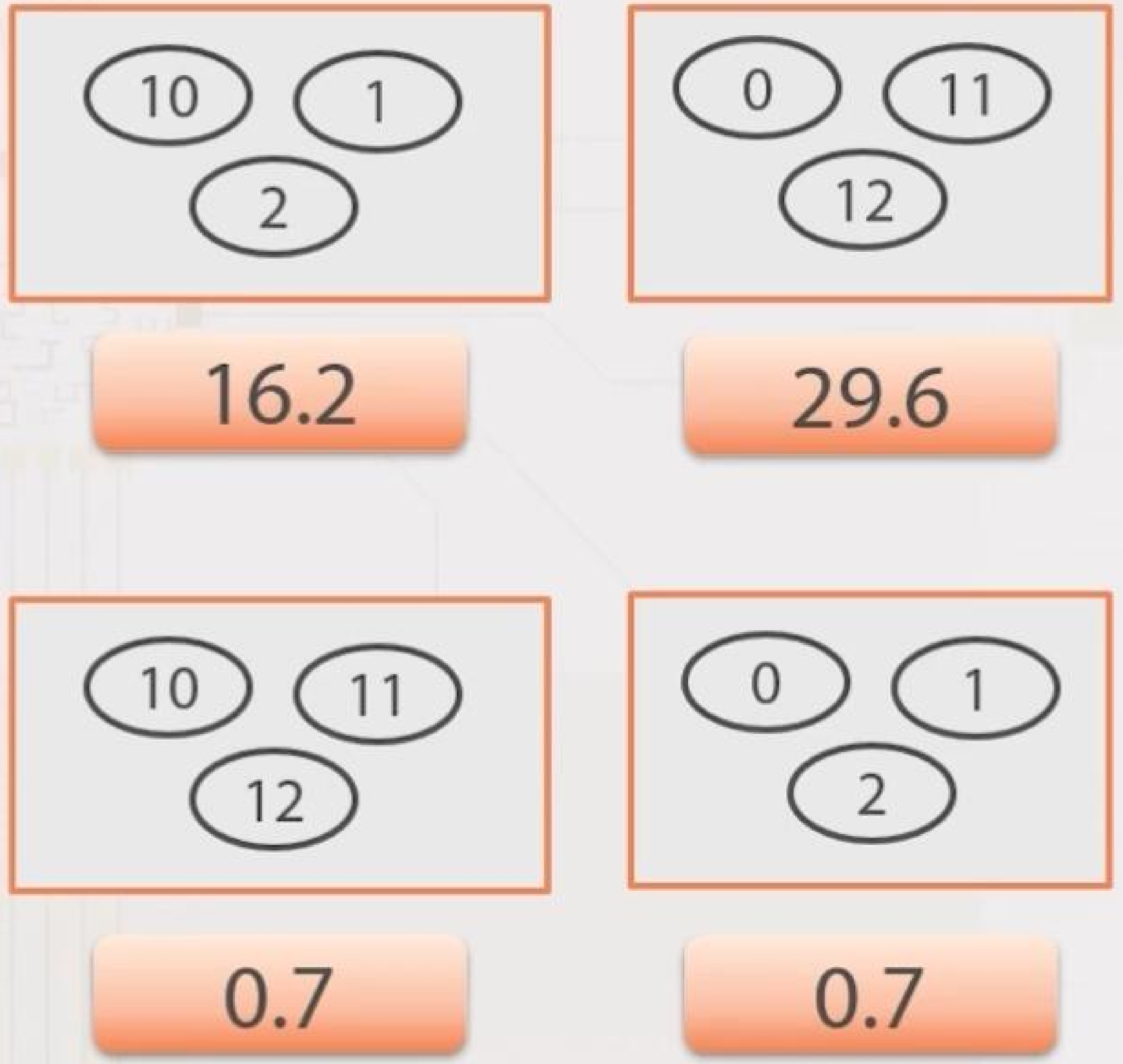
$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - y_R)^2$$

$$y_R = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i$$

- So we can measure the variance of answers in the node

How to Compare Two Splits?

$$H(R) = \mathbf{25.6}$$



- $Q(R) = 25.6 - \frac{1}{2}16.2 - \frac{1}{2}29.6$
- $Q(R) = 2.7$

- $Q(R) = 25.6 - \frac{1}{2}0.7 - \frac{1}{2}0.7$
- $Q(R) = 24.9$

Summary

- We can choose the split, so that it reduces the diversity of answers in the resulting nodes
- We use impurity criterion to measure the quality of the split
- There are different criterions that might be used.
  The most popular are:
  - Entropy and Gini for classification
  - Variance for regression