Introduction to Artificial Intelligence

COMP307/AIML420
Decision Tree Learning Methods

Dr Andrew Lensen

Andrew.Lensen@vuw.ac.nz

How do **humans** make decisions?

- Have a think about the last decision you made…
- What factors did you take into account?
- Can you make these into a set of rules?

How can we make this into an algorithm?

- It would be really useful to teach computers to “think” in this way
- Allows computers to automatically make decisions for us
- For example, “Should this credit card purchase be approved?”
- Today we will discuss decision trees, which do just this!

A real-world example:

- Which habitat is best for raising capybara offspring?

Table 1. Decision rules for allocating values of forage, shelter and rest and protection for offspring for different categories in which plant species were grouped according to criteria of Quintana and Bolkovic (unpublished data 2014).

Ecological requirements	Plant category		Mean size	Value	
Forage	Aquatic	*Oplimenopsis* spp., *Luziola peruviana*, *Leersia hexandra*	All	High	3
		All other	All	Low	1
	Broadleaf herbs		All	Low	1
	Sedges		Shorter than 1 m	High	3
			Taller than 1 m	Medium	2
	Terrestrial grasses		All	High	3
	Shrubs		All	Null	0
	Trees		All	Null	0
Shelter	Aquatic		All	Medium	2
	Herbaceous (broadleaf herbs, sedges, and grasses)		Shorter than 0.4 m	Null	0
			Between 0.4 and 0.8 m	Low	1
			Between 0.8 and 1.2 m	Medium	2
			Taller than 1.2 m	High	3
	Shrubs		All	High	3
	Trees		All	Medium	2
Rest and protection for offspring	Aquatic		All	High	3
	Herbaceous (broadleaf herbs, sedges, and grasses)		Shorter than 1 m	Low	1
			Taller than 1 m	High	3
	Shrubs		Shorter than 0.5 m	Low	1
			Between 0.5 and 1 m	Medium	2
			Taller than 1 m	High	3
	Trees		All	High	3

Schivo, Facundo, et al. "A Habitat Suitability Model for Capybara (Hydrochoerus Hydrochaeris) at Its Core Area in Argentina." Tropical Conservation Science, Mar. 2015, pp. 150–168, doi:10.1177/194008291500800113.

Outline

- What is a decision tree?
- How do we create a decision tree given a set of instances (inputs)?
- How can we use impurity to measure a decision tree node's quality?
- Using DTs for multi-class classification or with other purity measures

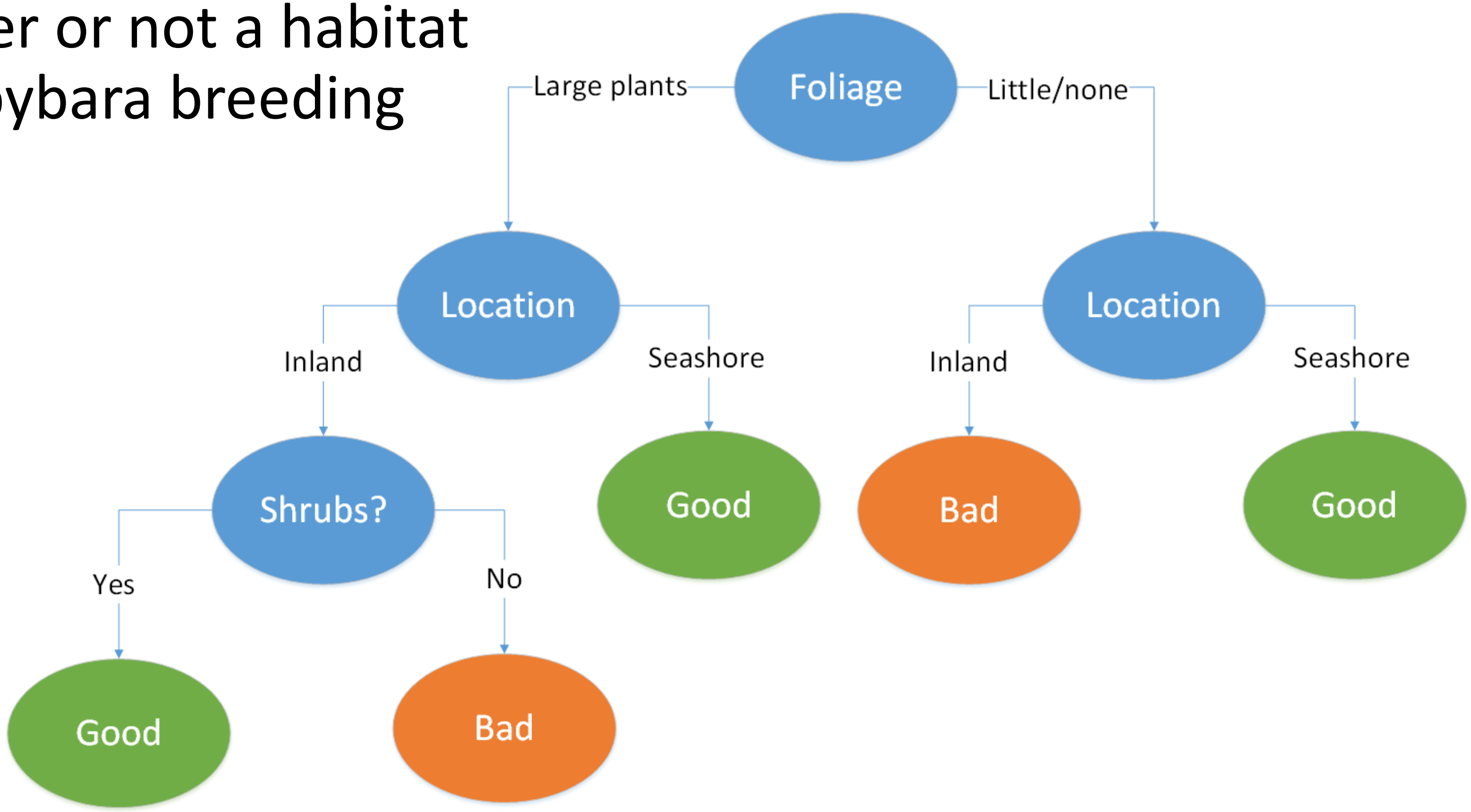
What is a Decision Tree?

- A symbolic classifier which uses a series of decisions/rules to make a classification
  - One "big" rule to make an overall decision
  - Easy to interpret?

- *Decision tree learning* is a method for creating decision trees
  - One of the first classification learning methods in AI!

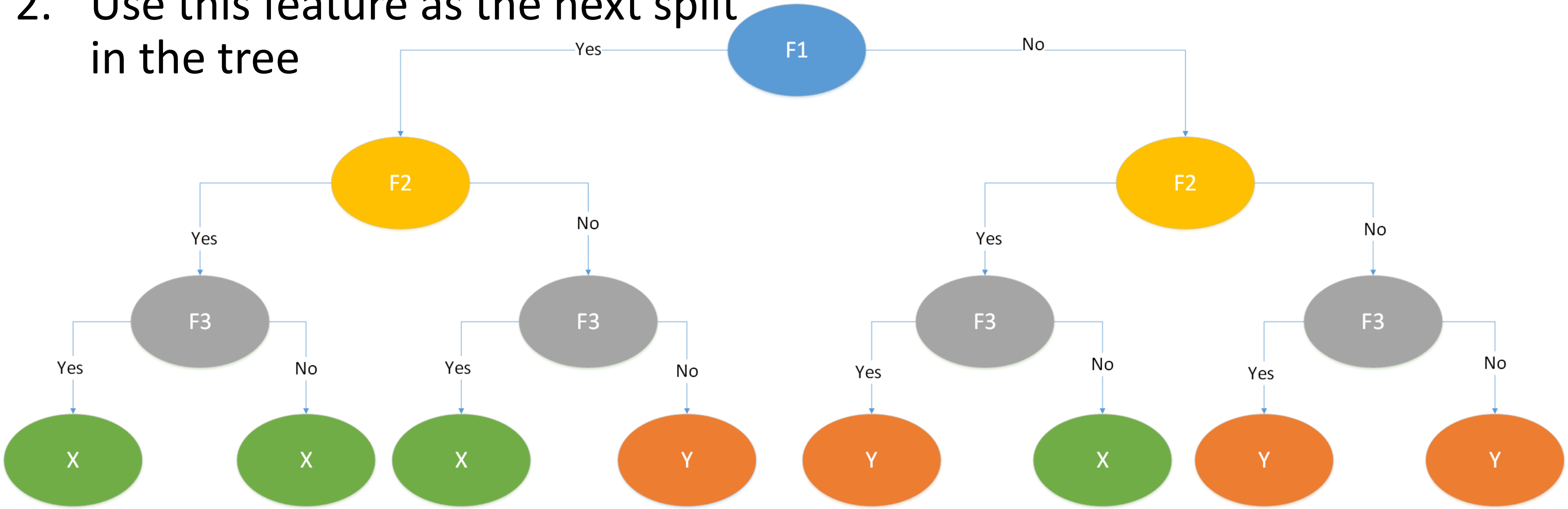
An example decision tree

- Decide whether or not a habitat is good for capybara breeding



Constructing Decision Trees

- Easy to build a naïve (**full**) decision tree:
    1. Pick one feature at a time
    2. Use this feature as the next split in the tree
    3. Repeat until all features used
    4. Label each leaf with the corresponding class.



What are the problems with this?

- No **learning** is taking place! Just *remembering* the inputs
  - Too specific…not likely to work well on future (unlabelled) inputs

How can we do better?

A smaller decision tree:

- Capture the common patterns across the inputs
- Discover some underlying rules which are more generic

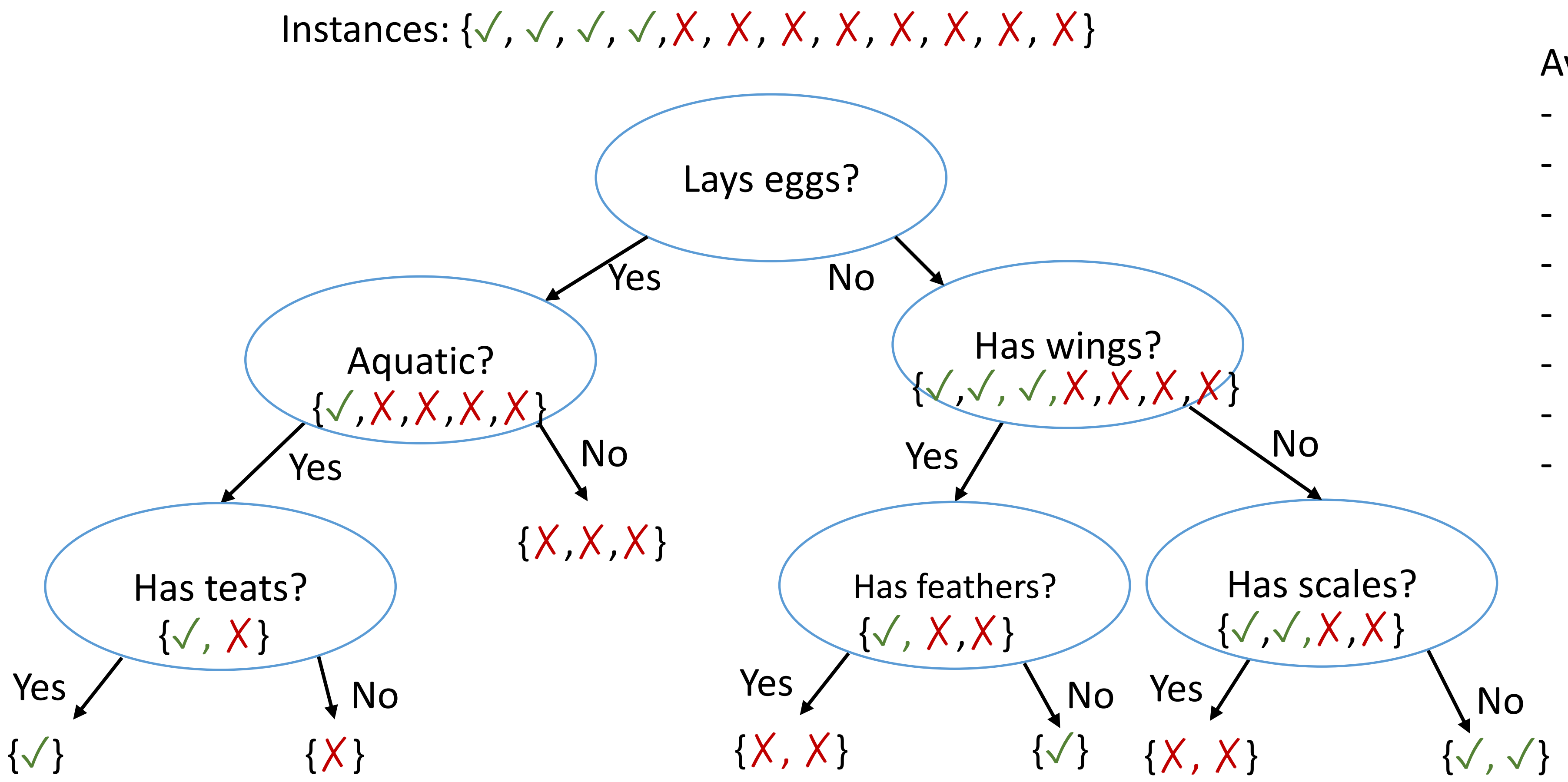
A better algorithm

Input: a set of instances defined by their feature values

Output: a decision tree classifier which performs classification

1. Compute if the set of instances is *pure* enough – if so, then stop!
2. Select the best (unused) feature to use as the next node – the one which would give the highest *purity*
3. Split the dataset into two sub-sets according to the chosen node
4. Recurse on each of the two sub-sets

Building a DT: is the animal a mammal?



Available features:

- Eggs
- Scales
- Aquatic
- Tail
- Legs
- Feathers
- Milk (has teats)
- Wings

https://archive.ics.uci.edu/ml/datasets/zoo

Measuring purity

- How do we measure the purity of a node?
  - Many different ways! Probability-based, information theory-based, …
  - More pure = can predict the class more confidently
  - Often easier to talk about impurity

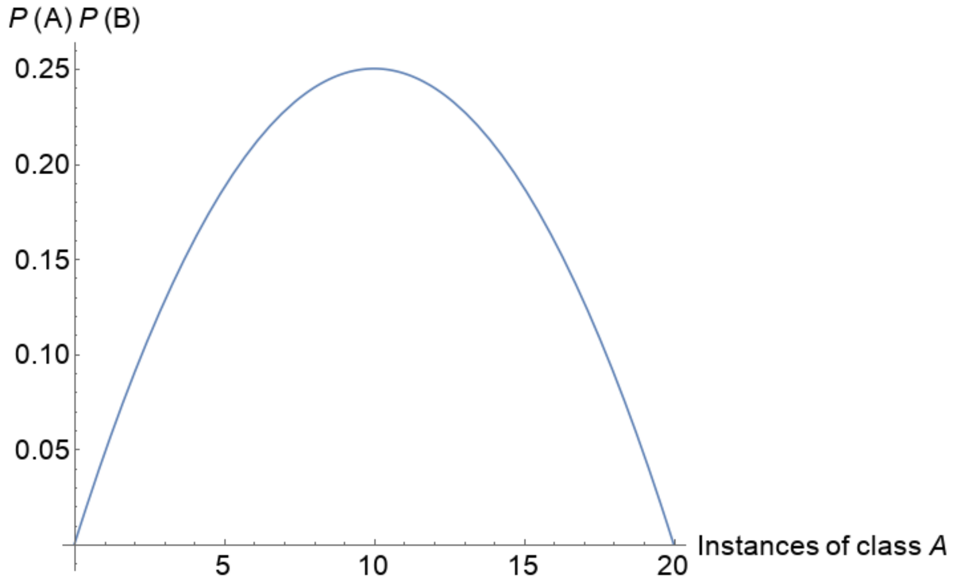
Most impure (least pure) | Less impure (more pure) | Least impure (most pure)

A probability-based impurity measure

- If we have class **A** and **B**, then we measure a node's impurity using P(**A**)P(**B**), where there are *m* instances of class **A** and *n* of class **B**:

$$\text{Impurity} = P(\boldsymbol{A})P(\boldsymbol{B}) = \frac{\boldsymbol{m}}{\boldsymbol{m+n}} \times \frac{\boldsymbol{n}}{\boldsymbol{m+n}} = \frac{\boldsymbol{mn}}{(\boldsymbol{m+n})^2}$$

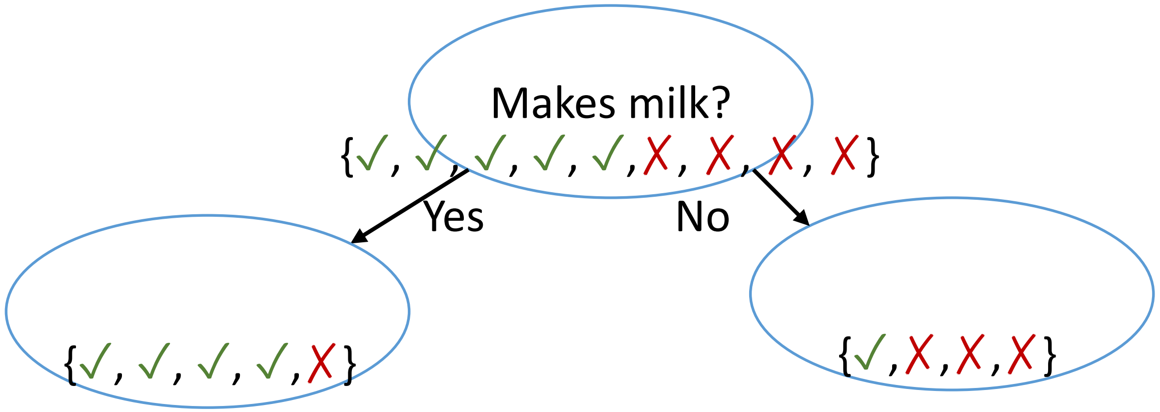
- For example, if we have 20 instances:



- This measure is *smooth*

Choosing the next feature split

- When we choose a feature to be the next node, we create two children nodes, which each have their own impurity
- How do we combine these two impurity values?



$$\text{Impurity} = \frac{4}{1+4} \times \frac{1}{1+4} = 0.16$$

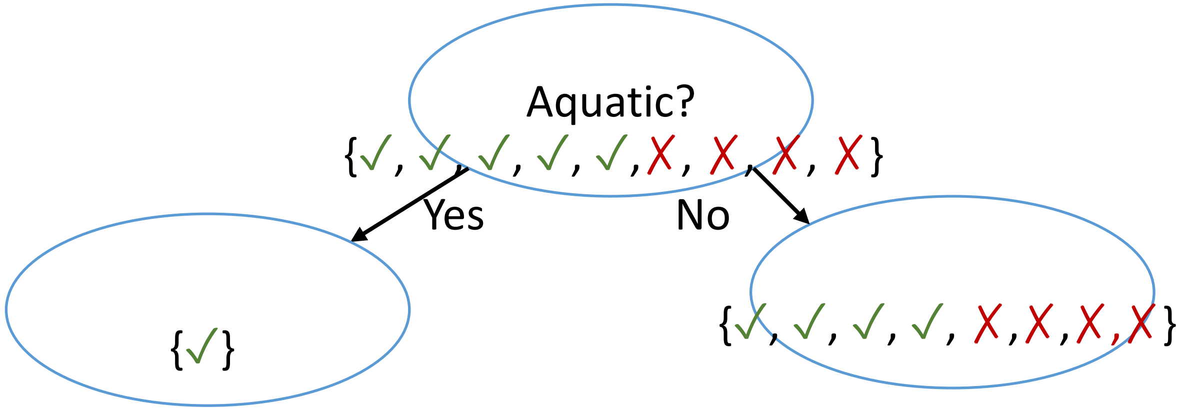
$$\text{Impurity} = \frac{1}{1+3} \times \frac{3}{1+3} = 0.1875$$

The simplest way: take the average of the two!

$$\text{Average Impurity} = \frac{0.16+0.1875}{2} = 0.17375$$

What is wrong with this?

Consider this example:

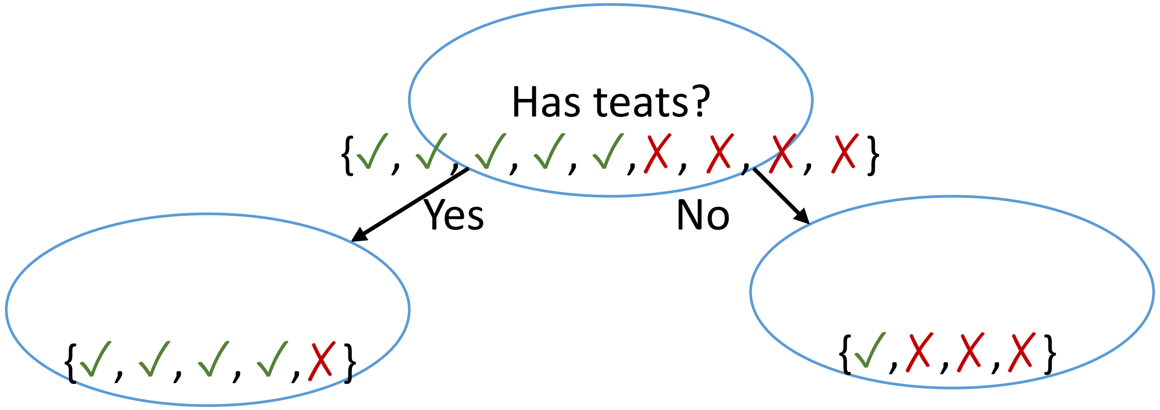


$$\text{Impurity} = \frac{1}{1} \times \frac{0}{1} = 0$$

$$\text{Impurity} = \frac{4}{8} \times \frac{4}{8} = 0.25$$

$$\text{Average Impurity} = \frac{0+0.25}{2} = 0.125$$

But wasn't this a better split?:



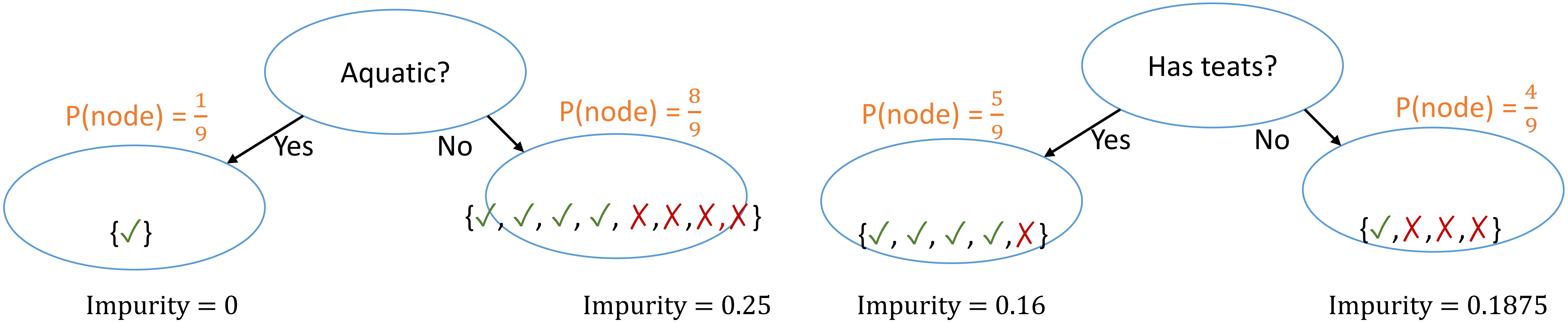
$$\text{Impurity} = \frac{4}{1+4} \times \frac{1}{1+4} = 0.16$$

$$\text{Impurity} = \frac{1}{1+3} \times \frac{3}{1+3} = 0.1875$$

Average Impurity = 0.17375!

The solution: **weighted** average impurity

- If we weight a child's impurity by how many instances it has, then the measure is not biased to uneven splits



Weighted average impurity = $\frac{1}{9} \times 0 + \frac{8}{9} \times 0.25$
≈ 0.222

Weighted average impurity = $\frac{5}{9} \times 0.16 + \frac{4}{9} \times 0.1875$
≈ 0.172

$$\text{Weighted average impurity} = \sum_i P(node_i) \times Impurity(node_i)$$

Advanced topics: More than two classes?

- Impurity = $P(\boldsymbol{A})P(\boldsymbol{B})$ works well for two classes
  - What about three? Impurity = $P(\boldsymbol{A})P(\boldsymbol{B})P(\boldsymbol{C})$
  - 10? Impurity = $P(\boldsymbol{A})P(\boldsymbol{B})P(\boldsymbol{C})P(\boldsymbol{D})P(\boldsymbol{E})$ …
  - Any problem?

Alternatives:

- Gini impurity: $\mathrm{Gini(X)} = \sum_{i=1}^{C} P(i) \sum_{j \neq i} P(j) = 1 - \sum_{i=1}^{C} P(i)^2$
- Entropy: $\mathrm{H}(X) = -\sum_{i=1}^{C} P(i) \log_2 P(i)$
- These often give very similar results!

Advanced topics: Numerical features (1)

- Many features are numerical, e.g. height, tail length, …
- How can our DT algorithm cope with this?

Instance #	Height (cm)	Tail length (cm)	Can fly?	Class
1	170	0	No	Mammal
2	19.5	1.1	Yes	Bird
3	90	4	No	Bird
4	15	30	Yes	Fish
5	4	70	No	Reptile

Advanced topics: Numerical features (2)

- Sort the feature we're considering
- Choose split points based on class boundaries
- Compute impurity for each split, choose best one

Instance #	Height (cm)	Tail length (cm)	Can fly?	Class
5	4	70	No	Reptile
4	15	30	Yes	Fish
2	19.5	1.1	Yes	Bird
3	90	4	No	Bird
1	170	0	No	Mammal

<15cm

<19.5cm

<170cm

Advanced topics: Overfitting

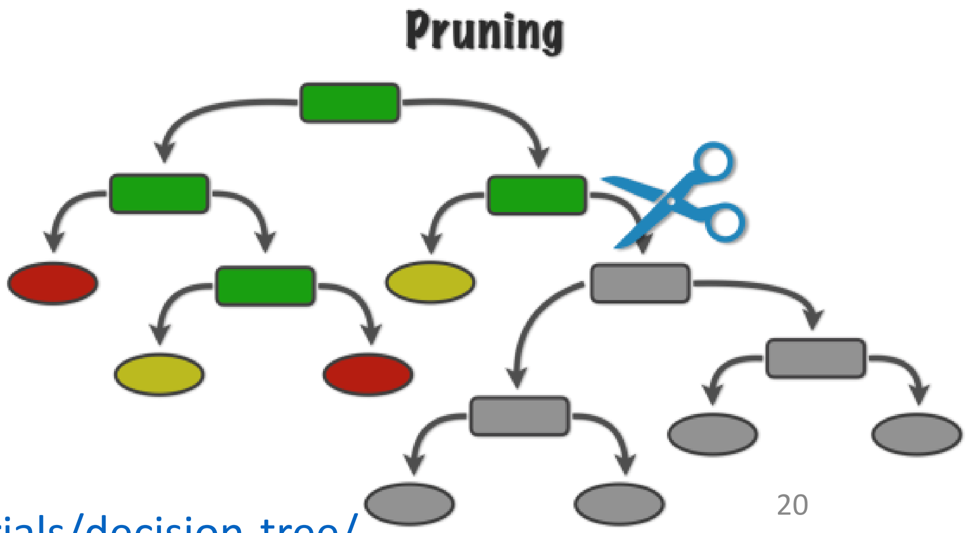
- The larger our tree, the more it will overfit, as it becomes too specific to the training set

How can we avoid this? Stop splitting earlier

- Need to be careful we don't stop too soon – otherwise nodes will not be pure enough → worse performance!

Some approaches:

- Stop at a given impurity threshold
- Stop at a pre-defined max depth
- Use pruning to simplify a completed tree



https://alanjeffares.wordpress.com/tutorials/decision-tree/