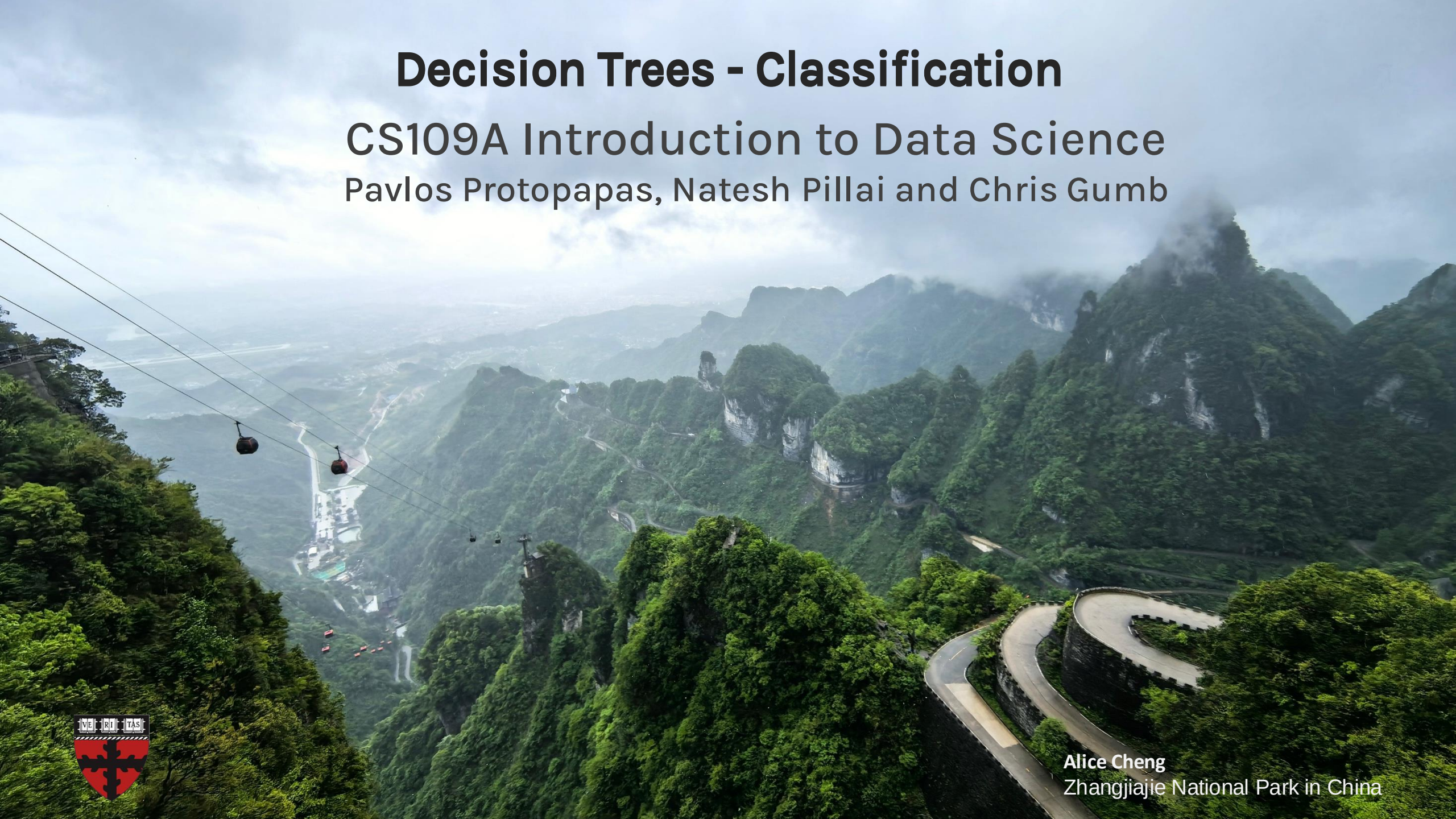
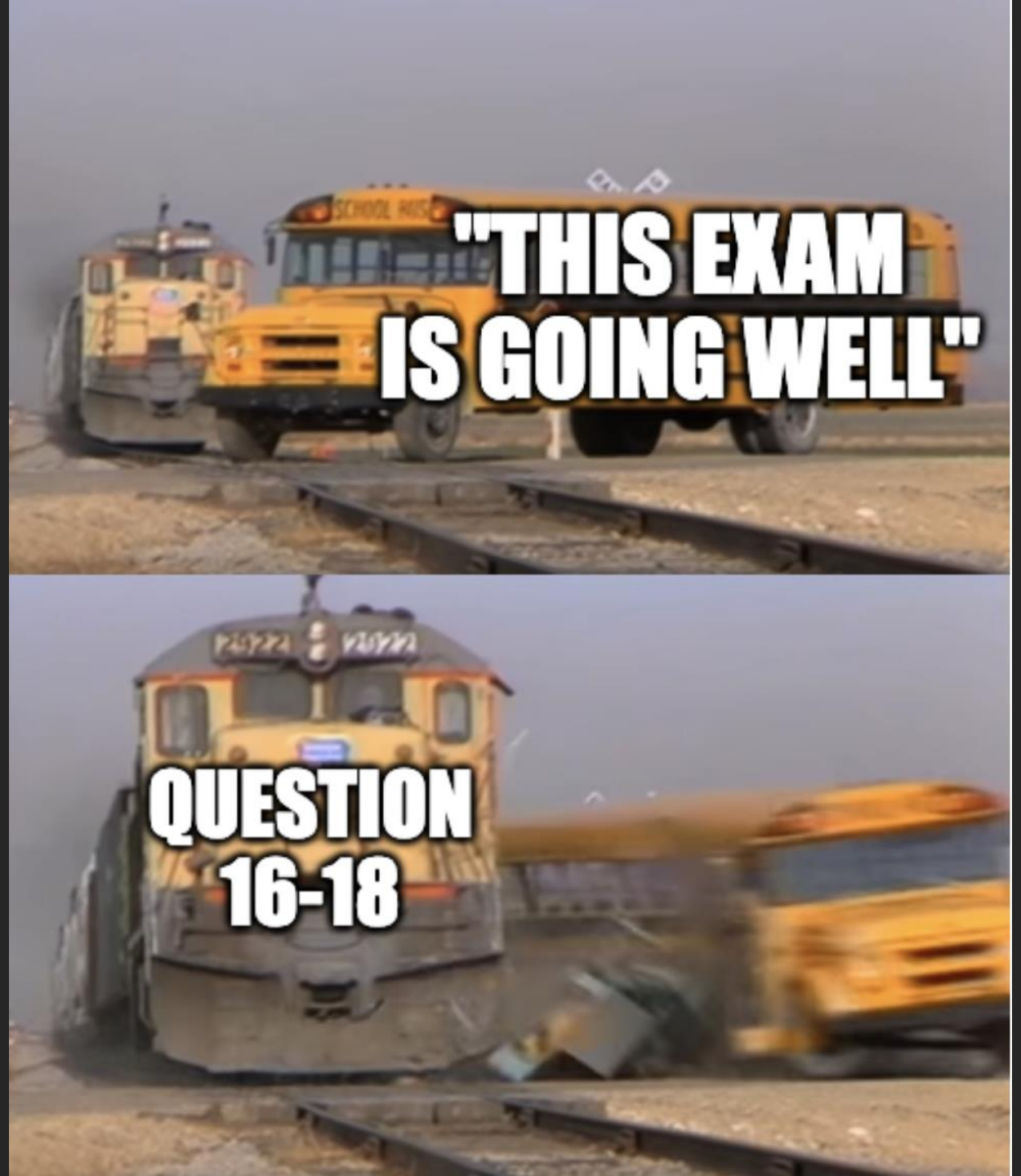
Decision Trees - Classification

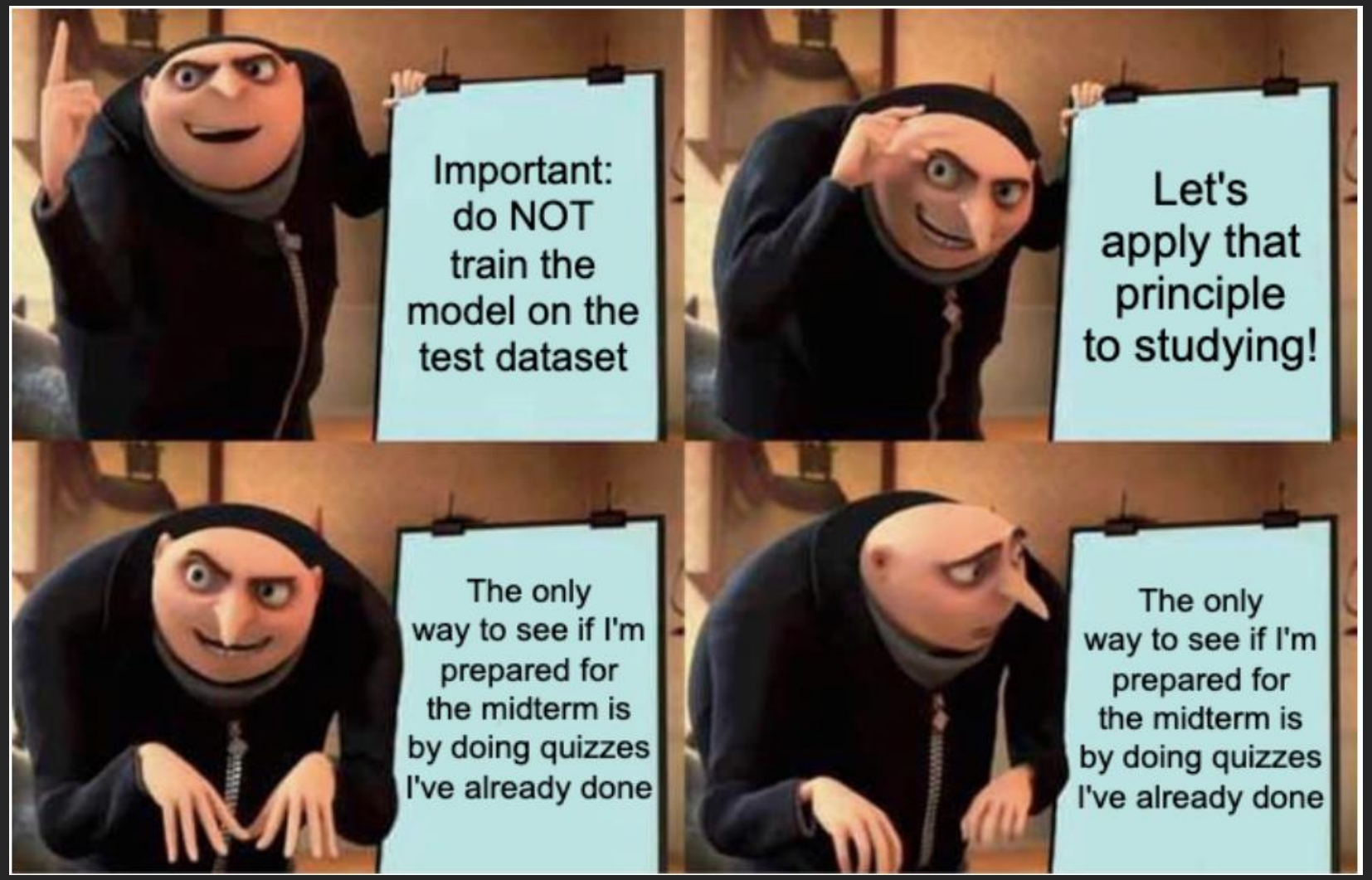
CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai and Chris Gumb



Alice Cheng
Zhangjiajie National Park in China


"THIS EXAM
IS GOING WELL"
QUESTION
16-18


Important: do NOT train the model on the test dataset
Let's apply that principle to studying!
The only way to see if I'm prepared for the midterm is by doing quizzes I've already done
The only way to see if I'm prepared for the midterm is by doing quizzes I've already done

Outline

- Motivation
- Decision Trees – Classification
  - Intuition
  - Predictions
  - Splitting Criteria
  - Stopping Conditions

Outline

- Motivation
- Decision Trees – Classification
  - Intuition
  - Predictions
  - Splitting Criteria
  - Stopping Conditions


When you're trying to be one with nature, but someone mentions food

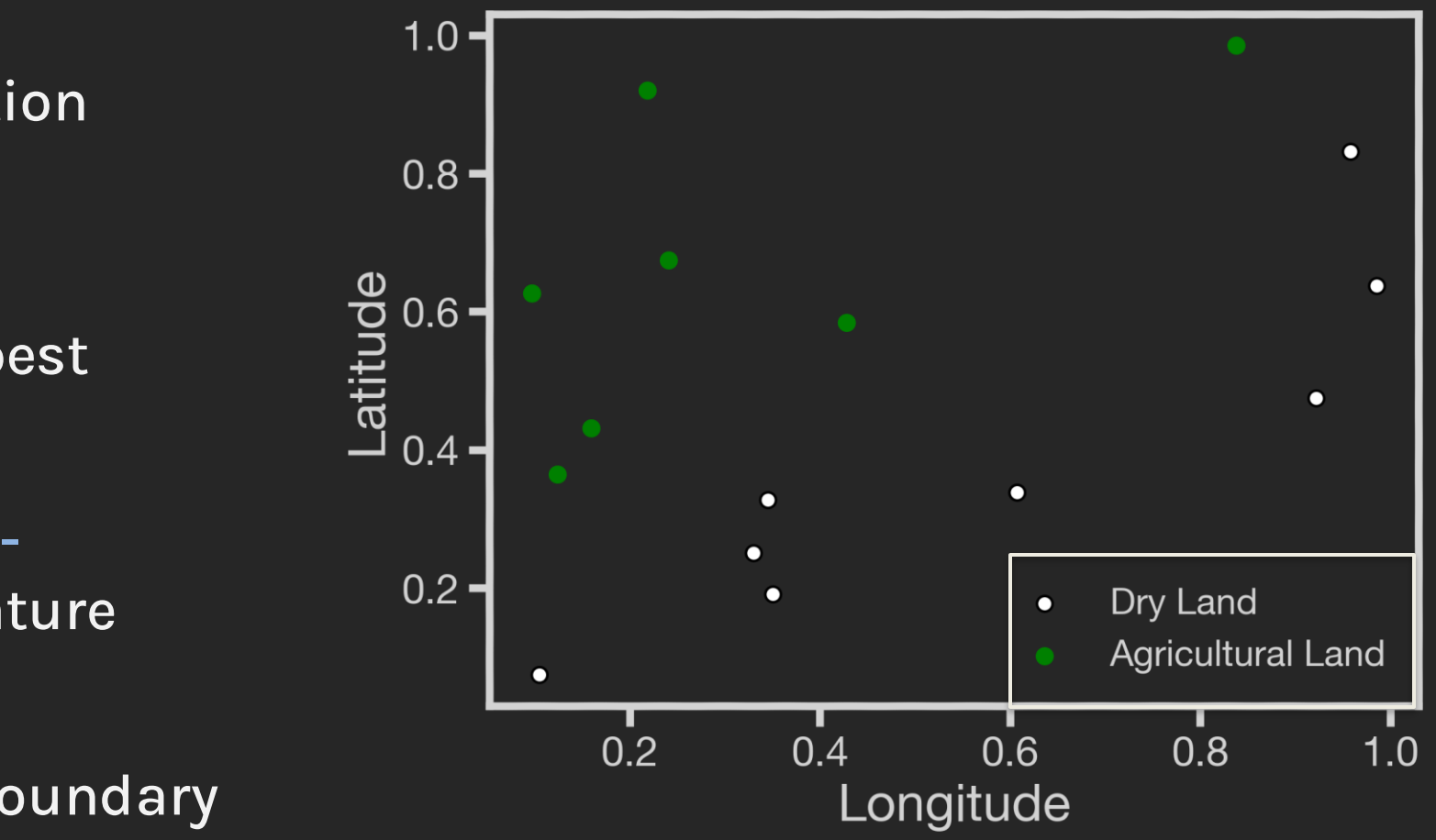
Motivation

The simplest and most straightforward classification model we know is logistic regression.

Logistic regression works best when:

a) The classes are well-separated in the feature space, and

b) The classification boundary has a nice geometry



Motivation

Classification Boundaries or **Decision Boundaries** are defined where the probabilities of being in class 1 and class 0 are equal, i.e.

$$P(Y = 1) = P(Y = 0)$$

Motivation

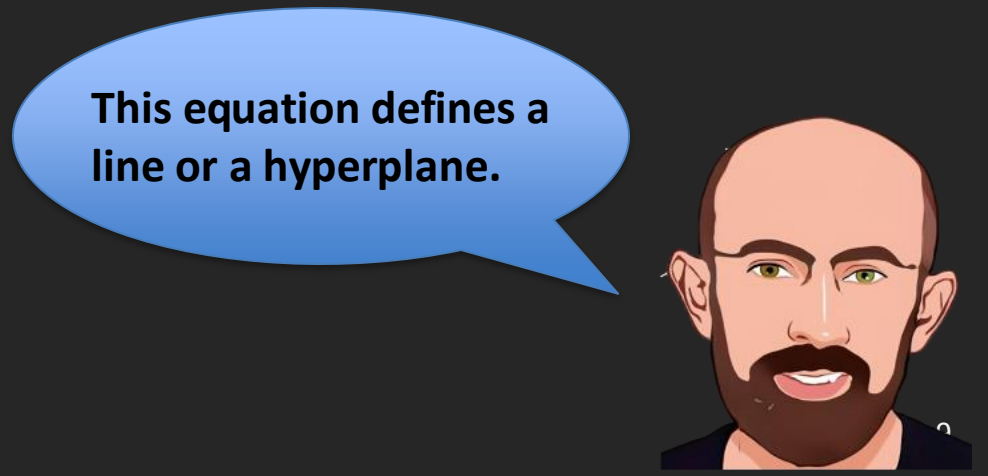
Classification Boundaries or **Decision Boundaries** are defined where the probabilities of being in class 1 and class 0 are equal, i.e.

$$P(Y = 1) = P(Y = 0) = 1 - P(Y = 1)$$

$$\Rightarrow P(Y = 1) = 0.5$$

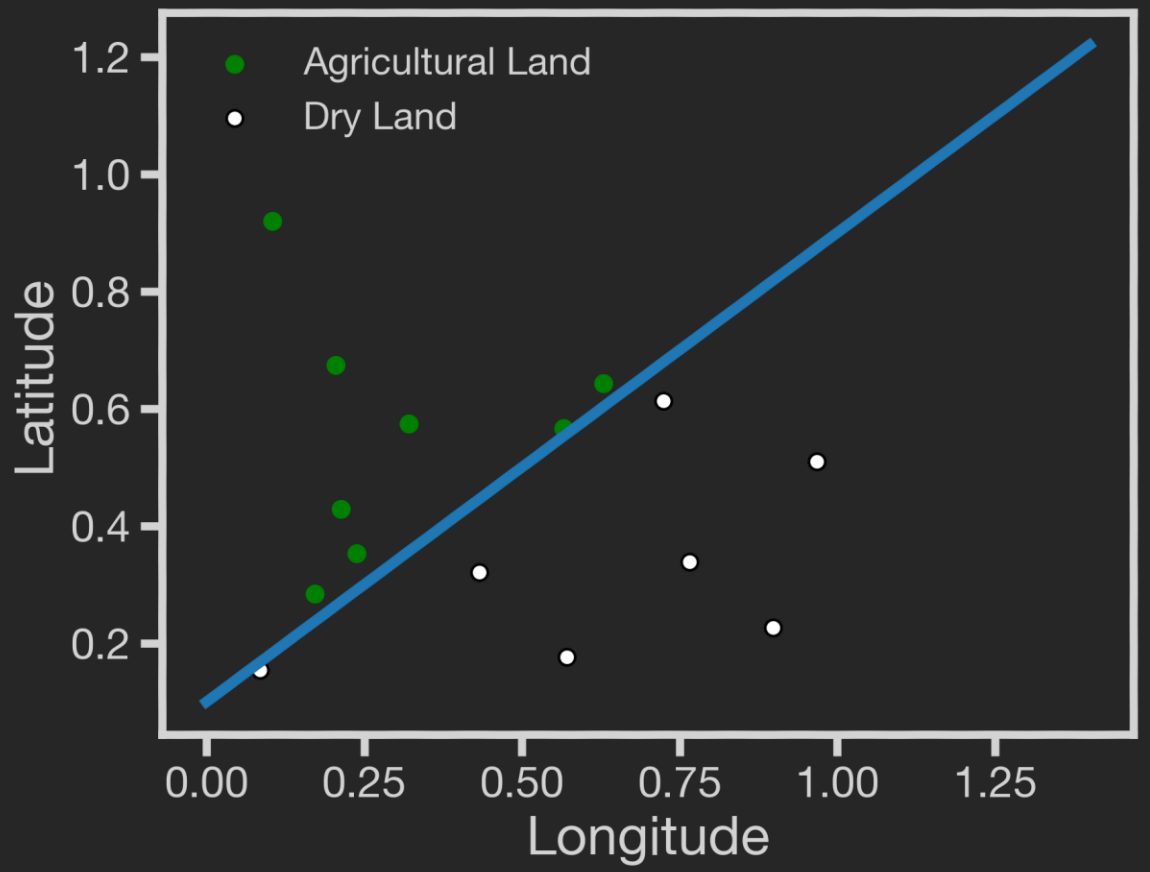
which is equivalent to when the log-odds are zero:

$$X\beta = 0$$



Motivation

For example, the equation that defines the decision boundary shown with a blue line in the figure is:



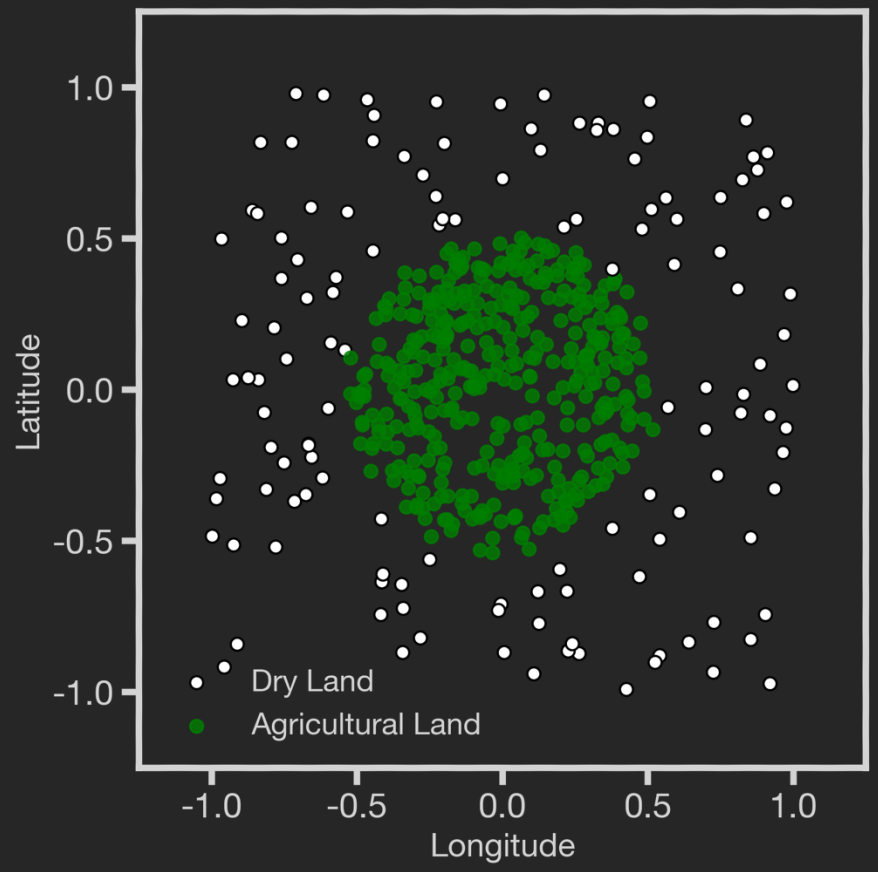
$$Latitude = 0.8\ Longitude + 0.1$$

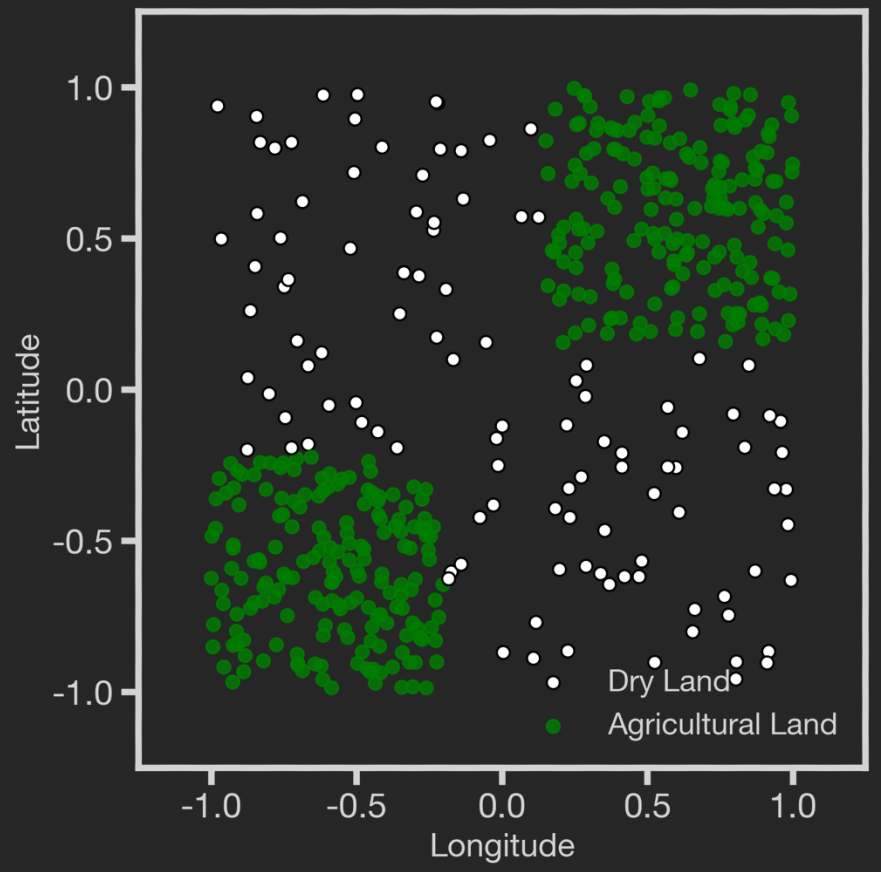
$$or$$

$$-0.8\ Longitude + Latitude - \ 0.1 = 0$$

Motivation

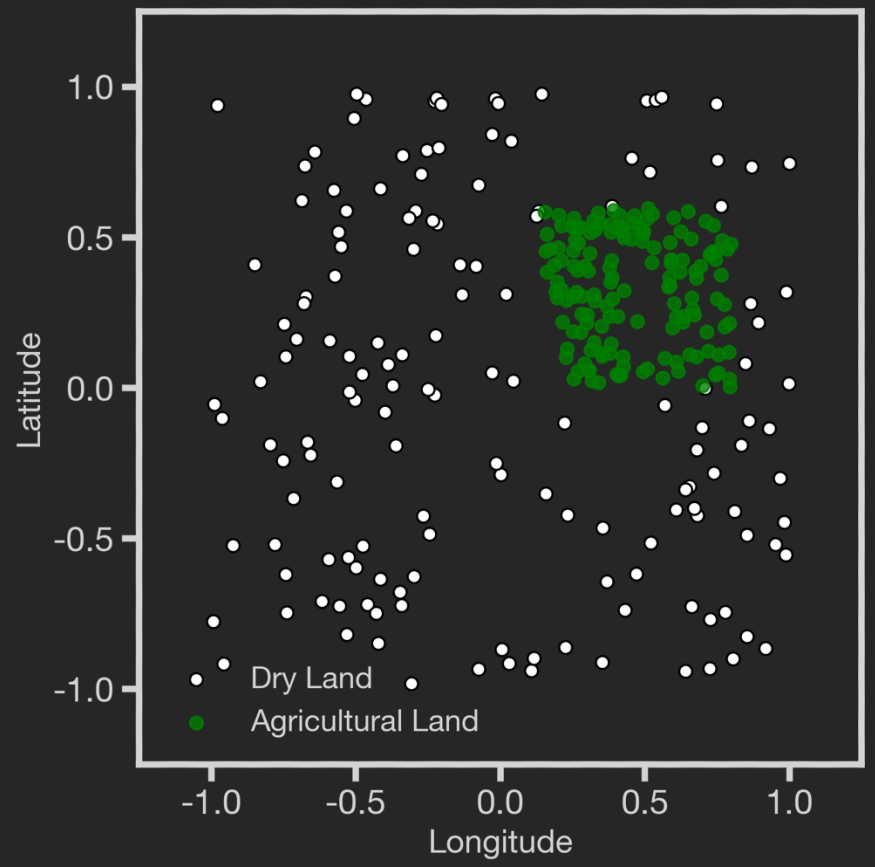
This works well when the classification boundary has a nice simple geometry, but what about these?

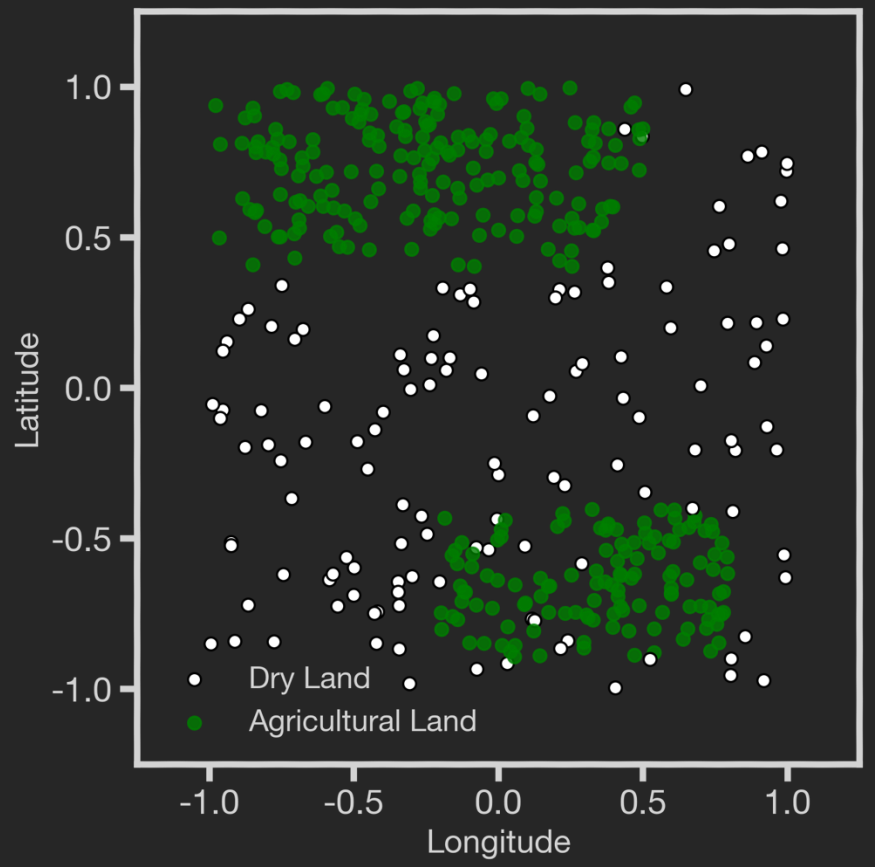




Motivation

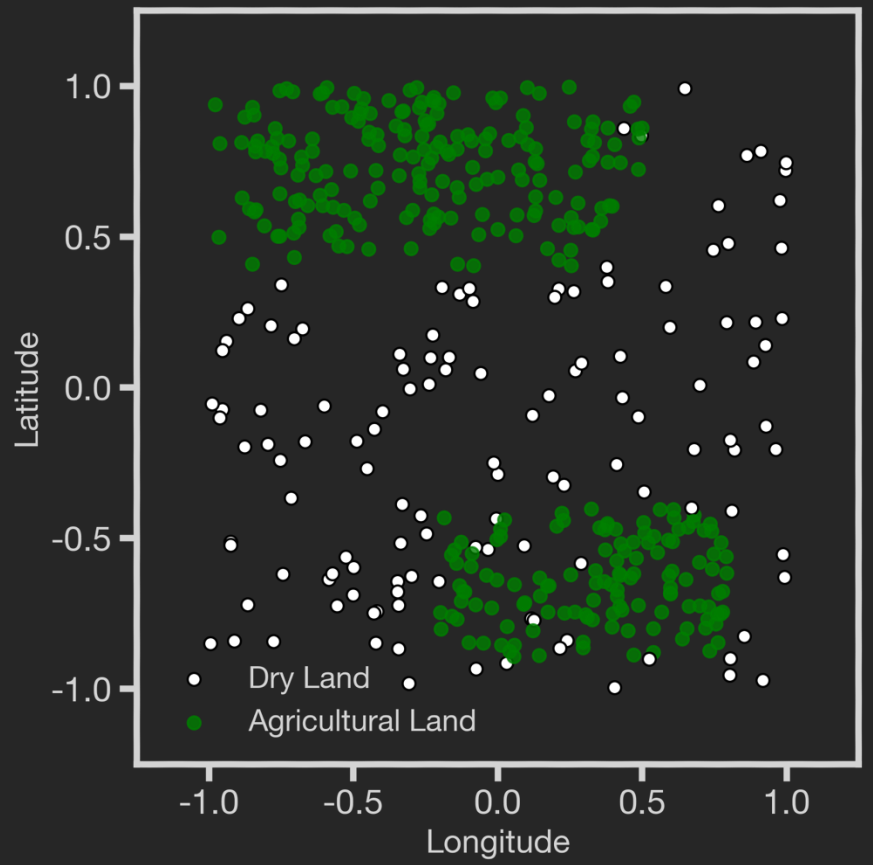
Or these?





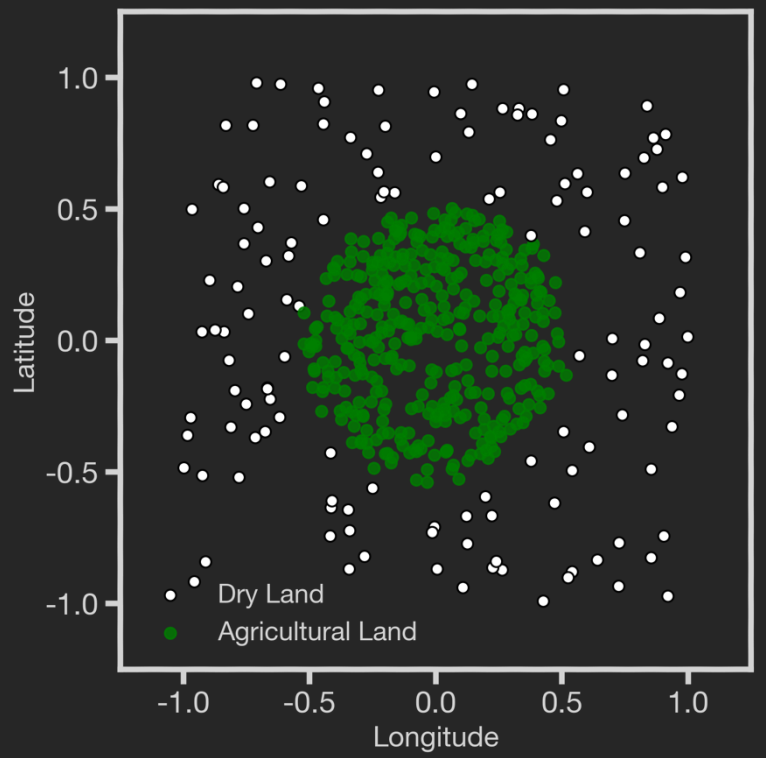
Motivation

In such datasets the classes are still well-separated in the feature space, but ***the decision boundaries cannot easily be described by a single equation***.



Motivation

Logistic regression models with linear boundaries are intuitive to interpret by examining the impact of each predictor on the log-odds of a positive classification. It is less straightforward to interpret nonlinear decision boundaries and how they interact with individual predictor values.



$$x_1^2 + x_2^2 - 0.25 = 0$$

Motivation

The modeling wish list: We would like to have models that

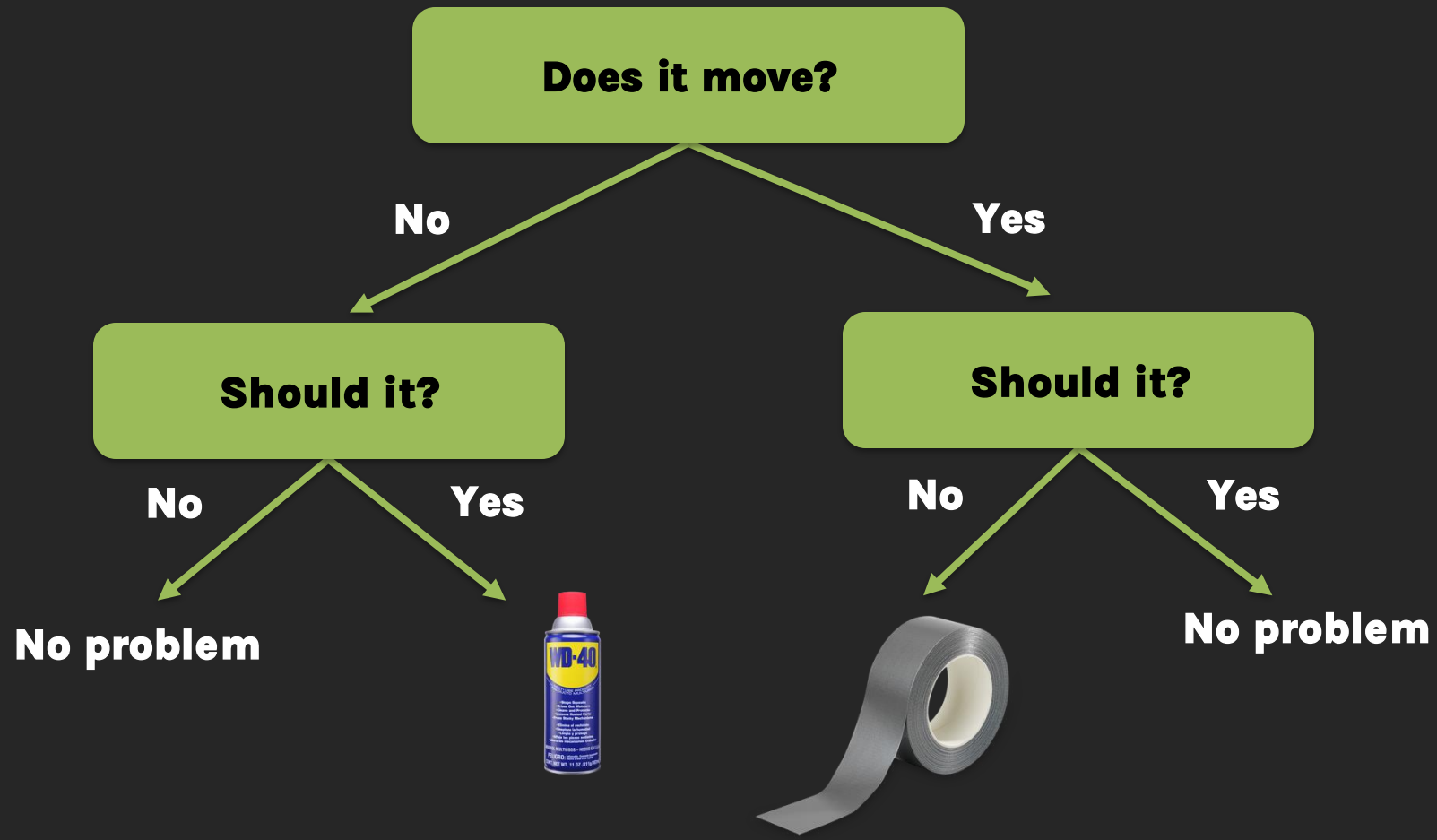
1. Allow for complex decision boundaries, and
2. Are also easy to interpret.

Interpretable Models

People in every walk of life have long been using interpretable models for differentiating between classes of objects and phenomena.

For example, the Engineering Flowchart:

Engineering Flowchart



Interpretable Models

It turns out that the simple flow chart in our example can be formulated as a mathematical model for classification, and this model has the properties we desire:

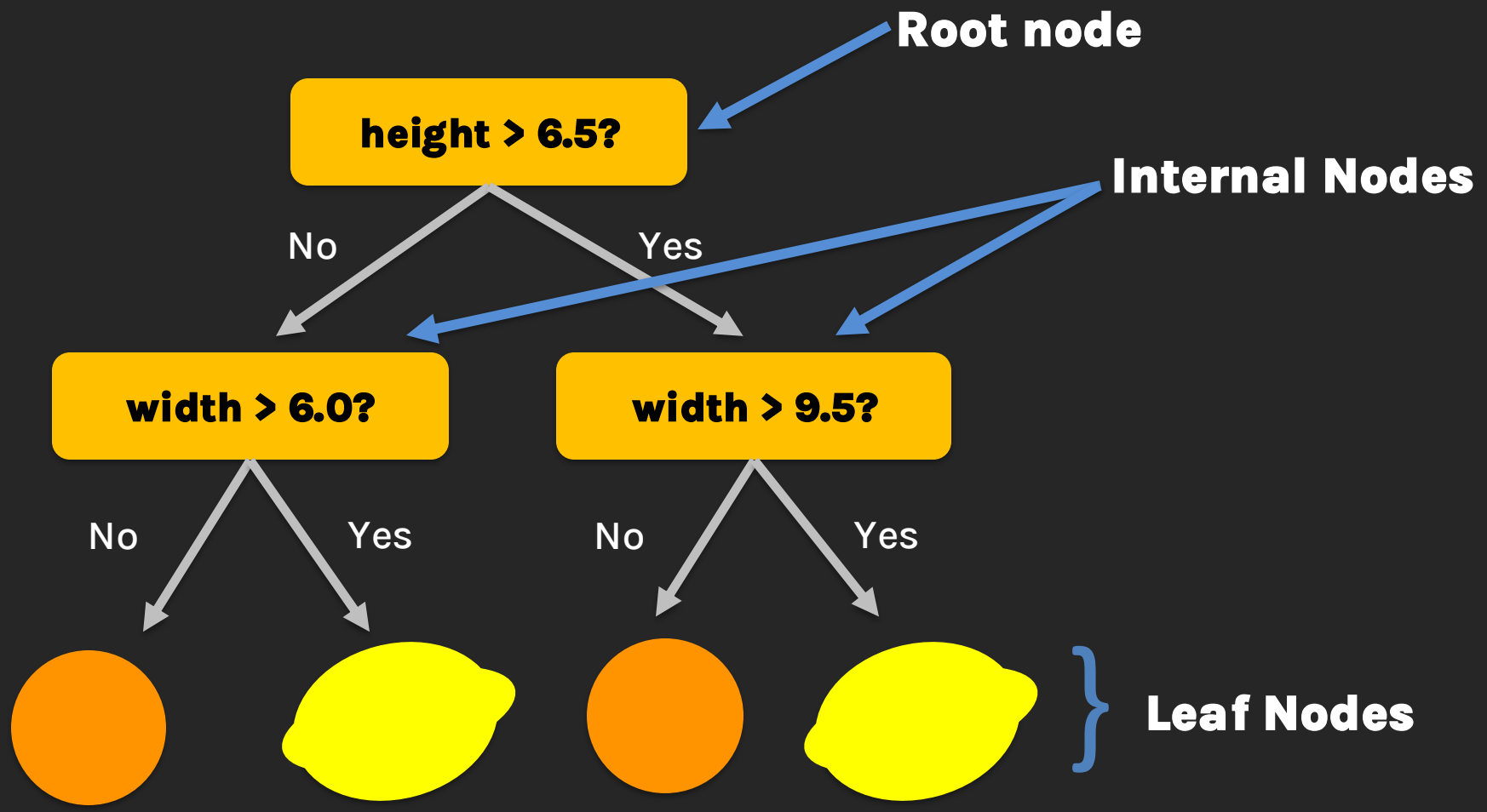
- The model is interpretable by humans.
- It has sufficiently complex decision boundaries.
- The decision boundaries are locally linear, which means each component of the decision boundary is simple to describe mathematically.

Outline

- Motivation
- Decision Trees – Classification
  - Intuition
  - Predictions
  - Splitting Criteria
  - Stopping Conditions

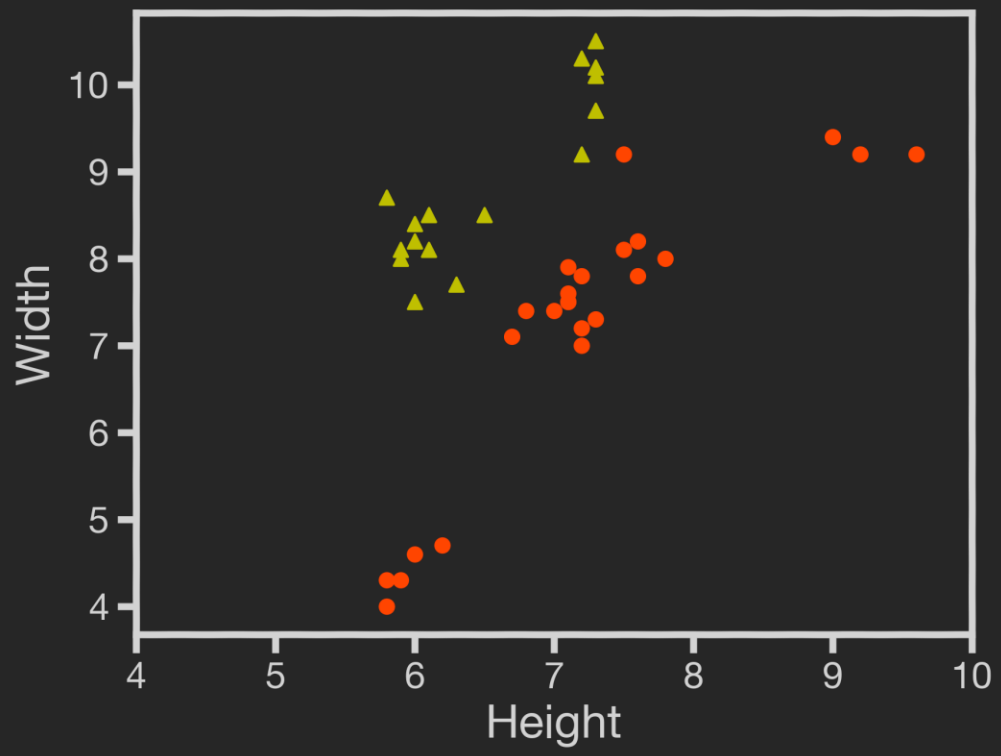
Example

A decision tree to classify fruits into lemons and oranges:



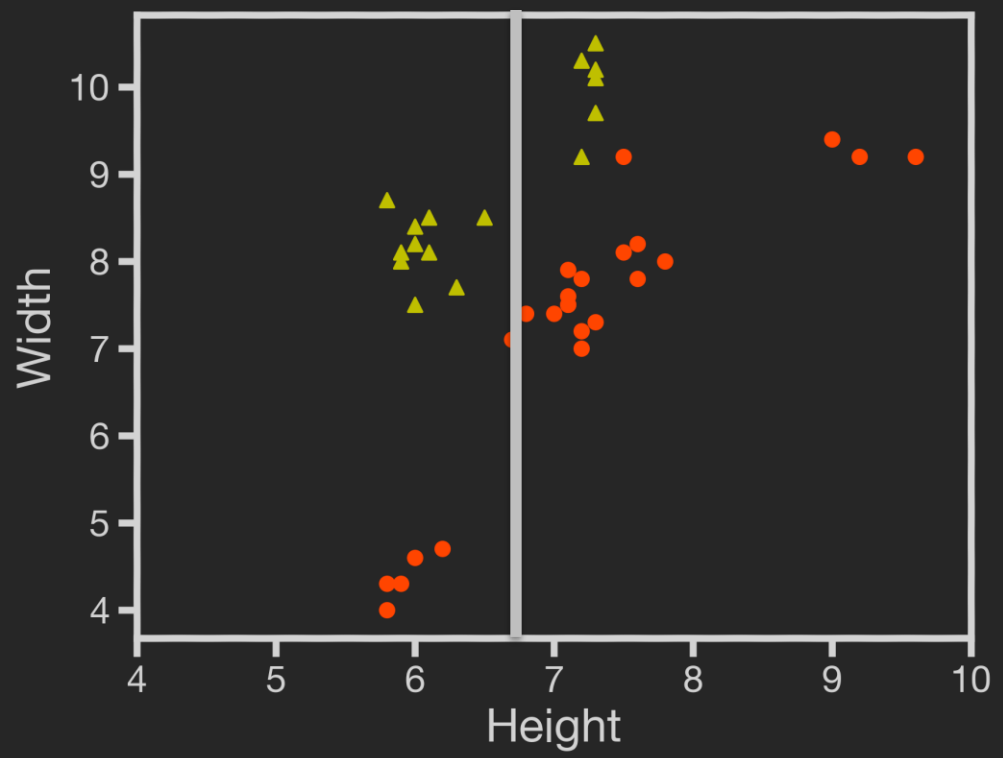
The Geometry of Flow Charts

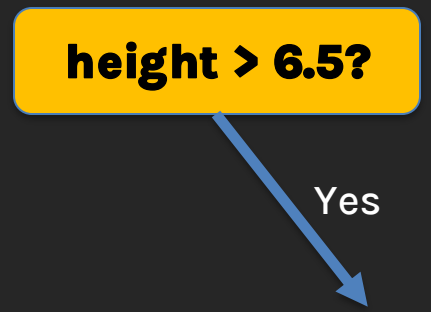
Every flow chart tree corresponds to a partition of the feature space by **lines parallel to the axis or (hyper) planes**. Conversely, every such partition can be written as a flow chart tree.



The Geometry of Flow Charts

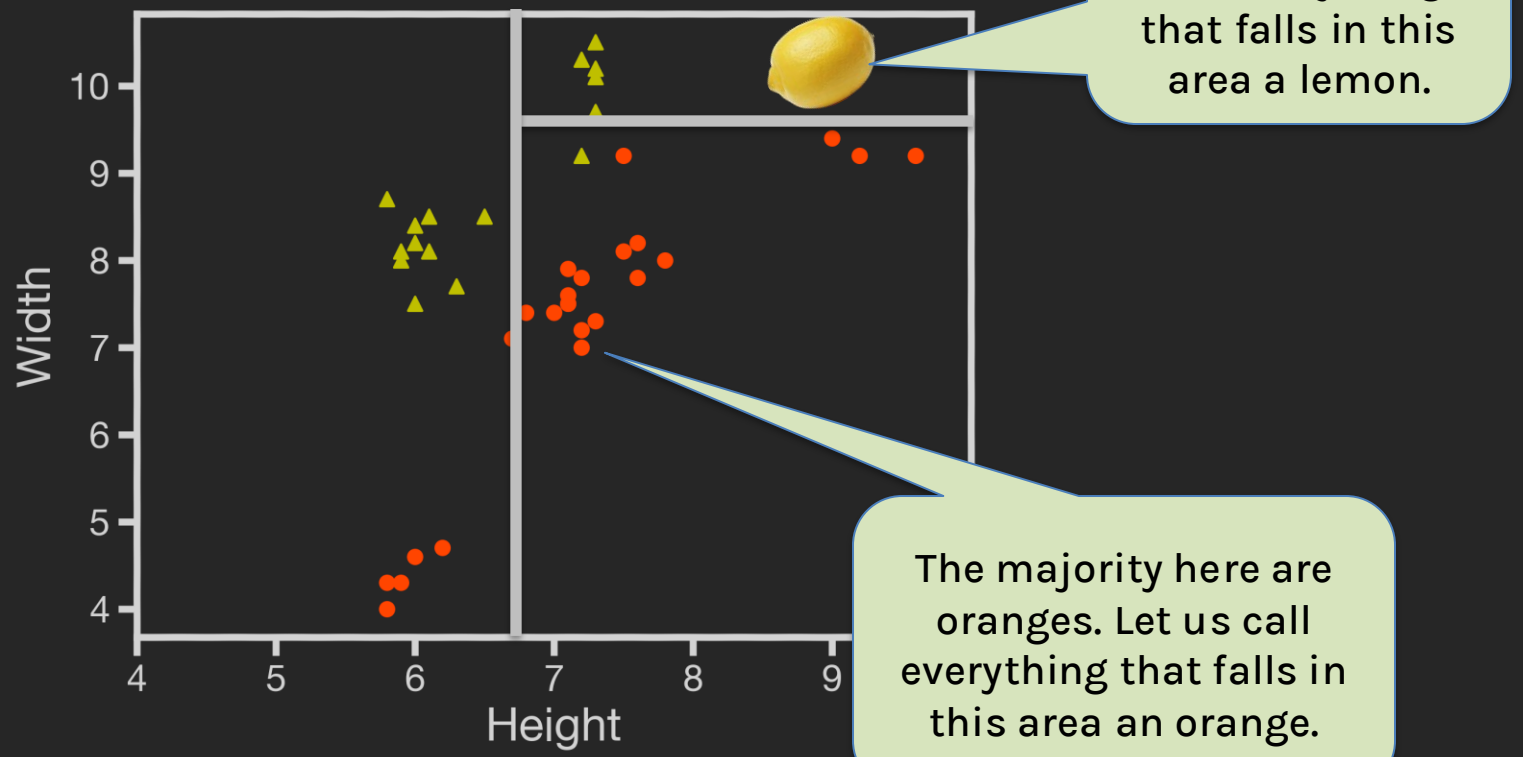
Every flow chart tree corresponds to a partition of the feature space by **lines parallel to the axis or (hyper) planes**. Conversely, every such partition can be written as a flow chart tree.

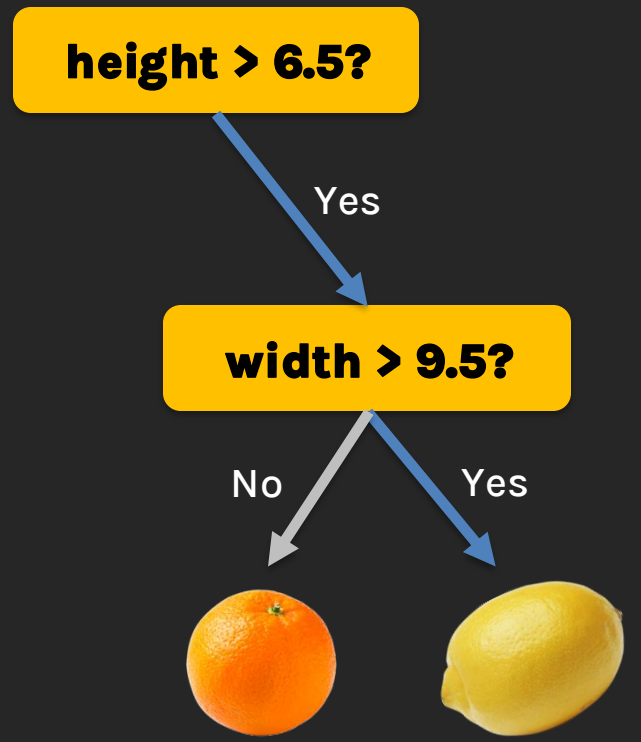




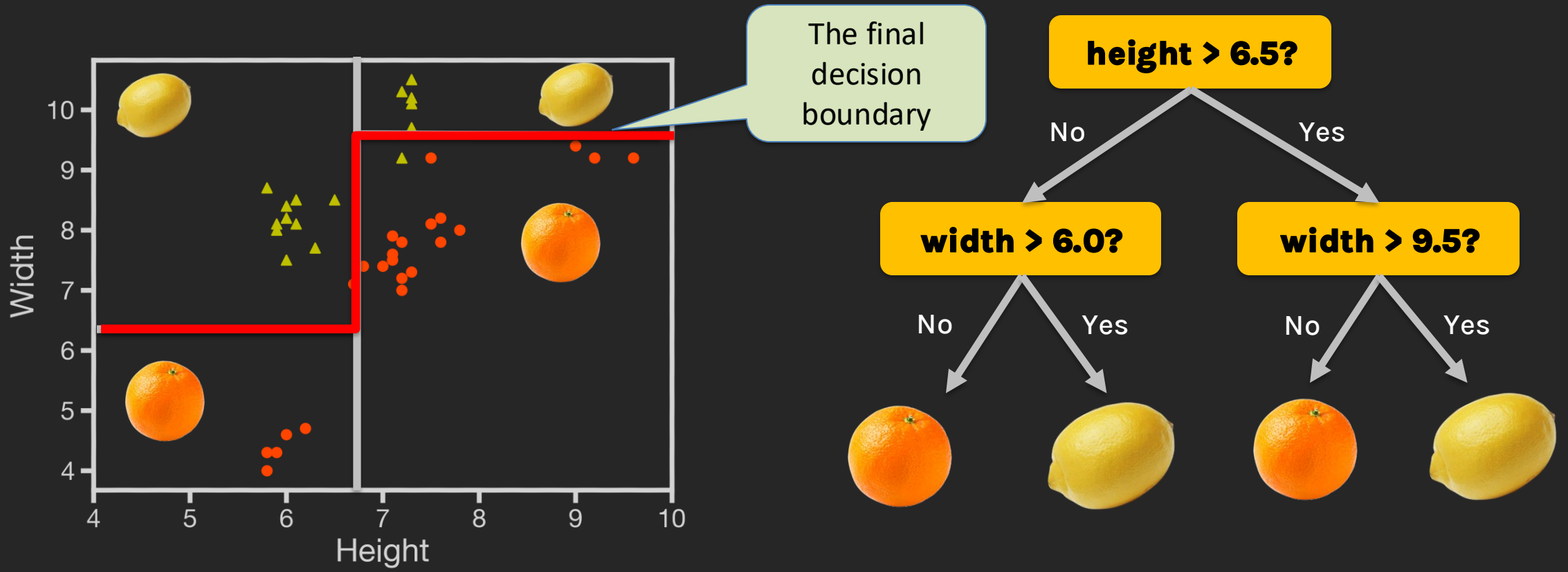
The Geometry of Flow Charts

Every flow chart tree corresponds to a partition of the feature space by **lines parallel to the axis or (hyper) planes**. Conversely, every such partition can be written as a flow chart tree.





The Geometry of Flow Charts

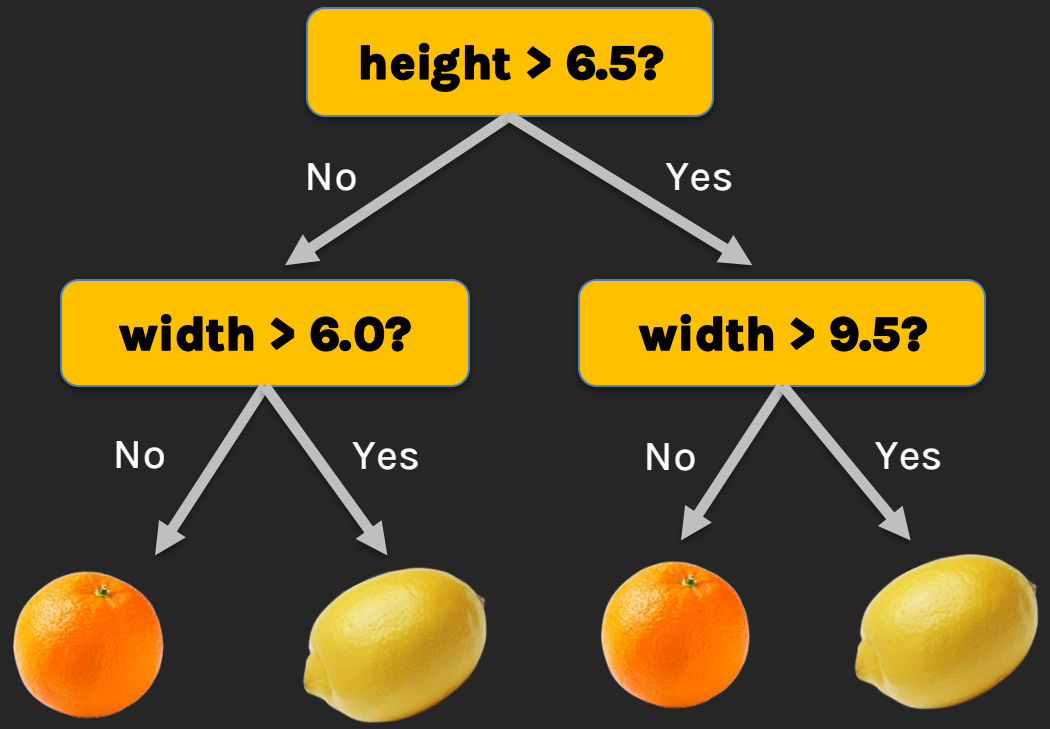


Outline

- Motivation
- Decision Trees – Classification
  - Intuition
  - Predictions
  - Splitting Criteria
  - Stopping Conditions

Predictions

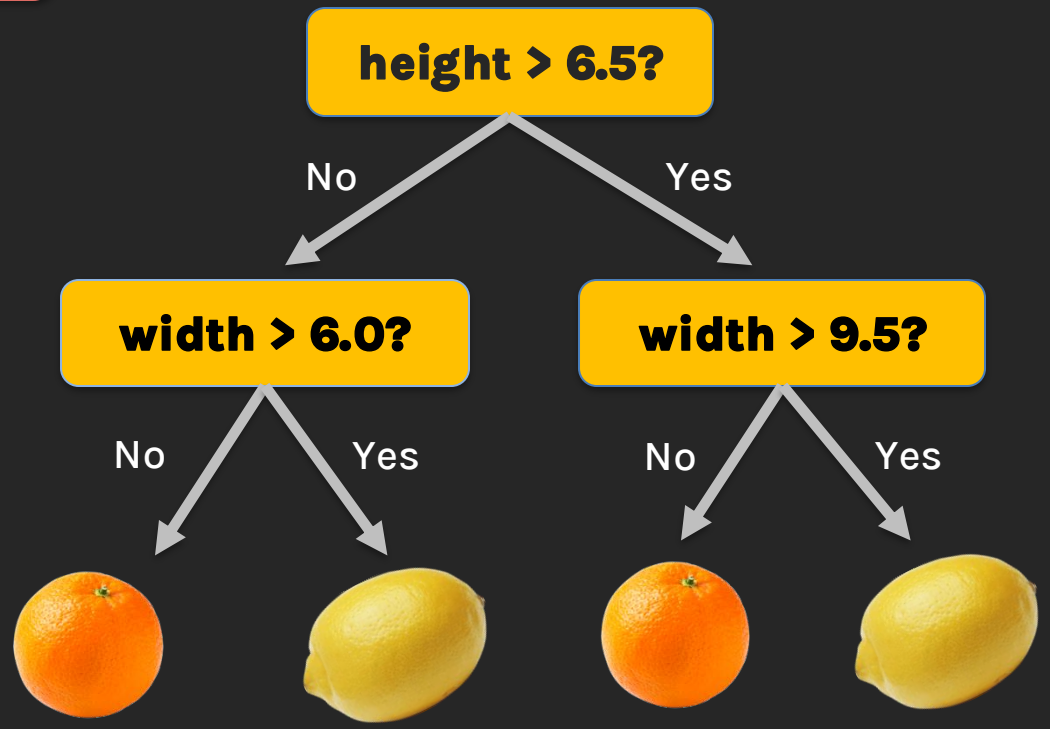
Let’s use the decision tree from before to make predictions.



Predictions

?

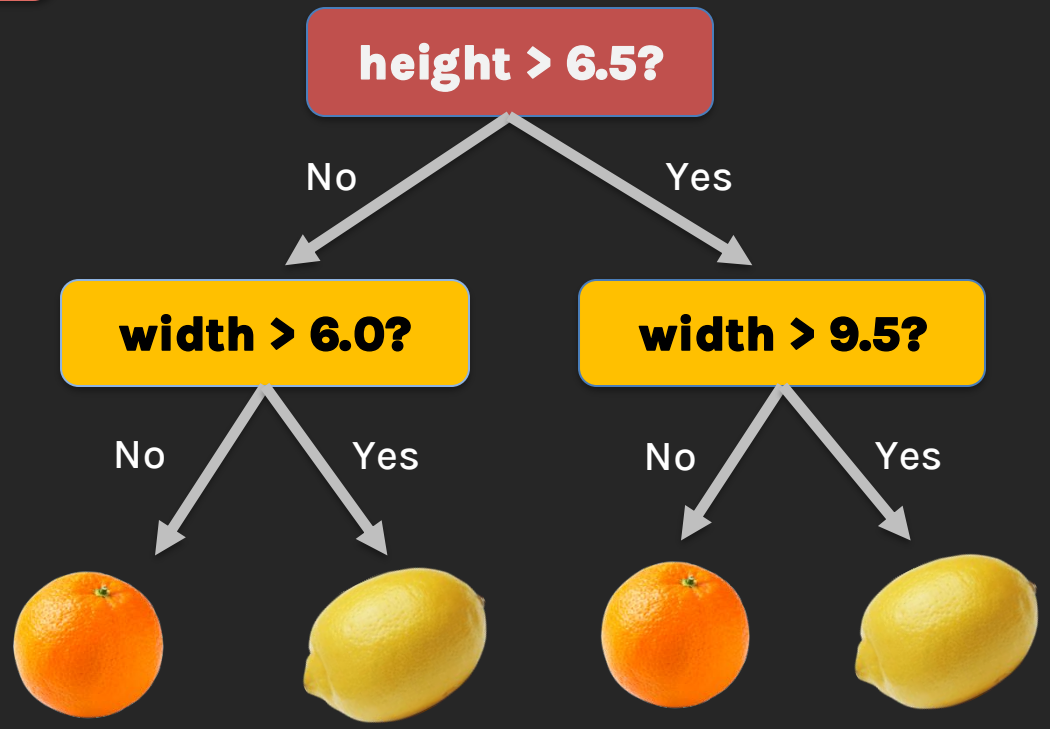
height = 5.9
width = 5.8



Predictions

?

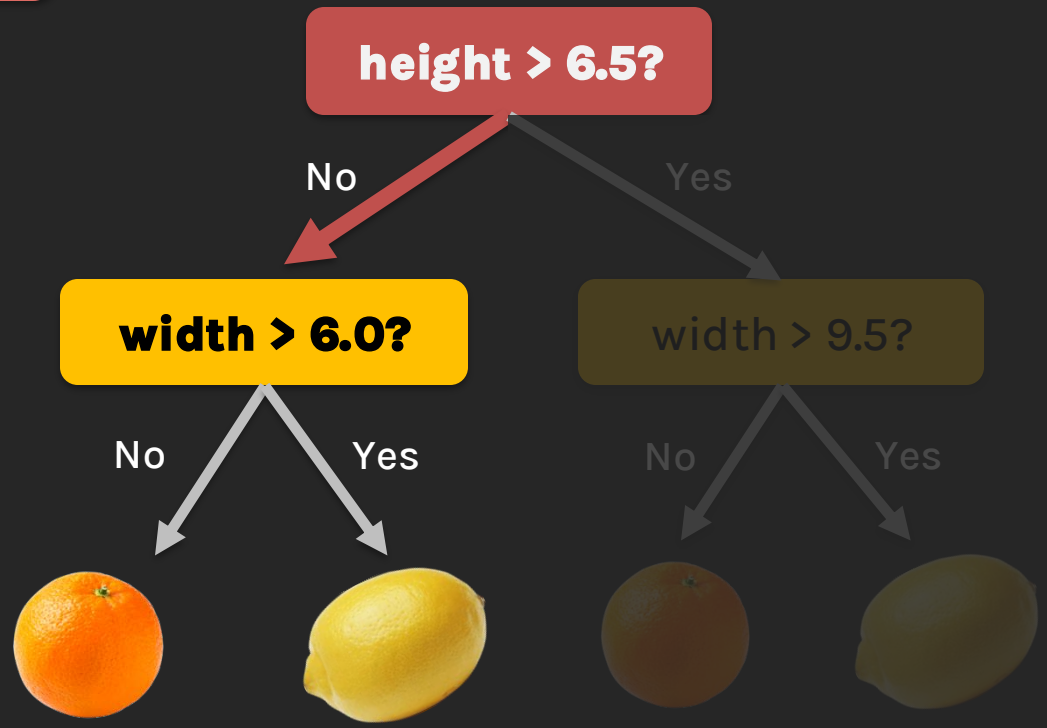
height = 5.9
width = 5.8



Predictions

?

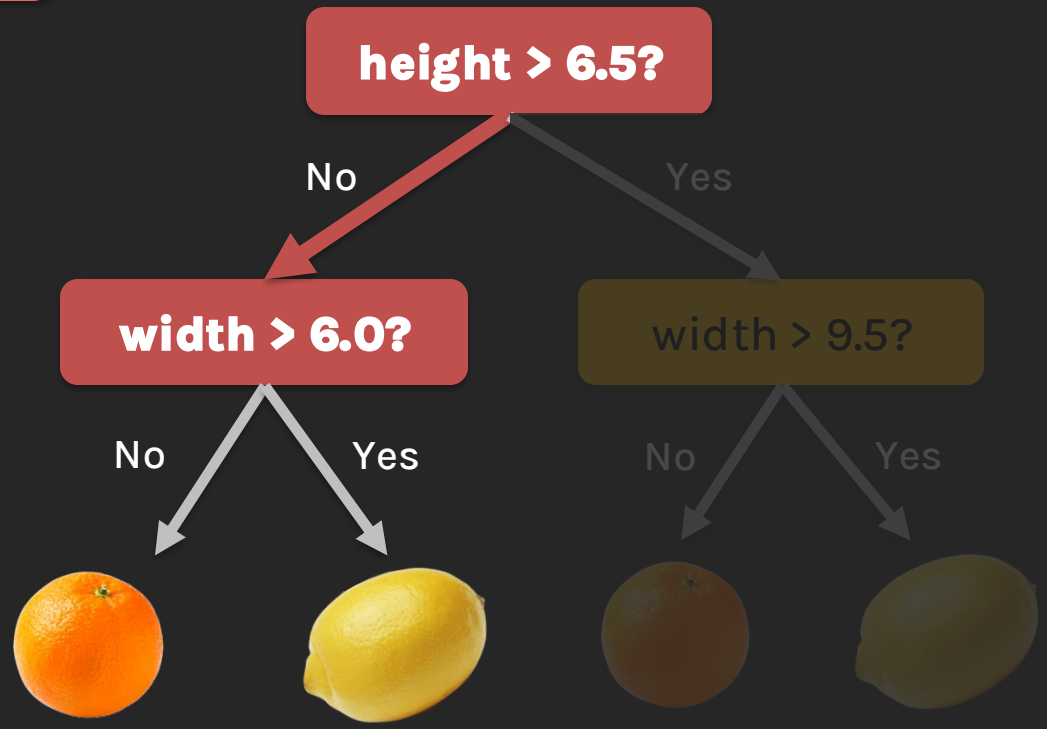
height = 5.9
width = 5.8



Predictions

?

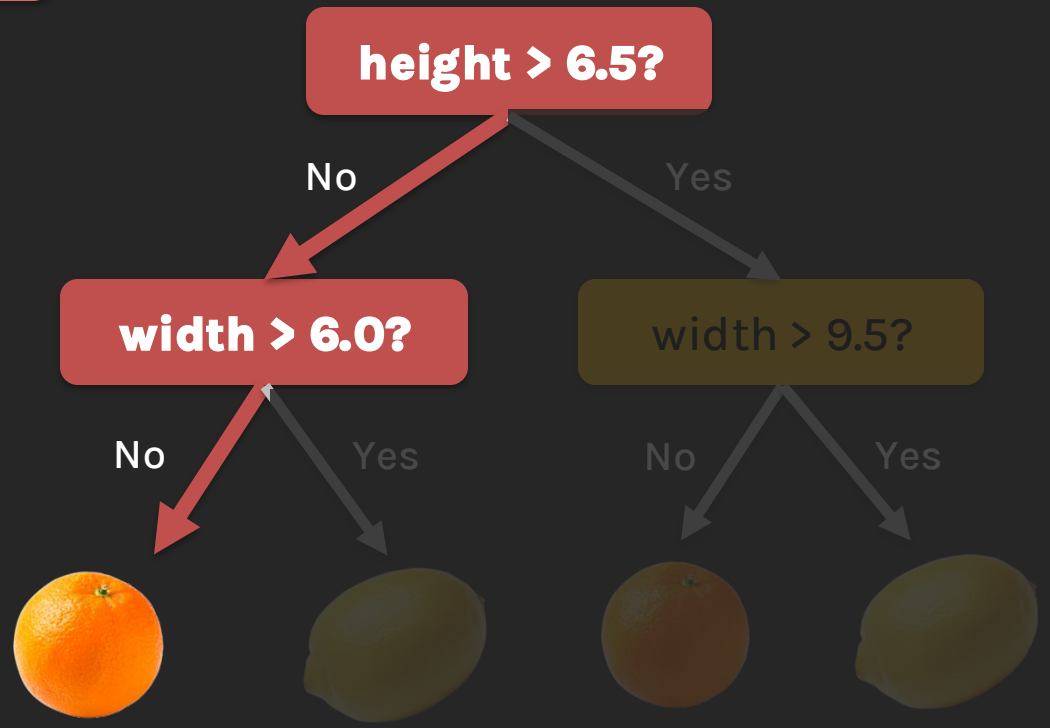
height = 5.9
width = 5.8



Predictions

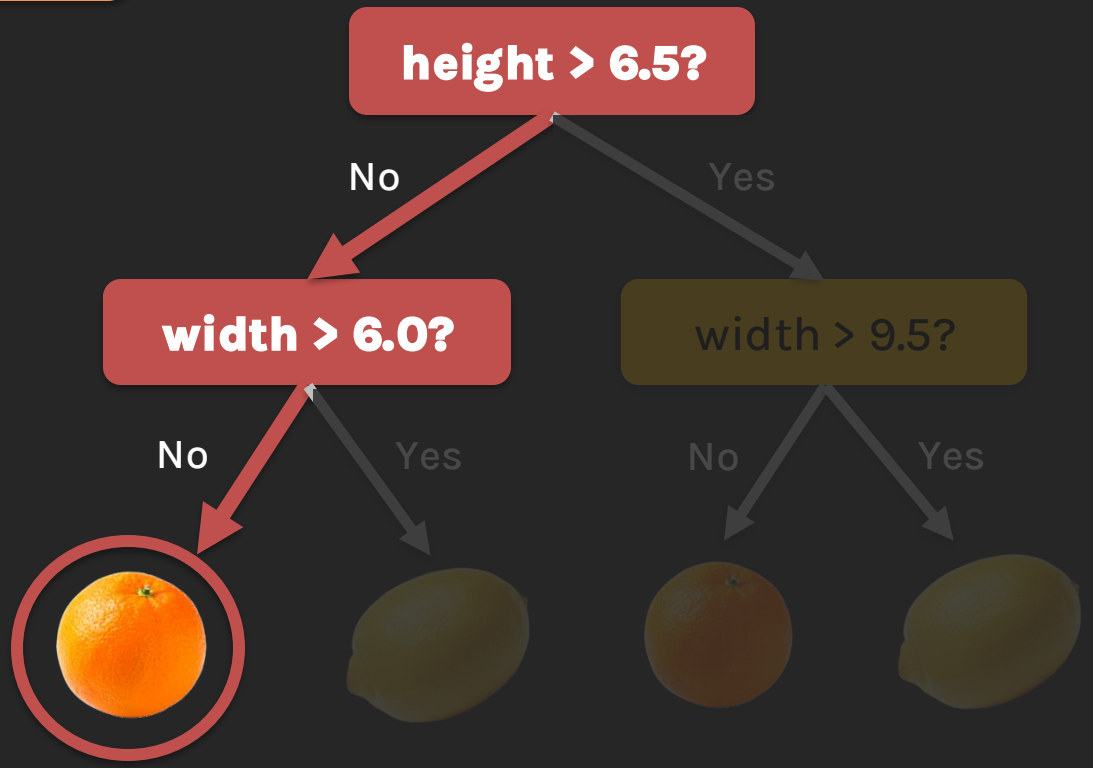
?

height = 5.9
width = 5.8



Predictions

height = 5.9
width = 5.8

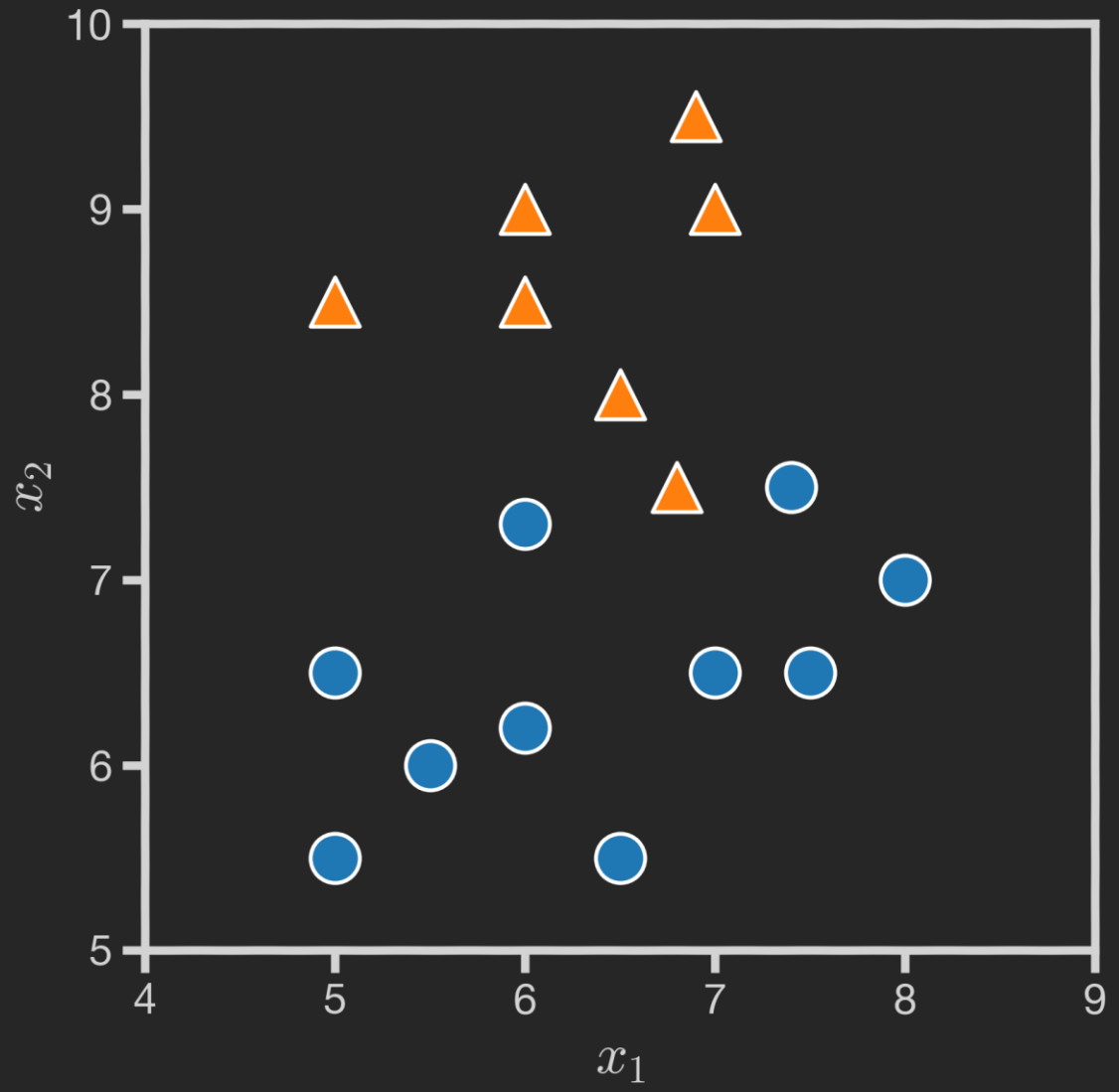


THIS METHOD IS ALSO CALLED TRAVERSING THE TREE

Outline

- Motivation
- **Decision Trees – Classification**
  - Intuition
  - Predictions
  - **Splitting Criteria**
  - Stopping Conditions

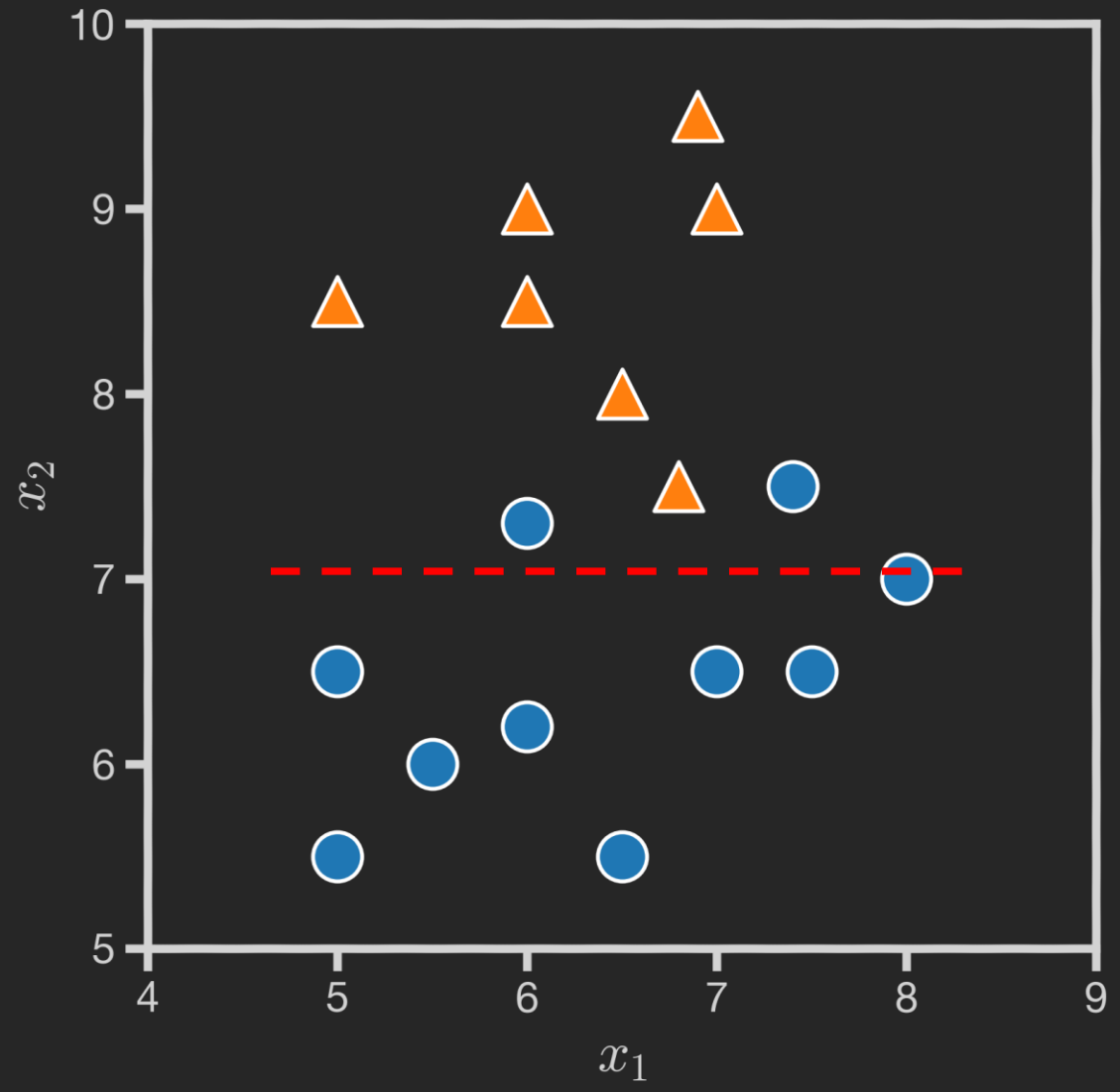
Splitting Criteria



Which gives a better split?

- Should it be split along x_1 or x_2?
- Where should we split?

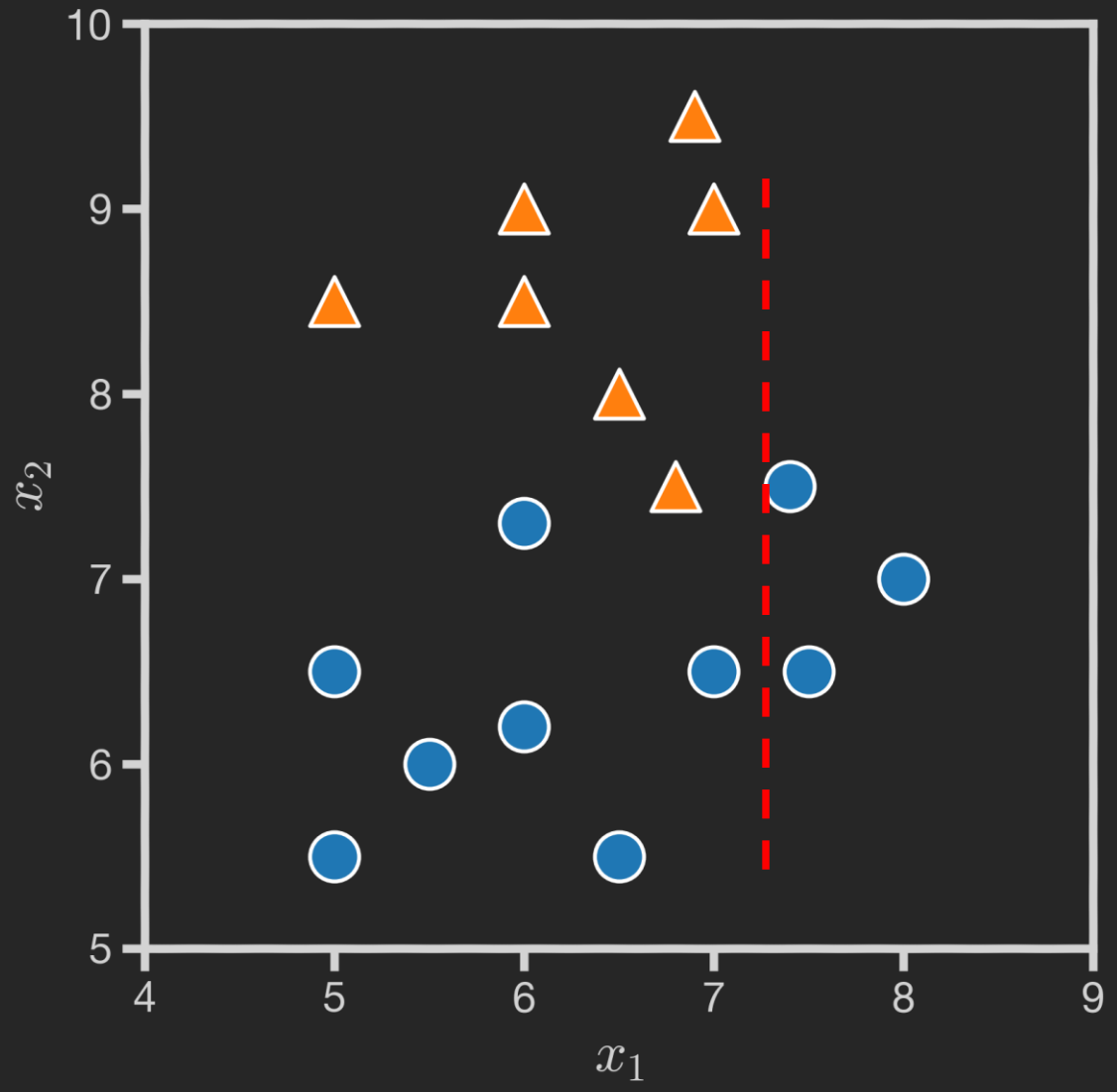
Splitting Criteria



Which gives a better split?

- Should it be split along x_1 or x_2?
- Where should we split?

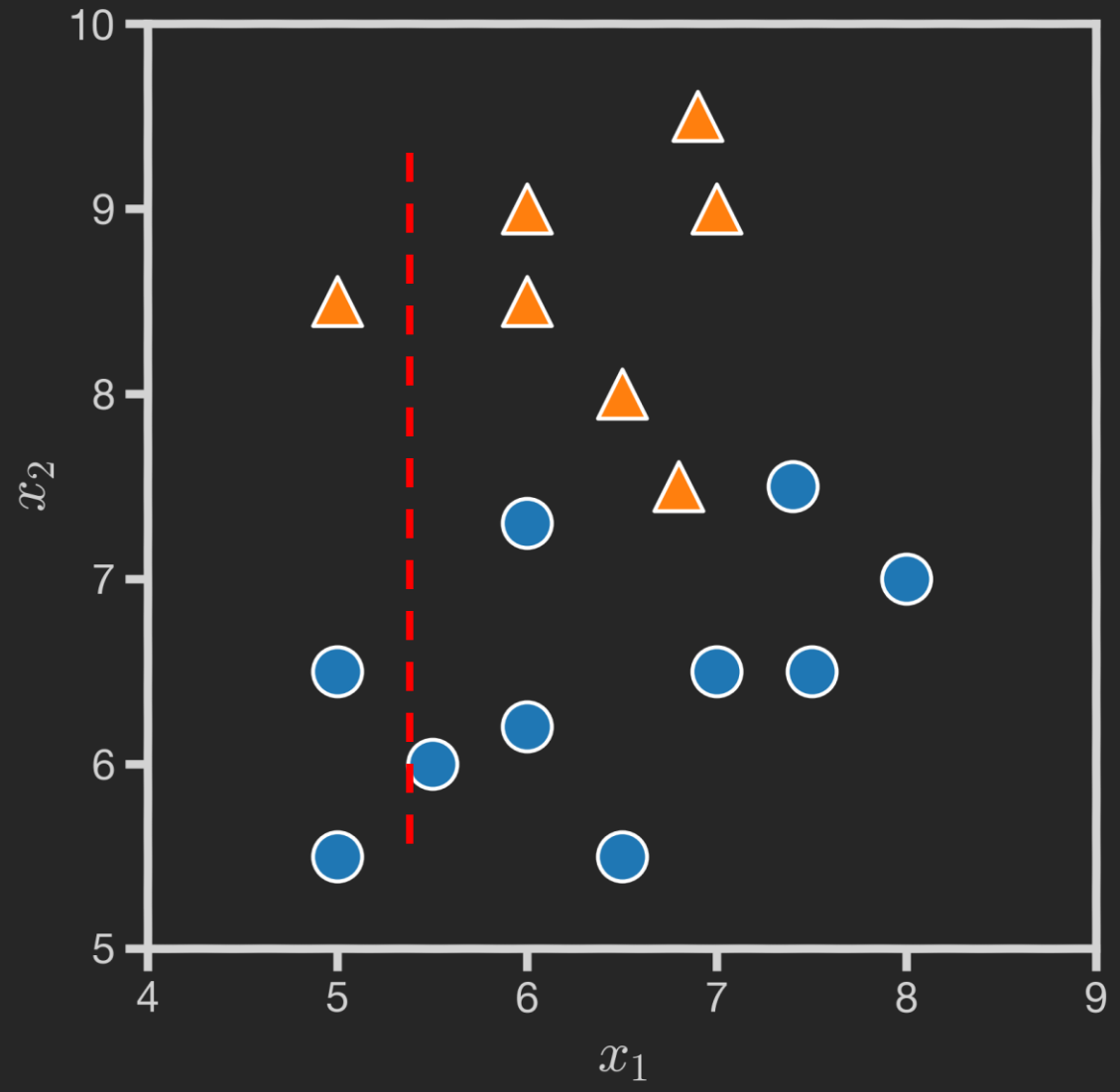
Splitting Criteria



Which gives a better split?

- Should it be split along x_1 or x_2?
- Where should we split?

Splitting Criteria



Which gives a better split?

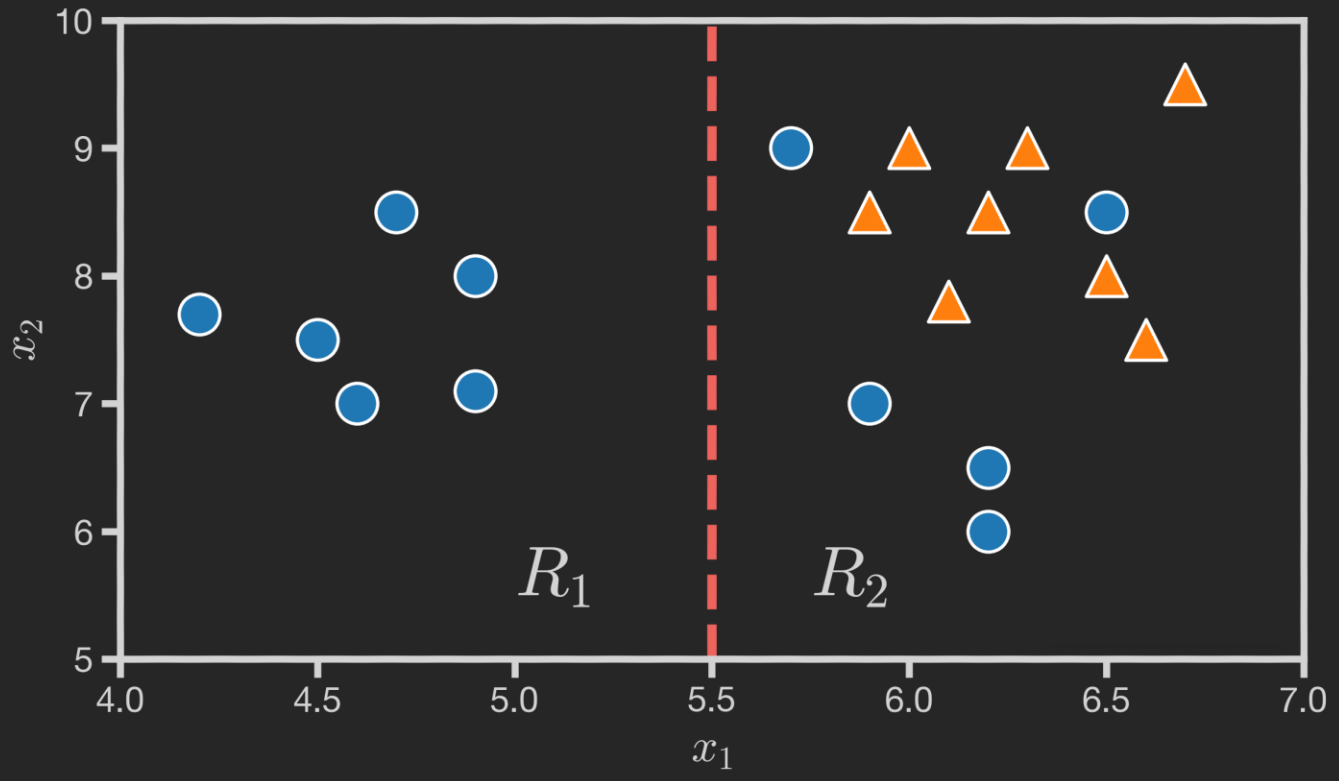
- Should it be split along x_1 or x_2?
- Where should we split?

Optimality of Splitting

While there is no ‘correct’ way to define an optimal split, there are some commonsense guidelines for every splitting criterion:

- The regions in the feature space should grow progressively **purer** with the number of splits. That is, we should see that each region ‘specializes’ towards a single class.
- We should end up with **no empty regions** – every region should contain training points.
- The splitting criterion of a split could take a **differentiable or non-differentiable form**.

Classification Error



Consider region R_2:

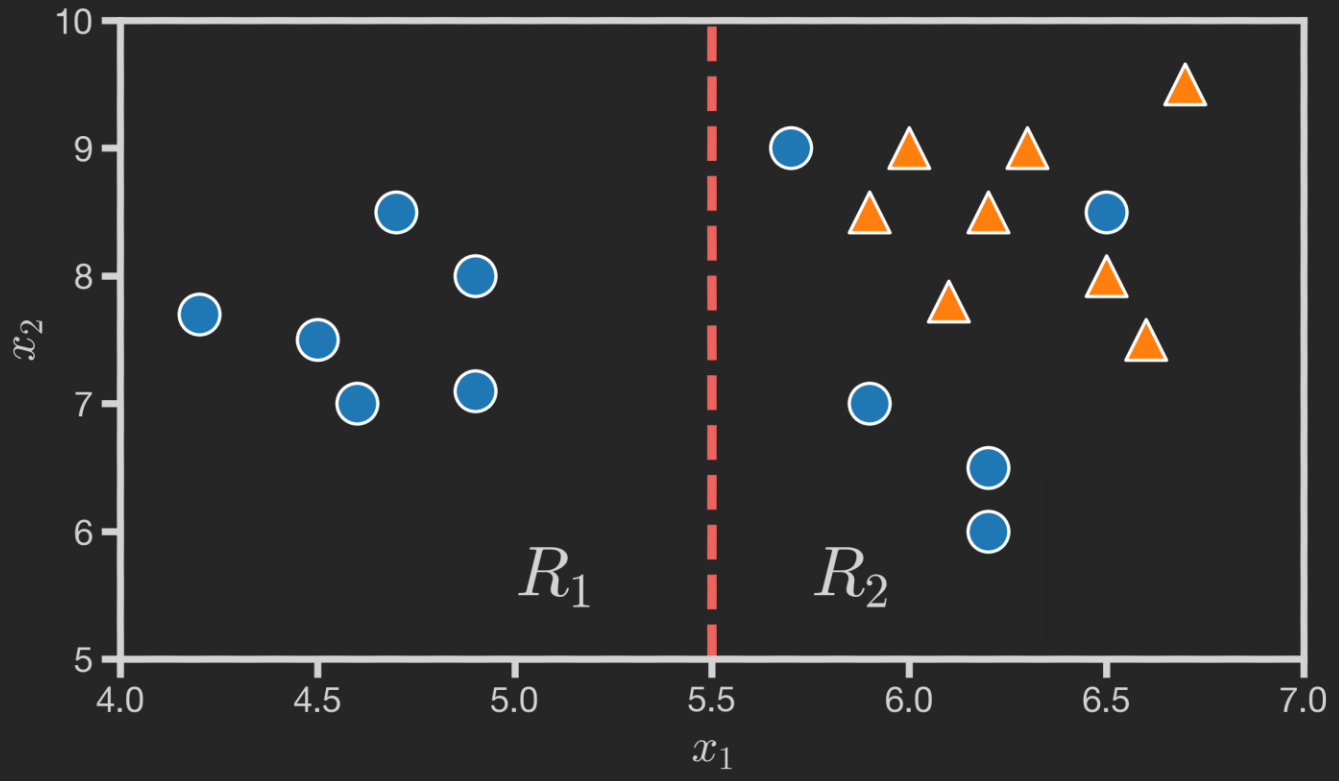
$$\text{Classification error} = \frac{error}{total}$$

$$= \frac{Number\ of\ minority\ class\ data\ points}{Total\ number\ of\ data\ points}$$

$$= \frac{Number\ of\ \bullet}{Total\ number\ of\ data\ points}$$

$$= 1 - \frac{Number\ of\ majority\ data\ points}{Total\ number\ of\ data\ points}$$

Classification Error



Classification error in region R_2:

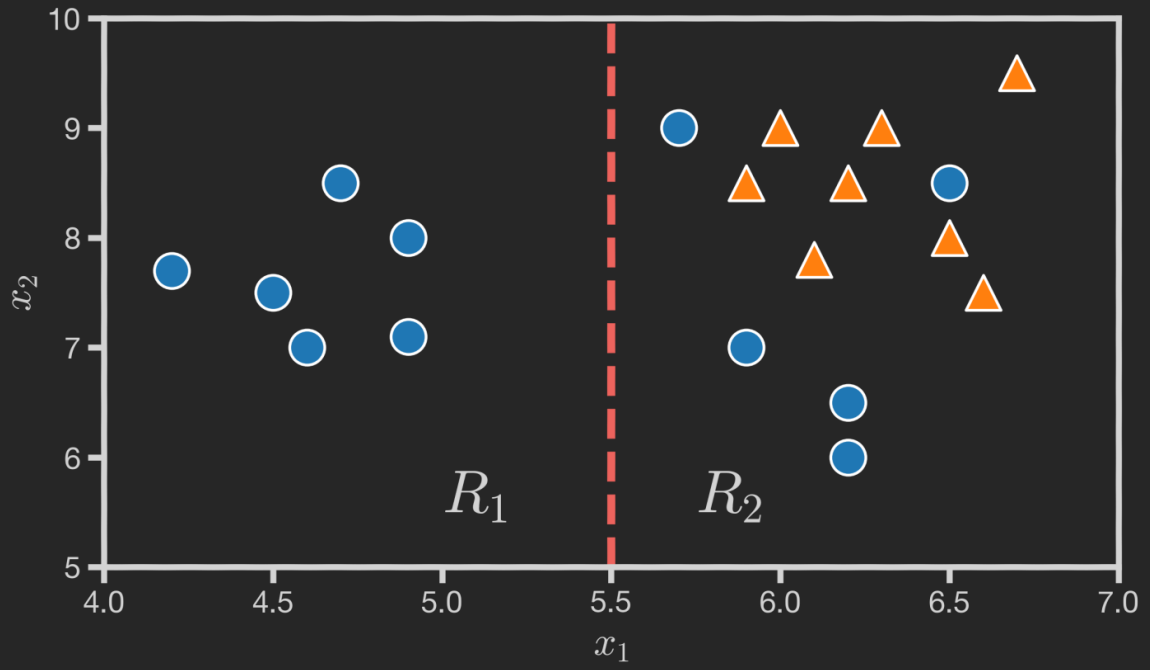
$$= 1 - \frac{Number\ of\ majority\ data\ points}{Total\ number\ of\ data\ points}$$

$$= 1 - \Psi(▲|R_2)$$

$$= 1 - \max_k(\Psi(k|R_r))$$

$\Psi(k|R_r)$ is the proportion of training points in R_2 that are labeled class k.

Classification Error



	●	▲	$Error = 1 - \max_k(\Psi(k\|R_r))$
R_1	6	0	$1 - max\left(\frac{6}{6}, \frac{0}{6}\right) = 0$
R_2	5	8	$1 - max\left(\frac{5}{13}, \frac{8}{13}\right) = \frac{5}{13} = 0.38$

Classification Error

In general:

Assume we have P **predictors (or features)** and K **classes**. Suppose we select the p^{th} predictor and split a region along the **threshold** t_p.

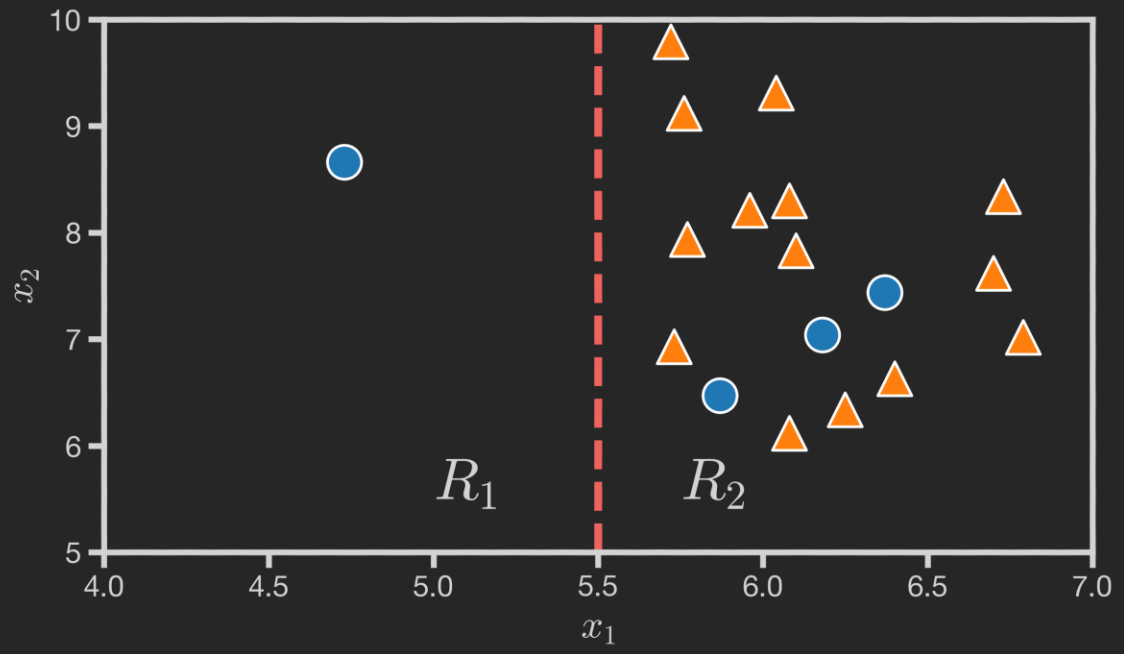
We can assess the quality of this split by calculating the **classification error** made by each newly created region:

$$Error\ (R_r|\ p, t_p) = 1\ - \max_k(\Psi(k|R_r))$$

where $\Psi(k|R_r)$ is the proportion of training points in R_r that are labeled class k.

Classification Error

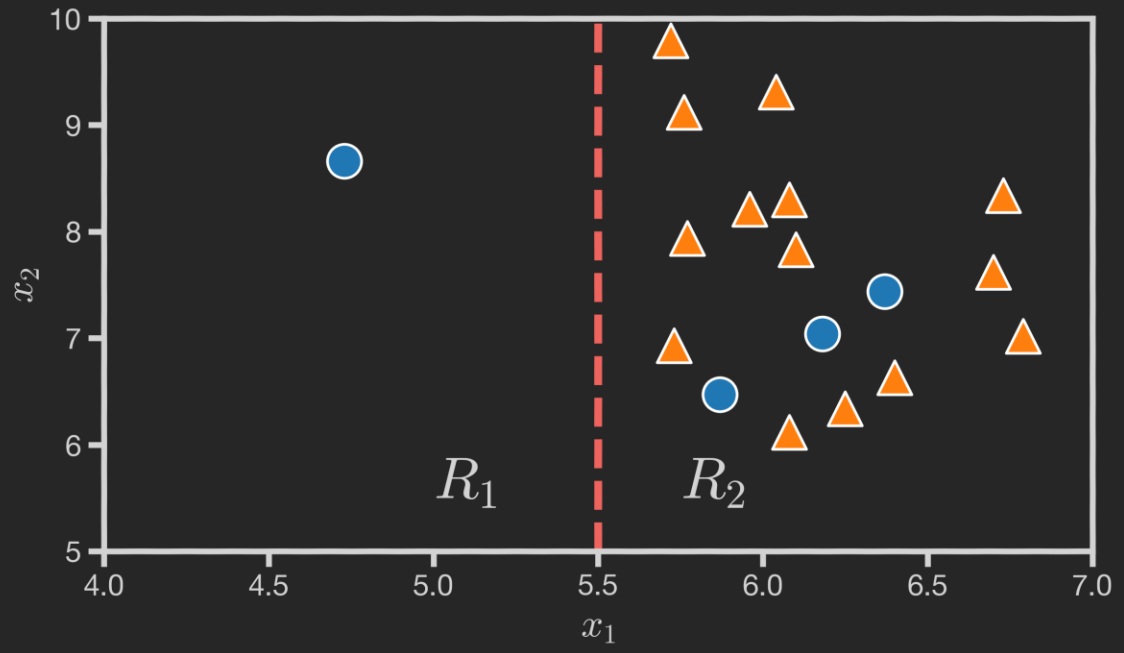
Now calculate the error for the following split:



	●	▲	$Error = 1 - \max_k(\Psi(k\|R_r))$
R_1	1	0	$1 - max\left(\frac{1}{1}, \frac{0}{1}\right) = 0$
R_2	3	14	$1 - max\left(\frac{3}{17}, \frac{14}{17}\right) = \frac{3}{17} = 0.18$

R_1 has a smaller error that R_2.
Does that mean this is a good split?

Classification Error



We need to take the **weighted average** over both regions so the number of points in each region is taken into consideration:

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

where N_r is the number of training points inside region R_r.

Gini Index

Assume we have P predictors and K classes. Suppose we select the p^{th} predictor and split a region along the threshold t_p.

We can assess the quality of this split by measuring the Gini Index made by each newly created region by calculating:

1 – sum of the squares of the proportions of points from the k-th class in the region r (PSI).

$$Gini\ (R_r|\ p, t_p) = 1\ - \sum_k \Psi(k|R_r)^2$$

Question: What is the effect of squaring the proportions of each class?

Game time

Which of the following statements best describes the effect of squaring the proportions of each class when calculating the Gini Index?

Options:

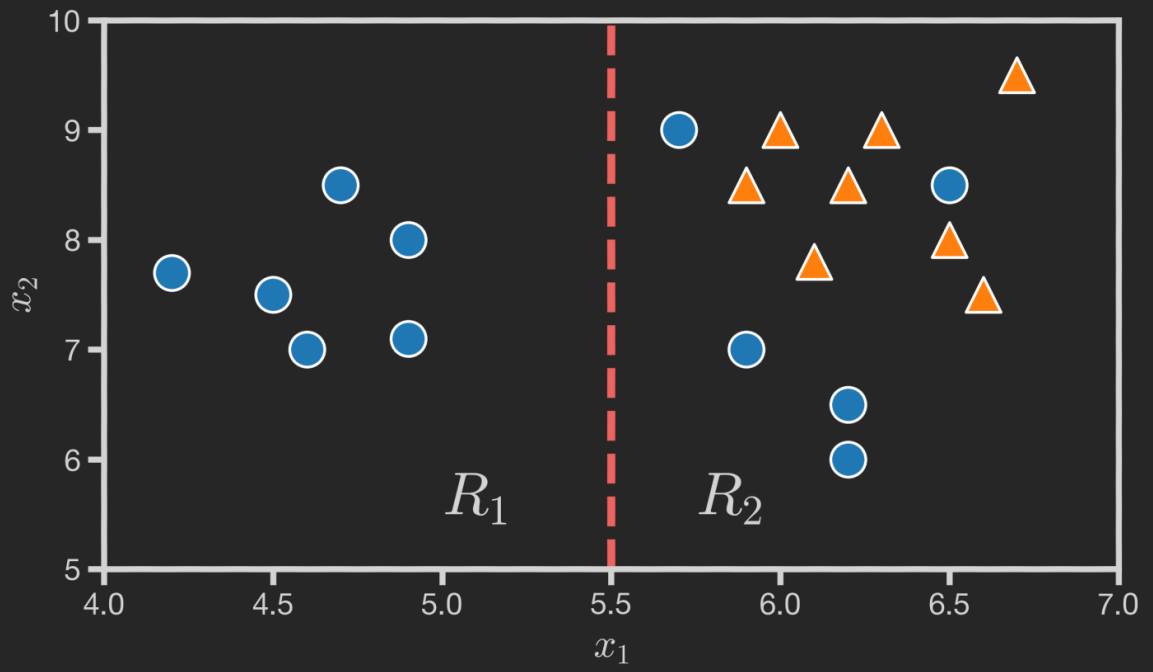
A) Squaring the proportions magnifies the influence of majority classes in a region, thereby overstating their significance.

B) The squaring process accentuates the contrast between pure regions and mixed ones, enabling the index to better differentiate between quality splits.

C) The act of squaring diminishes the index's sensitivity to minor fluctuations in class distributions, making it more stable.

D) Squaring the proportions directly correlates the Gini Index with the count of predictors $\boldsymbol{P}$, biasing the selection of split predictors.

Gini Index



	●	▲	$Gini = 1 - \sum_k \Psi(k\|R_r)^2$
R_1	6	0	$1 - \left[\left(\frac{6}{6}\right)^2 + \left(\frac{0}{6}\right)^2\right] = 0$
R_2	5	8	$1 - \left[\left(\frac{5}{13}\right)^2 + \left(\frac{8}{13}\right)^2\right] = \frac{80}{169} = \mathbf{0.47}$

Compared to 0.38 when we used **classification error**

Gini Index

We can now try to find the predictor p and the threshold t_p that minimize the **weighted average Gini Index** over the two regions:

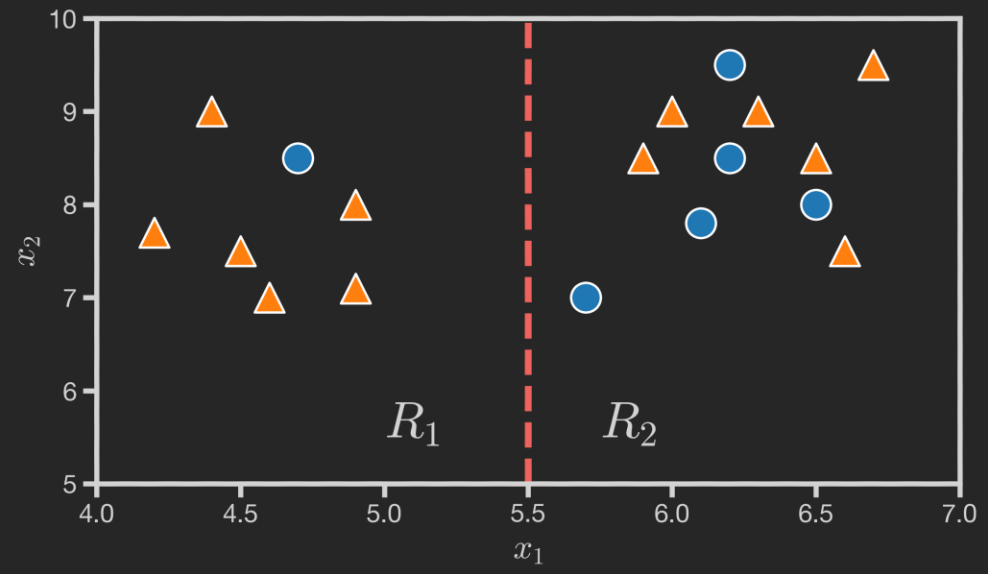
$$\min_{p,t_p}\left[\frac{N_1}{N}Gini\left(R_1|p\,,t_p\right)+\frac{N_2}{N}Gini\left(R_2|p\,,t_p\right)\right]$$

where $N_{1,2}$ is the number of training points inside region $R_{1,2}$.

Information Theory

The last metric for evaluating the quality of a split is motivated by metrics of uncertainty in information theory.

Question: In the below plot, which region is 'purer'?



While both regions are impure, R_1 clearly sends a **stronger 'signal'** for class 2 than R_2.

Information Theory

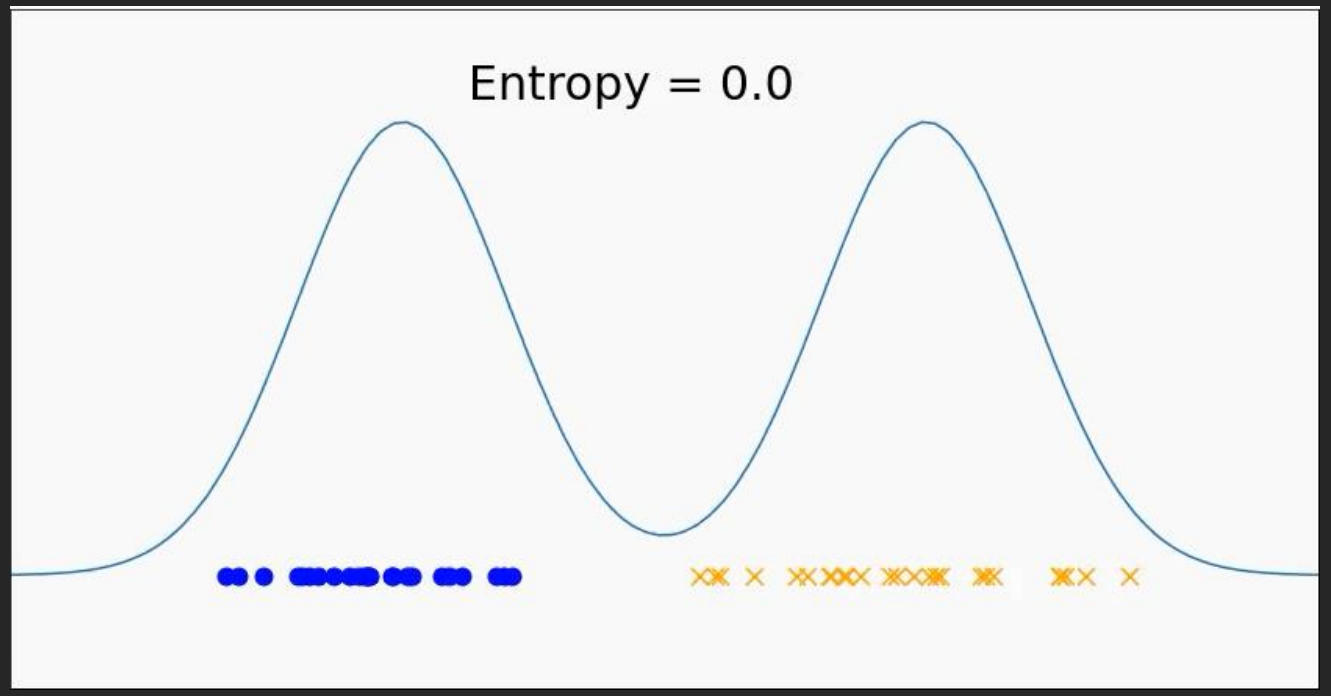
One way to quantify the strength of a signal in a particular region is to analyze the **distribution of classes** within the region. We compute the **entropy** of this distribution.

For a random variable with a discrete distribution, the entropy is computed by:

$$H(x) = -\sum_{x \epsilon X} \Psi(x) \log_2 \Psi(x)$$

Information Theory

$$H(x) = -\sum_{x \epsilon X} \Psi(x) \log_2 \Psi(x)$$



Entropy

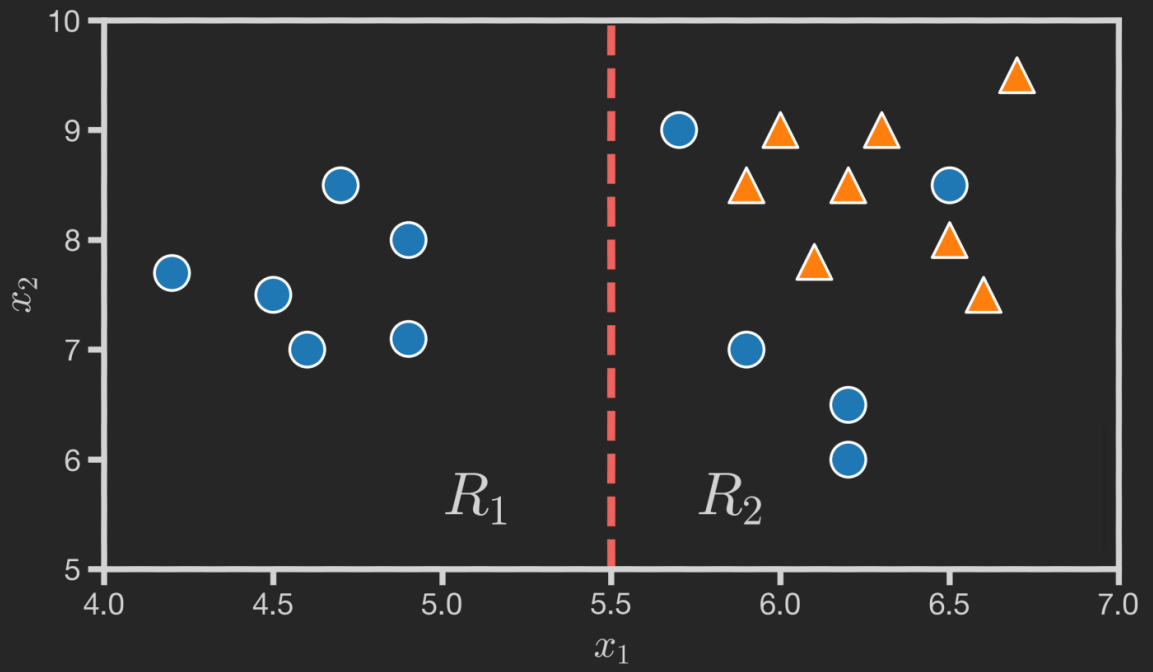
Assume we have P predictors and K classes. Suppose we select the p^{th} predictor and split a region along the **threshold** t_p.

We can assess the quality of this split by measuring the entropy of the class distribution in each newly created region by calculating:

$$Entropy\,(R_r|\,p,\,t_p) = -\sum_k \Psi(k|R_r)\log_2\Psi(k|R_r)$$

Note: We are actually computing the conditional entropy of the distribution of training points amongst the K classes given that the point is in region r.

Entropy



	●	▲	$Entropy = -\sum_k \Psi(k\|R_r)\log_2 \Psi(k\|R_r)$
R_1	6	0	$-\left[\frac{6}{6}\log_2\frac{6}{6}+\frac{0}{6}\log_2\frac{0}{6}\right]=0$
R_2	5	8	$-\left[\frac{5}{13}\log_2\frac{5}{13}+\frac{8}{13}\log_2\frac{8}{13}\right]\approx 1.38$

Entropy

Error Measurement	Value
Classification Error	0.38
Gini Index	0.47
Entropy	1.38

Entropy

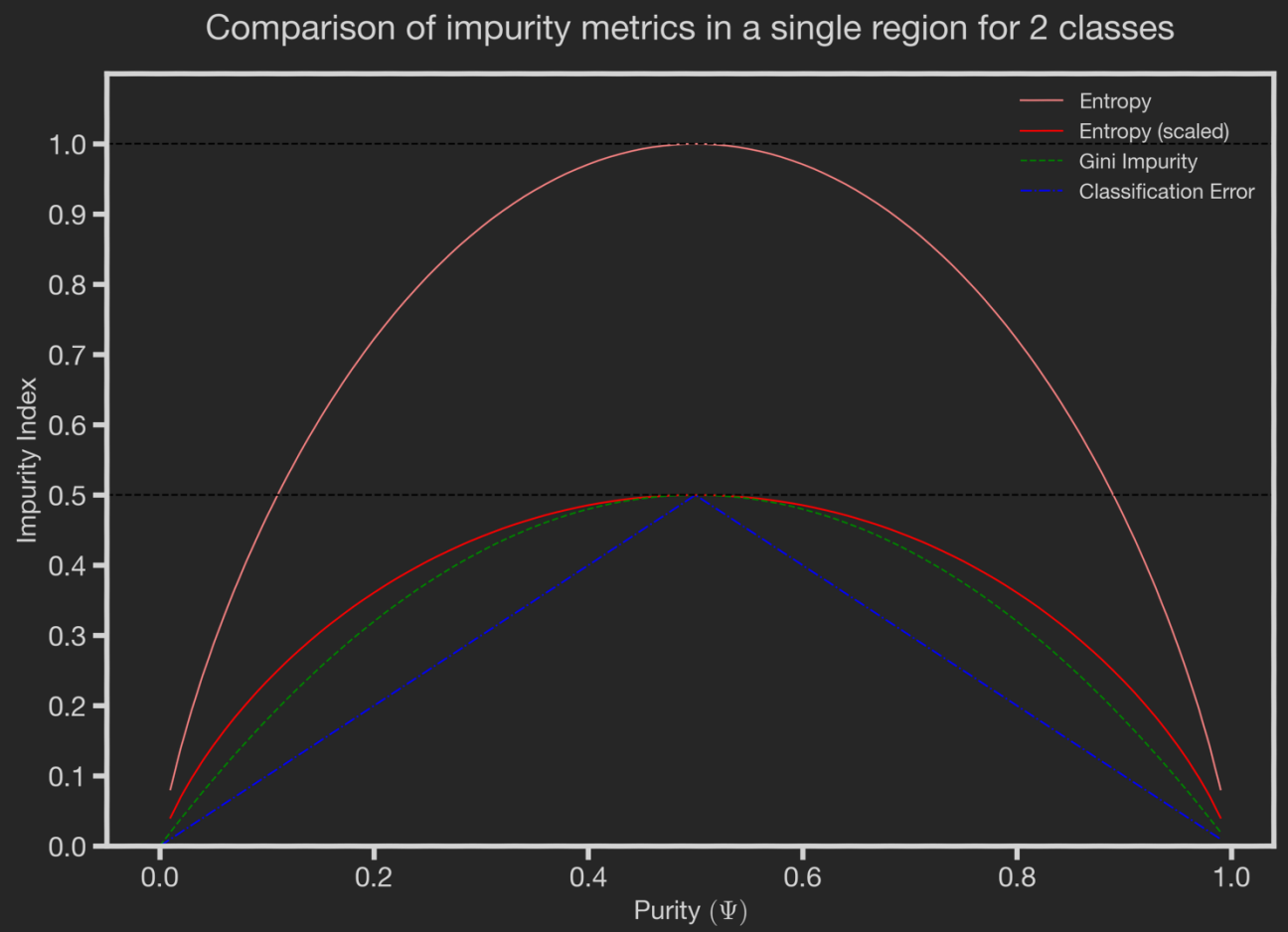
We can now try to find the predictor p and the threshold t_p that minimize the **weighted average entropy** over the two regions:

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

where $N_{1,2}$ is the number of training points inside region $R_{1,2}$.

Comparison of Criteria

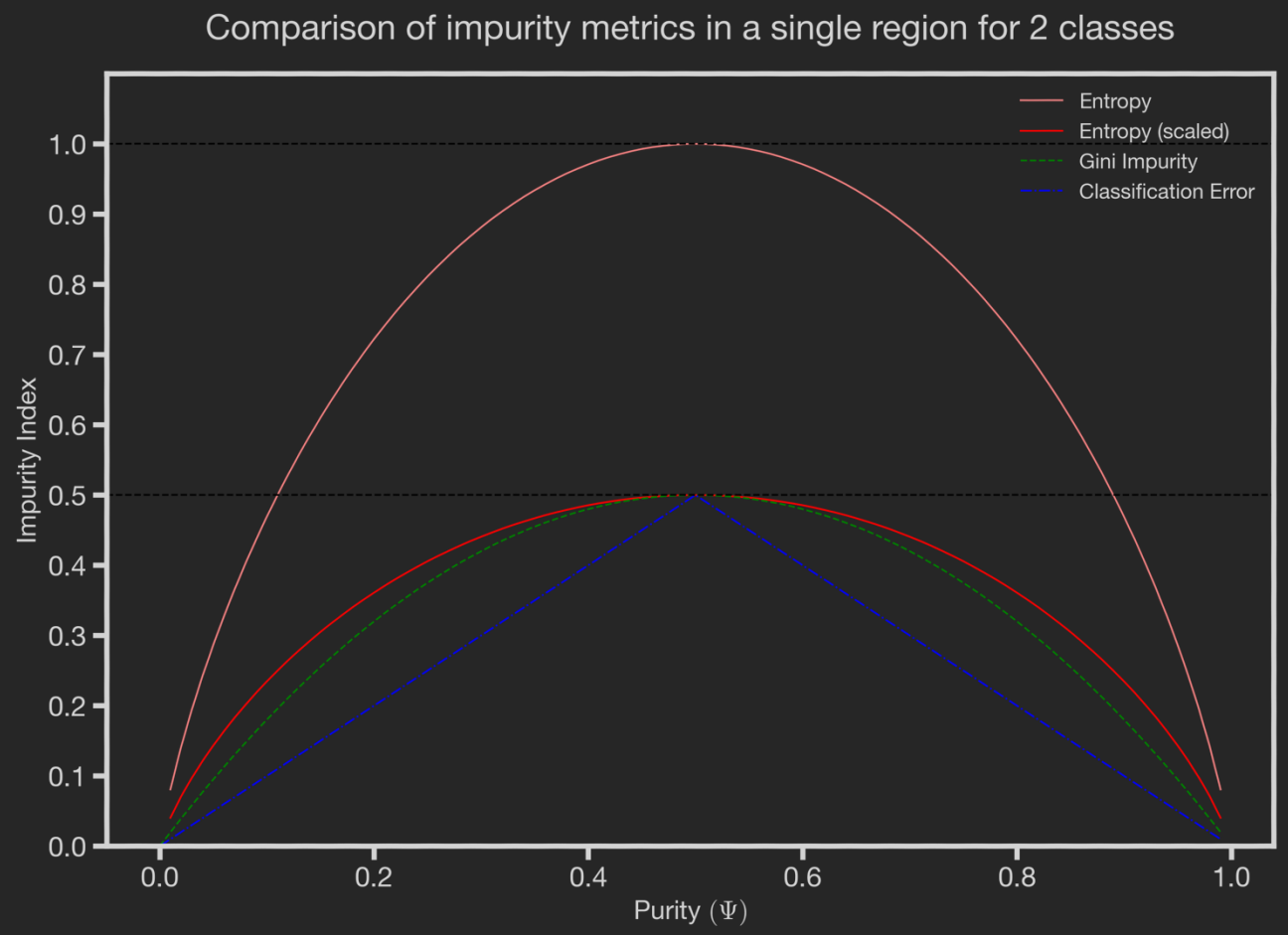
Question: Which of the three criteria fits our guideline the best?



Note: Sklearn uses the unscaled entropy as a splitting criterion

Comparison of Criteria

Question: Which of the three criteria fits our guideline the best?



Entropy **penalizes impurity the most**

Learning the Model

Learning the ‘optimal’ decision tree for any given set of data is NP complete (intractable) for numerous simple definitions of ‘optimal’. Instead, we will use a greedy algorithm.

1. Start with an empty decision tree (undivided feature space)
2. Choose the ‘optimal’ predictor on which to split and choose the ‘optimal’ threshold value for splitting.
3. Recurse on each new node until the stopping condition is met.
4. For the case of classification, predict each region to have a class label based on the largest class of the training points in that region (majority class).

Learning the Model

Recurse on each new node until the **stopping condition** is met.