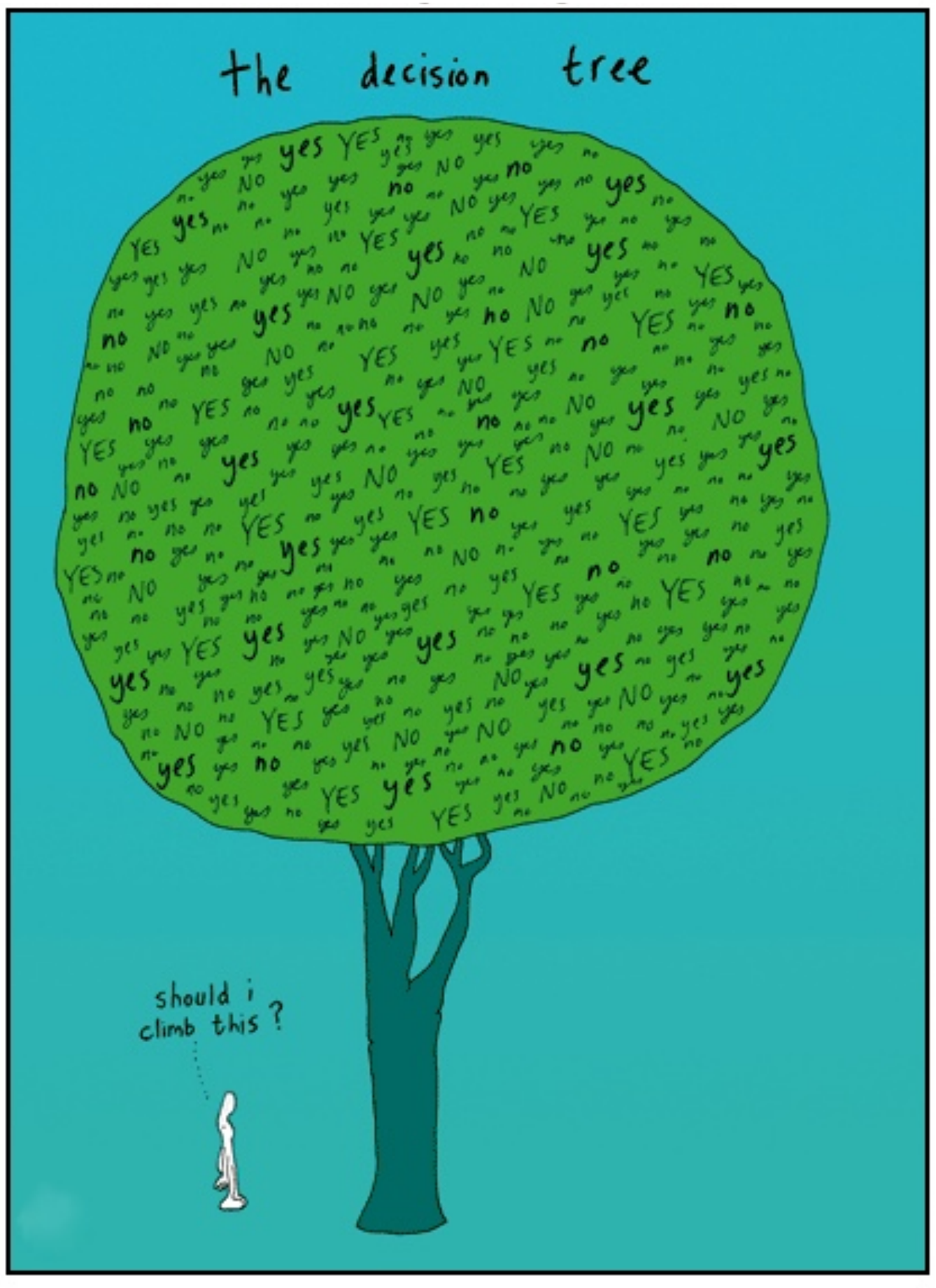
DECISION TREES

Dr. Brian J. Spiering



Where we are getting lunch?

What are the features of choosing lunch?

How much time?

Heathy or not?

Time

Little

Lots

Healthy

Healthy

Yes

No

Yes

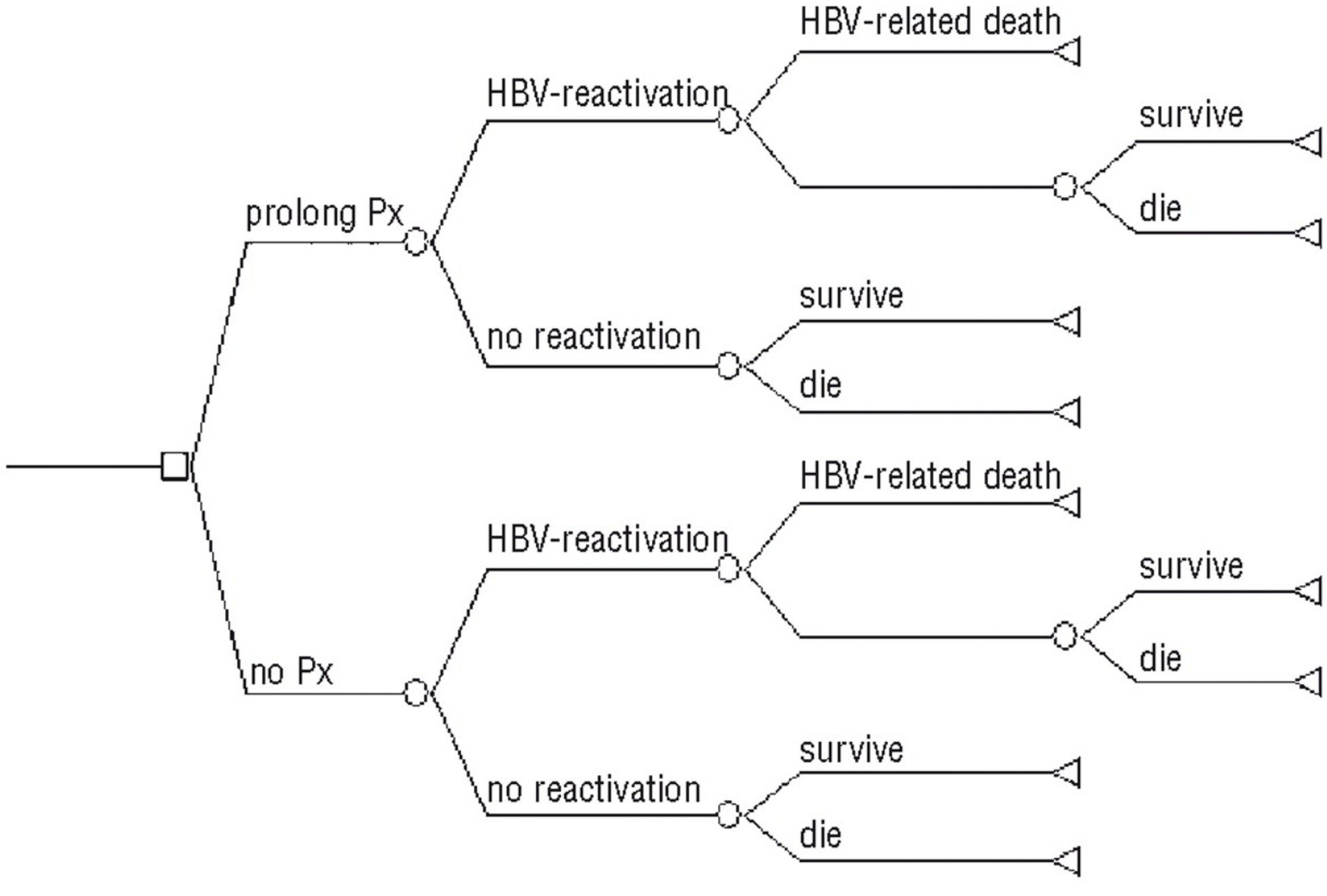
No

Outline

- Lecture
  - Overview
  - Building Decision Trees
  - Extending to Random Forest
- Lab
  - Walk through example
  - Small group activity

What is a Decision Tree?

A sequence of if-then choices which determine a path that ultimately leads to an outcome.



When to use a Decision Tree?

A supervised learning method,
thus need labeled data

What are the parts of a Decision Tree?

Root Node

Time

Little

Lots

Meeting

Yes

No

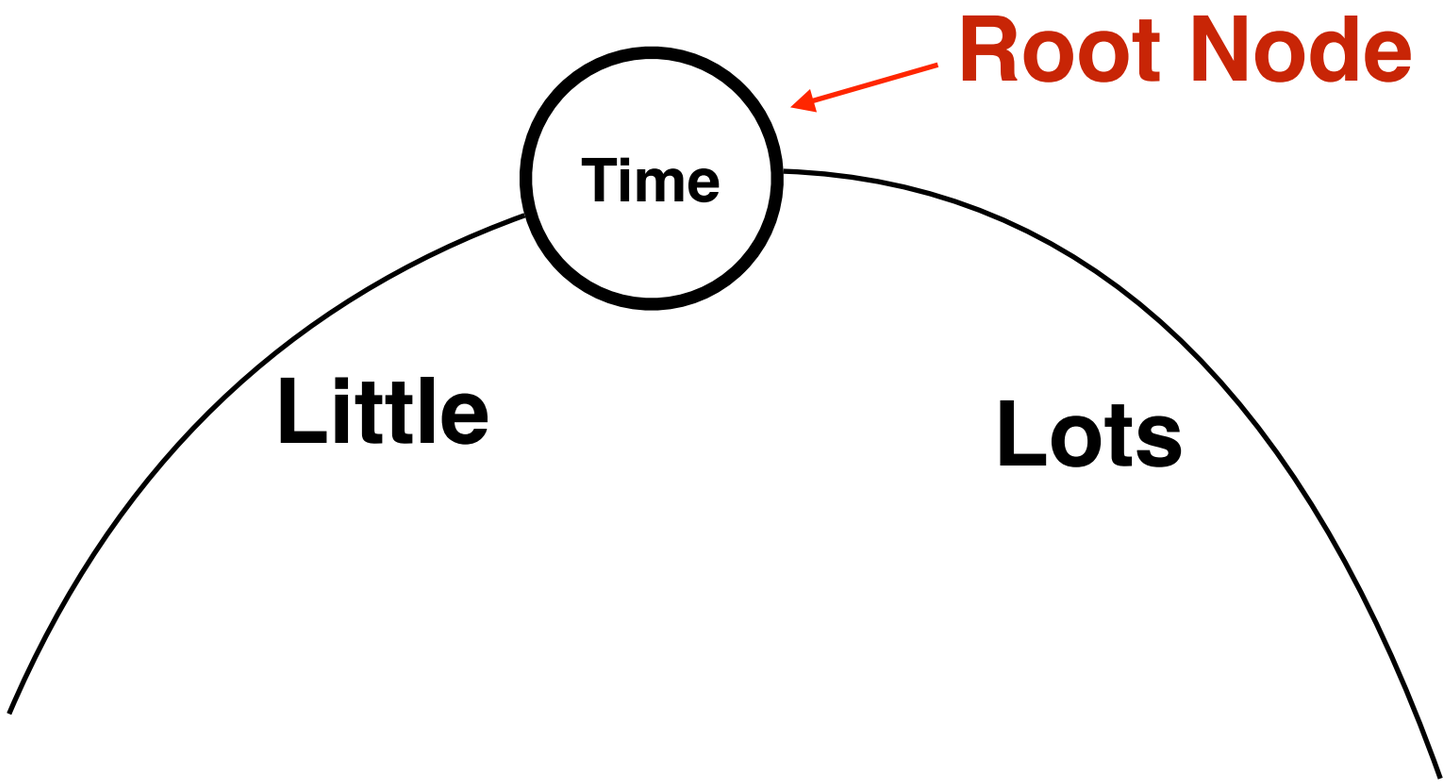
Terminal node (leaf)

Internal node

Meeting

Yes

No



Root Node

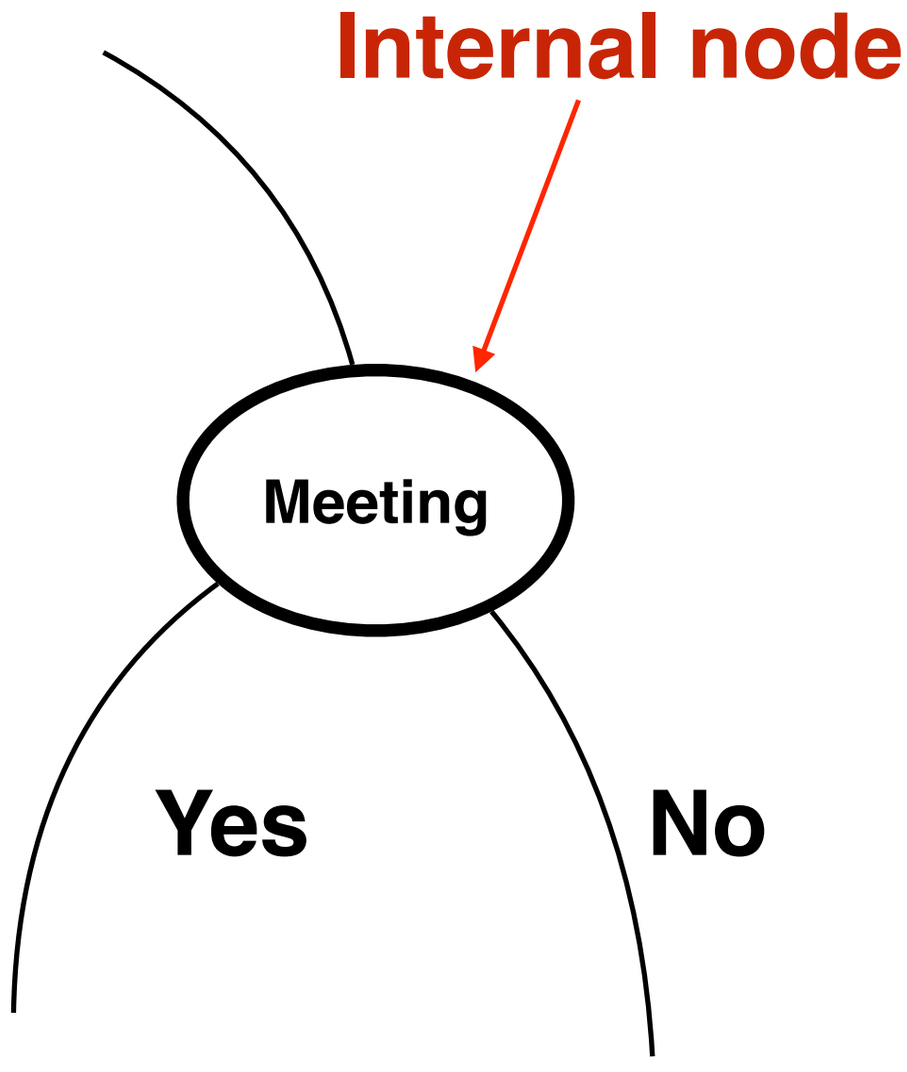
Topmost decision node

Single best predictor

Internal Node

A branch in the decision tree

aka, decision node

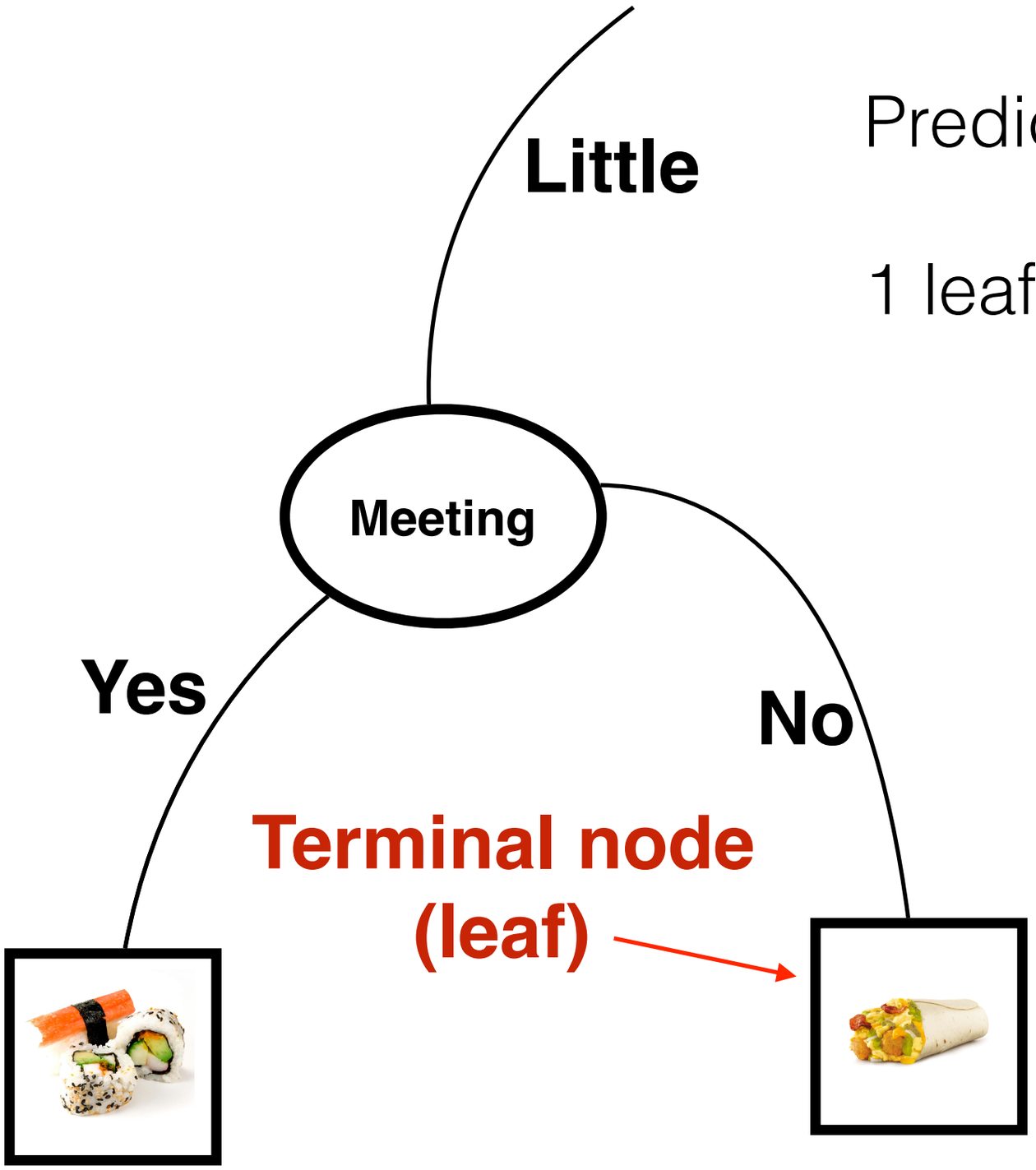


Terminal Node

Represents a decision or prediction

Predicted value for a region

1 leaf per region



Tinder
for
Kittens

SAN FRANCISCO
SPCA

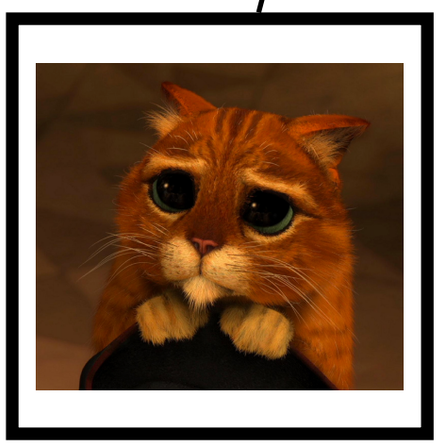
Swipe left or right?

How do we start to construct a decision tree?

Start simple: Make 1 choice

Claws

Yes:
Swipe Left



No:
Swipe Right

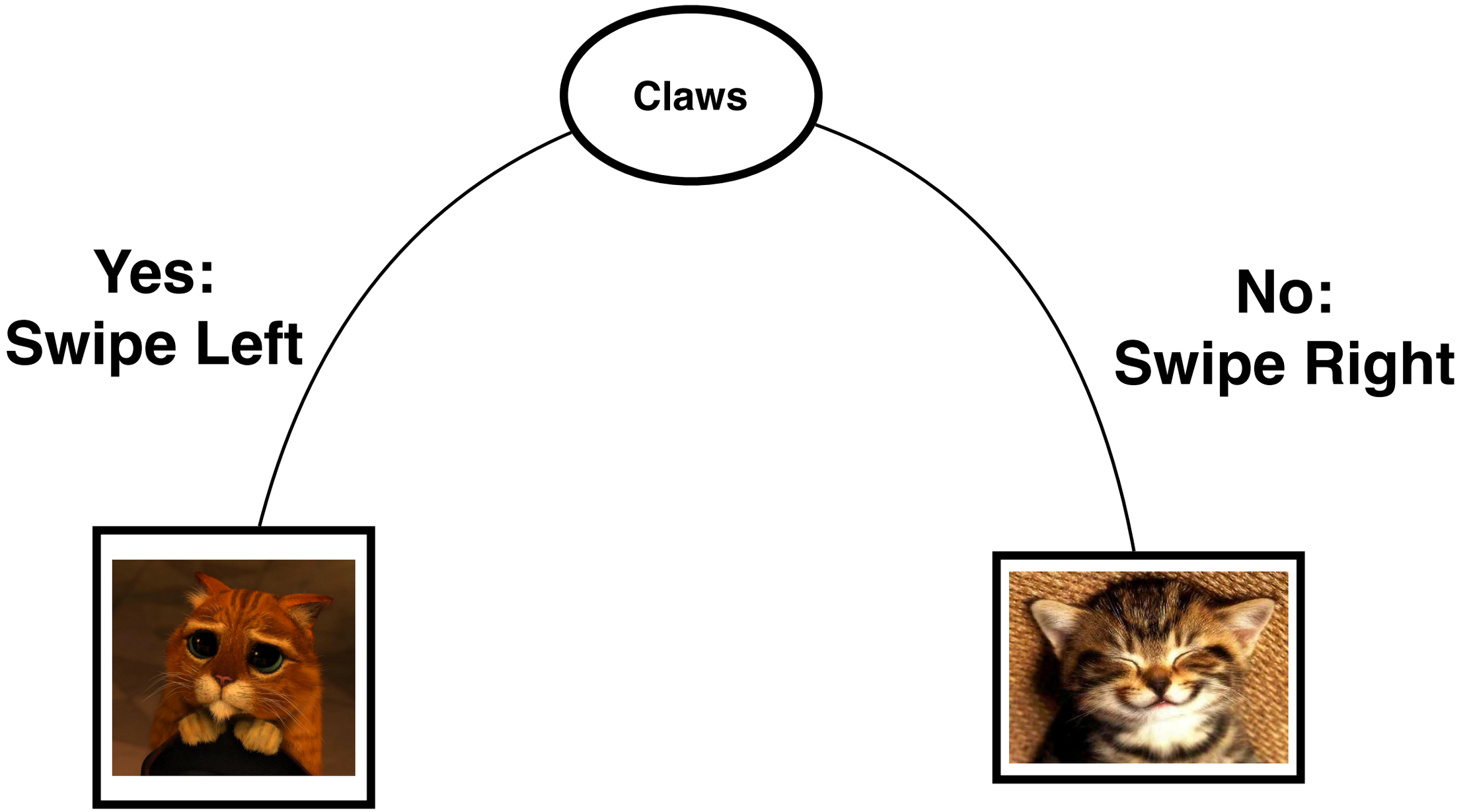
Decision Stumps

- One-level decision tree
- aka, 1-Rules
- Pick the most important feature that divides the data

Decision Stumps
are
weak learners.

Weak learners
are better than
random guessing.

The tree predicts the same label for each bottommost (leaf) partition.

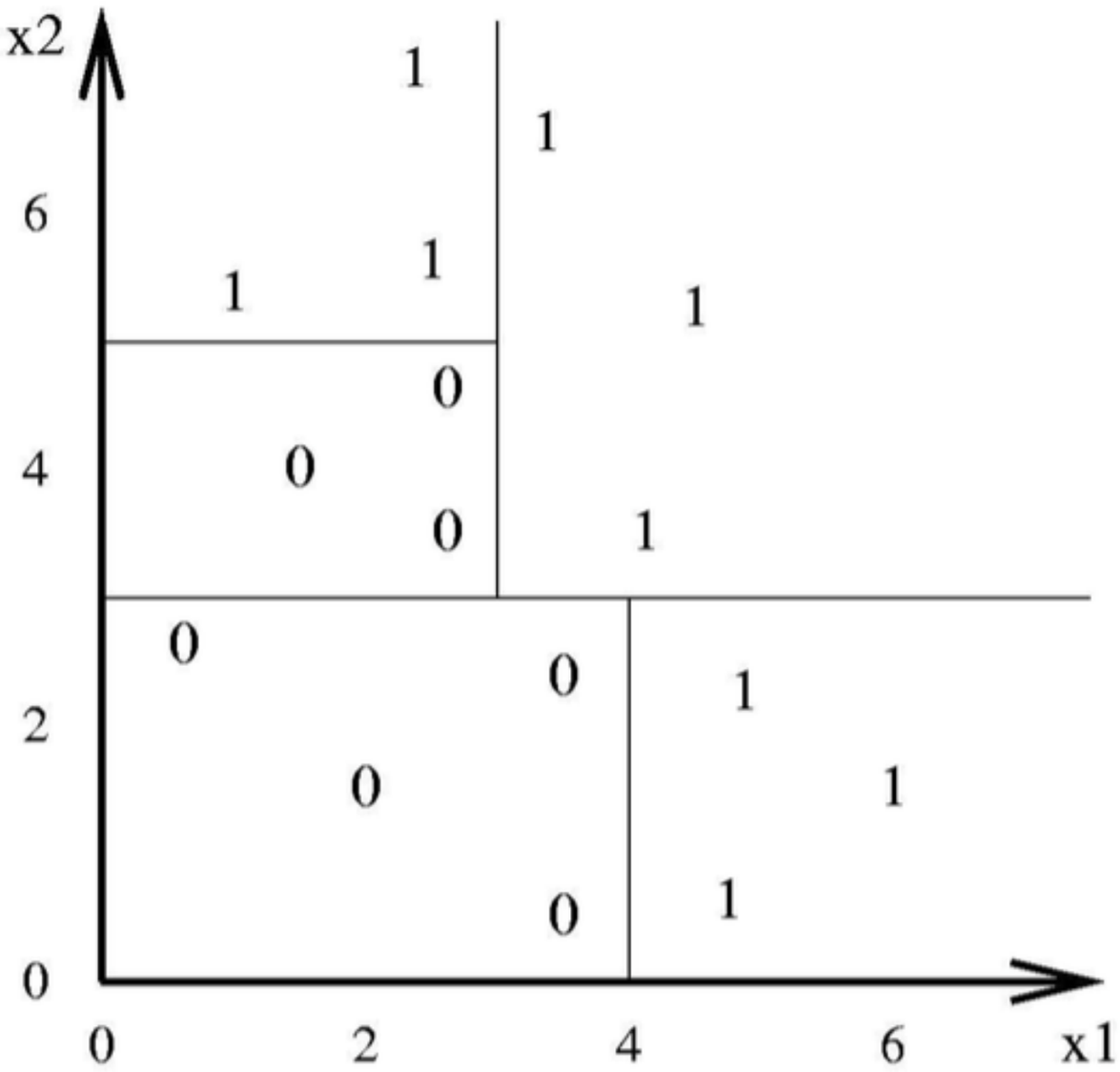


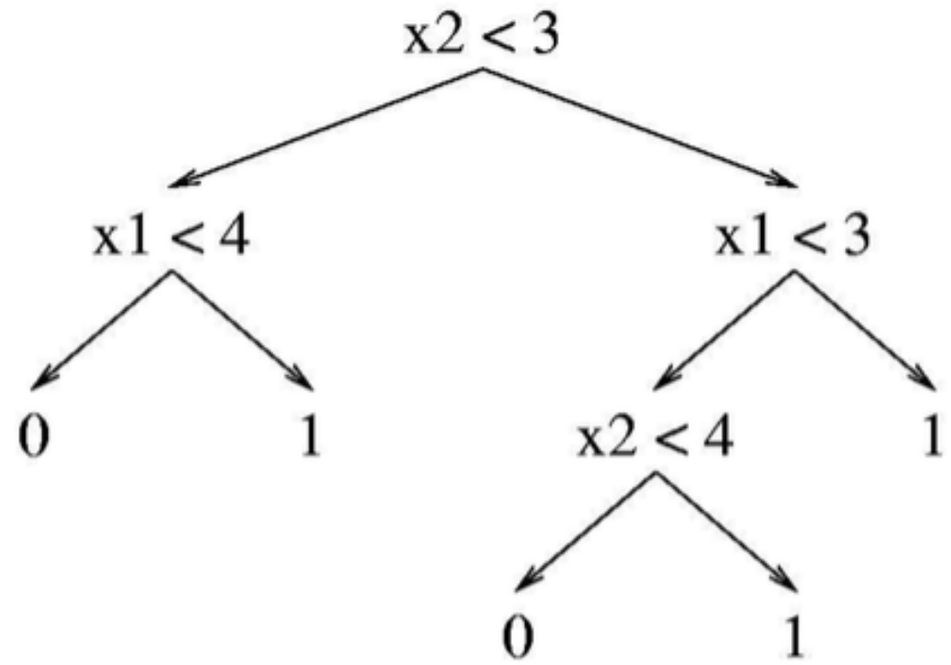
Divide up the input space

&

assign predictions to pieces of the space

Decision Trees Boundaries





Divide the feature space in axis parallel rectangle

Label each rectangle into a class

Each branch/subset should be as **pure** as possible

Each partition is chosen **greedily** by selecting the best split from a set of possible splits, in order to minimize the node impurity at a leaf.

Node impurity:

A measure of the homogeneity of the labels at the node

Gini Impurity

How often a randomly chosen element from the set would be incorrectly labeled, if it were randomly labeled according to the distribution of labels in the subset.

Gini Impurity

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - {f_i}^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} {f_i}^2 = 1 - \sum_{i=1}^{m} {f_i}^2$$

Summing the probability of each item being chosen times the probability of a mistake in categorizing that item.

Information Gain

The difference between the parent node impurity and the weighted sum of the two child node impurities.

Split s partitions the dataset D of size N into two datasets:

D_{left} and D_{right} of sizes N_{left} and N_{right}

The information gain is:

$$IG(D, s) = Impurity(D) - \frac{N_{left}}{N} Impurity(D_{left}) - \frac{N_{right}}{N} Impurity(D_{right})$$

Entropy

From Information Theory

The higher the entropy the more the information content.

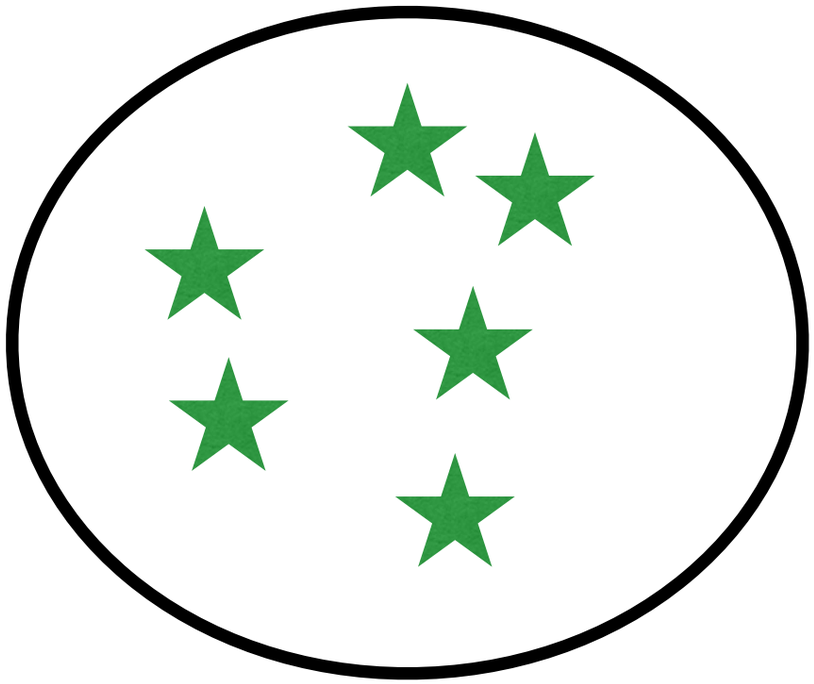
Entropy

$$I_E(f) = -\sum_{i=1} f_i \log_2 f_i$$

Entropy

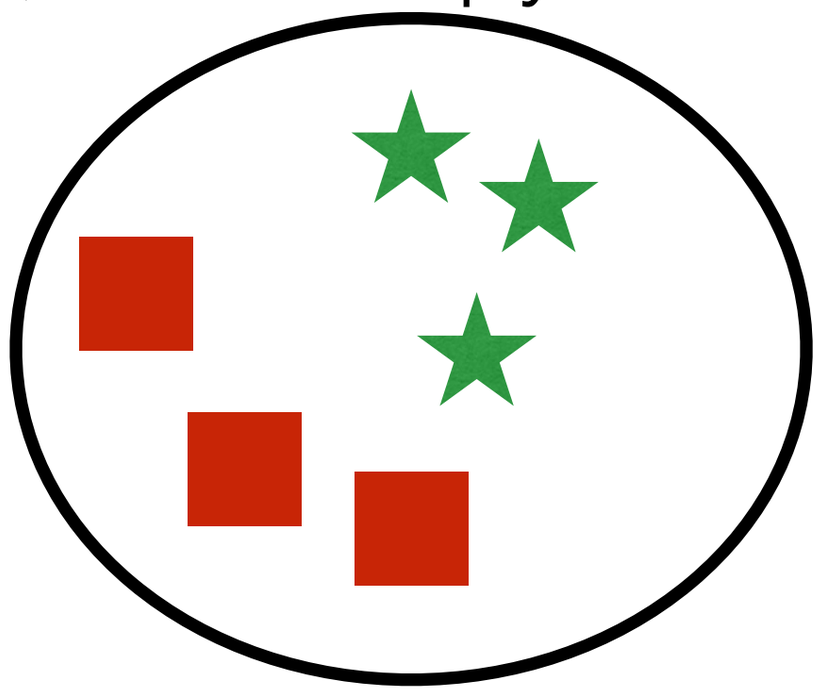
If sample is completely homogeneous, the entropy = 0.

$$-1 \log_2 1 = 0$$



If sample is completely heterogeneous, the entropy = 1.

$$-0.5 \log_2 0.5 = 1$$



Gini vs. Entropy

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i)$$

$$I_E(f) = -\sum_{i=1}^{m} f_i \log_2 f_i$$

Gini is the default (and slightly faster).

It doesn’t matter (in most cases).

When to stop splitting?

1. Minimal information gain for additional split
2. Reached maximum numbers of splits (aka, depth of tree)
3. No split has minimal number of instances in child nodes

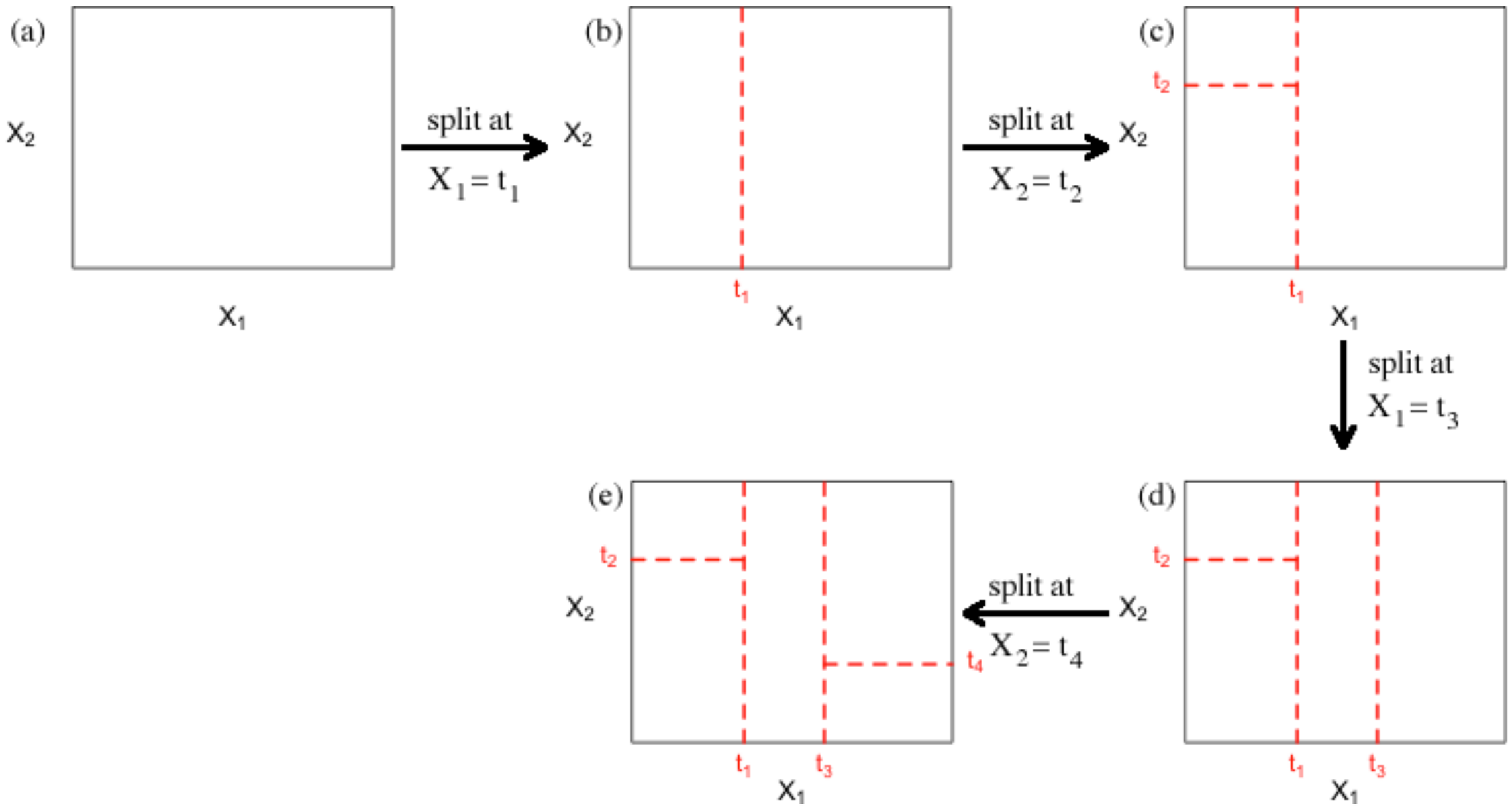
Building Decision Trees

A top-down, greedy algorithm

that performs

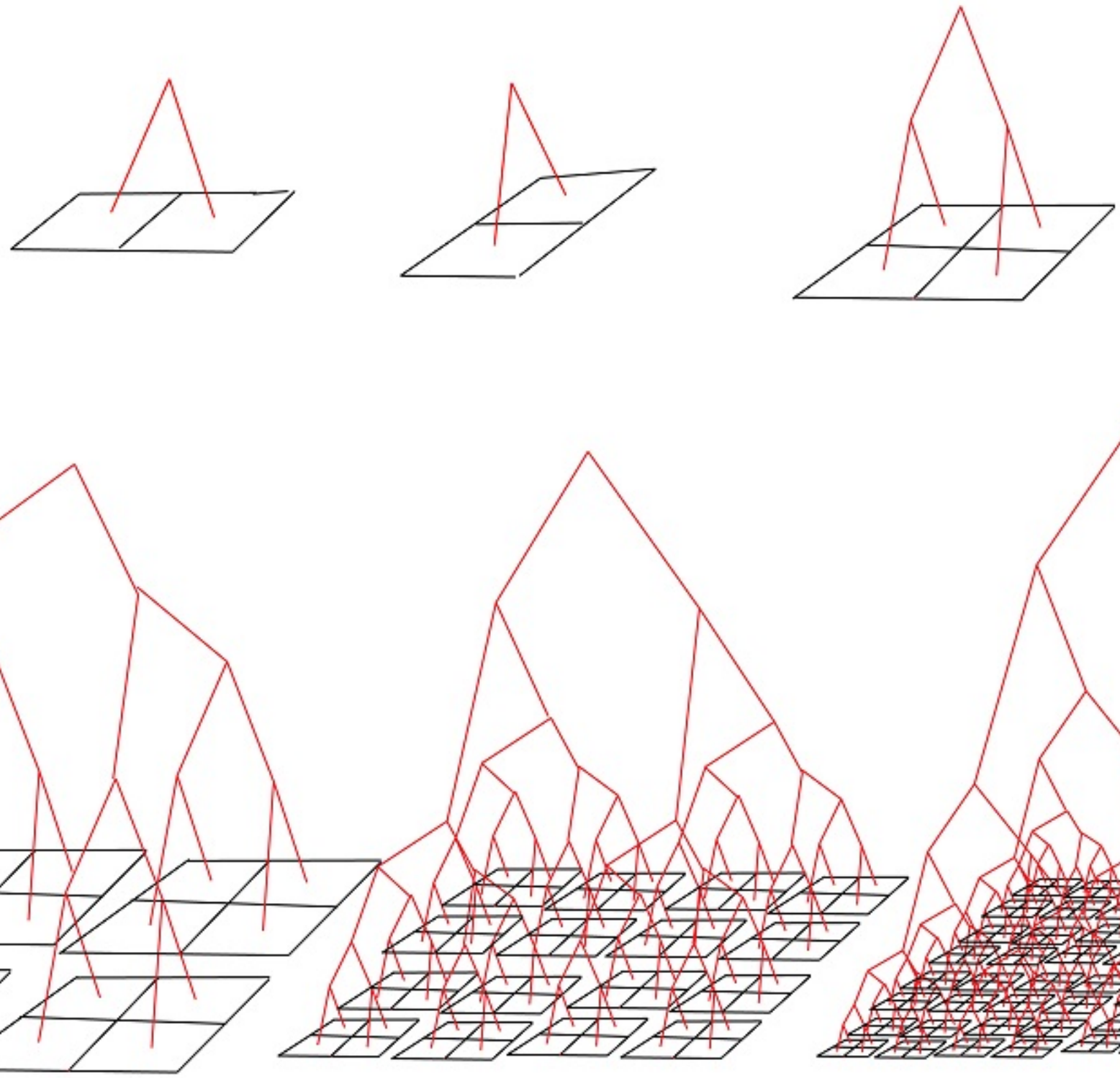
recursive binary partitioning

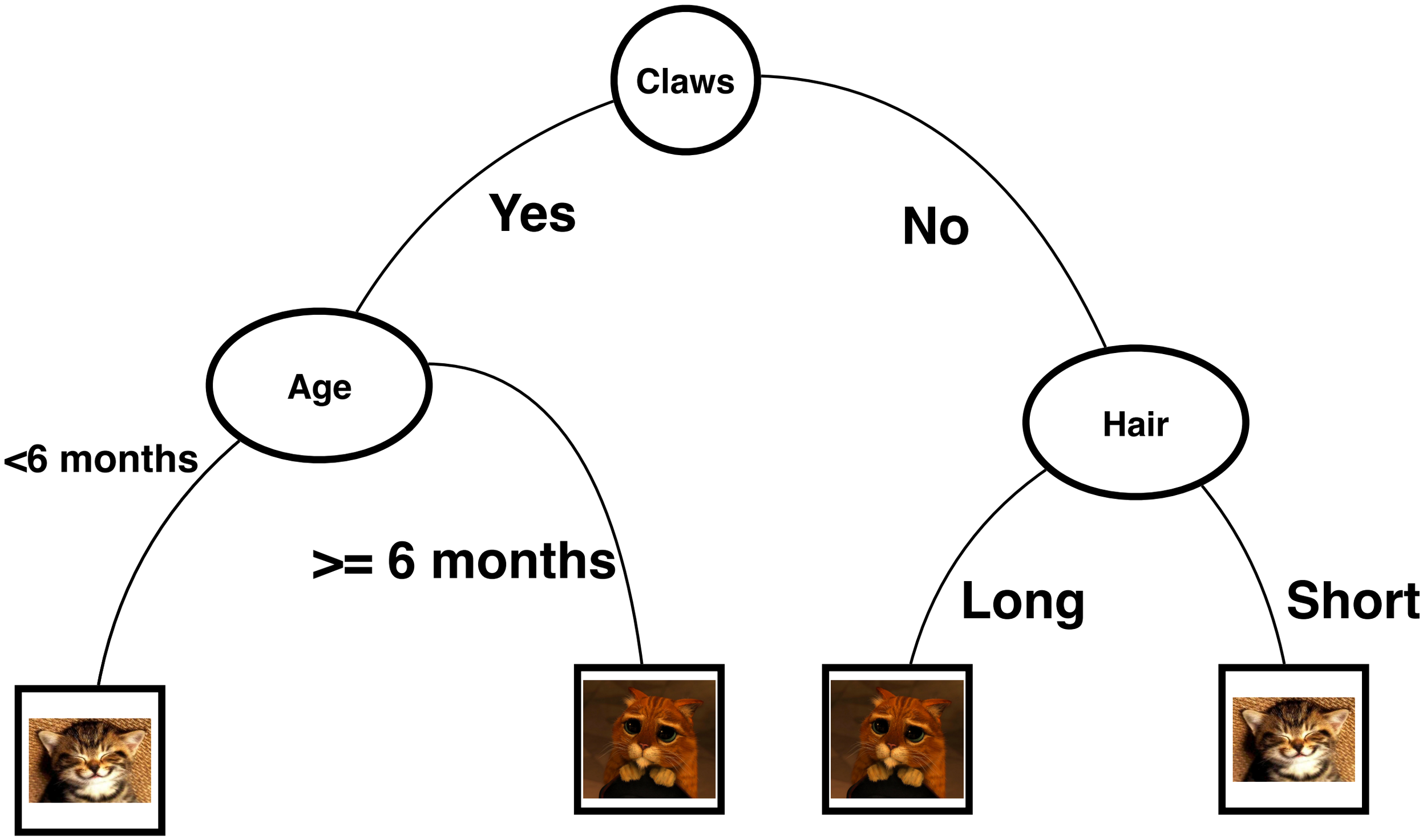
Recursive Binary Partitioning



Parellel Axis Splits

Making 2 dimensional surfaces with Binary Trees




Claws
Yes
No
Age
<6 months
>= 6 months
Hair
Long
Short

Regression Trees

When the target is continuous

The goal is minimizing variance at each node

Regression Trees

The variance reduction of a node N is defined as the total reduction of the variance of the target variable x due to the split at this node.

$$I_V(N) = \frac{1}{|S|}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(x_i-x_j)^2-\left(\frac{1}{|S_t|}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(x_i-x_j)^2+\frac{1}{|S_f|}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(x_i-x_j)^2\right)$$

S are presplit sample indices
S_t are indices where the split is true
S_f are indices where the split is false

Okay,
Decisions Trees are

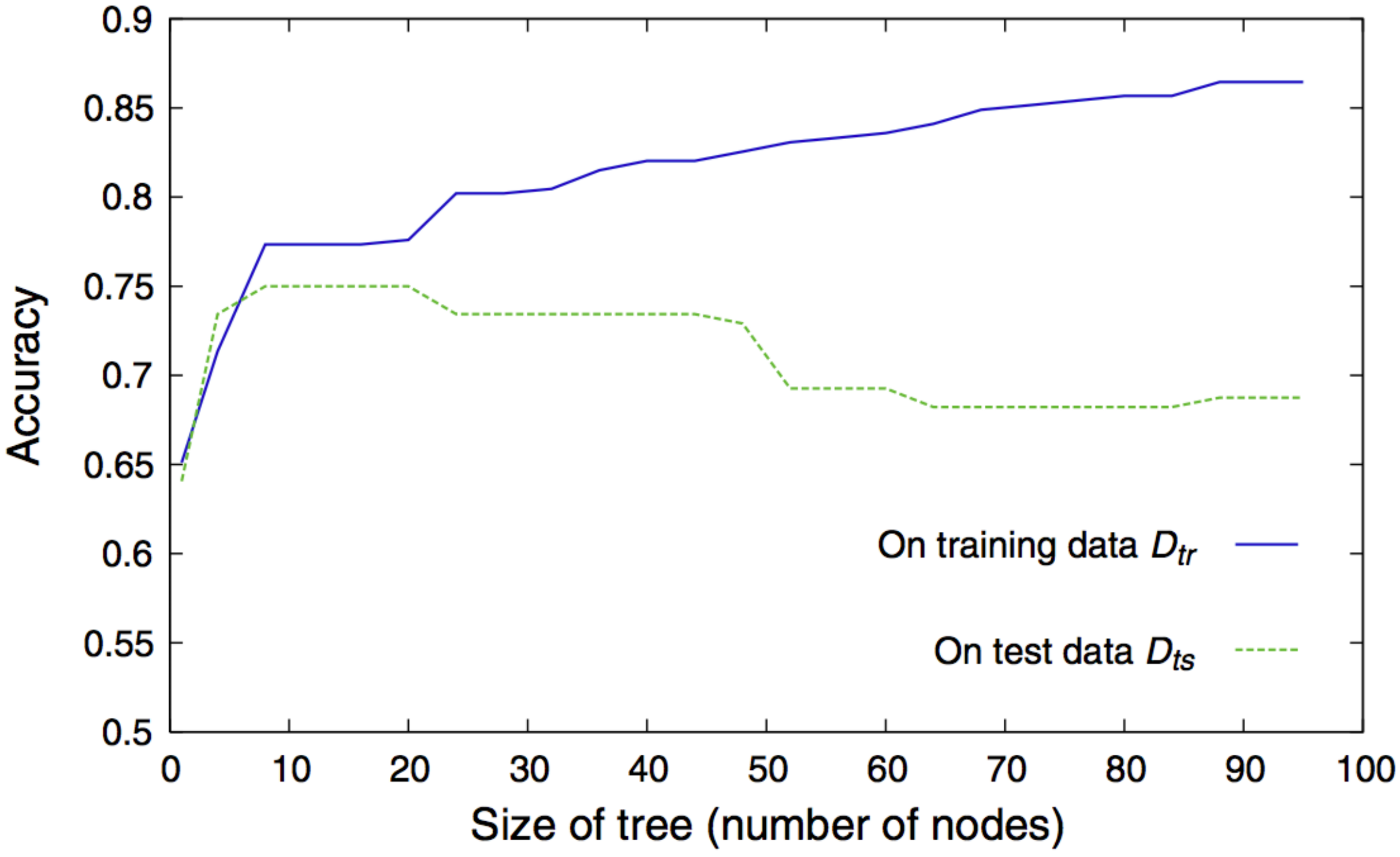
What is the downside?

Overfitting:

Good predictions on the training, poor performance on transfer

Overfitting

Generalization is the **most important** criteria.


Accuracy
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
10
20
30
40
50
60
70
80
90
100
Size of tree (number of nodes)
On training data D_{tr}
On test data D_{ts}

Sources of Overfitting

- Noise
- Small number of examples associated with each leaf
- Accidental patterns

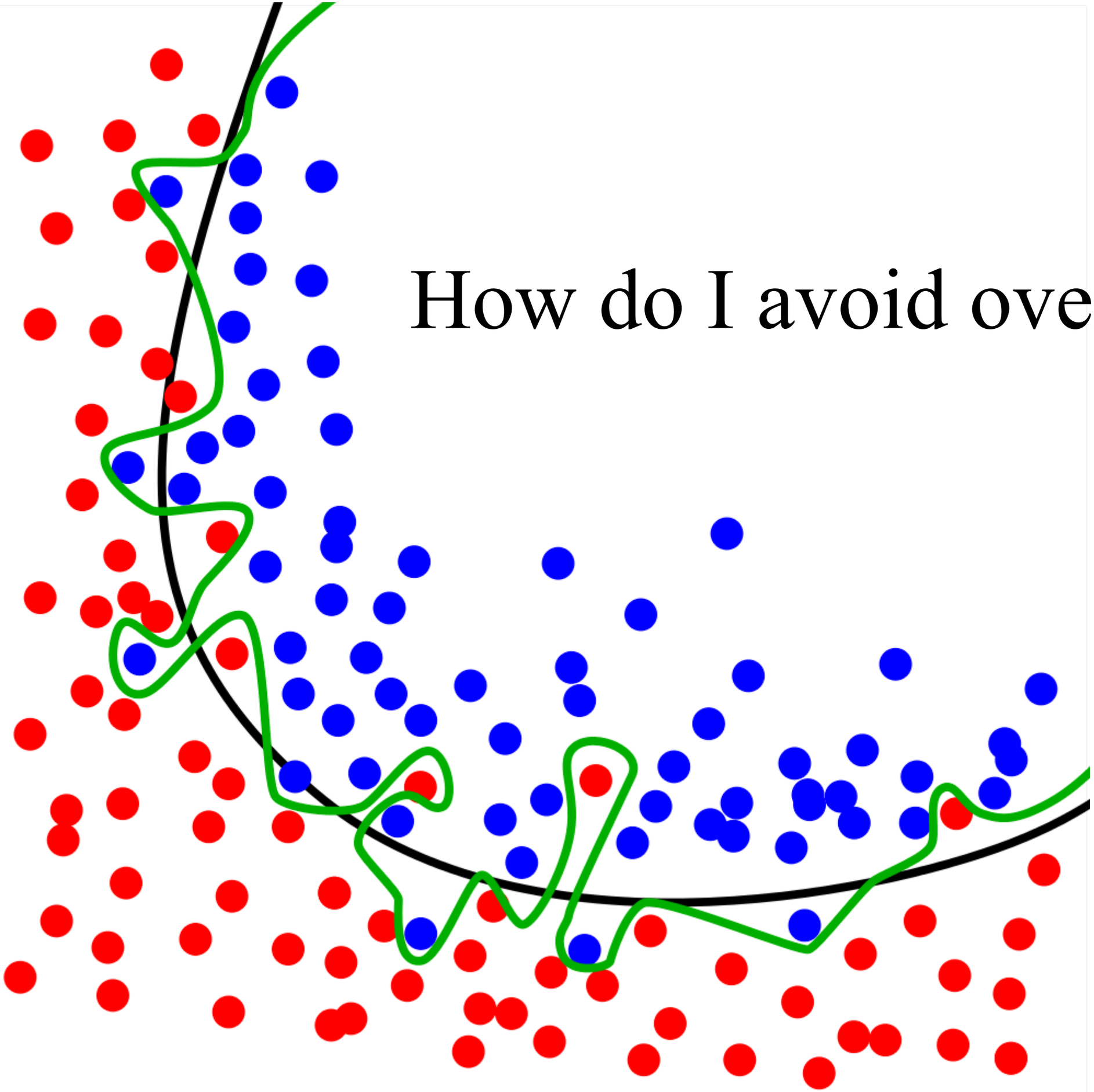
Avoiding Overfitting

1. Stop growing the tree when data split is not statistically significant

2. Grow tree fully, then post-prune

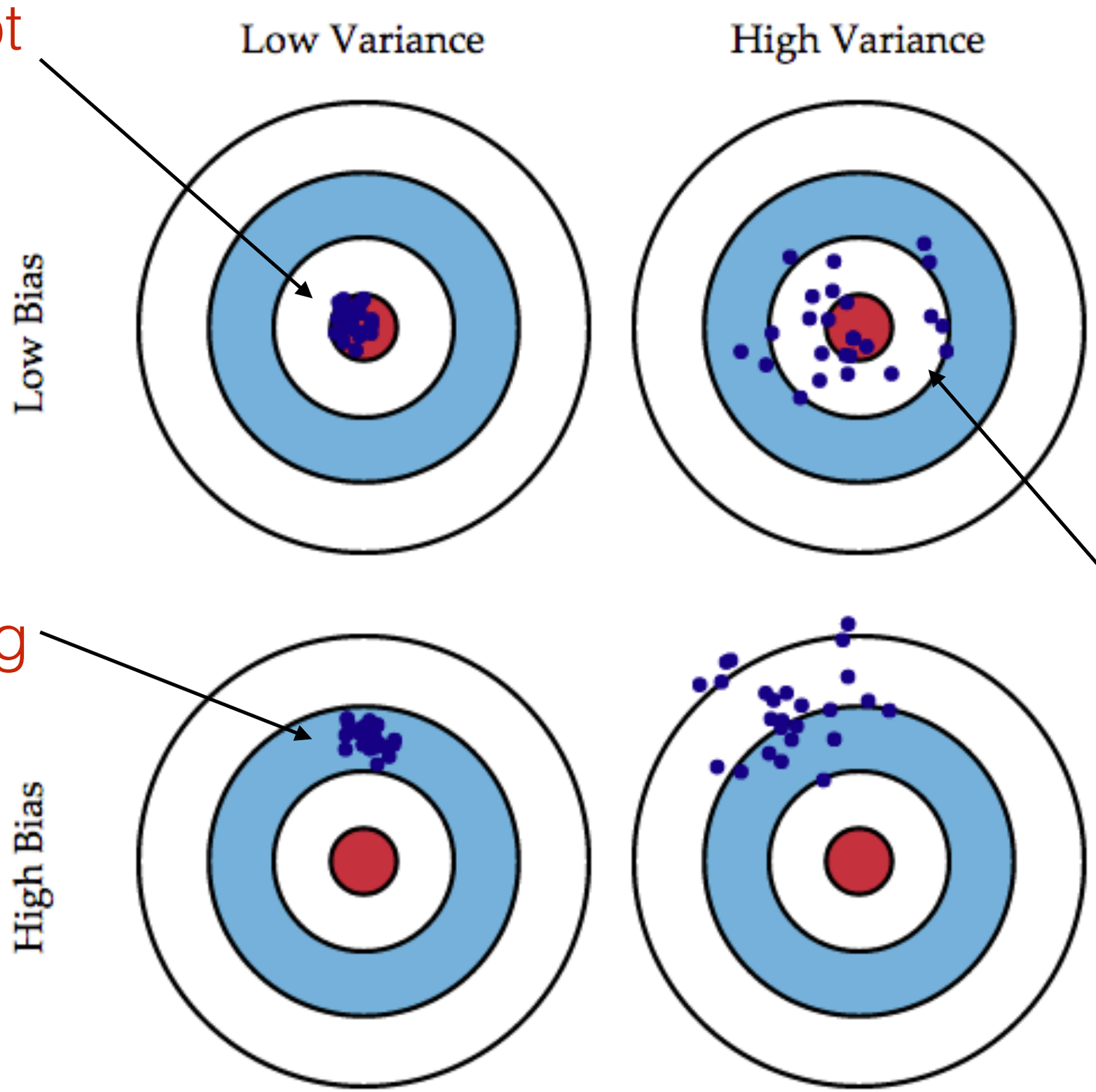
Stopping criteria

1. Number of training examples per split drops below a threshold
2. Nonsignificant purity reduction



How do I avoid overfitting?

Sweet Spot

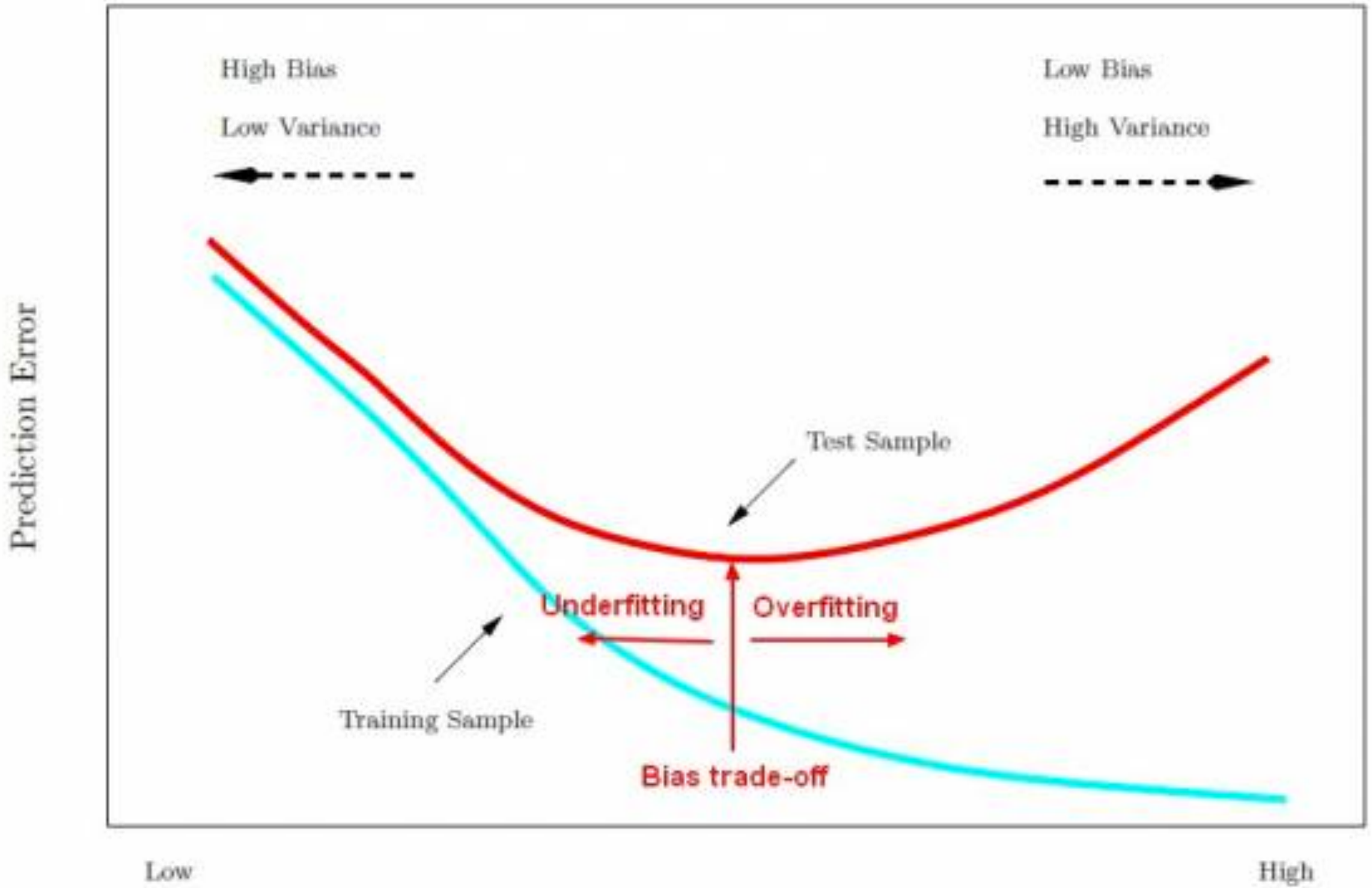


Underfitting

Overfitting

Bias: Systematic error from the truth

Variance: Error due to variability of the model


High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Underfitting
Overfitting
Training Sample
Bias trade-off
Low
High
Model Complexity

Decision trees are high variance models:

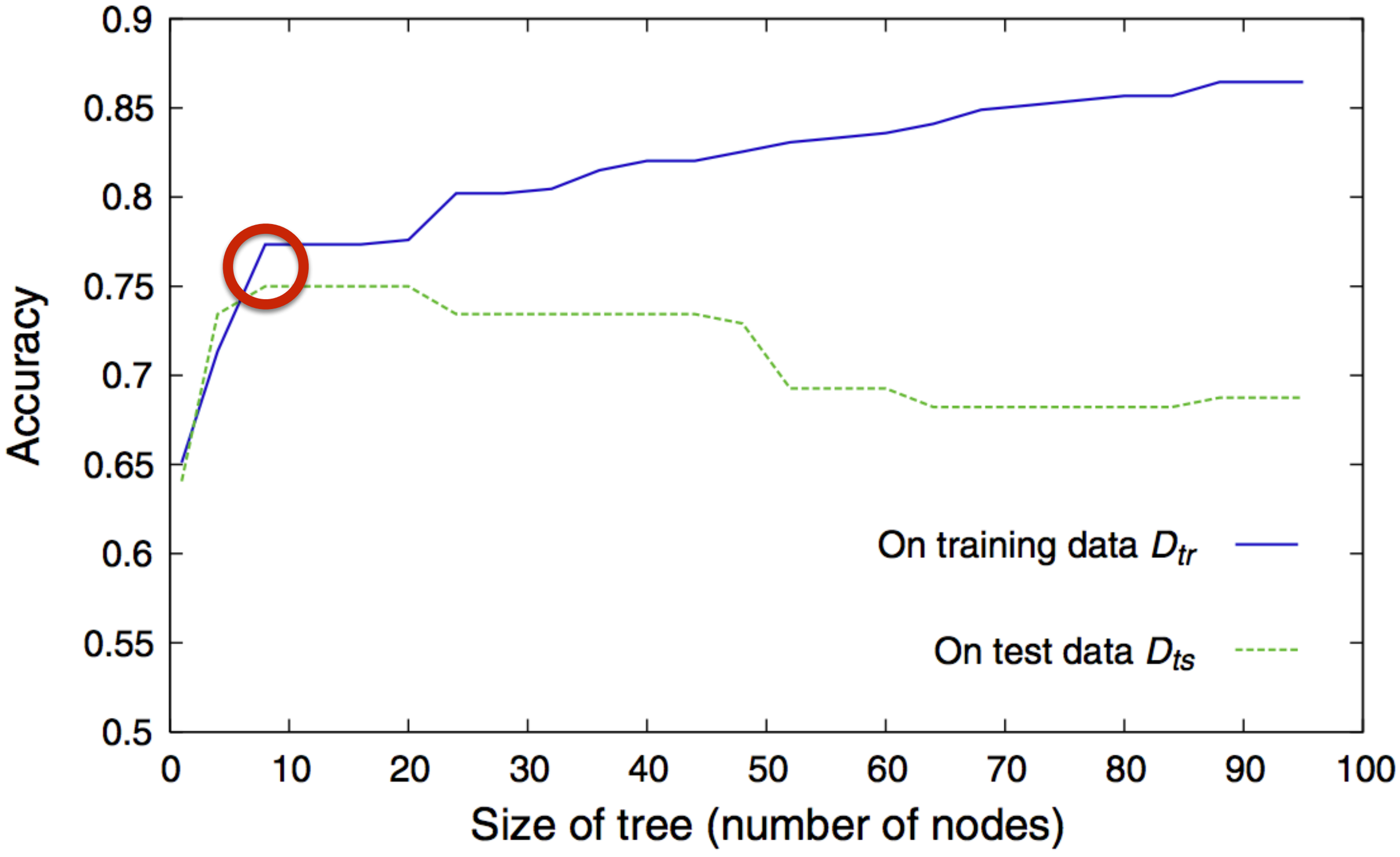
A small change in the learning sample can result in a very different tree.

Avoiding Overfitting

A smaller tree with fewer splits might lead to lower variance and better interpretation at the cost of a little bias.

Pruning

- Reduce the size of decision trees
- Remove sections add provide little power to classify instances
- Reduces the complexity
- Improves predictive accuracy by the reduction of overfitting


Accuracy
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
10
20
30
40
50
60
70
80
90
100
Size of tree (number of nodes)
On training data D_{tr}
On test data D_{ts}

RANDOM FOREST

Random Forests: **Overview**

- Produce multiple decision trees (ensembles) which are then combined to yield a single consensus
- One of the most successful machine learning models
- Combine many decision trees in order to reduce the risk of overfitting. Decreases variance.

Random Forests: **Algorithm**

- A set of decision trees separately
  (the training can be done in parallel)

- Injects randomness into the training process so that each decision tree is a bit different.

- Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data.

Bagging
(**B**ootstrap **AGG**regat**ING**)

- The average model has the same bias as the original method but zero variance

- Simulate sampling from nature by bootstrap sampling

- Bootstrap sampling = sampling with replacement of N objects (N is the size of the data)

Random Forests

- Combine bagging and random attribute subset selection:
  - Build the tree from a bootstrap sample
  - Select the best split among a random subset of k attributes (Instead of choosing the best split among all attributes)

Prediction Aggregation

Classification:

Majority vote.

Each tree's prediction is counted as a vote for one class. The label is predicted to be the class which receives the most votes.

Regression:

Averaging.

Each tree predicts a real value. The label is predicted to be the average of the tree predictions.

Decision Trees: Pros

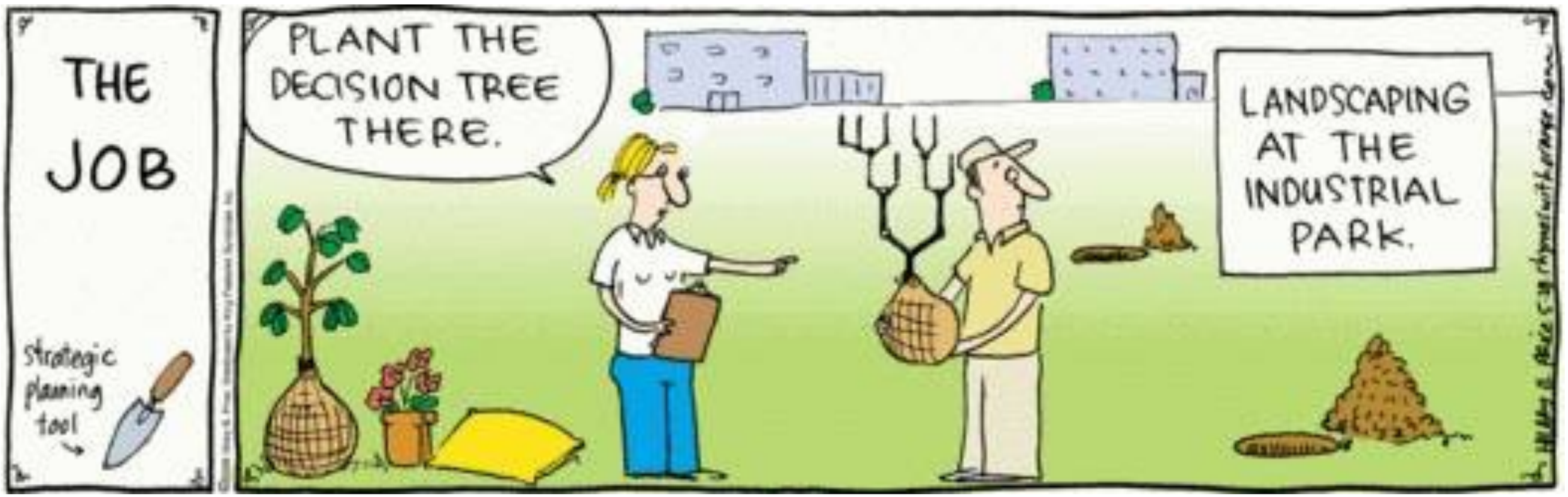
- Easy to interpret
- Handle numerical and categorical features
- Do not require preprocessing of data
  (e.g., feature scaling)
- Able to capture conditional probabilities and feature interactions

Decision Trees: Cons

- Overfitting
- Fragile
- Problems out-of-sample prediction

Summary

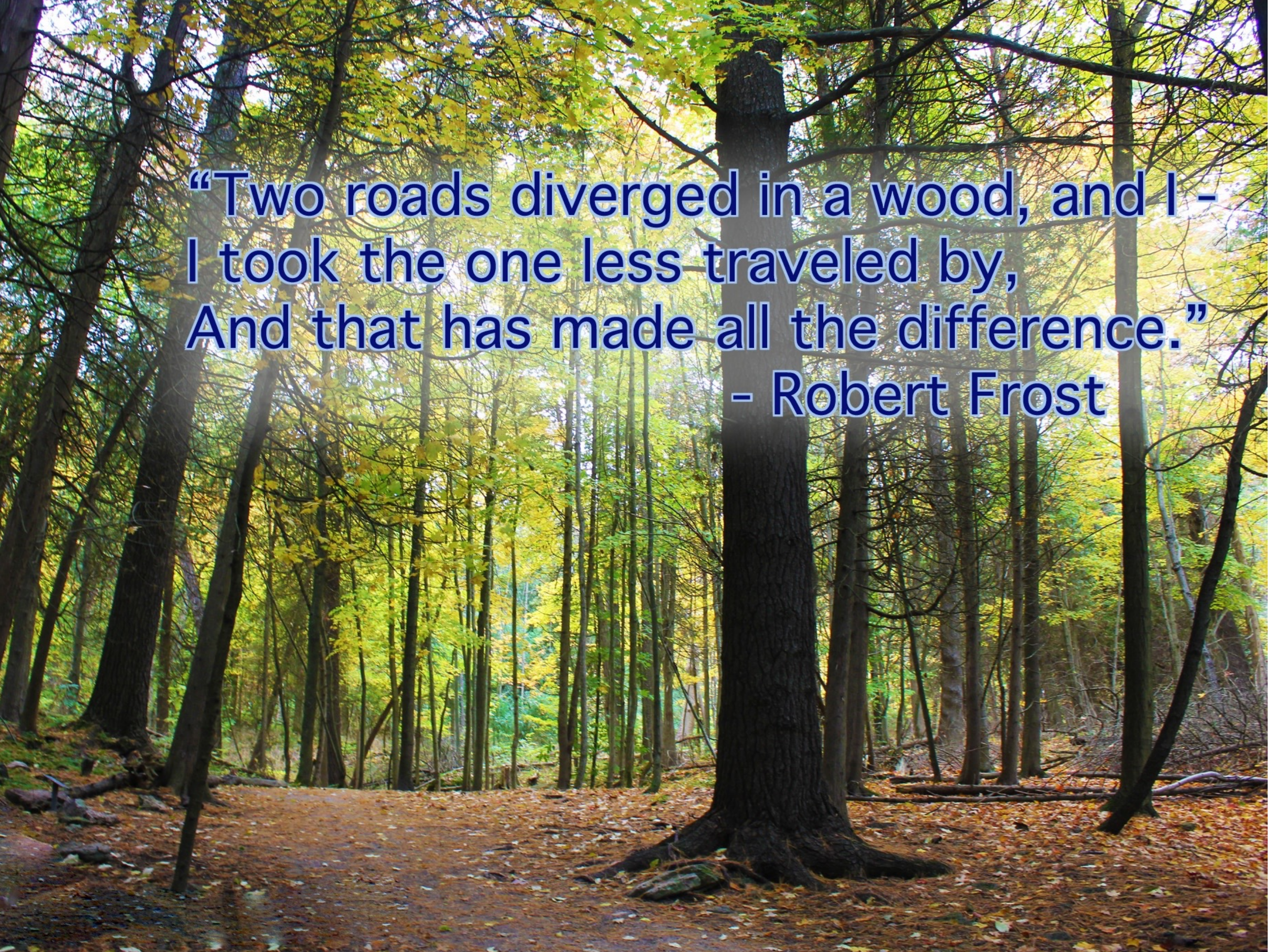
- A top-down, greedy algorithm that performs recursive binary partitioning
- Looking for node purity:
    - Classification: Gini Impurity or Entropy
    - Regression: Variance
- Overfitting as a limitation
- Random forest extension to reduce overfitting

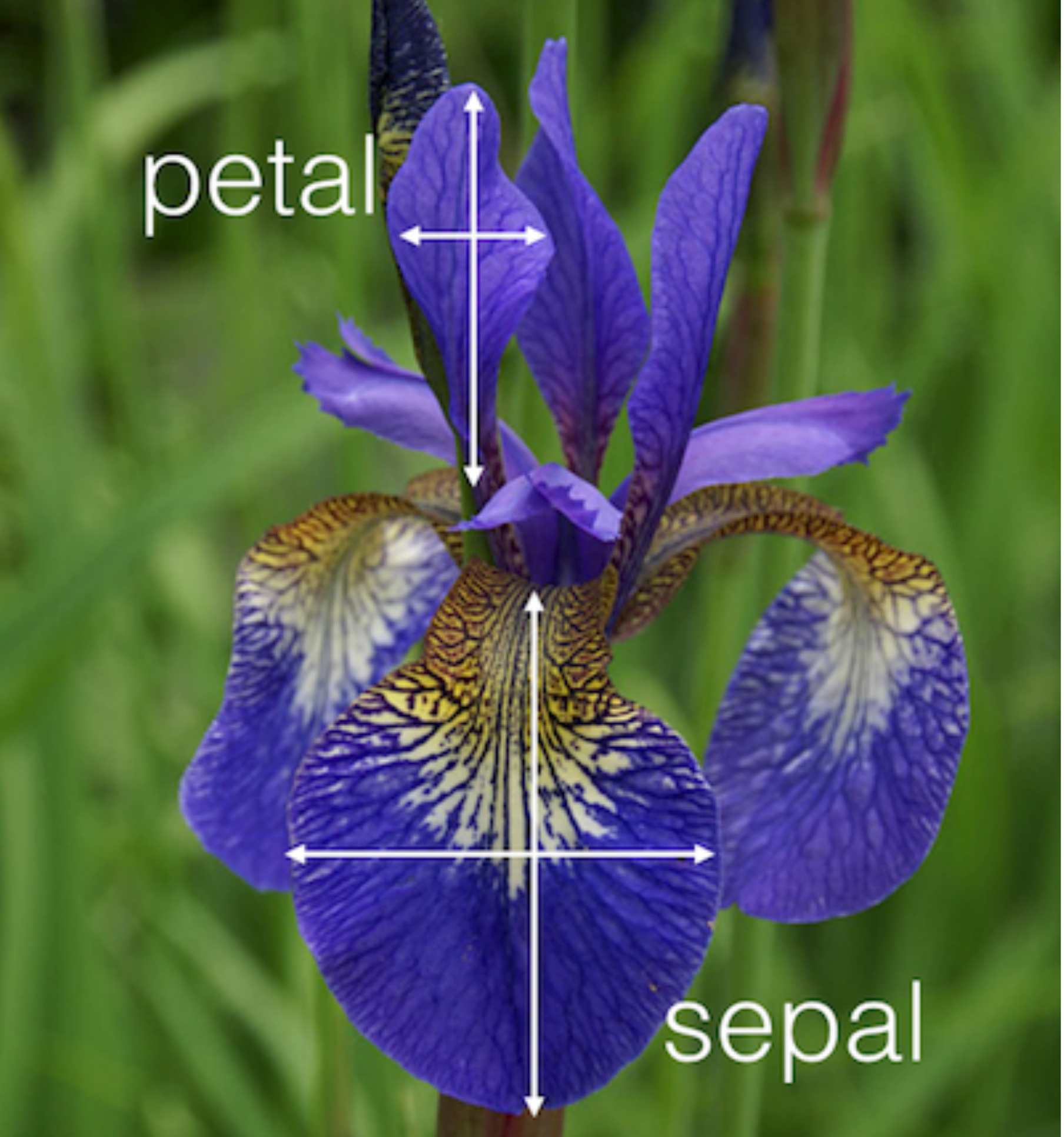

THE JOB
strategic planning tool
PLANT THE DECISION TREE THERE.
LANDSCAPING AT THE INDUSTRIAL PARK.

Questions?

Lab

- Part 1: Work through example together
- Part 2: Small group activity


“Two roads diverged in a wood, and I –
I took the one less traveled by,
And that has made all the difference.”
- Robert Frost



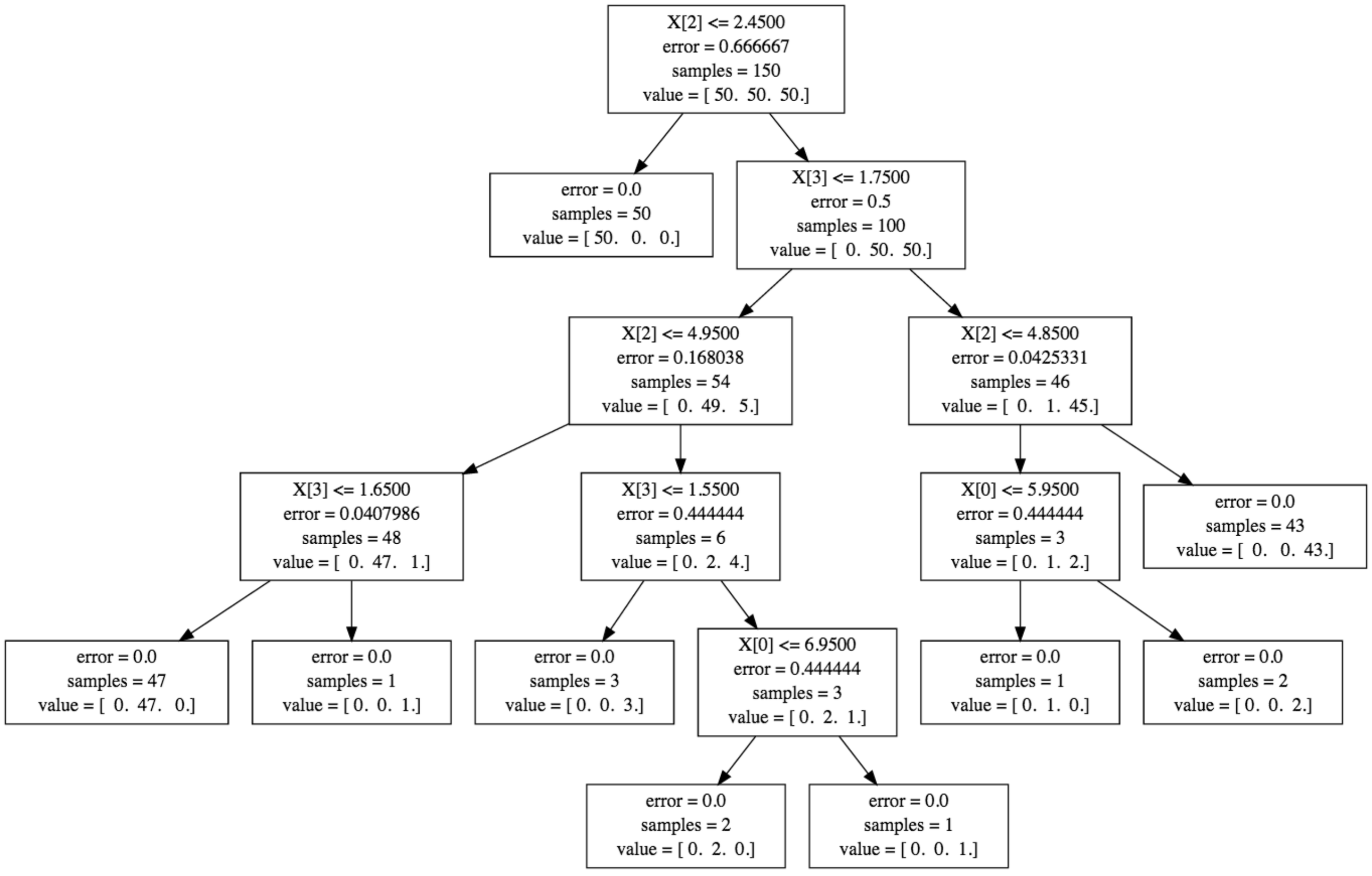
Iris Dataset

Iris setosa

Iris versicolor

Iris virginica

bit.ly/dt-intro


X[2] <= 2.4500
error = 0.666667
samples = 150
value = [50. 50. 50.]
error = 0.0
samples = 50
value = [50. 0. 0.]
X[3] <= 1.7500
error = 0.5
samples = 100
value = [0. 50. 50.]
X[2] <= 4.9500
error = 0.168038
samples = 54
value = [0. 49. 5.]
X[2] <= 4.8500
error = 0.0425331
samples = 46
value = [0. 1. 45.]
X[3] <= 1.6500
error = 0.0407986
samples = 48
value = [0. 47. 1.]
X[3] <= 1.5500
error = 0.444444
samples = 6
value = [0. 2. 4.]
X[0] <= 5.9500
error = 0.444444
samples = 3
value = [0. 1. 2.]
error = 0.0
samples = 43
value = [0. 0. 43.]
error = 0.0
samples = 47
value = [0. 47. 0.]
error = 0.0
samples = 1
value = [0. 0. 1.]
error = 0.0
samples = 3
value = [0. 0. 3.]
X[0] <= 6.9500
error = 0.444444
samples = 3
value = [0. 2. 1.]
error = 0.0
samples = 1
value = [0. 1. 0.]
error = 0.0
samples = 2
value = [0. 0. 2.]
error = 0.0
samples = 2
value = [0. 2. 0.]
error = 0.0
samples = 1
value = [0. 0. 1.]

http://bit.ly/predict-lab

Questions to answer

- What is the best way to summarize this data?
- What is the number predictor of breast cancer?
- How good is the best fitting model?
    - Which metrics would you choose?

Errata

CART: Classification and Regression Trees

Both an algorithm and generic acronym for decision trees

The split chosen at each tree node is chosen from the set:

$$\underset{s}{\operatorname{argmax}}\, IG(D, s)$$

where **IG**(D,s) is the **Information Gain** when a split s is applied to a dataset D

Node Impurity

Measure Name: Entropy

Task: Classification

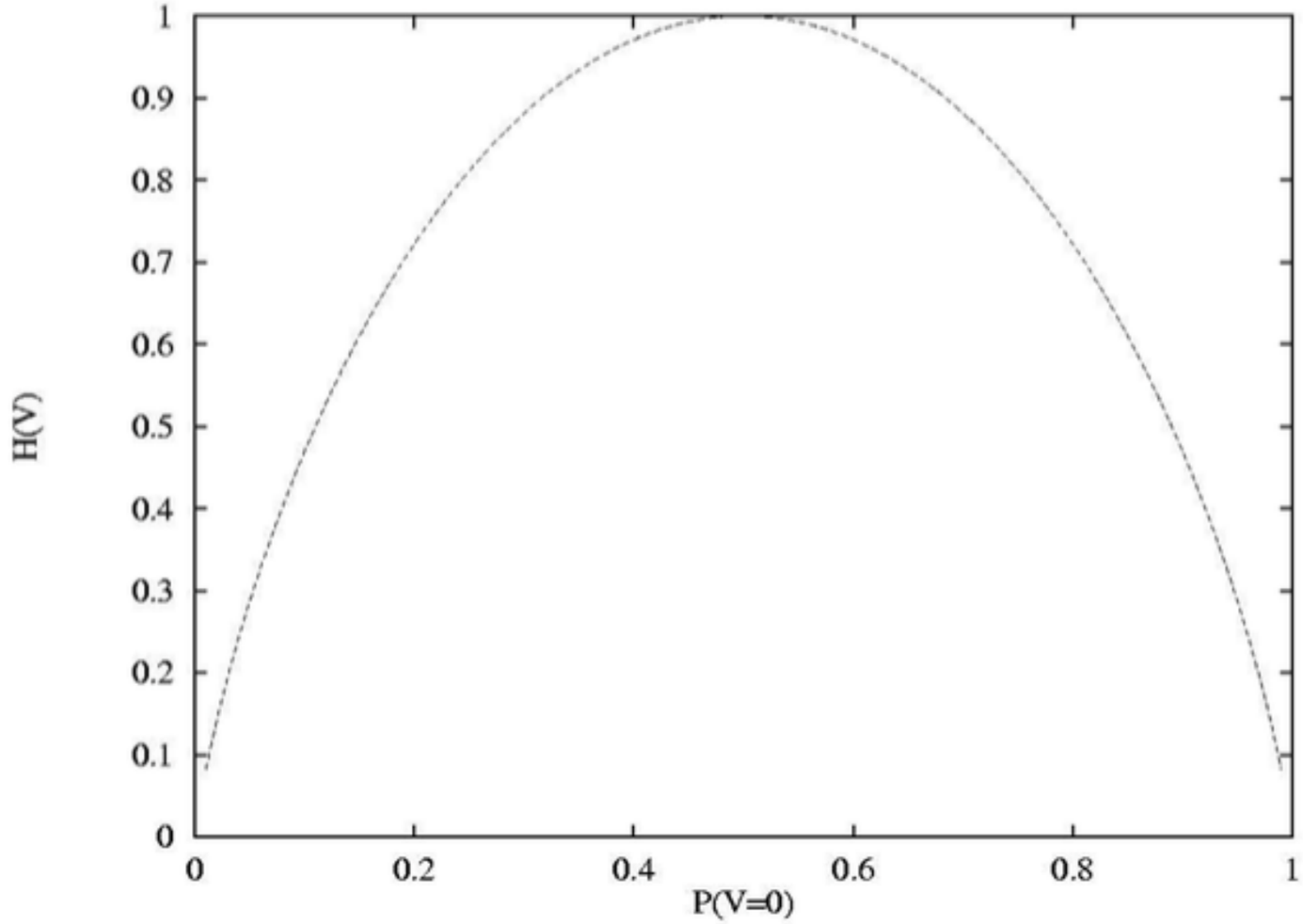
Algorithm: ID3, C4.5 and C5.0

Entropy

The *entropy* of V, denoted $H(V)$ is defined as follows:

$$H(V) = \sum_{v=0}^{1} -P(H = v) \lg P(H = v).$$

This is the average surprise of describing the result of one "trial" of V (one coin toss).



Node Impurity

Measure Name: Gini impurity

Task: Classification

Algorithm: CART (classification and regression tree)

Boosting

- Boosting type algorithms
- Grows several models sequentially
- Ex: Adaboost, MART
- Decrease mainly bias

Rule Post-Pruning

1. Convert the tree to equivalent set of rules
2. Prune each rule independently of others
3. Sort final rules into desired sequence for use