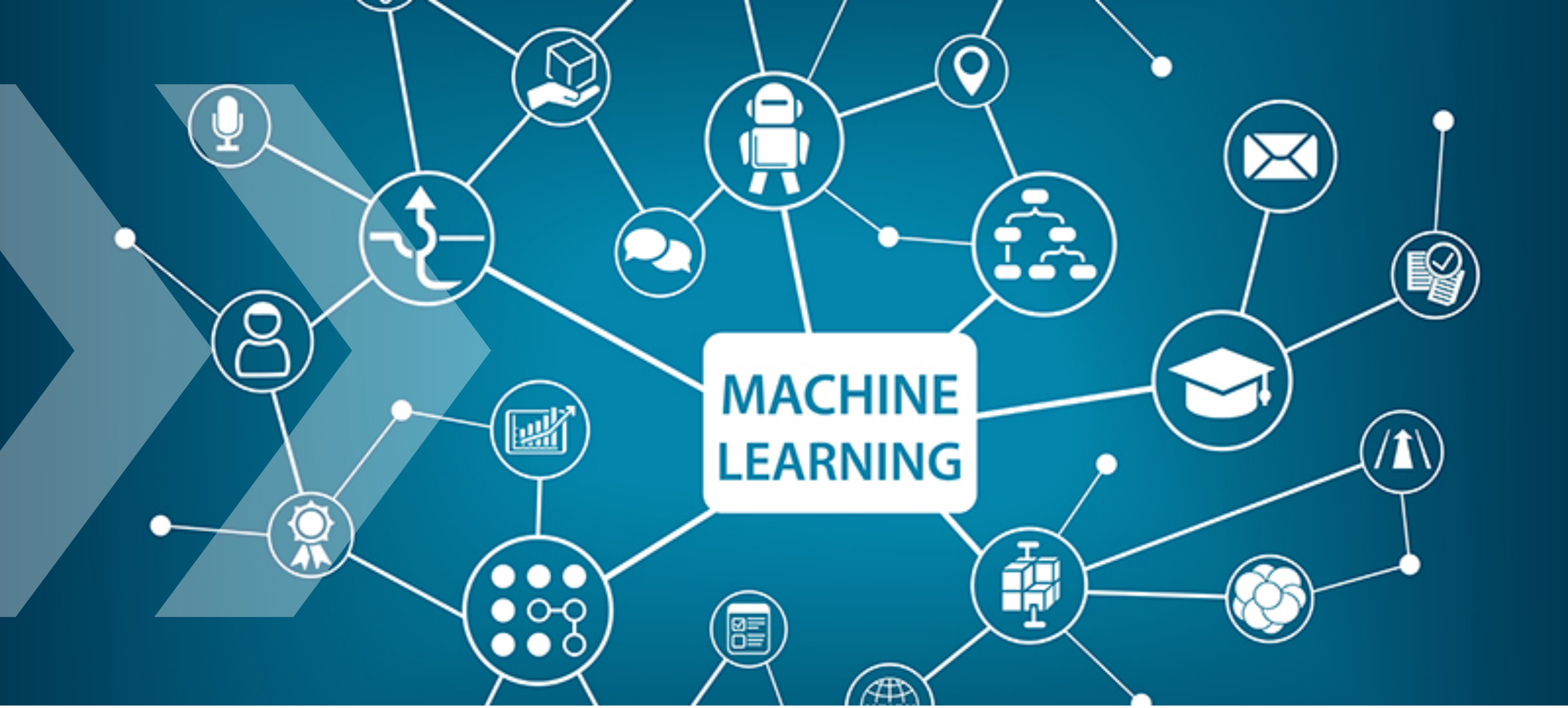
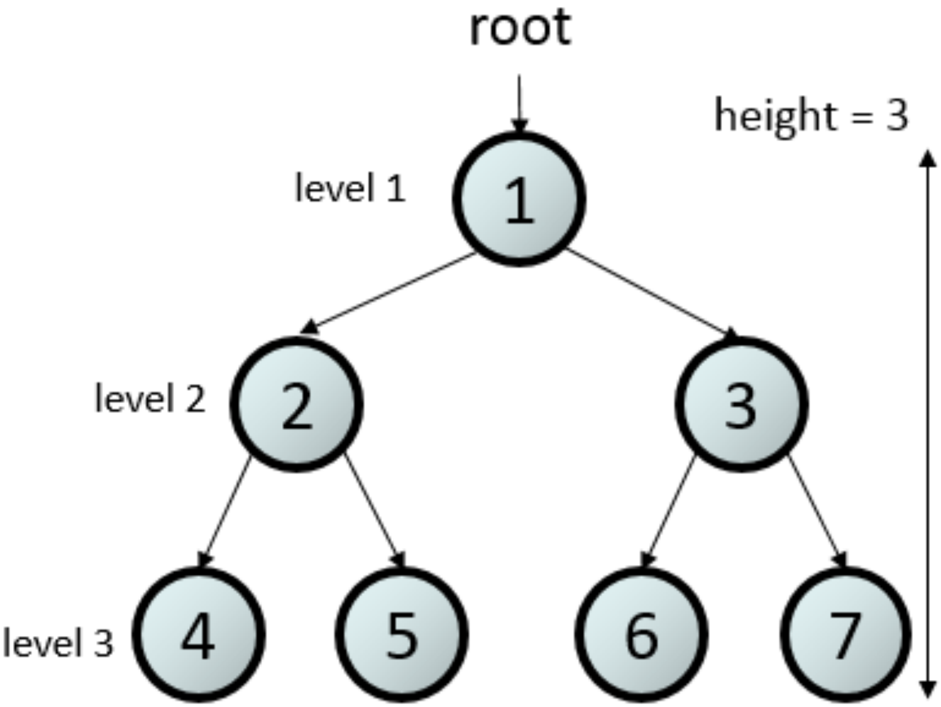


CS489/698: Intro to ML

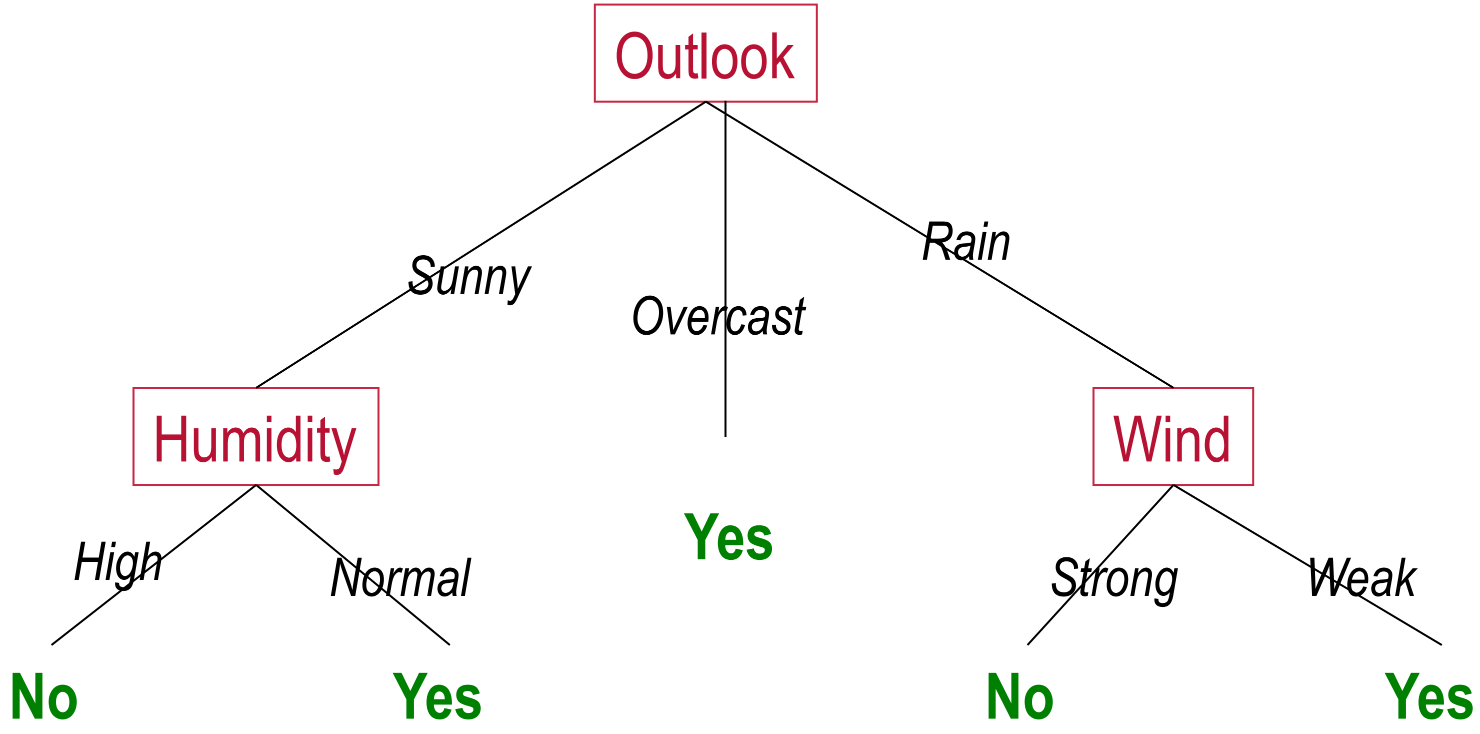
Lecture 16: Decision Trees

UNIVERSITY OF WATERLOO

Trees Recalled

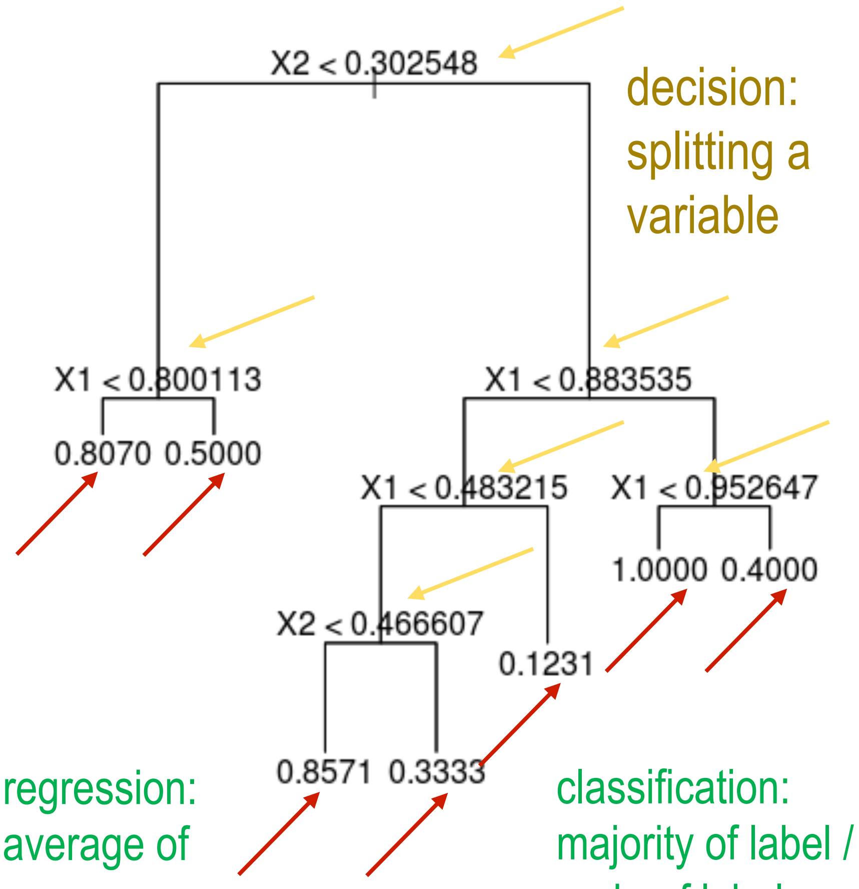


Example: EnjoySport?



Decision trees can represent any boolean function

Classification And Regression Tree

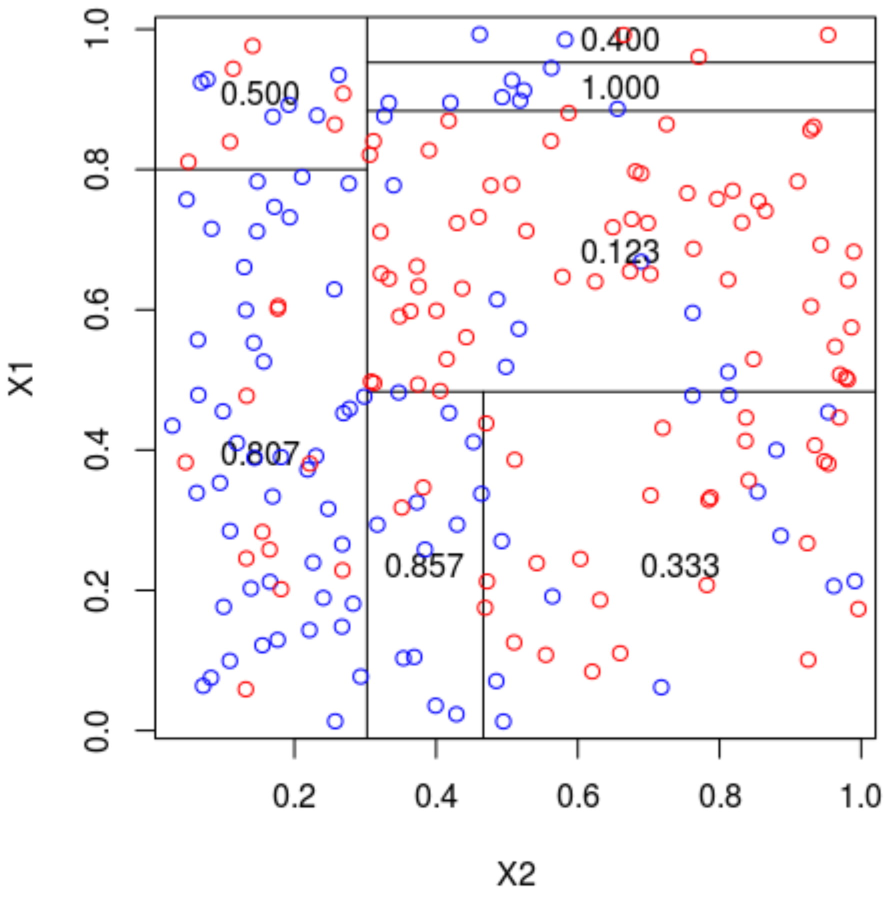


decision: splitting a variable

regression: average of responses

classification: majority of label / prob. of label

training data



LEARNing a Decision Tree

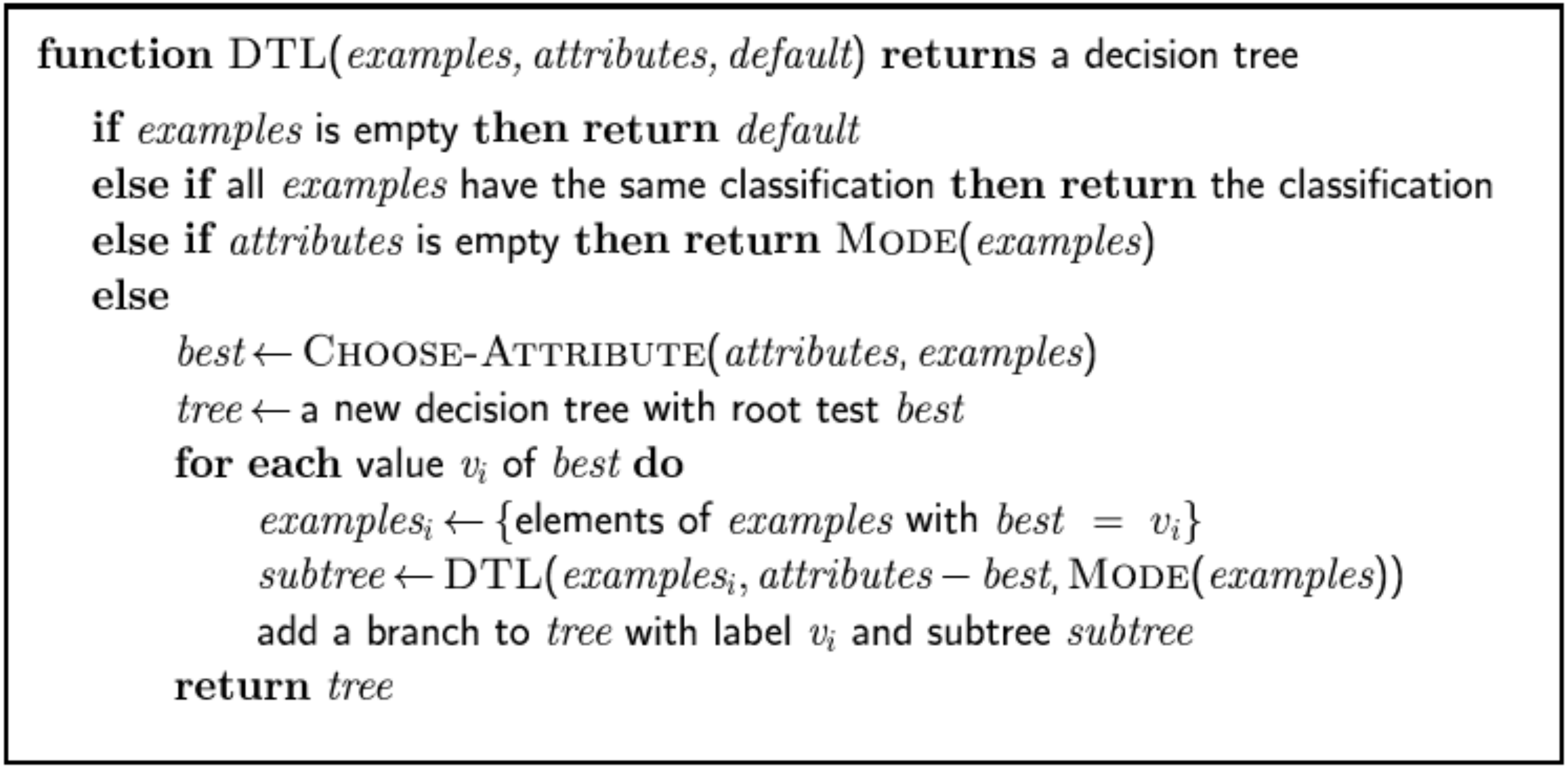
- Which variables to split in each stage?
- What threshold to use?

setup a cost/ objective

NP-Hard …

- When to stop? → regularization: early stopping / pruning
- What to put at the leaves? → regression / classification / other

Algorithm



```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value $v_i$ of best do
            $examples_i$ ← {elements of examples with best = $v_i$}
            subtree ← DTL($examples_i$, attributes − best, MODE(examples))
            add a branch to tree with label $v_i$ and subtree subtree
        return tree
```

Which and How

assuming features
are ordinal

$$(j^*, t^*) = \arg \min_{j=1,\dots,d} \min_{t \in T_j} \ell(\{(\mathbf{x}_i, y_i) : x_{ij} \le t\}) + \ell(\{(\mathbf{x}_i, y_i) : x_{ij} > t\})$$

greedily choose a
variable to split

greedily choose a
threshold to split

partition training data
into left and right

cost / loss

- For categorical features, simply try each
- What should T_j be?

Stopping criterion

- Maximum depth exceeded
- Maximum running time exceeded
- All children nodes are sufficiently homogeneous
- All children nodes have too few training examples
- Cross-validation
- Reduction in cost is small

$$\Delta = \ell(\mathcal{D}) - \left(\frac{|\mathcal{D}_L|}{|\mathcal{D}|} \ell(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} \ell(\mathcal{D}_R) \right)$$

Regression cost

$$\ell(\mathcal{D}) = \min_{y} \sum_{i=1} (y_i - y)^2 = \sum_{i \in \mathcal{D}} (y_i - \bar{y})^2$$

average of y_i in D stored on leaves

- Can of course use other loss than least-squares
- Can also fit any regression model on D

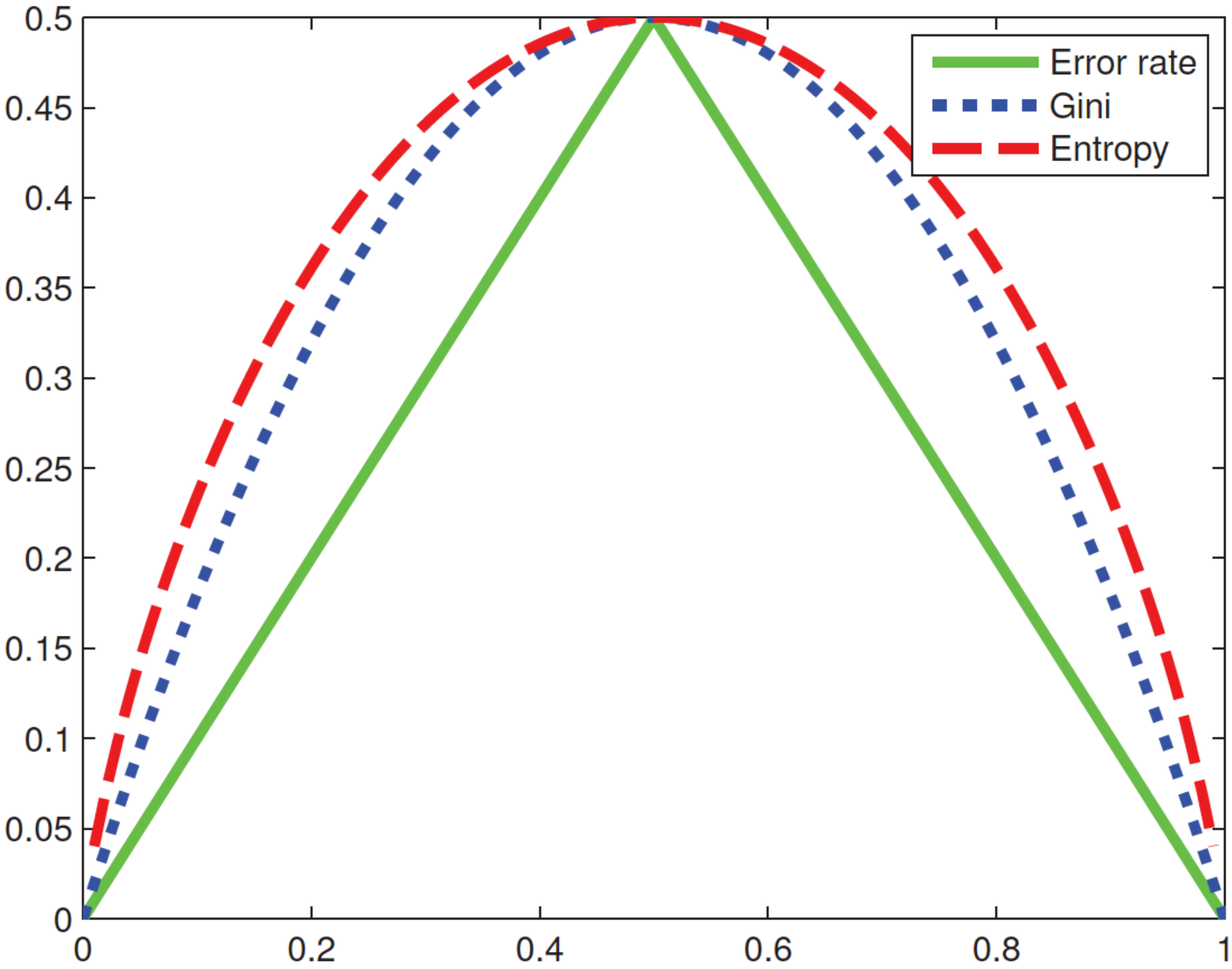
Classification cost

$$\hat{p}_c = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} 1(y_i = c) \qquad \hat{y} = \arg\max_c \hat{p}_c$$

majority vote

- Misclassification error: $\ell(\mathcal{D}) = 1 - \hat{p}_{\hat{y}}$
- Entropy: $\ell(\mathcal{D}) = -\sum_{c=1}^{C} \hat{p}_c \log \hat{p}_c$
- Gini index: $\ell(\mathcal{D}) = \sum_{c=1}^{C} \hat{p}_c(1-\hat{p}_c) = 1 - \sum_{c=1}^{c} \hat{p}_c^2$

Comparison

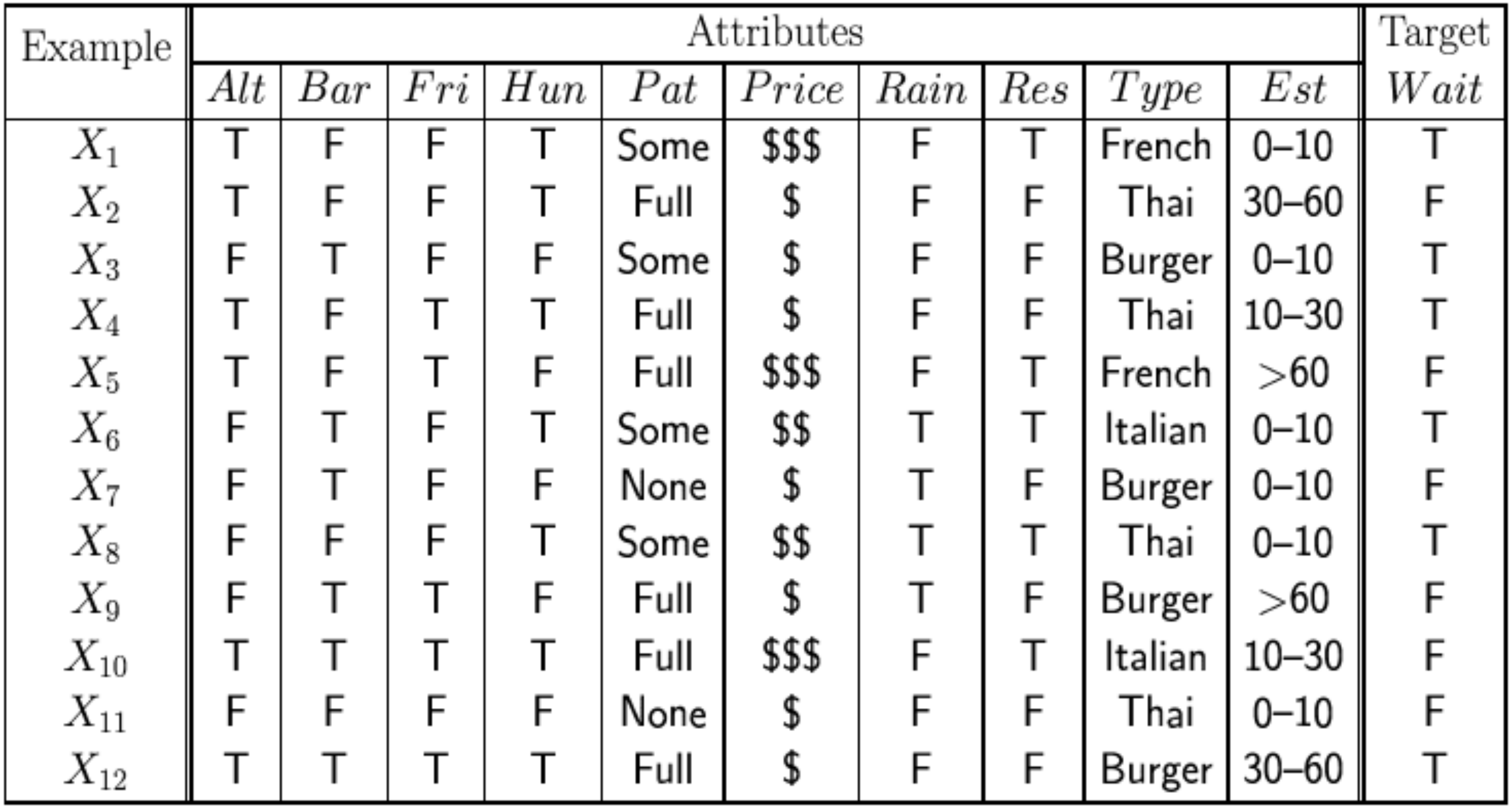


$$\min\{\hat{p}, 1-\hat{p}\}$$

$$2\hat{p}(1-\hat{p})$$

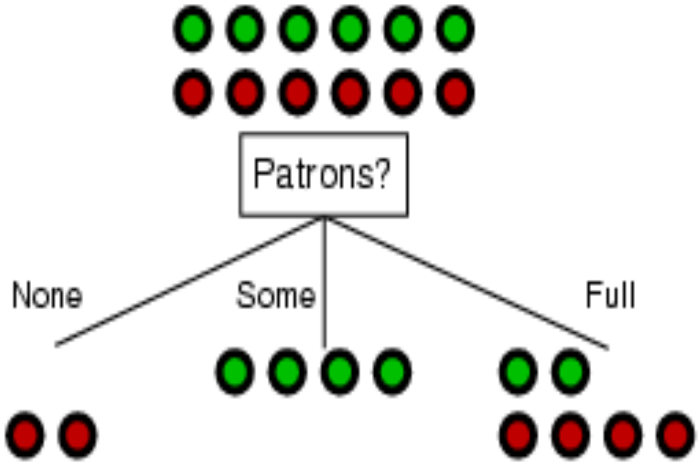
$$\hat{p}\log\hat{p}+$$
$$(1-\hat{p})\log(1-\hat{p})$$

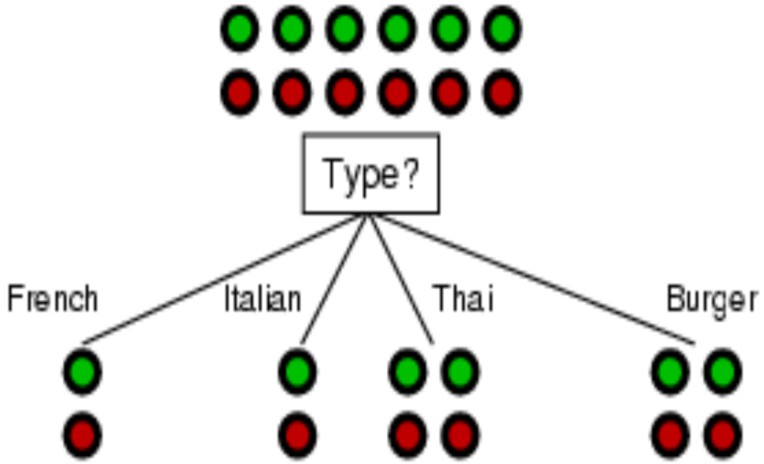
Example



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

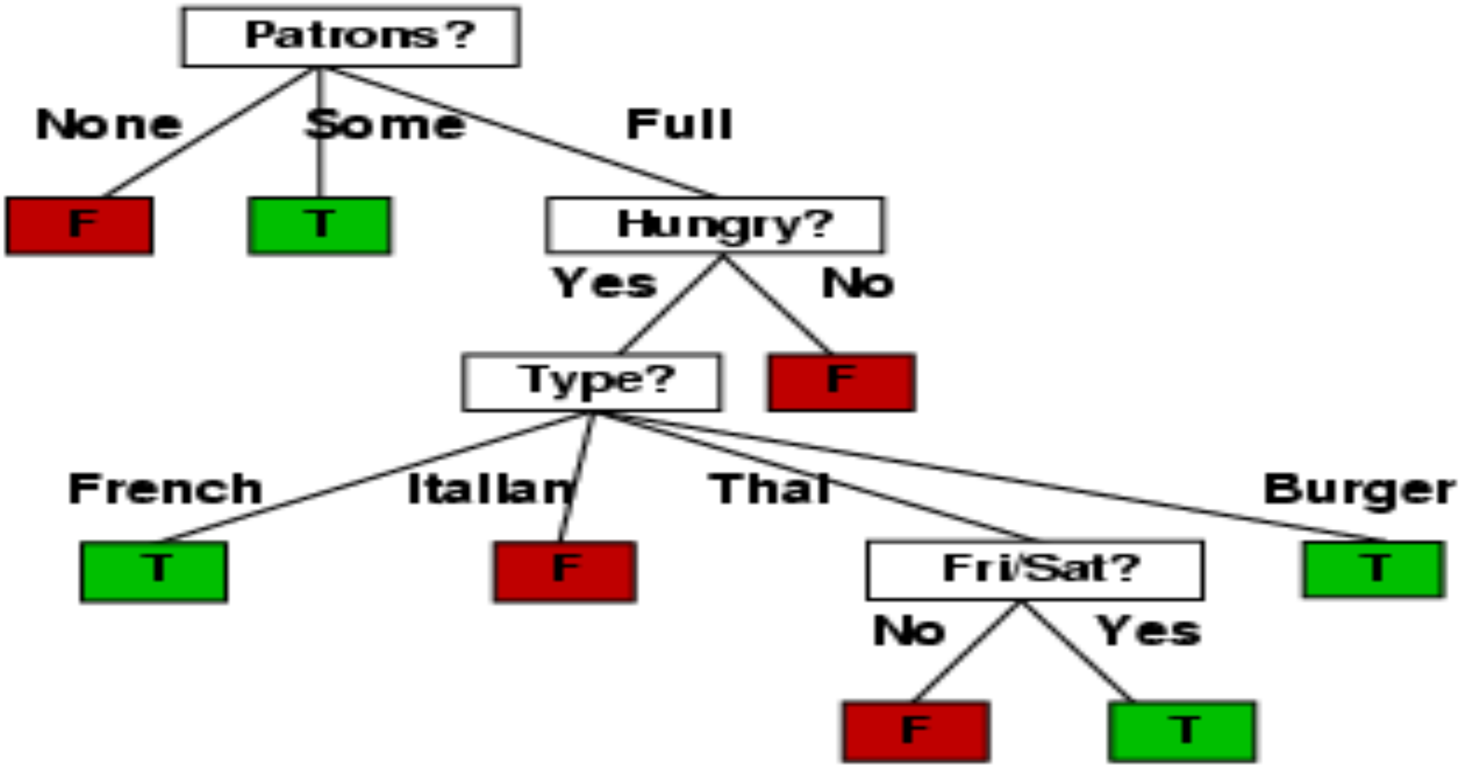
Type or Patrons





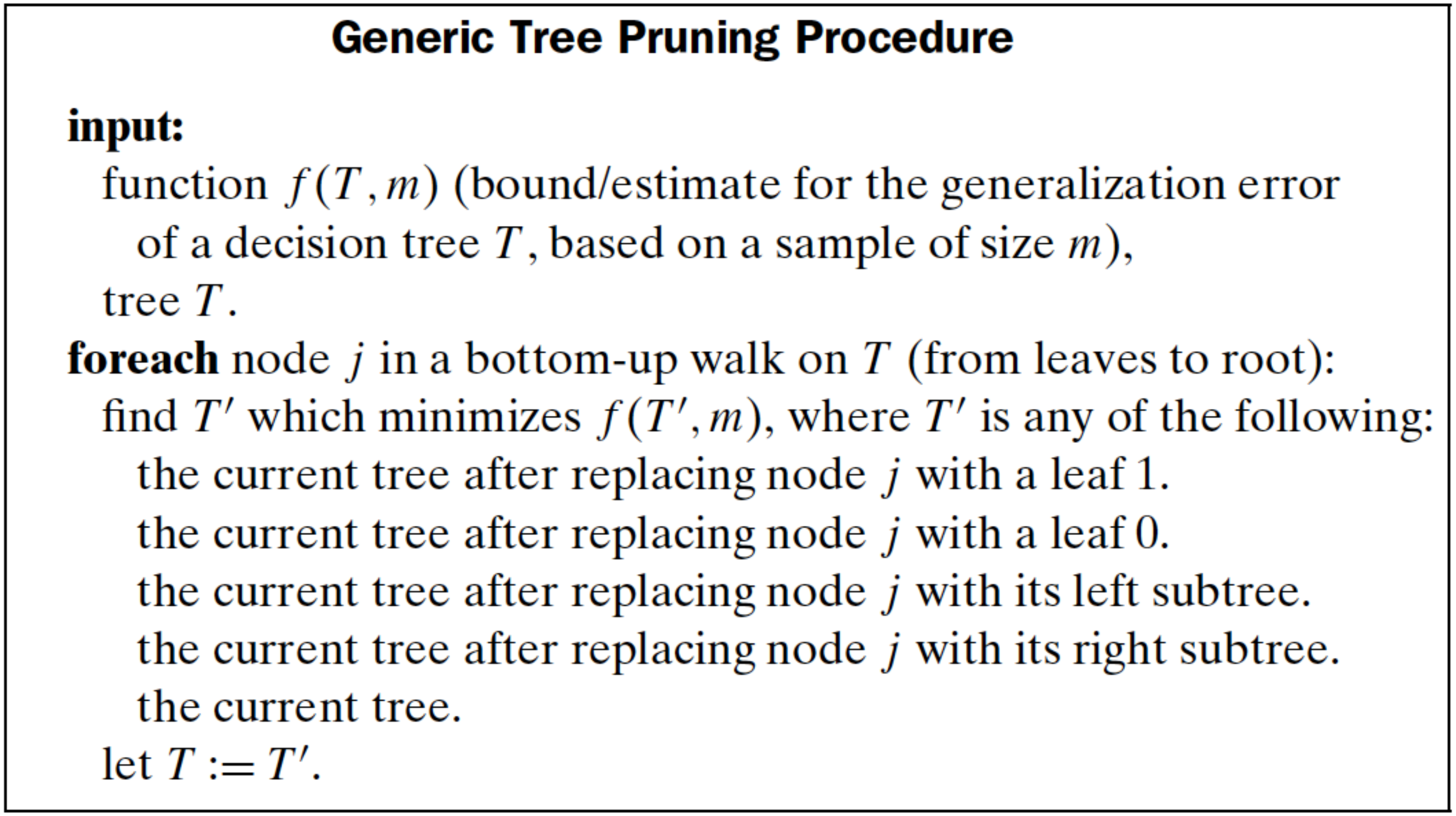
- A better feature split should lead to nearly all positives or all negatives

Result


Patrons?
None
Some
Full
F
T
Hungry?
Yes
No
Type?
F
French
Italian
Thai
Burger
T
F
Fri/Sat?
T
No
Yes
F
T

Pruning

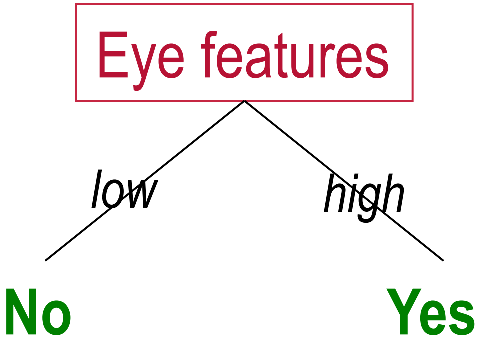
- Early stopping can be myopic
- Grow a full tree and then prune in bottom-up



Generic Tree Pruning Procedure

```
input:
  function $f(T, m)$ (bound/estimate for the generalization error
    of a decision tree $T$, based on a sample of size $m$),
  tree $T$.
foreach node $j$ in a bottom-up walk on $T$ (from leaves to root):
  find $T'$ which minimizes $f(T', m)$, where $T'$ is any of the following:
    the current tree after replacing node $j$ with a leaf 1.
    the current tree after replacing node $j$ with a leaf 0.
    the current tree after replacing node $j$ with its left subtree.
    the current tree after replacing node $j$ with its right subtree.
    the current tree.
  let $T := T'$.
```

Decision Stump



- A binary tree with depth 1
- Performs classification based on one feature
- Easy to train but underfits; interprettable

Questions?