Decision trees and forests

Machine Learning

Agenda

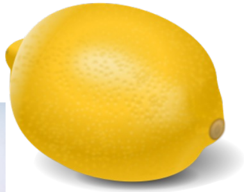
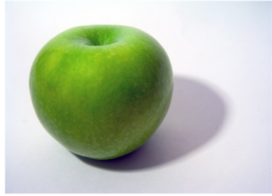
- What is a decision tree?
- Learning DTs
- DT properties
- More about selecting questions
- Controlling overfitting
- The more the merrier: Ensembles of trees

Learning rules

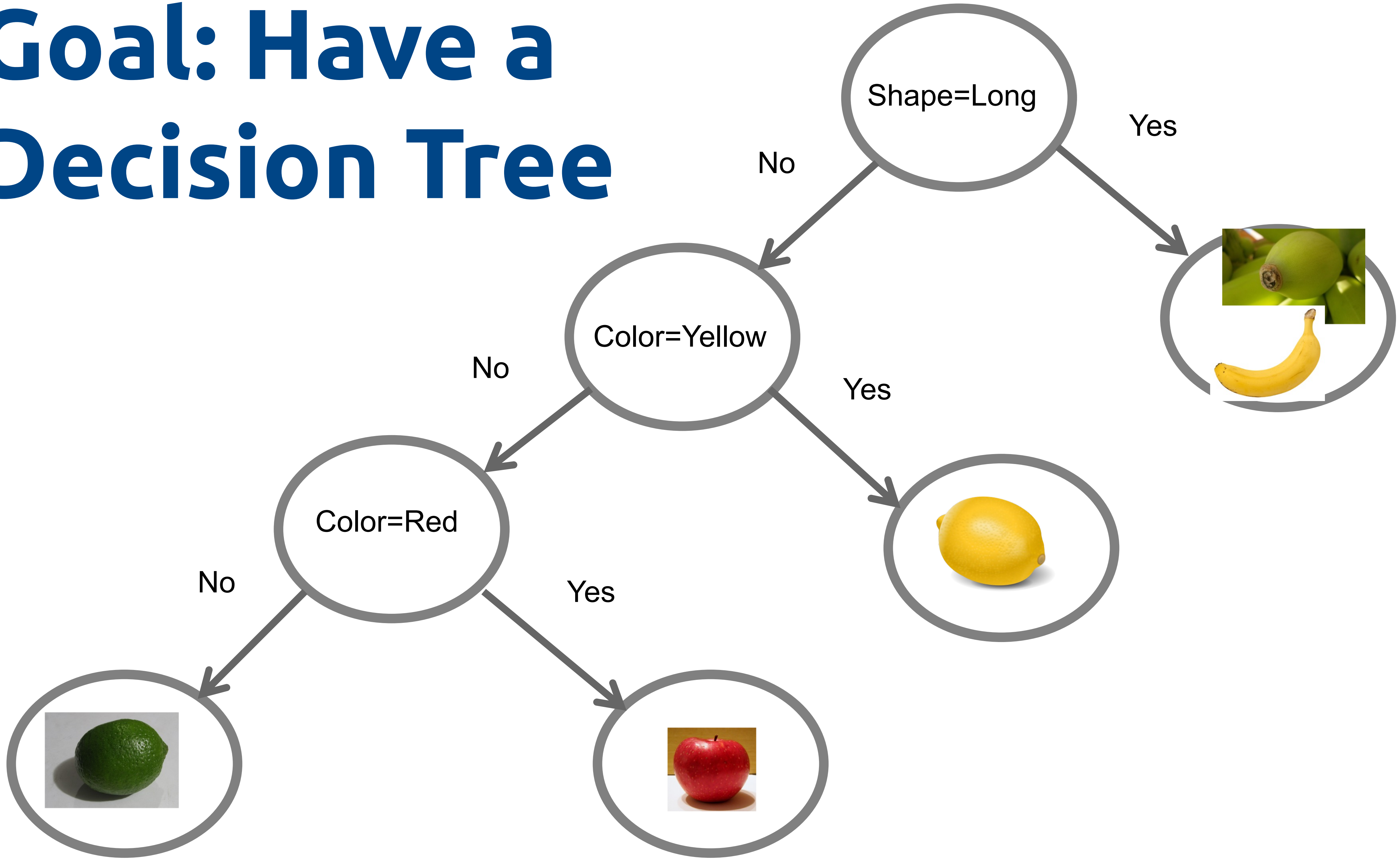
```
If condition A:
    If condition B:
        Action 1
    Else:
        Action 2
Else:
    Action 3
```

Fruit classification

Shape	**Color**	***Target***
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana


Goal: Have a Decision Tree



Agenda

- What is a decision tree?
- Learning DTs
- DT properties
- More about selecting questions
- Controlling overfitting
- The more the merrier: Ensembles of trees

How can we pick a good decision tree?

Build all possible trees

Evaluate how ‘good’ each one is

Pick the best one!

How can we pick a good decision tree?

- Number of possible trees grows exponentially with number of features
- Can't check them all and see which one works best
- Need to build a tree incrementally

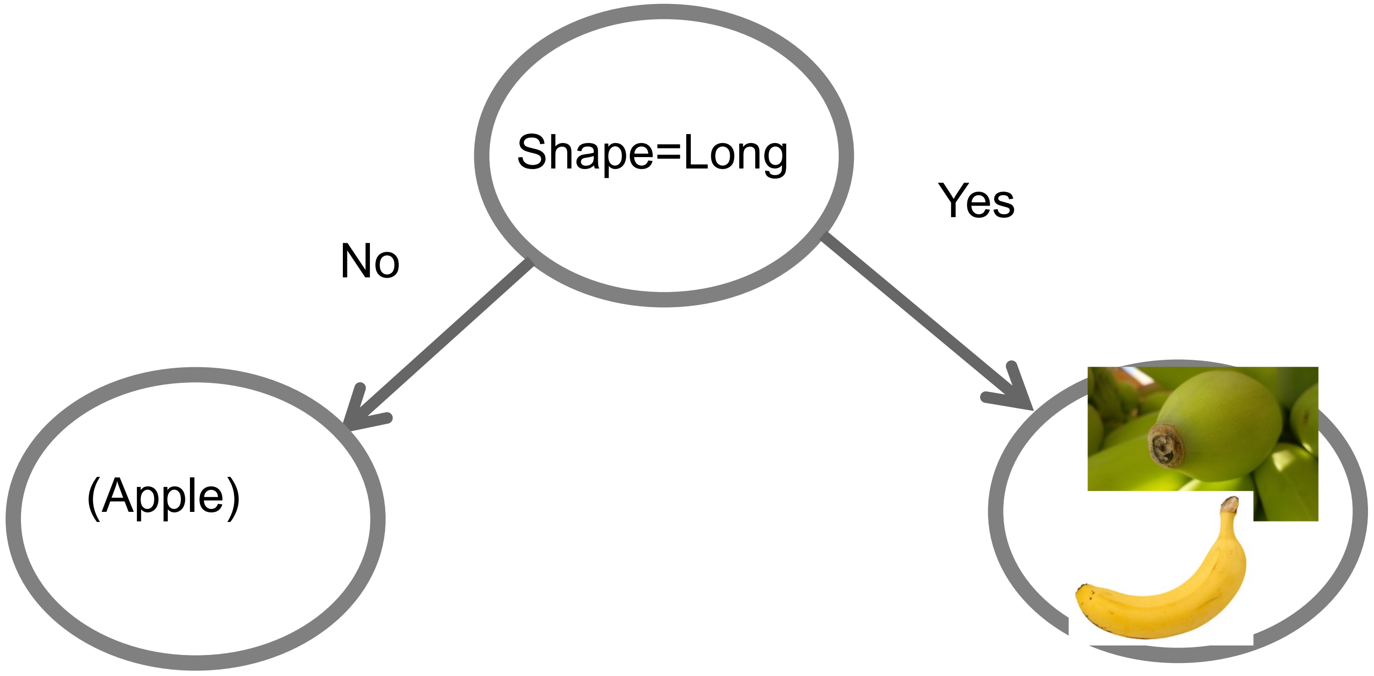
Which question to ask first?

- It's best to ask important questions first
- Which questions are important?
  - The ones which help us classify
- if we had to classify data based only on one question, which question would do best?

Step by step

Q	Correct
Long?	4
Green?	2
Red?	3
Yellow?	3

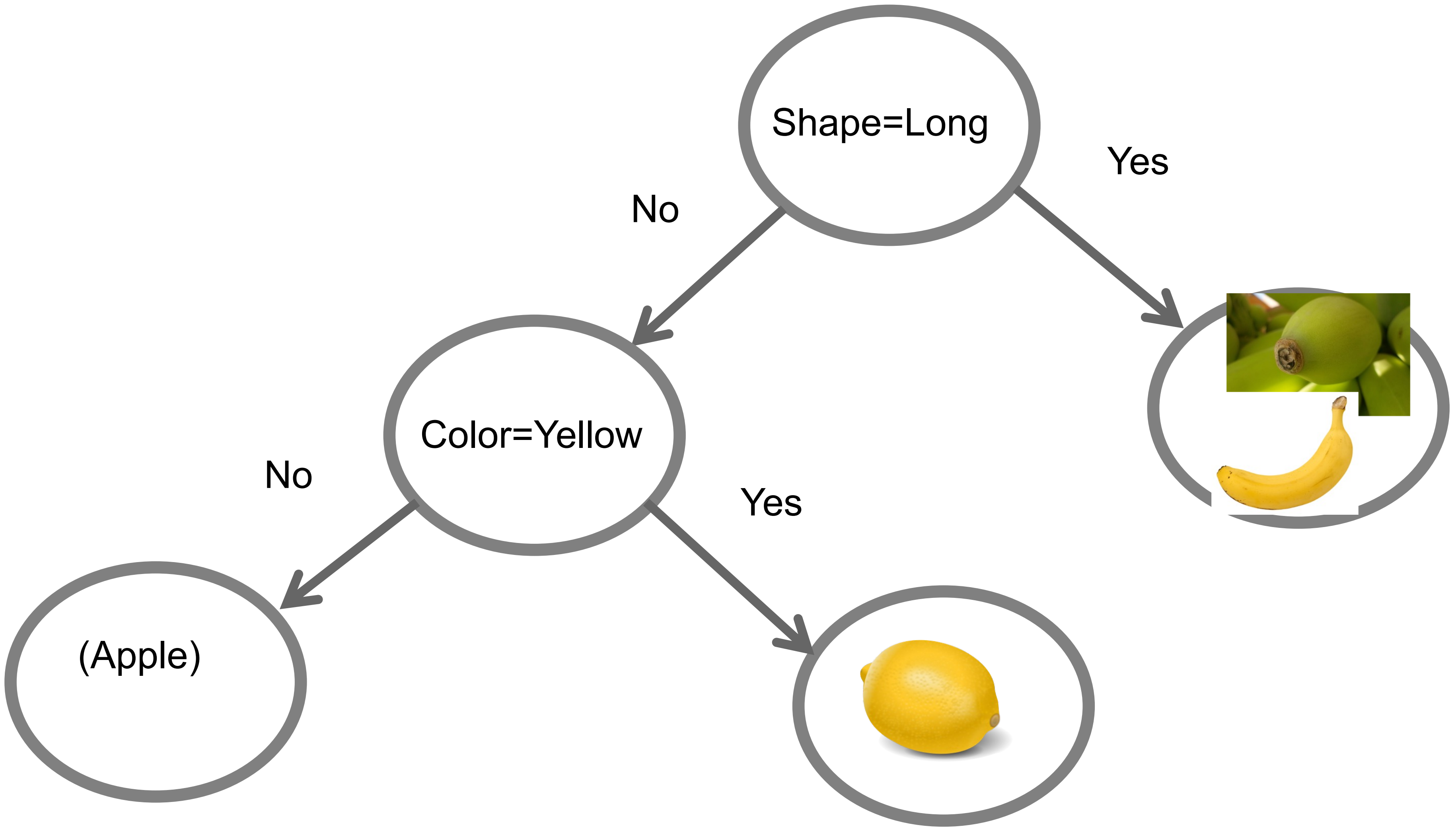
Shape	**Color**	***Target***
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana


Shape=Long
No
Yes
(Apple)

Long=No

Q	Correct
Green?	2
Red?	2
Yellow?	3

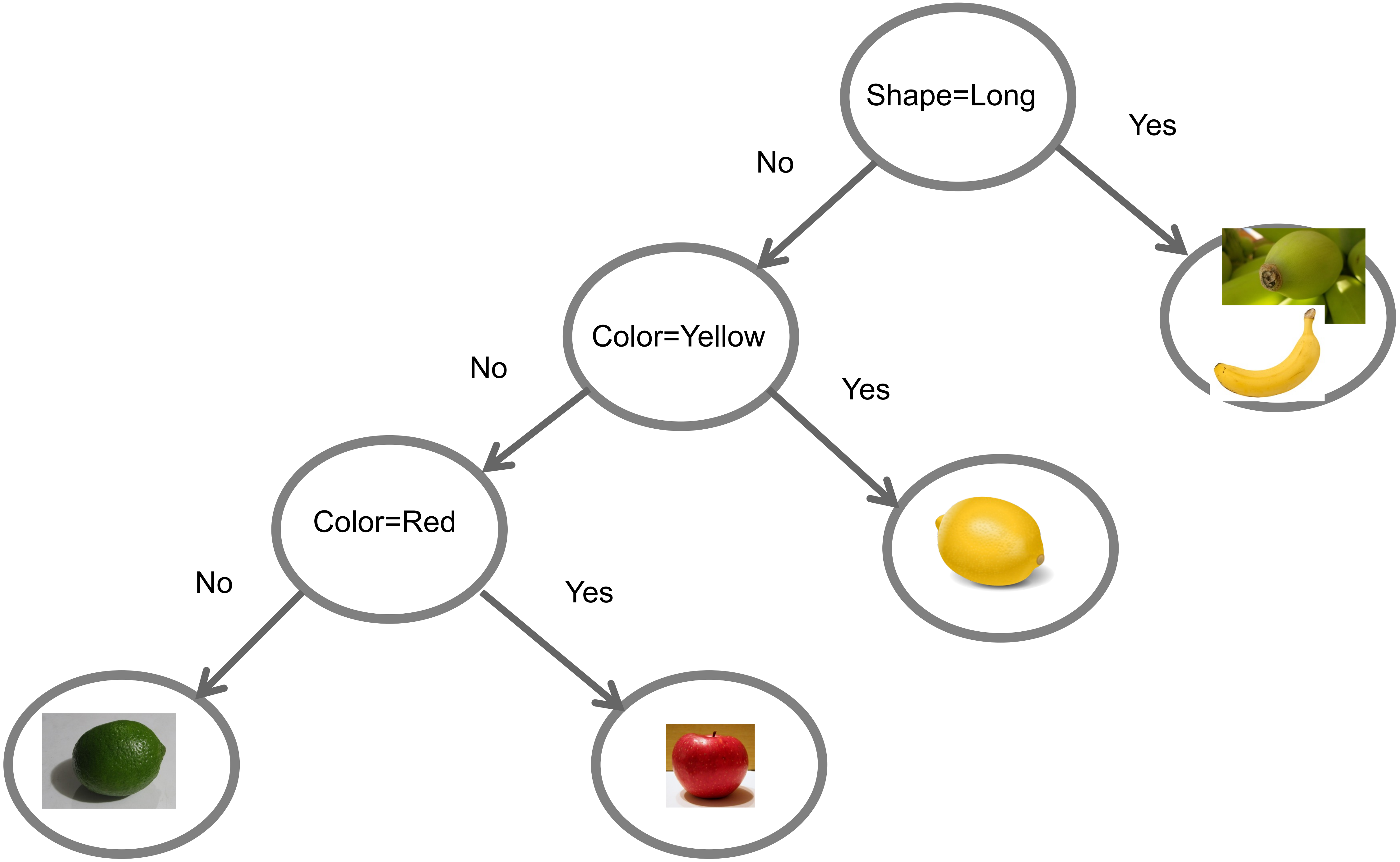
Shape	Color	*Target*
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple


Shape=Long
No
Yes
Color=Yellow
No
Yes
(Apple)

Yellow=No

Q	Correct
Red?	2
Green?	2

Shape	Color	*Target*
Round	Green	Lime
Round	Green	Apple
Round	Red	Apple


Shape=Long
Yes
No
Color=Yellow
No
Yes
Color=Red
No
Yes

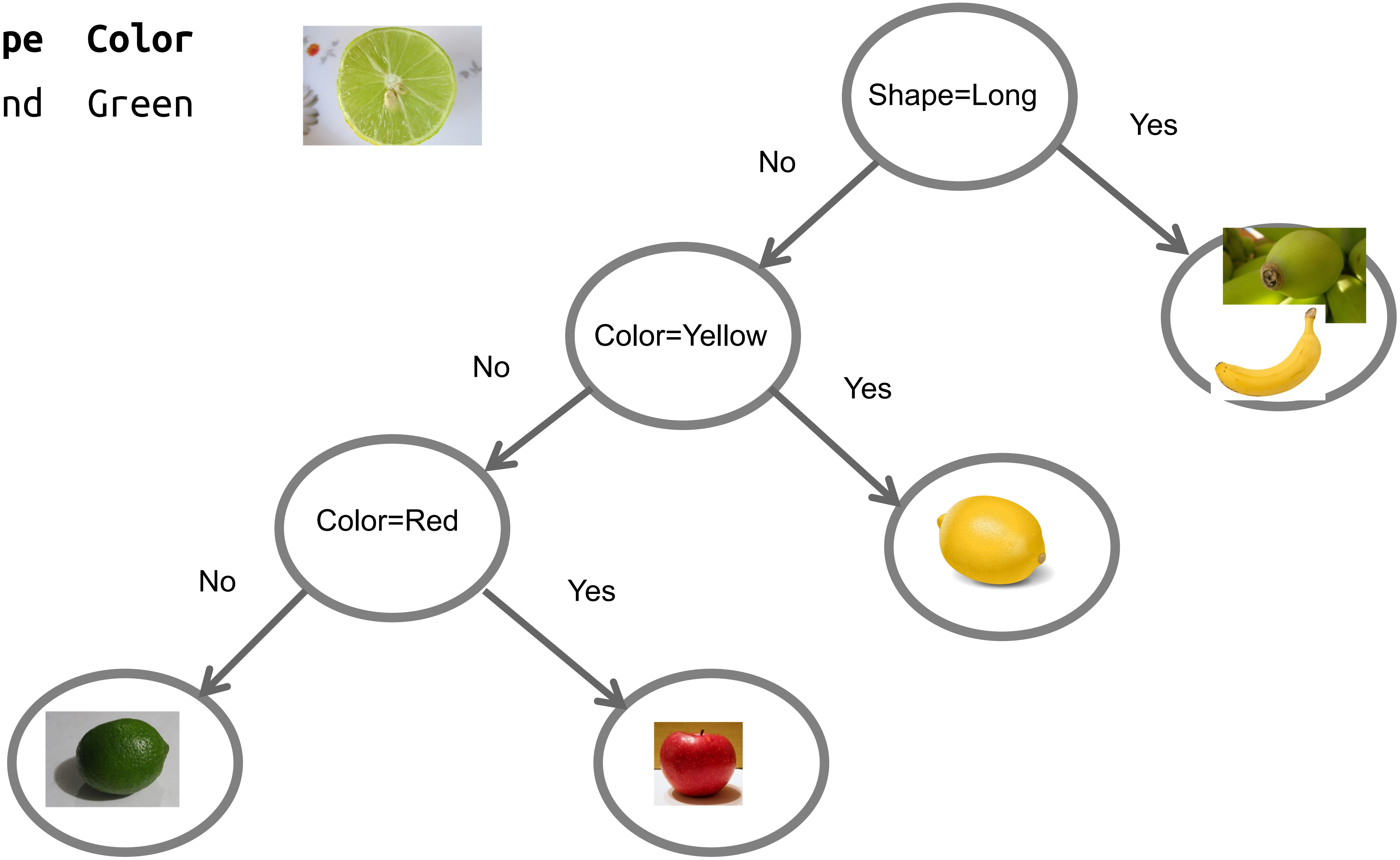
Building a decision tree

- If all examples have same label
    - Create leaf node with label
- Otherwise
    - Choose most important question
    - Split data into two parts (**NO** and **YES**) according to question
    - Remove question from question set
    - Iterate (Recursive algorithm):
        - Left branch ← Apply algorithm to **NO** examples
        - Right branch ← Apply algorithm to **YES** examples
    - Create node with (question, left branch, right branch)

Applying a decision tree

Shape	Color
Round	Green

Shape	Color
Round	Green



Using a decision tree

Given a tree and an example

- If tree is leaf node:
  - Prediction ← label
- Otherwise ask the question about example
  - If **NO**
    - Prediction ← apply algo with left branch
  - If **YES**
    - Prediction ← apply algo with right branch

Agenda

- What is a decision tree?
- Learning DTs
- DT properties
- More about selecting questions
- Controlling overfitting
- The more the merrier: Ensembles of trees

Digression: Recursion

- We build and use DT with recursive functions
- Recursive function
  - Base case
  - Recursive call – applies itself
- Example: 6!

```
def factorial(n):
        if n == 0:
                return 1
        else:
                return n * factorial(n-1)
```

Trees and recursion

- Trees are recursively defined data structures
    - Base case = leaf node
    - Recursive case = branch node
- Good match for recursive functions
- More during tutorial

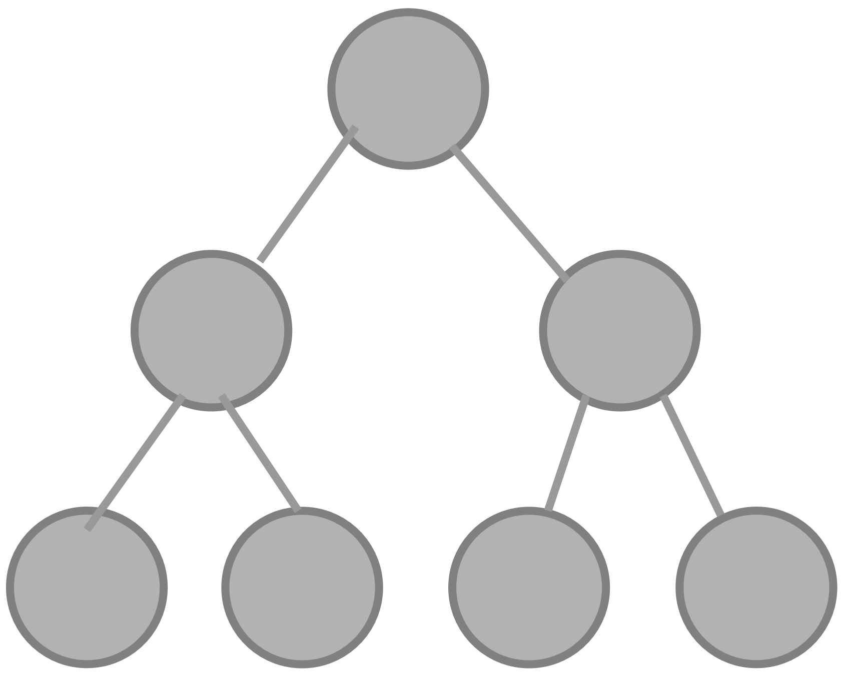
Decision Tree efficiency

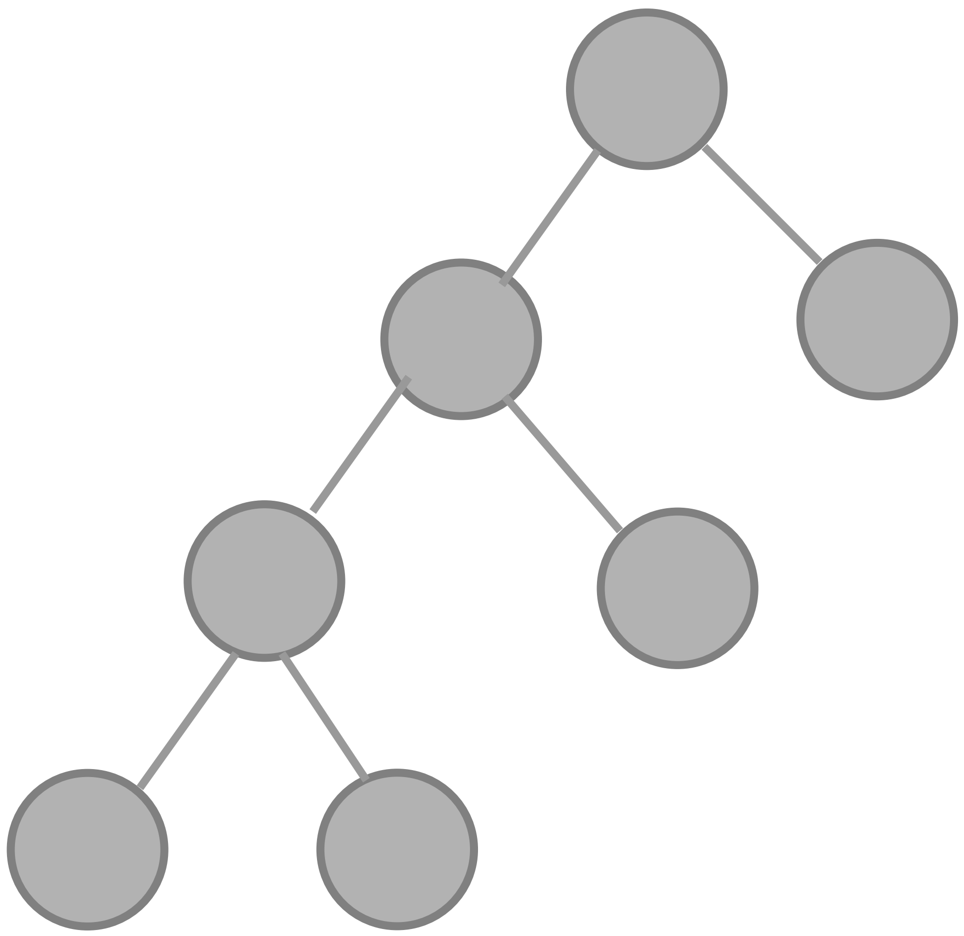
- You build a tree A with **100** nodes, and a tree B with **1000** nodes
- Predicting the targets of a set of 1 million examples takes 1 second with tree A
- How long would it take with tree B?

Decision Tree speed

- Depends on number of questions needed to get to a leaf node
- Which depends on the **depth** of the tree

Which tree has more depth?

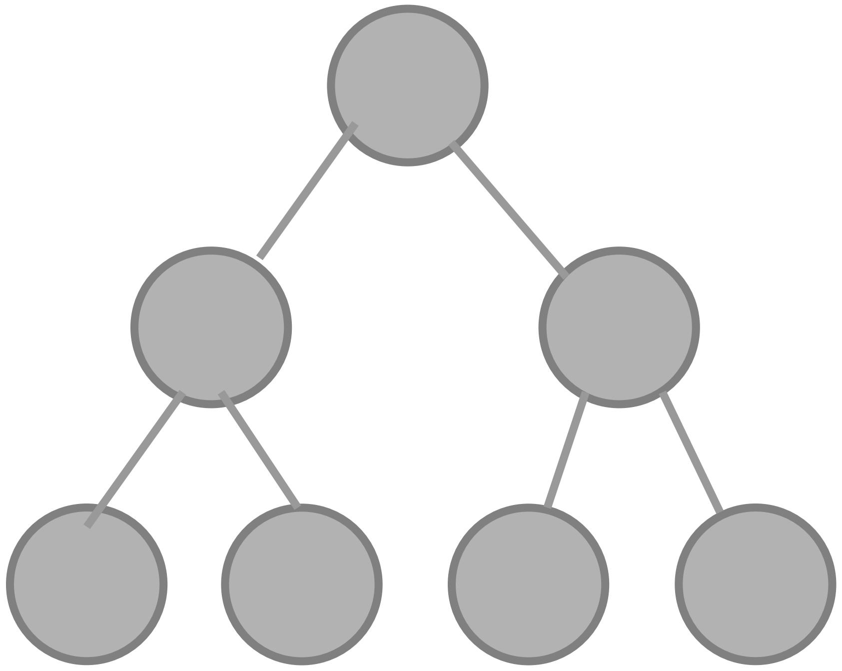


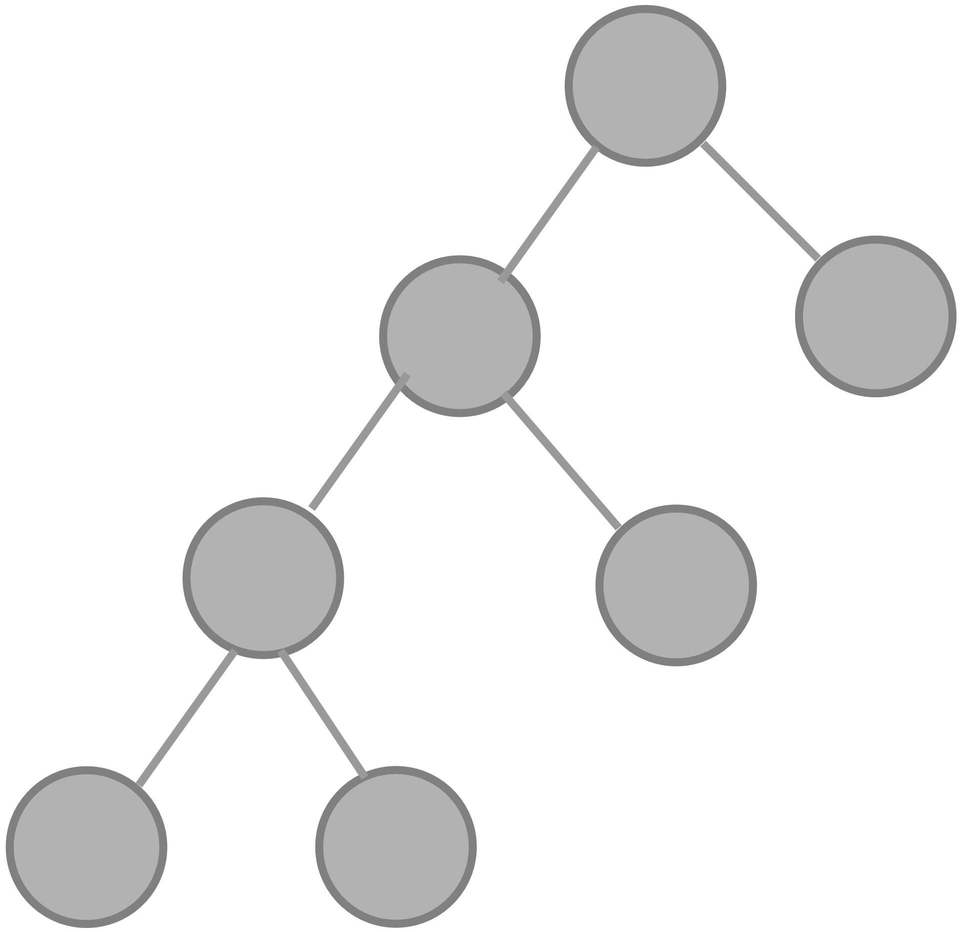


Decision Tree speed

- Depends on number of questions needed to get to a leaf node
- Which depends on the **depth** of the tree
  - Which depends on the **balance** of the tree

Which tree has more balance?





Depth of balanced tree

- In a balanced binary tree, each time you ask a question
- You **halve** the number of remaining questions
  - Just like the optimal Guess Who strategy!

Repeated halving

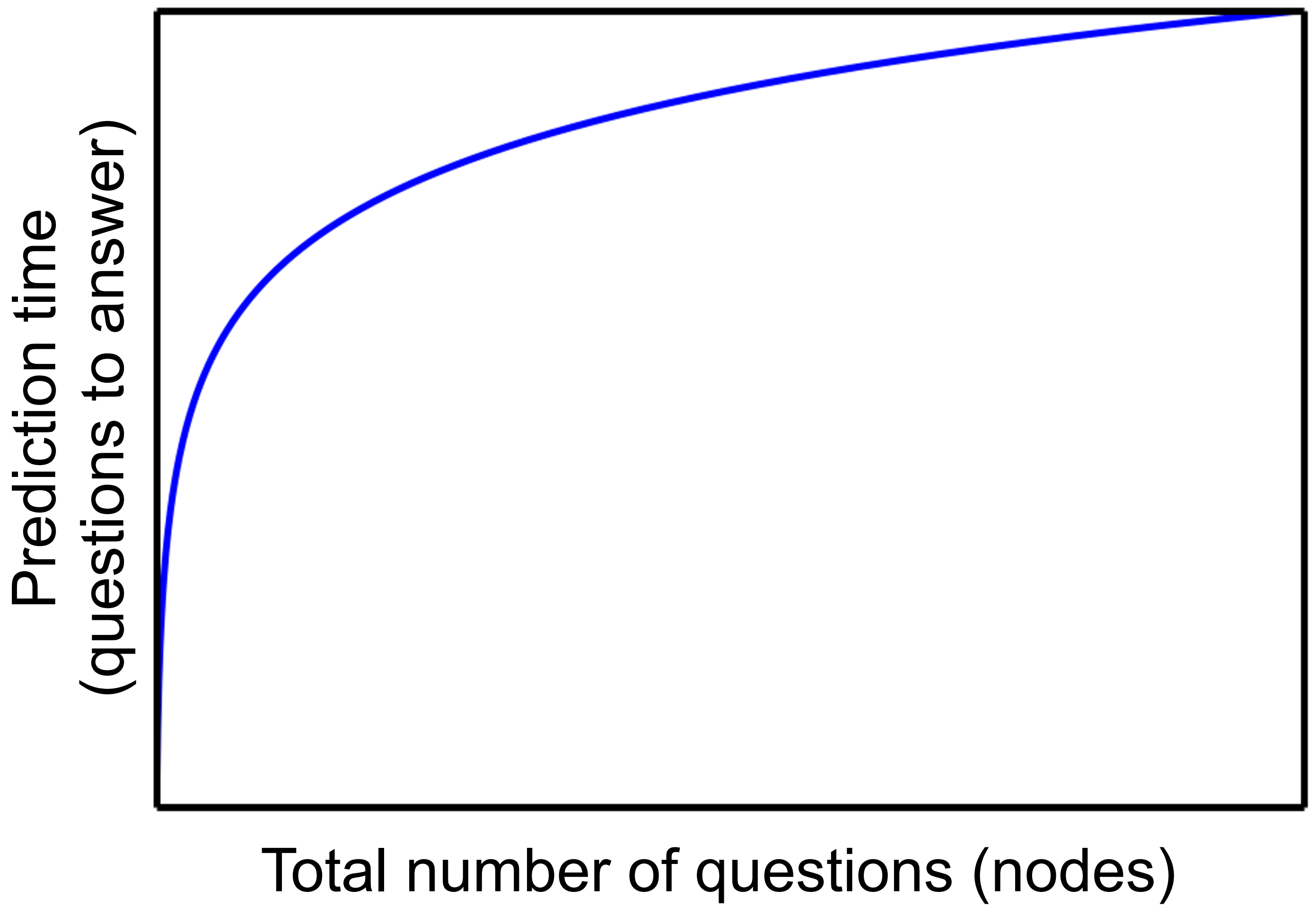
- How many halvings of **N** to get to **1**?
- How many doublings of **1** to get to **N**?

$$(((1 \times 2) \times 2) \times 2) = 8$$

$$2^3 = 8$$

$$\log_2(8) = 3$$

Prediction speed (balanced trees)



Agenda

- What is a decision tree?
- Learning DTs
- DT properties
- More about selecting questions
- Controlling overfitting
- The more the merrier: Ensembles of trees

Generating Yes/No Questions

How to generate a Question depends on the variable type

What is a possible question for a categorical variable with 5 types (A, B, C, D, and E)?

Generating Yes/No Questions

How to generate a Question depends on the variable type

What is a possible question for a continuous variable (for example, age)?

How do we generate questions?

- Categorical values
  - Binarize (convert to 1/0 or YES/NO)
- Numerical values
  - Discretize
  - Questions of the form: $\text{is } x_i <= \text{threshold}_j?$
  - Thresholds: based on data

Discretization

YearsEducation
13
13
9
7
13
14

<=7	**<=9**	**<=13**	**<=14**
No	No	Yes	Yes
No	No	Yes	Yes
No	Yes	Yes	Yes
Yes	Yes	Yes	Yes
No	No	Yes	Yes
No	No	No	Yes

Measures of Node Impurity

1. Misclassification
2. Entropy
3. Gini Impurity

Misclassification impurity

Proportion of misclassified examples in node P

$$I(P) = 1 - \boxed{\max_i (P_i)}$$

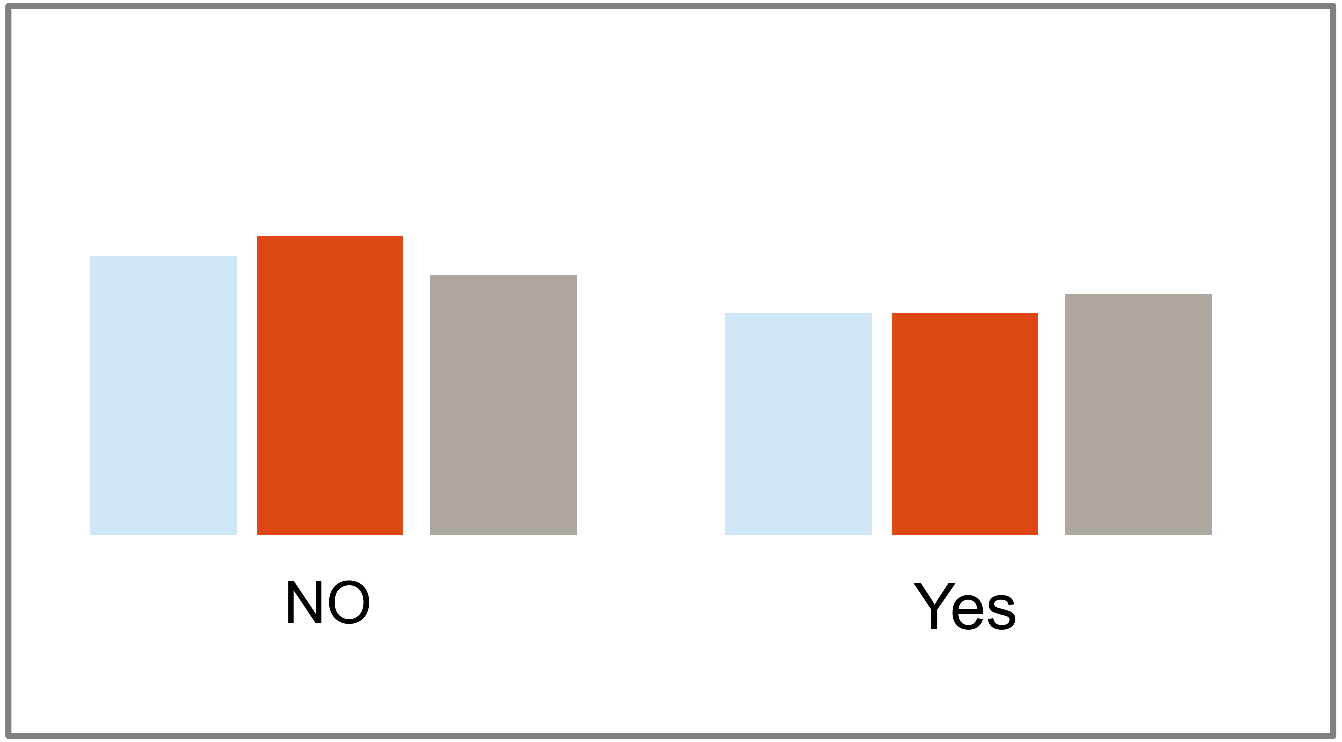
Across all i labels in node P, the proportion correctly labeled by the best label

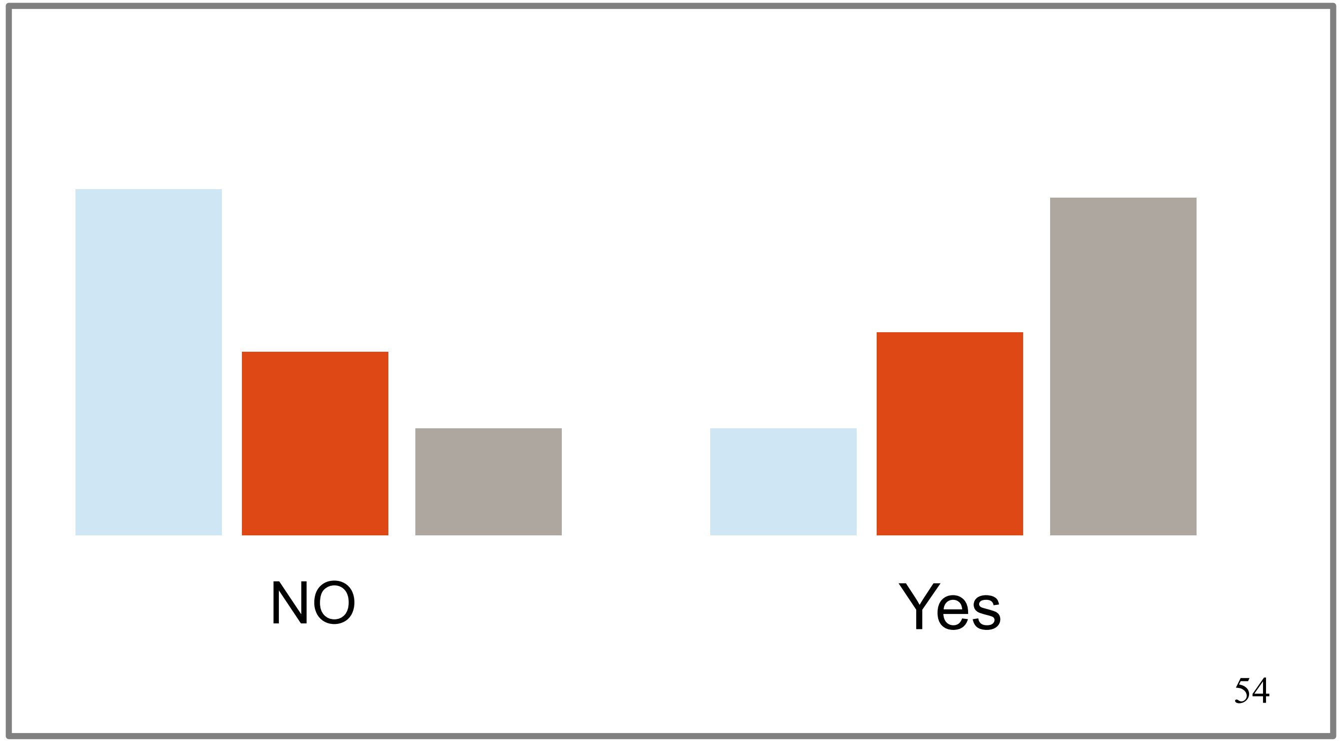
Entropy

Measure of the uniformity of a distribution

Entropy

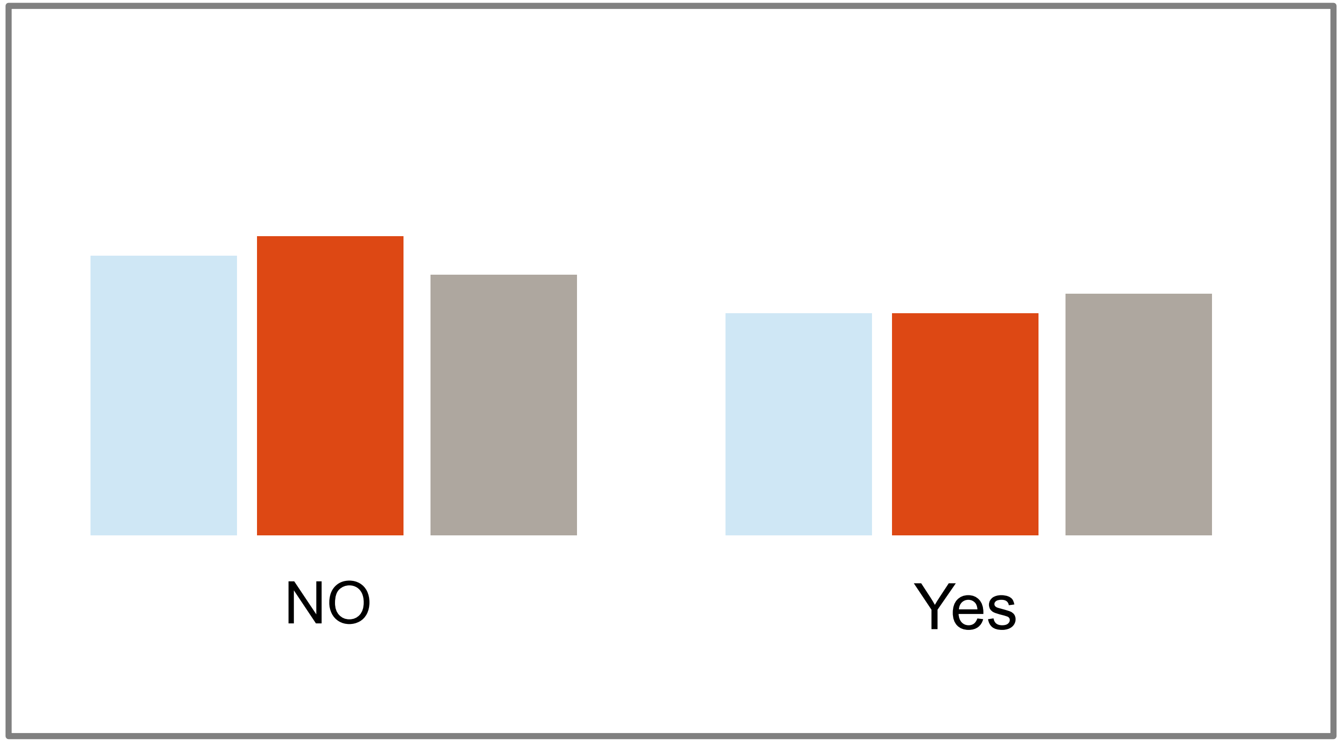
Measure of uniformity of distribution



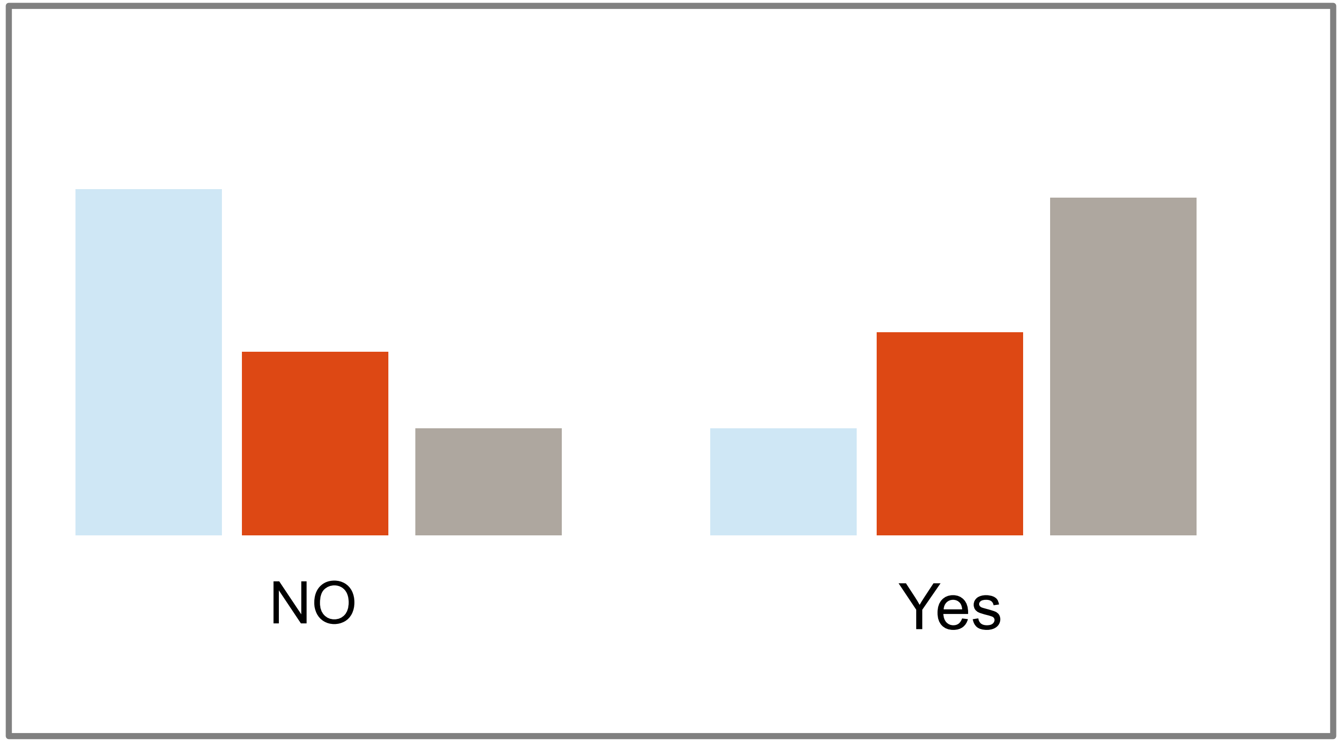


Which node has higher average entropy in the two branches?

Node A

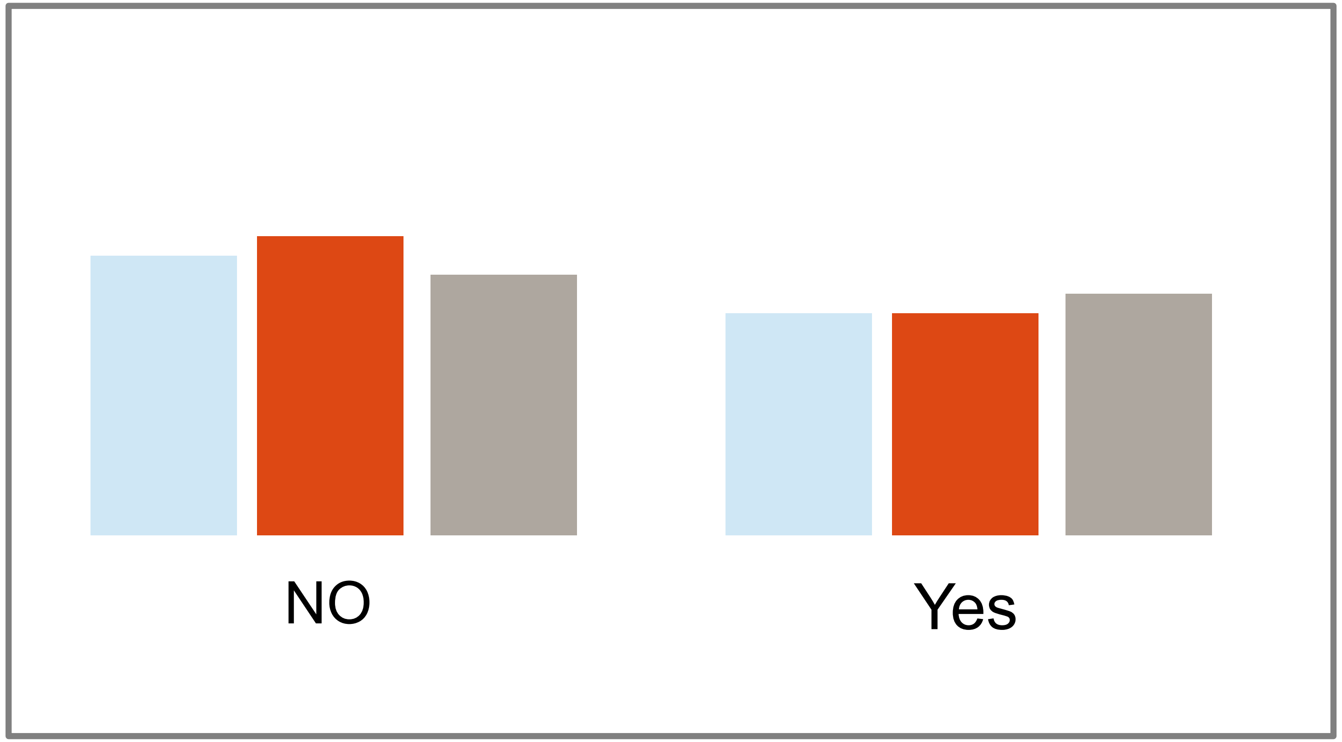


Node B

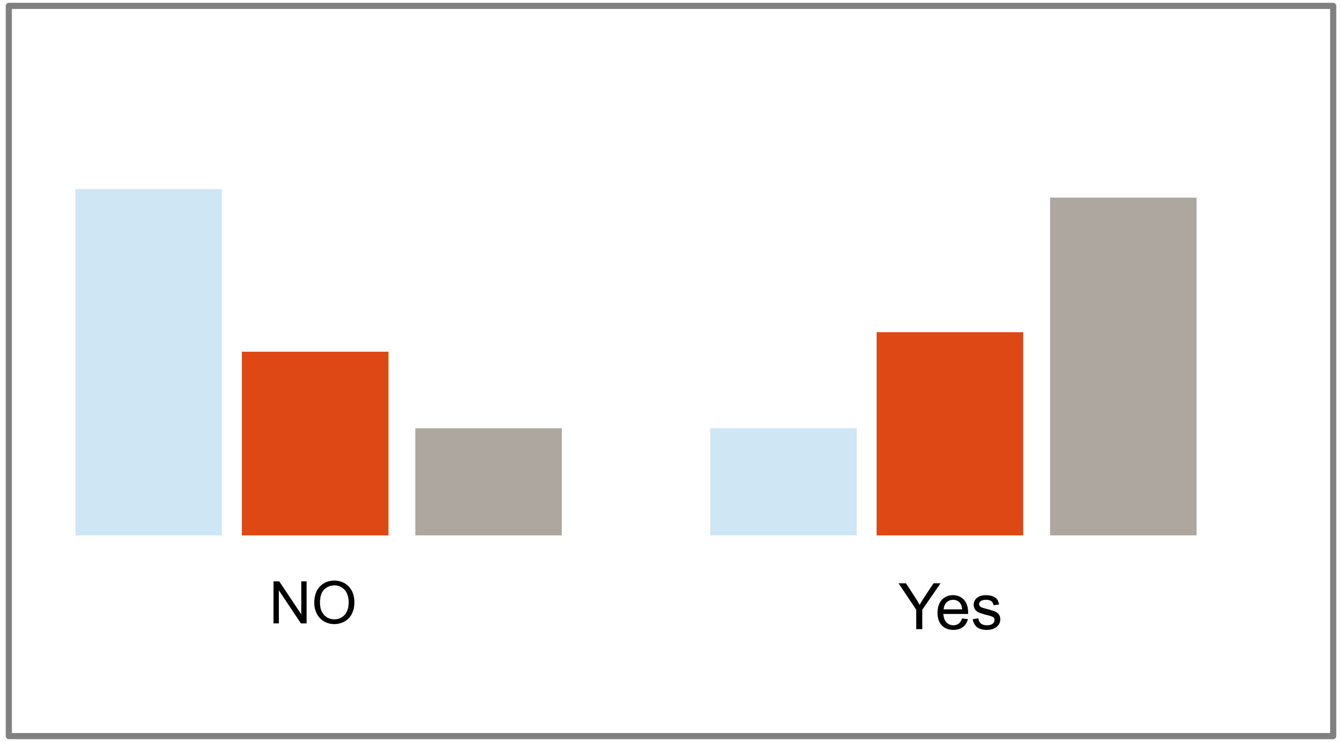


Which node has a lower impurity (based on entropy)?

Node A



Node B



Entropy

A measure of uncertainty, where more uniform distributions have more uncertainty.

The probability of label i

$$I_H(P) = -\sum_i P_i \log_2(P_i)$$

The log of the probability of label i

Gini impurity

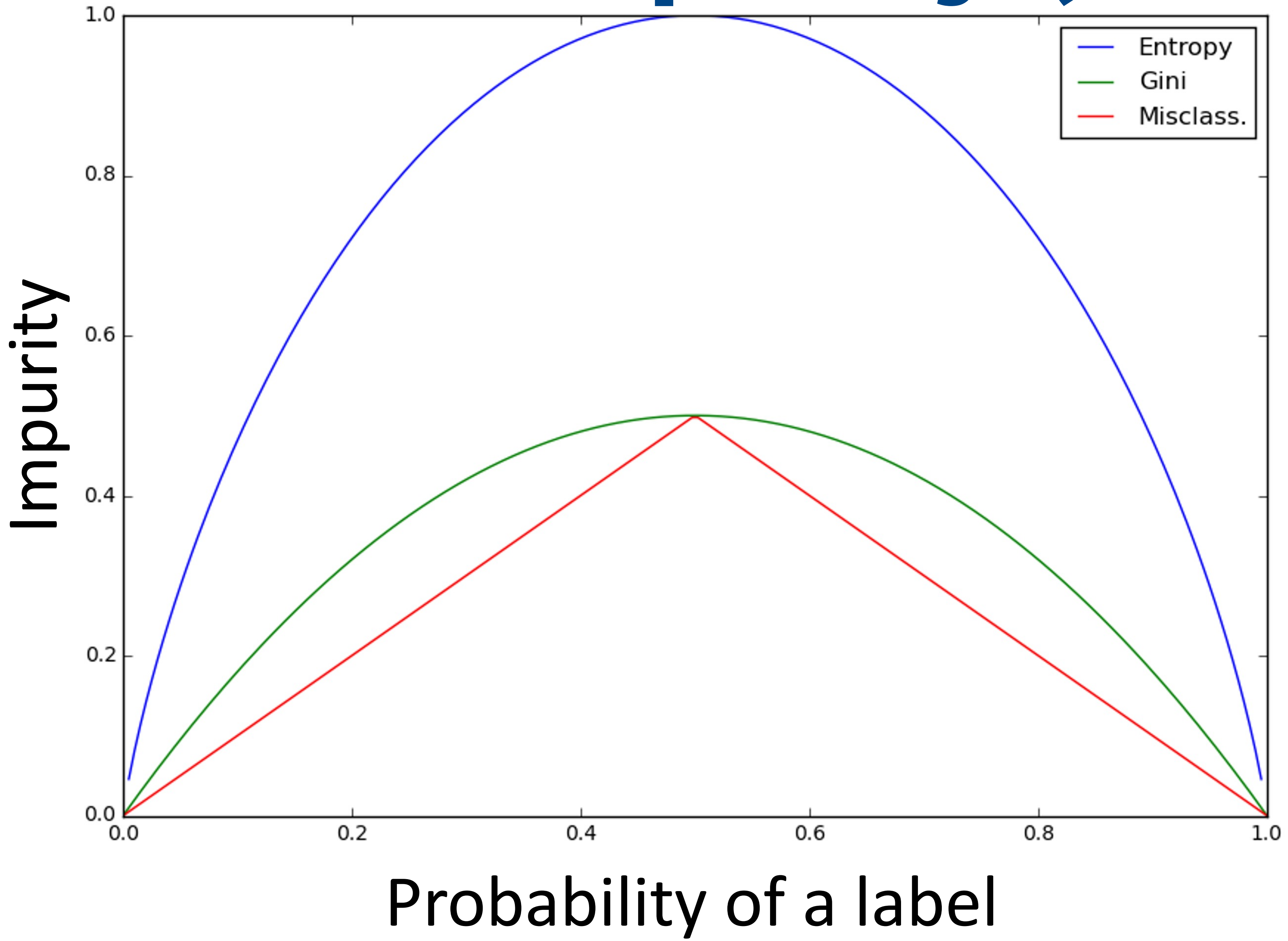
How often a random element would be labeled incorrectly if labels were assigned at random from given distribution.

The probability of label i

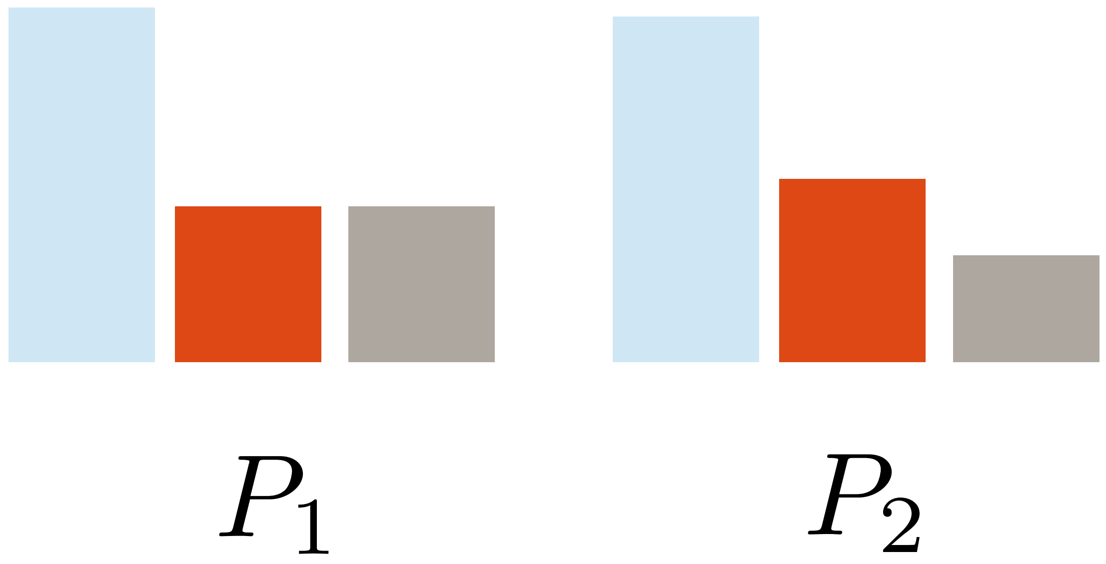
$$I_G(P) = \sum_i \boxed{P_i} \boxed{(1 - P_i)}$$

The inverse of the probability of label i

Measures of impurity (2 labels)



Impurity with 3 classes



Measure	P_1	P_2
Gini	0.560	0.540
Entropy	1.371	1.295
Misscl.	0.400	0.400

Impurity Measures

- Gini vs Entropy
  - Little impact on overall performance
- Misclassification impurity not in common use

Question Effectiveness

$$G(q,n) = f(q(n)_{\text{left}})I(q(n)_{\text{left}}) + f(q(n)_{\text{right}})I(q(n)_{\text{right}})$$

The effectiveness G() of a question *q* given the items at node *n* is the sum of:

- The number of items in the left sub-tree *f(left)* multiplied by the impurity of the left sub-tree *I(left)*
- The number of items in the right sub-tree *f(right)* multiplied by the impurity of the right sub-tree *I(right)*

Where $q(n)_{left}$ indicates the set of examples where the answer is NO for question q applied to node n and $q(n)_{right}$ indicates the set of examples where the answer is YES for question q applied to node n.

Selecting a question

$$G(q,n) = f(q(n)_{\text{left}})I(q(n)_{\text{left}}) + f(q(n)_{\text{right}})I(q(n)_{\text{right}})$$

- Minimize resulting impurity of the split
- Weighted by relative size of left/right branch

$$\hat{q} = \arg\min_{q} G(q,n)$$

Agenda

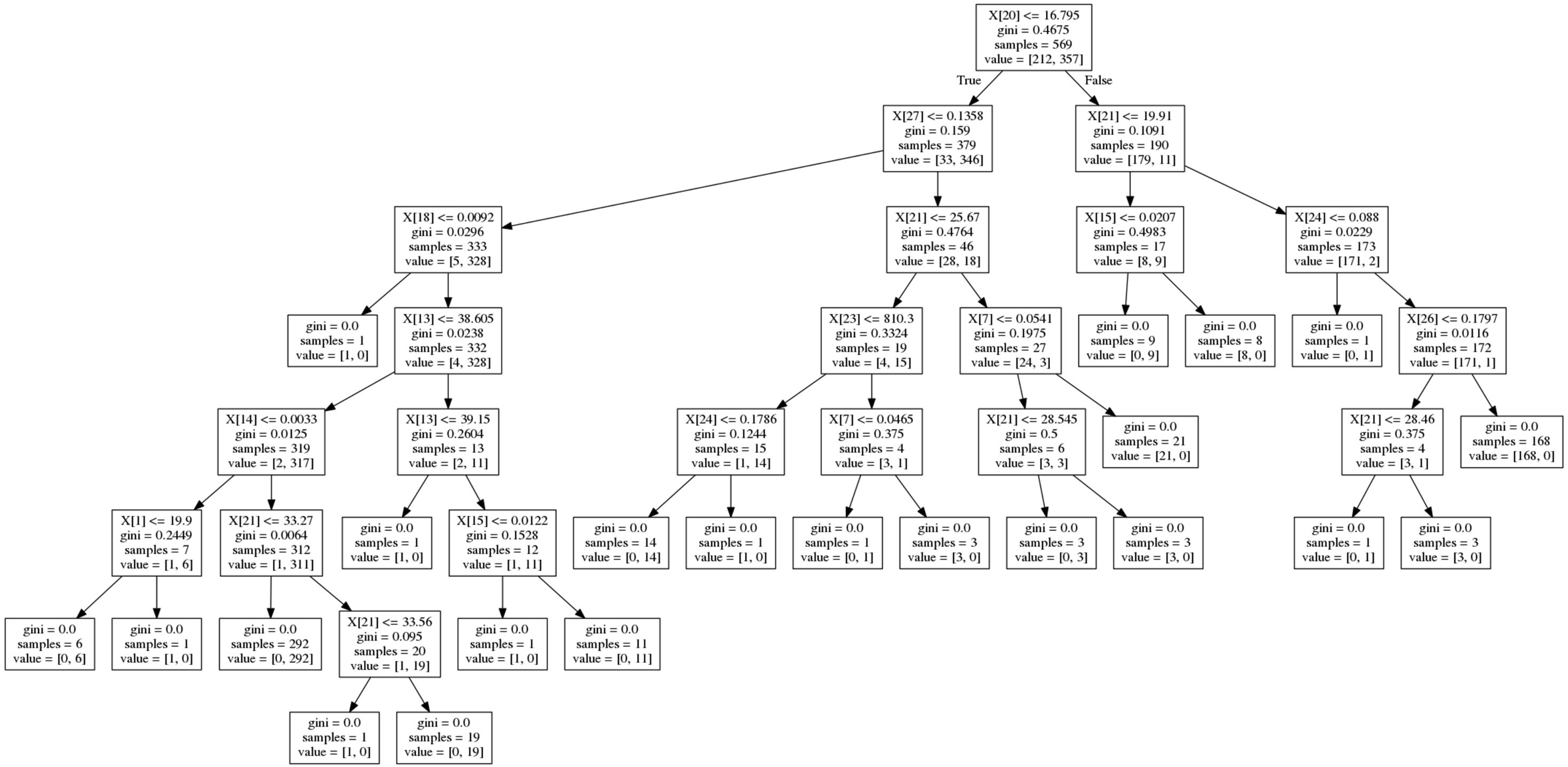
- What is a decision tree?
- Learning DTs
- DT properties
- More about selecting questions
- Controlling overfitting
- The more the merrier: Ensembles of trees

What are the advantages of Decision Trees

Breast Cancer Wisconsin (Diagnostic) Database

- mean radius
- mean texture
- mean perimeter
- mean area
- mean smoothness
- mean compactness
- mean concavity
- mean concave points
- mean symmetry
- mean fractal dimension
- radius error
- texture error
- perimeter error
- area error
- smoothness error
- compactness error
- concavity error
- concave points error
- symmetry error
- fractal dimension error
- worst radius
- worst texture
- worst perimeter
- worst area
- worst smoothness
- worst compactness
- worst concavity
- worst concave points
- worst symmetry
- worst fractal dimension

Interpretability?

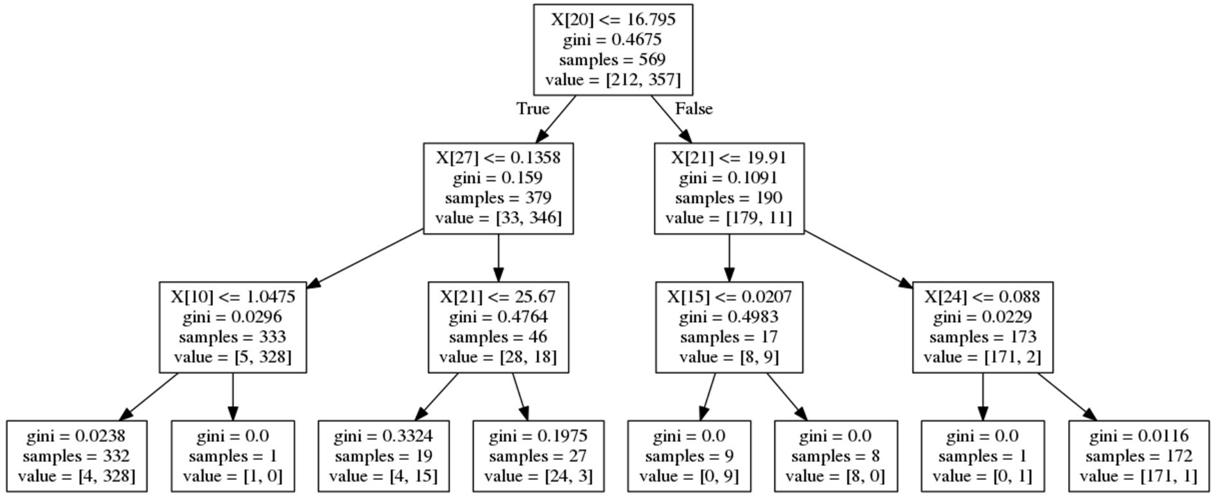


Drawback

- Very intricate tree shape
- Sensitive to minor details of data
- Tend to overfit

- We can try limiting the depth of the tree
  - Better interpretability
  - Less chance of overfitting

depth=3



- 20 worst radius
- 27 worst concave points
- 21 worst texture
- 10 radius error
- 21 worst texture
- 15 compactness error
- 24 worst smoothness

What about fixing the sensitivity to minor details of the data?

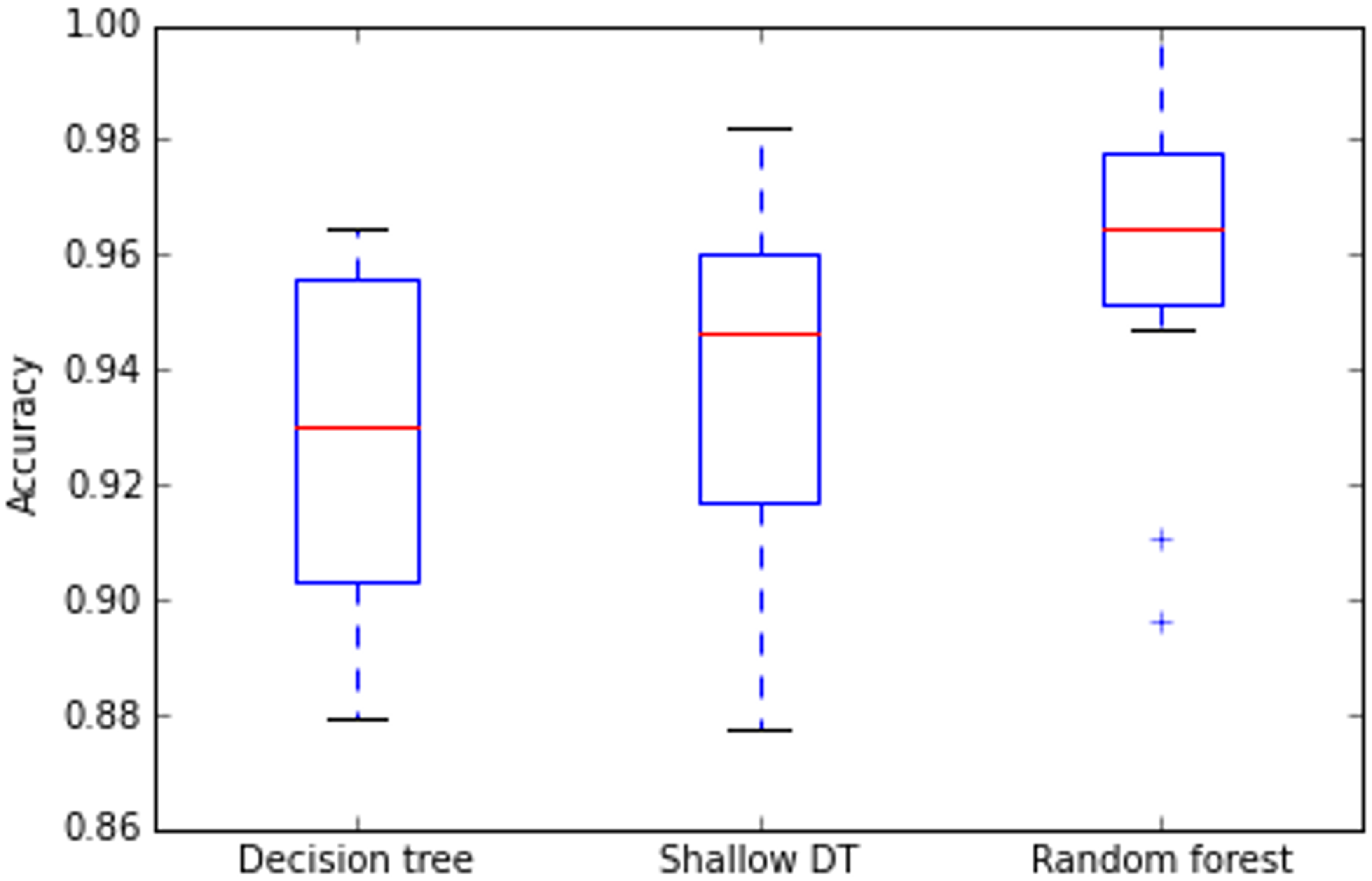
Ensembles of trees

- 10 trees are better than one.
- With original data of *m* features and *n* items, for each tree in the forest generate a new dataset by sampling
  - Randomly sample *n* items with replacement
  - Randomly sample < *m* features
  - Train the decision tree on this new dataset
- Prediction: Majority vote

Random Forest

- An example of a bagging algorithm
  - *Bootstrap* sampling of the data
  - *Aggregating* the results of multiple models
- Bagging helps reduce overfitting and sensitivity to minor data details
  - Cost: loss of interpretability

10 different train/test splits



Summary

- DT implement nested if-then-else rules
- Impurity criteria used to choose questions
- Control overfitting
  - Limit depth
  - Ensembles of DTs: Random forests

Image credits

- Red apple http://upload.wikimedia.org/wikipedia/commons/2/24/Redapple.jpg
- Banana http://upload.wikimedia.org/wikipedia/commons/8/8a/Banana-Single.jpg
- Lime http://upload.wikimedia.org/wikipedia/commons/5/55/Lime_closeup.jpg
- Lemon https://openclipart.org/image/300px/svg_to_png/189589/lemon-citrina.png
- Green apple http://upload.wikimedia.org/wikipedia/commons/5/55/GreenApple.png
- Green Banana http://pixabay.com/en/green-bananas-tip-garden-banana-108109/
- Green lemon http://pixabay.com/static/uploads/photo/2013/12/15/11/33/lemon-228857_640.jpg