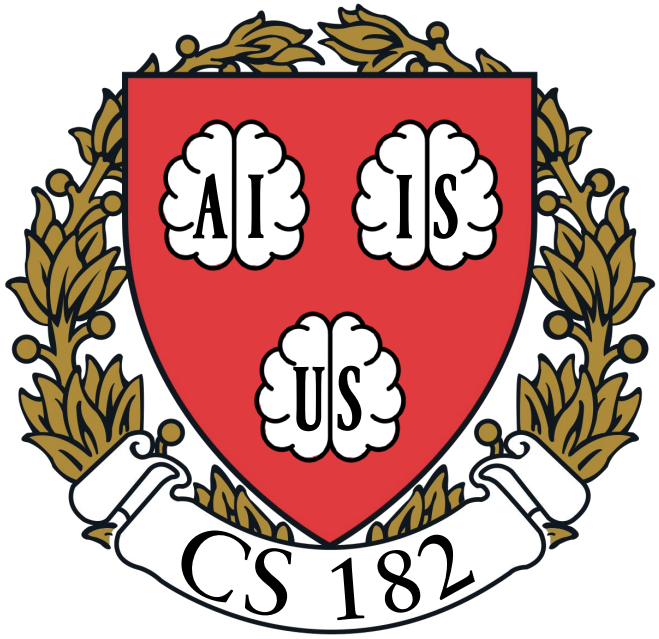


Fall 2021 | Lecture 16

Decision Trees

Ariel Procaccia | Harvard University

SUPERVISED LEARNING

- We are given a training set of n examples $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)$ where each pair was generated by an unknown function $y = f(x)$
- The goal is to find a hypothesis $h \in \mathcal{H}$ that approximates f, where $\mathcal{H}$ is called the hypothesis space
- h is chosen to be a best-fit function for which each $h\left(x^{(i)}\right)$ is "close" to $y^{(i)}$
- h generalizes well if it gives accurate predictions on a fresh test set

CLASSIFICATION

- Classification is the task of learning f whose range is a discrete, finite set
- Such a function is called a classifier
- When the cardinality of the range is 2 then the task is known as binary classification, otherwise it's called multi-class classification

EXAMPLE: IMAGE CLASSIFICATION

$(\boldsymbol{x}^{(1)}, dog)$

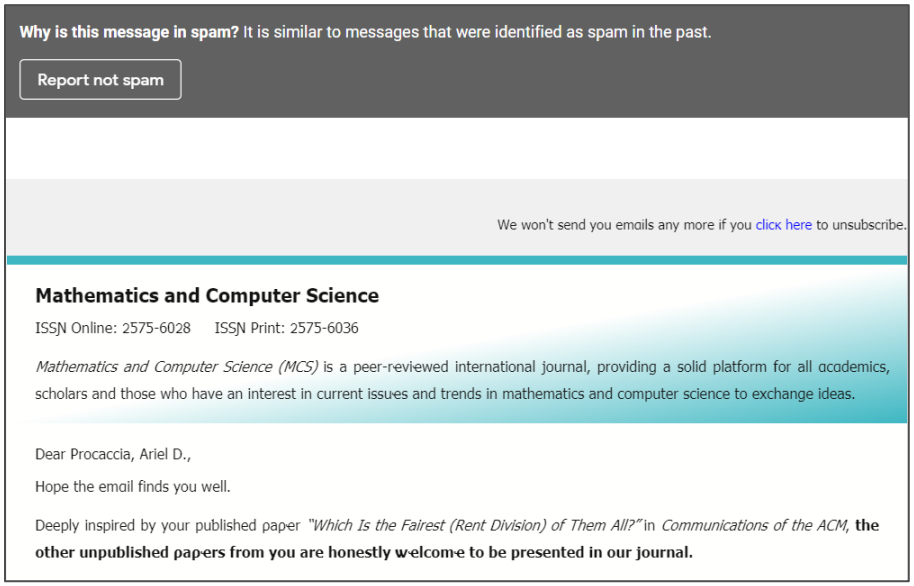
$(\boldsymbol{x}^{(2)}, cat)$

$(\boldsymbol{x}^{(3)}, dog)$

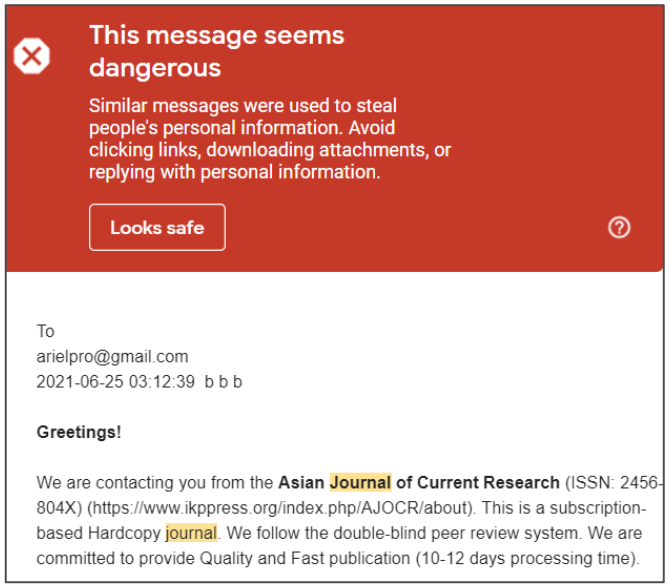
$(\boldsymbol{x}^{(4)}, dog)$

$f(\boldsymbol{x}) = ?$

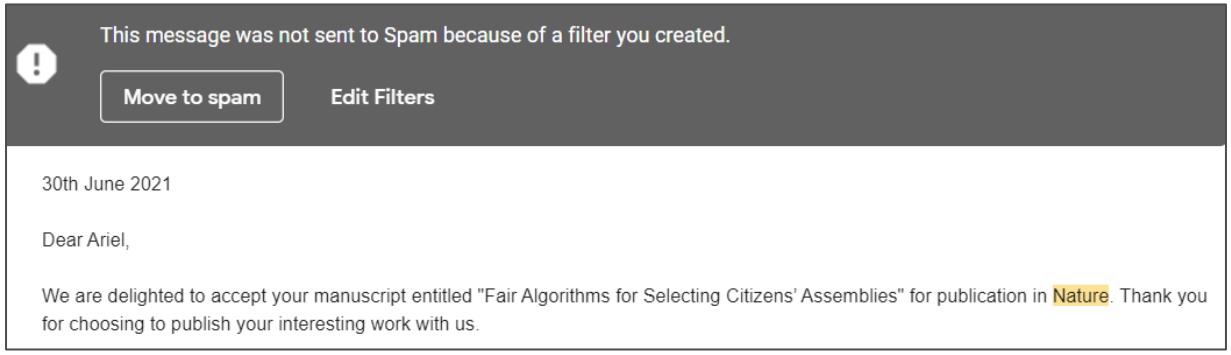
EXAMPLE: SPAM FILTER



$(\boldsymbol{x}^{(1)}, spam)$



$(\boldsymbol{x}^{(2)}, spam)$

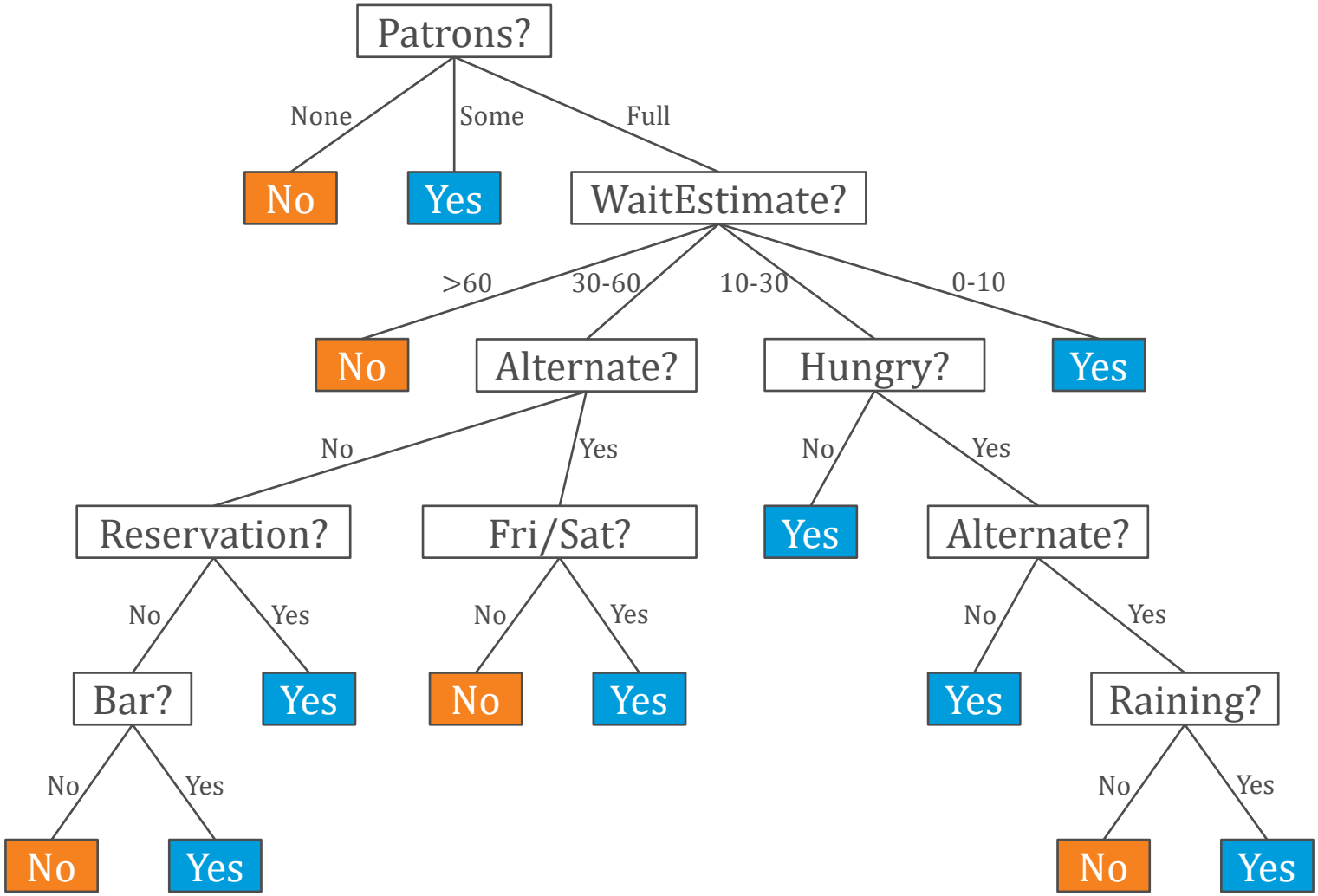


$f(\boldsymbol{x}) = ?$

EXAMPLE: RESTAURANT WAITING

Example	Input Features										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
$\boldsymbol{x}^{(1)}$	*Y*	*N*	*N*	*Y*	*Some*	$$$	*N*	*Y*	*French*	0-10	$y^{(1)} = Y$
$\boldsymbol{x}^{(2)}$	*Y*	*N*	*N*	*Y*	*Full*	$	*N*	*N*	*Thai*	30-60	$y^{(2)} = N$
$\boldsymbol{x}^{(3)}$	*N*	*Y*	*N*	*N*	*Some*	$	*N*	*N*	*Burger*	0-10	$y^{(3)} = Y$
$\boldsymbol{x}^{(4)}$	*Y*	*N*	*Y*	*Y*	*Full*	$	*Y*	*N*	*Thai*	10-30	$y^{(4)} = Y$
$\boldsymbol{x}^{(5)}$	*Y*	*N*	*Y*	*N*	*Full*	$$$	*N*	*Y*	*French*	>60	$y^{(5)} = N$
$\boldsymbol{x}^{(6)}$	*N*	*Y*	*N*	*Y*	*Some*	$$	*Y*	*Y*	*Italian*	0-10	$y^{(6)} = Y$
$\boldsymbol{x}^{(7)}$	*N*	*Y*	*N*	*N*	*None*	$	*Y*	*N*	*Burger*	0-10	$y^{(7)} = N$
$\boldsymbol{x}^{(8)}$	*N*	*N*	*N*	*Y*	*Some*	$$	*Y*	*Y*	*Thai*	0-10	$y^{(8)} = Y$
$\boldsymbol{x}^{(9)}$	*N*	*Y*	*Y*	*N*	*Full*	$	*Y*	*N*	*Burger*	>60	$y^{(9)} = N$
$\boldsymbol{x}^{(10)}$	*Y*	*Y*	*Y*	*Y*	*Full*	$$$	*N*	*Y*	*Italian*	0-30	$y^{(10)} = N$
$\boldsymbol{x}^{(11)}$	*N*	*N*	*N*	*N*	*None*	$	*N*	*N*	*Thai*	0-10	$y^{(11)} = N$
$\boldsymbol{x}^{(12)}$	*Y*	*Y*	*Y*	*Y*	*Full*	$	*N*	*N*	*Burger*	30-60	$y^{(12)} = Y$

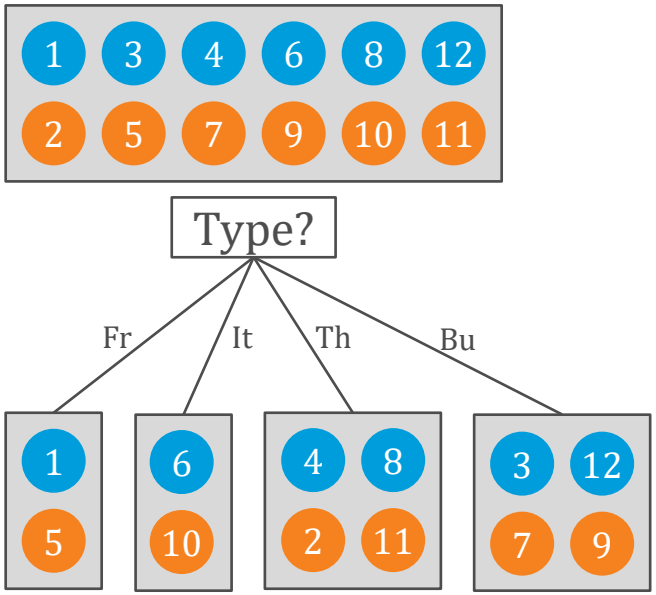
DECISION TREES

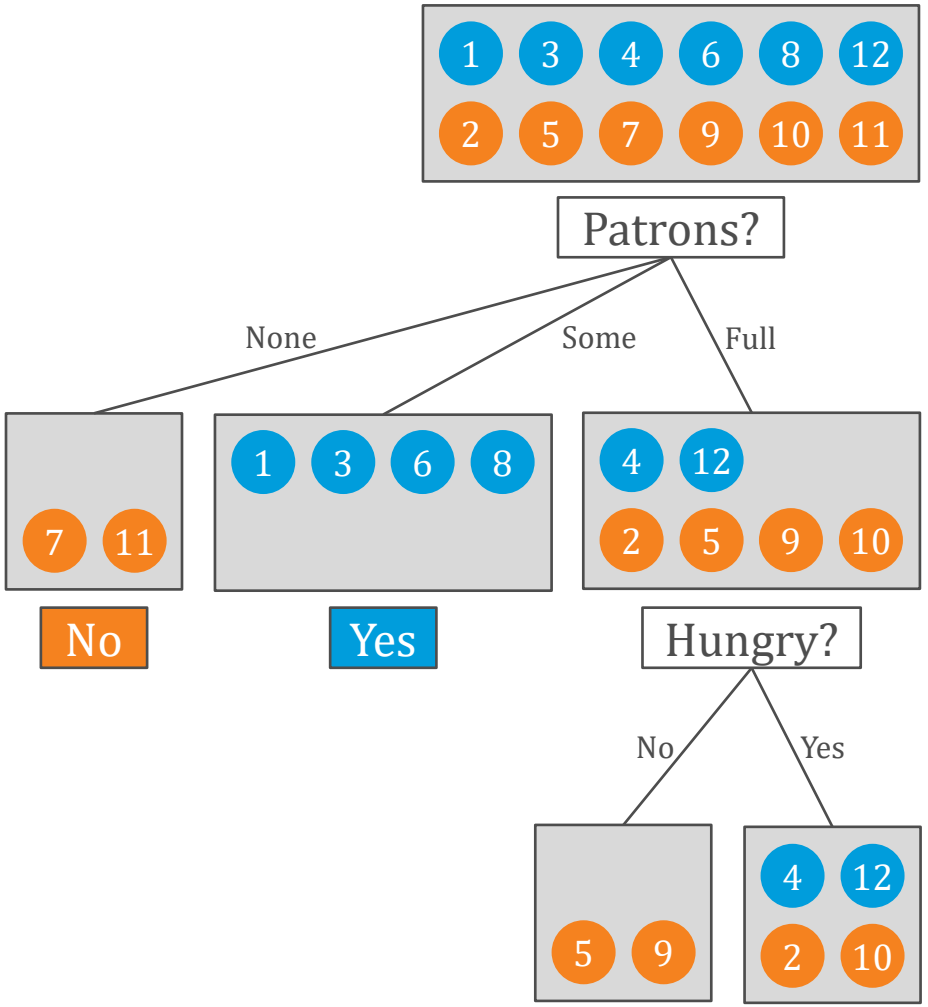


DECISION TREES

- A decision tree reaches an output (in the leaves) through a sequence of tests on the input attributes (in internal nodes)
- Decision trees can represent any classifier, but some may require a large tree
- Poll 1: Which of the following Boolean functions can be represented via a tree of linear size?
  - Unanimity ✓
  - Parity
  - Majority
  - None of the above

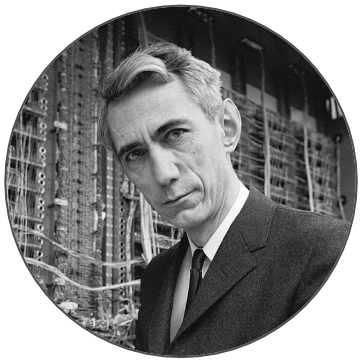
SPLITTING ON FEATURES





LEARNING DECISION TREES

```
function LEARN-DT(examples, features, parent_examples)
    if examples = ∅ then return PLURALITY-VALUE(parent_examples)
    else if all examples have the same label then return
    that label
    else if features = ∅ then return PLURALITY-VALUE(examples)
    else
        A ← argmax_{a∈features} IMPORTANCE(a, examples)
        tree ← new decision tree with root test A
        for each value v of A do
            new_examples ← {e ∈ examples: e.A = v}
            subtree ←LEARN-DT(new_examples, features \ {A}, examples)
            add branch to tree with label A = v and subtree
        return tree
```



Claude Shannon

1916-2001

Mathematician and electrical engineer, father of information theory. Also remembered as a prankster.

INFORMATION GAIN

- To instantiate the IMPORTANCE function we will use the notion of entropy, which is measured in bits
- The entropy of random variable V that takes each value v with probability $P(v)$ is

$$H(V) = \sum_v P(v) \log \frac{1}{P(v)} = -\sum_v P(v) \log P(v)$$

- The entropy of a fair coin flip is 1 bit:

$$H(Fair) = -(0.5 \log 0.5 + 0.5 \log 0.5) = 1$$

- The entropy of a biased coin with 99% heads is:

$$H(Biased) = -(0.99 \log 0.99 + 0.01 \log 0.01) \approx 0.08$$

- Denote the entropy of a Bernoulli random variable that is true with probability q by

$$B(q) = -(q \log q + (1-q) \log(1-q))$$

INFORMATION GAIN

- If a training set contains p positive examples and n negative examples, the entropy of the output variable is

$$H(Output) = B\left(\frac{p}{p+n}\right)$$

- Feature A with d values divides the training set into d subsets, each with p_k positive examples and n_k negative examples
- The entropy after testing A is

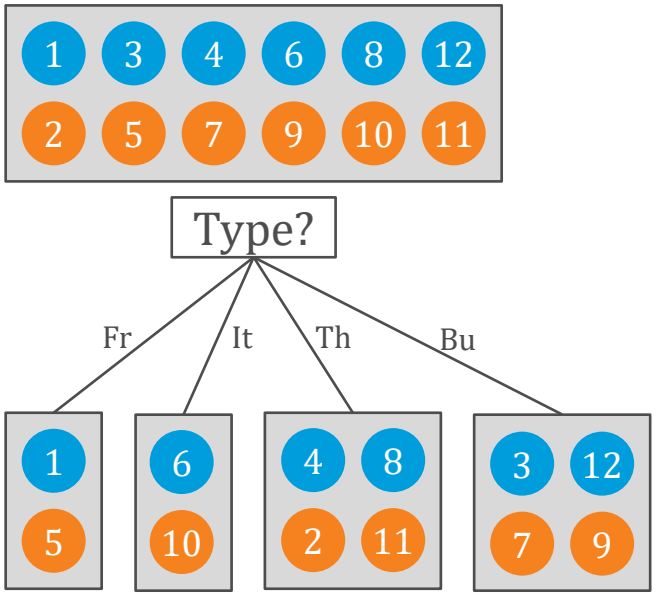
$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right)$$

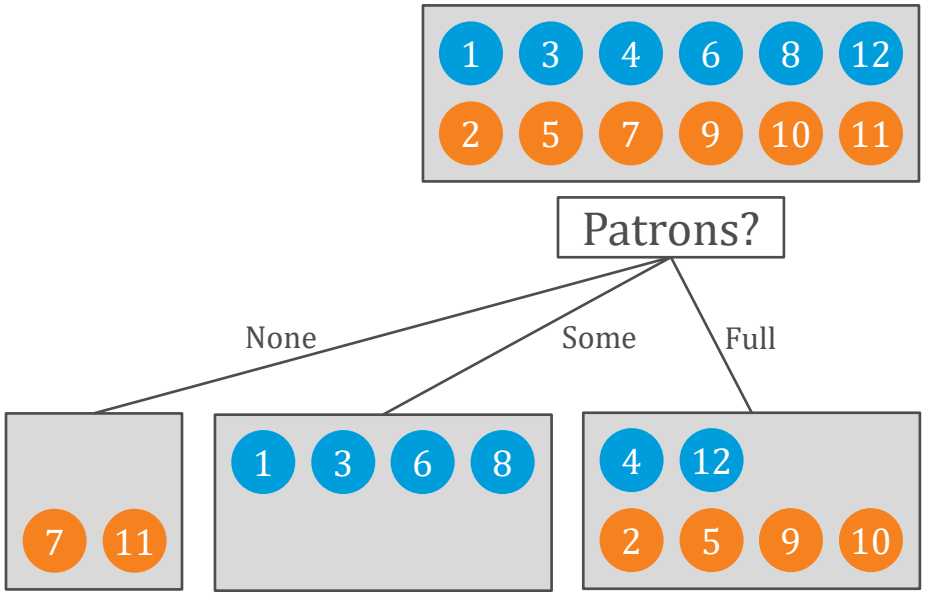
- The information gain from testing A is

$$Gain(A) = B\left(\frac{p}{p+n}\right) - Remainder(A)$$

- In LEARN-DT, we can measure IMPORTANCE based on information gain

INFORMATION GAIN: EXAMPLE

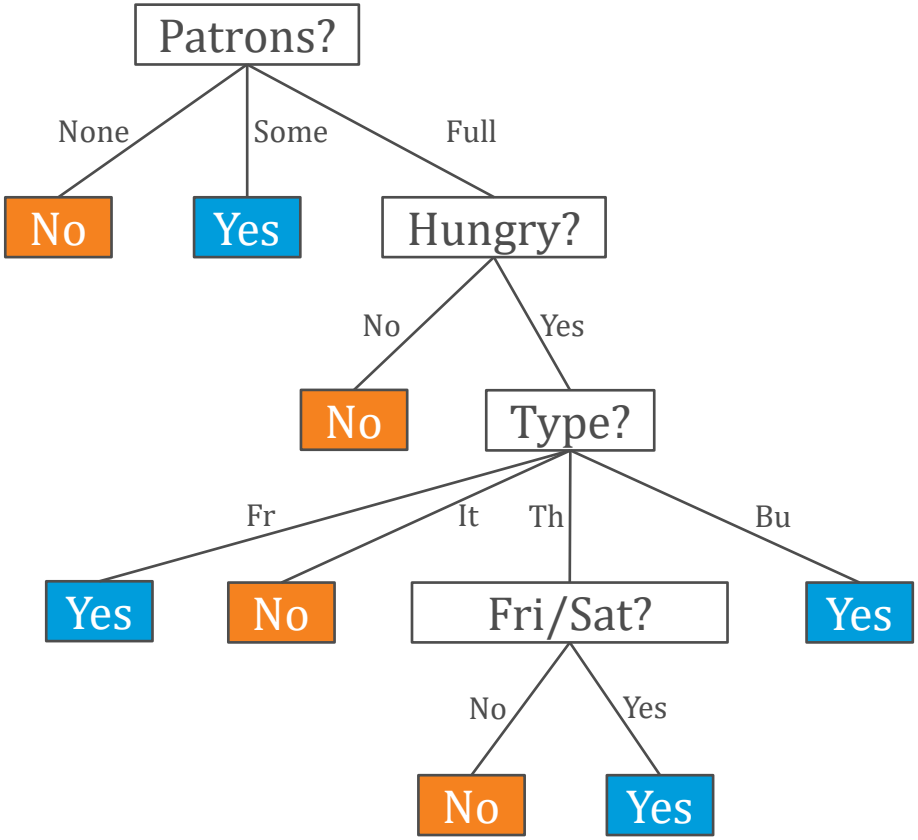




$$Gain(Type) = 1 - \left[\frac{2}{12}B\left(\frac{1}{2}\right) + \frac{2}{12}B\left(\frac{1}{2}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) + \frac{4}{12}B\left(\frac{2}{4}\right)\right] = 0$$

$$Gain(Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right)\right] \approx 0.541$$

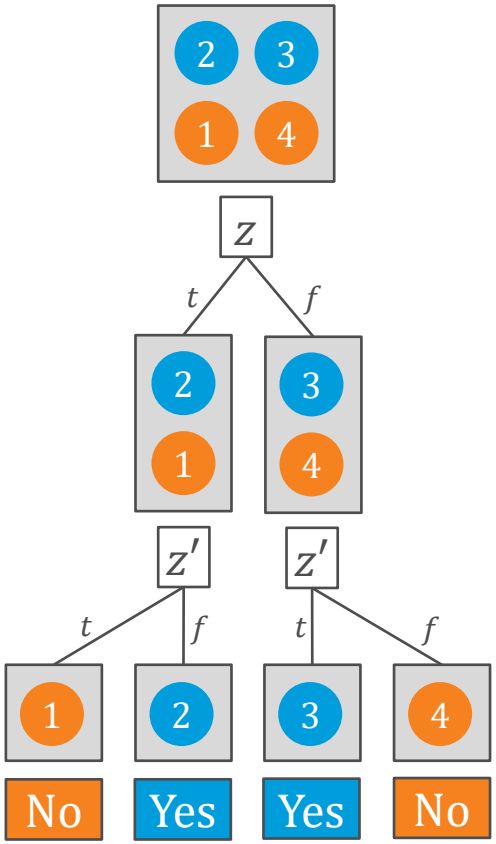
LEARNING DECISION TREES: EXAMPLE



The output of LEARN-TD is simpler than the original tree!

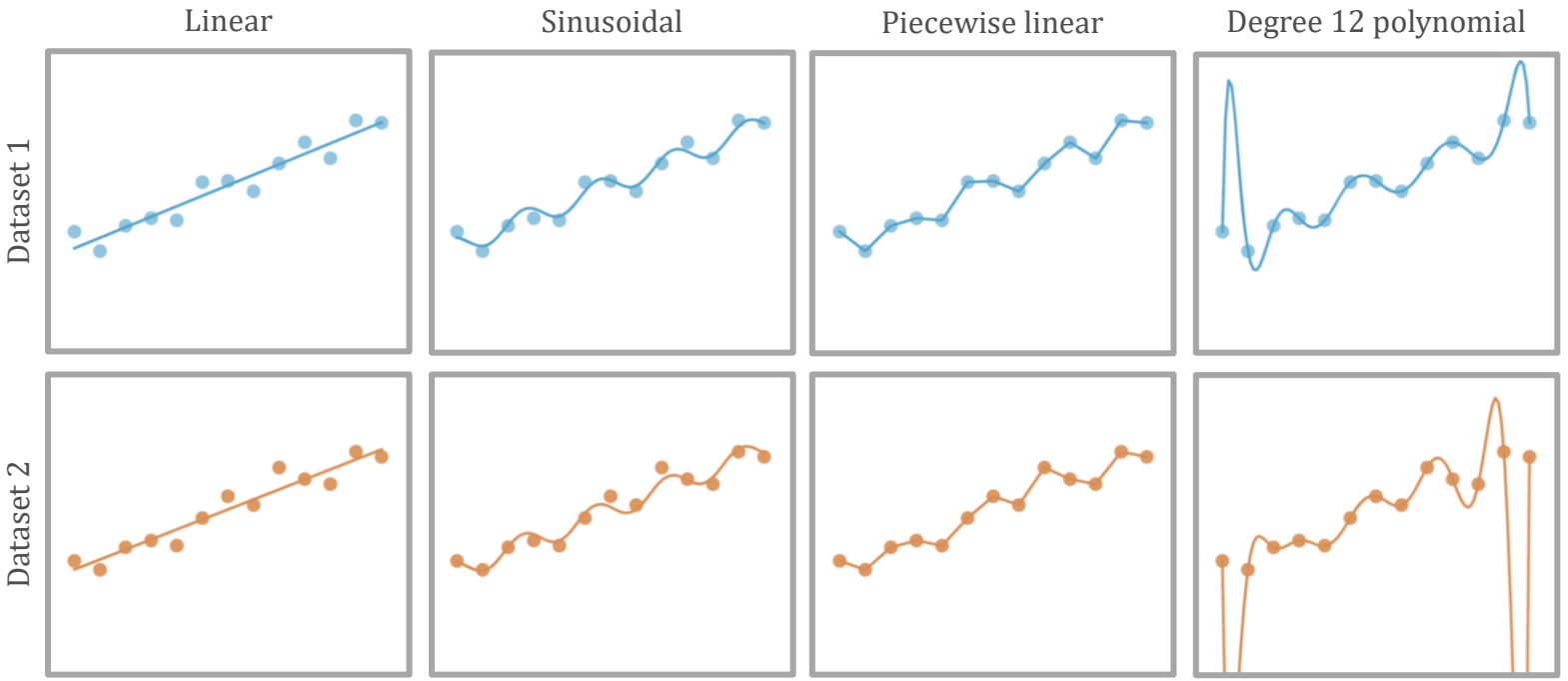
EARLY STOPPING

Example	Input features		Output
	z	z'	$z \oplus z'$
$\boldsymbol{x}^{(1)}$	t	t	f
$\boldsymbol{x}^{(2)}$	t	f	t
$\boldsymbol{x}^{(3)}$	f	t	t
$\boldsymbol{x}^{(4)}$	f	f	f



Should we stop Learn-TD when the information gain is low to avoid overfitting? We may miss situations where combos of features are informative!

MODEL SELECTION



Degree 12 polynomials exhibit overfitting

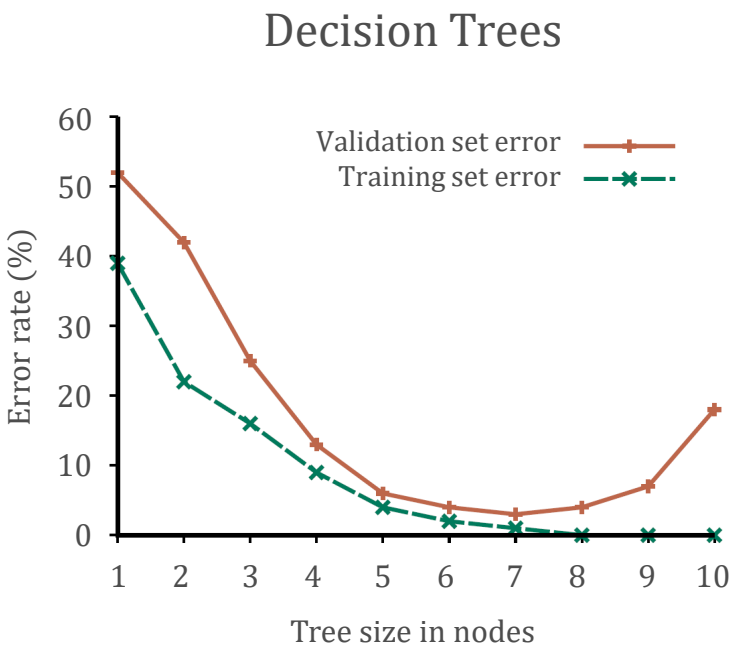
MODEL SELECTION

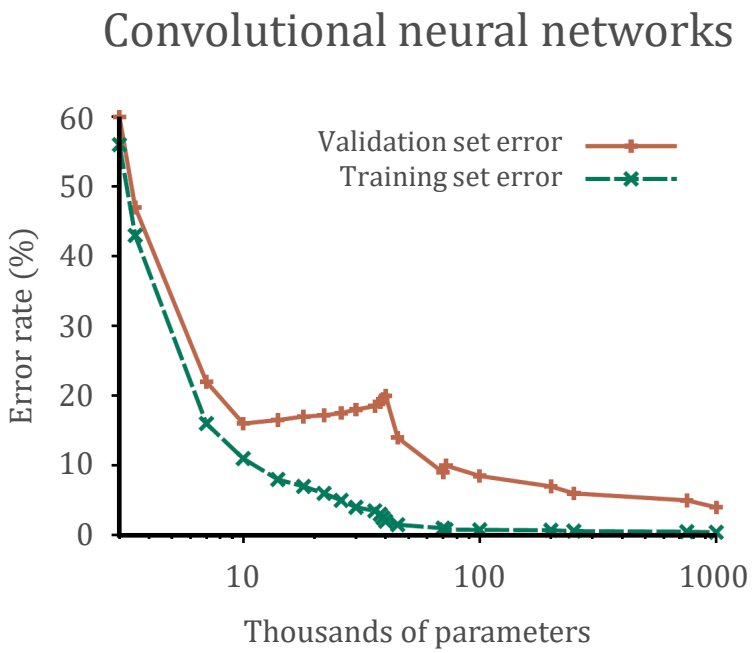
- Let us think of the quality of the "fit" of hypothesis h as error rate, i.e., the probability that $h(\boldsymbol{x}) \neq f(\boldsymbol{x})$
- We divide the data into three sets:
  1. Training set to train candidate models
  2. Validation set to choose among different models or hypothesis classes
  3. Test set to perform an unbiased evaluation of the best model
- The same examples can be used both for training and validation through k-fold cross-validation

MODEL SELECTION

- We refer to the "complexity" of the hypothesis class (e.g., number of nodes in a decision tree) as the model size
- Poll 2: As the model size grows (check all possible options):
  - Training error decreases, validation error decreases ✓
  - Training error decreases, validation error increases ✓
  - Training error increases, validation error decreases
  - Training error increases, validation error increases

MODEL SELECTION





The graph on the right is an example of one of the great mysteries of deep learning!