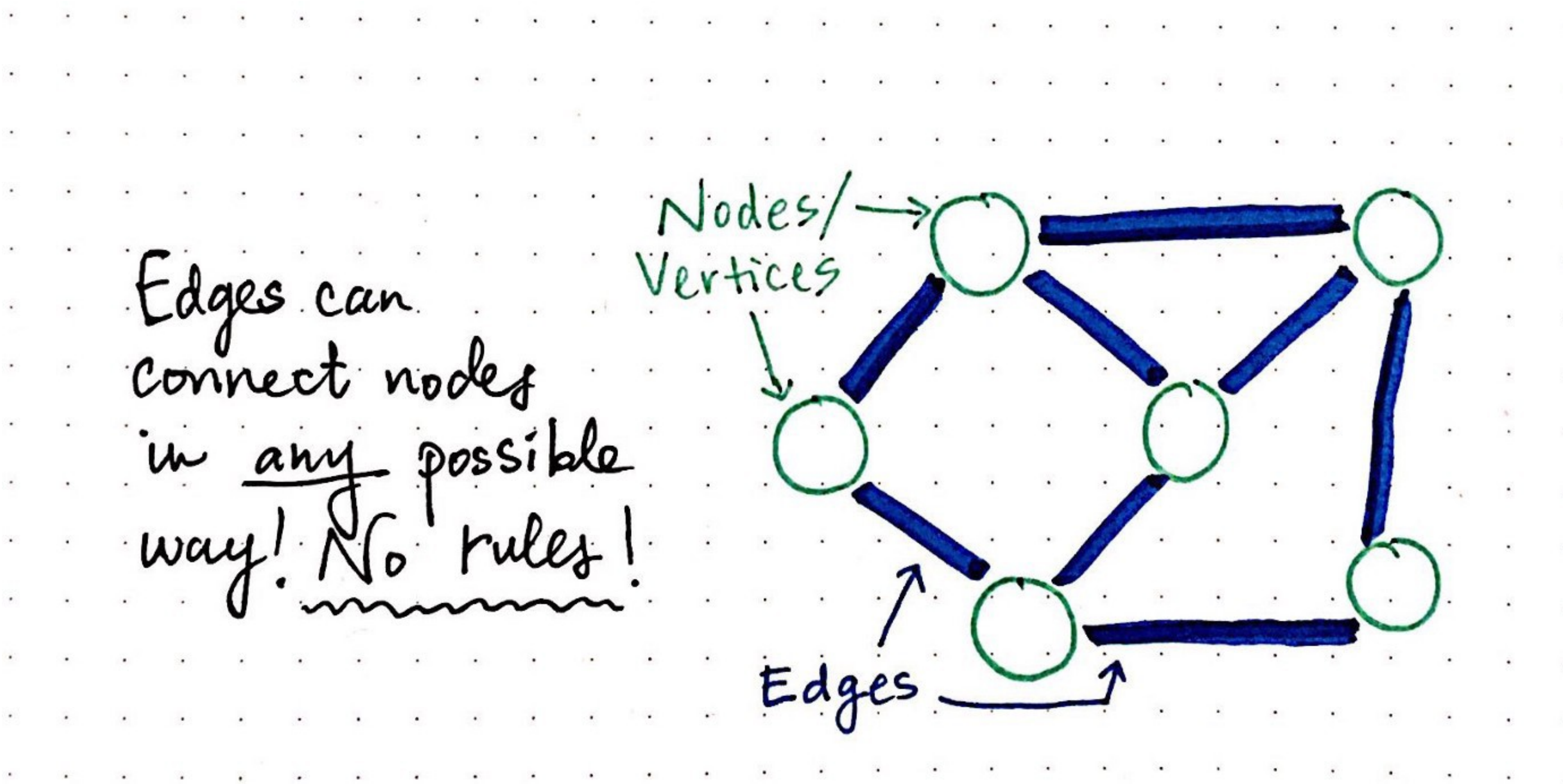
349:Machine Learning

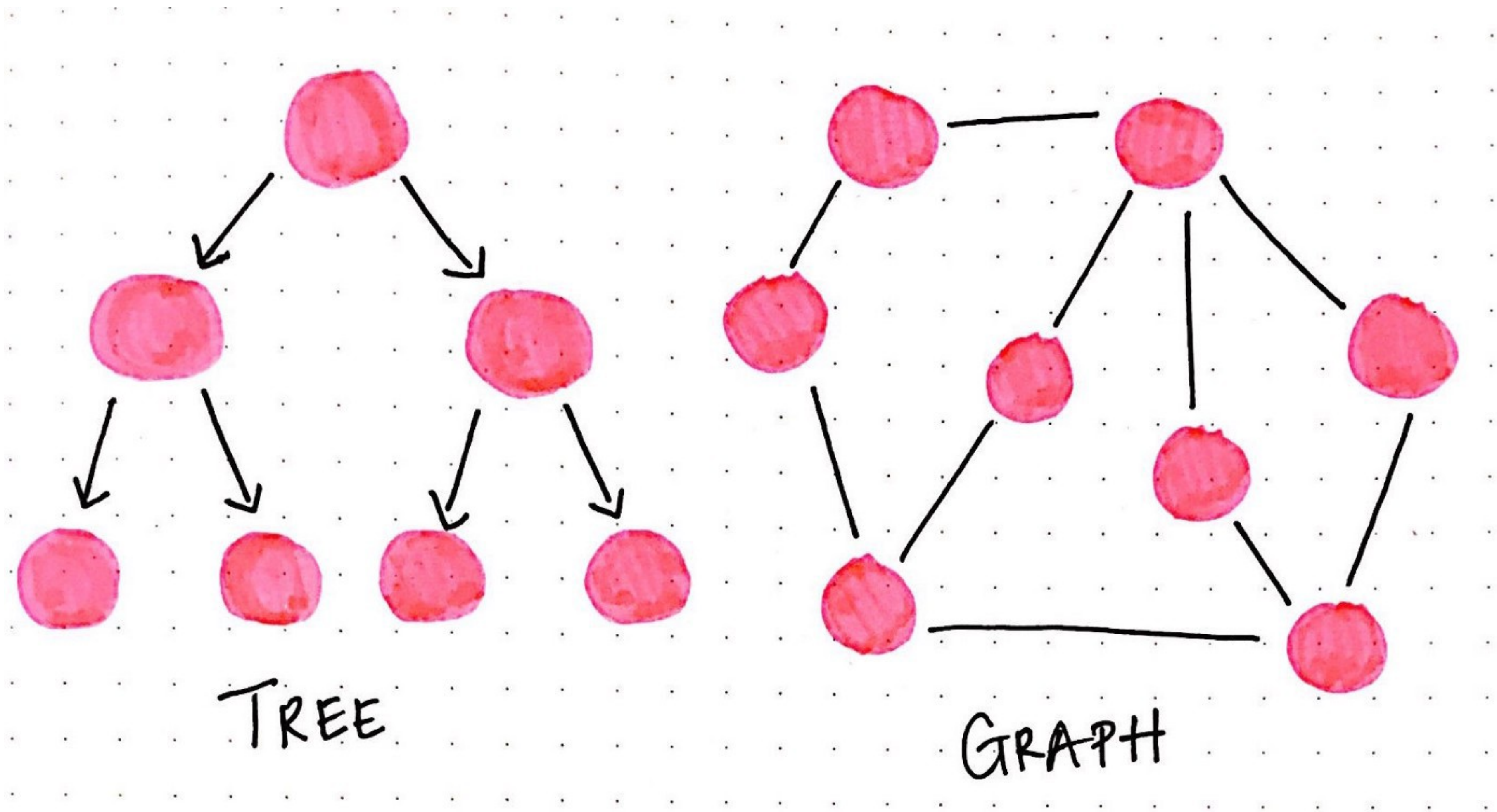
Fall 2024

Decision Trees
Part 1

Primer on Graphs



Primer on Graphs



General Learning Task

There is a set of possible examples $X = \{\vec{x}_1, \ldots \vec{x}_n\}$

Each example is an n-tuple of attribute values

$$\vec{x}_1 = < a_1, \ldots, a_k >$$

There is a target function that maps X onto some finite set Y

$$f : X \rightarrow Y$$

The DATA is a set of duples <example, target function values>

$$D = \{< \vec{x}_1, f(\vec{x}_1) >, \ldots < \vec{x}_m, f(\vec{x}_m) >\}$$

Find a hypothesis ***h*** such that...

$$\forall \vec{x}, h(\vec{x}) \approx f(\vec{x})$$

Attribute-based representations

Examples described by attribute values (Boolean, discrete, continuous, etc.)
E.g., situations where I will/won't wait for a table:

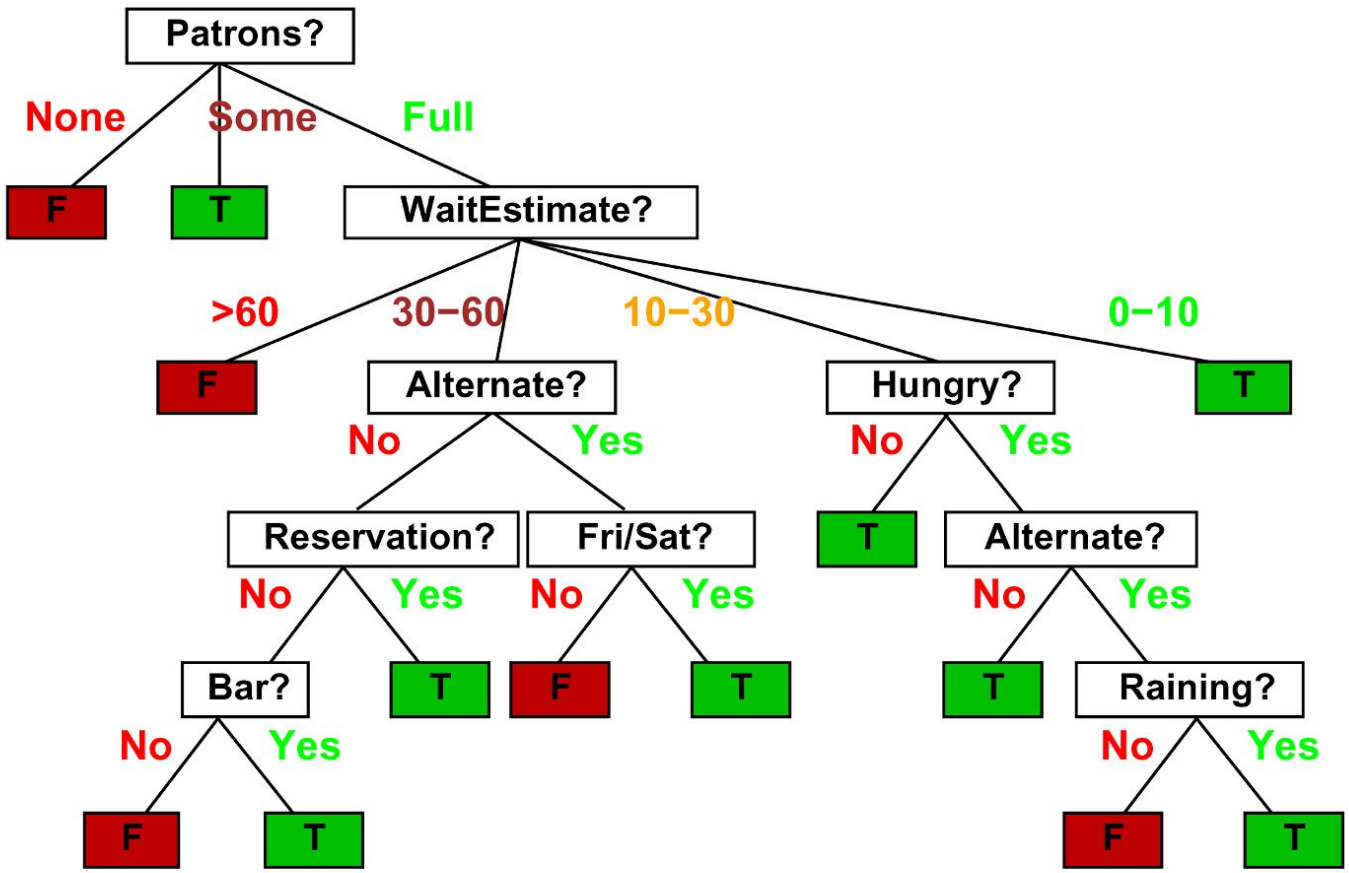
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*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*

Classification of examples is positive (T) or negative (F)

Decision Tree

One possible representation for hypotheses

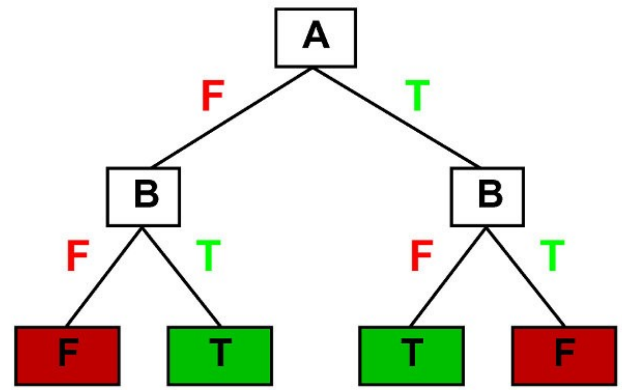
E.g., here is the "true" tree for deciding whether to wait:



Expressiveness of D-Trees

Decision trees can express any function of the input attributes.
E.g., for Boolean functions, truth table row $\rightarrow$ path to leaf:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Trivially, there is a consistent decision tree for any training set w/ one path to leaf for each example (unless f nondeterministic in x) but it probably won't generalize to new examples

Prefer to find more **compact** decision trees

Another Example

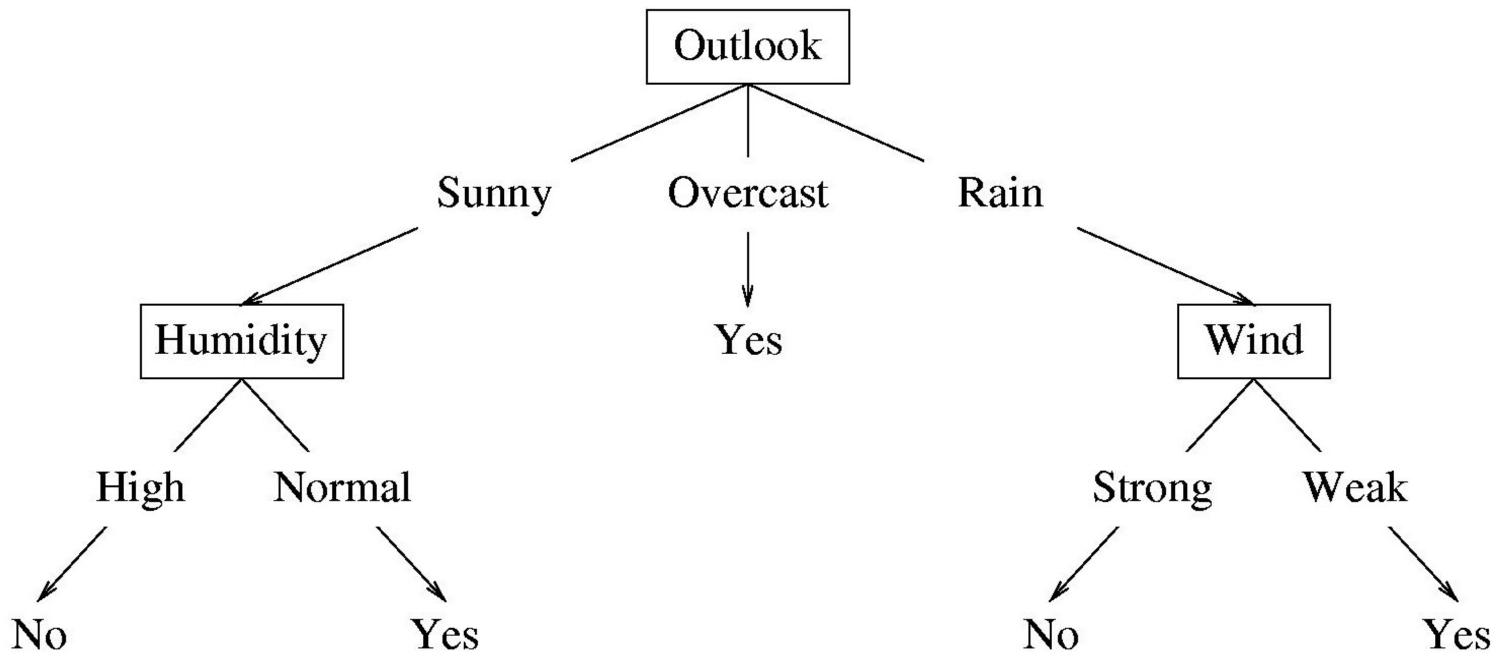
- Columns denote features X_i
- Rows denote labeled instances
- Class label denotes whether a tennis game was played

$\langle \boldsymbol{x}_i, y_i \rangle$

Predictors				Response
Outlook	**Temperature**	**Humidity**	**Wind**	**Class**
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

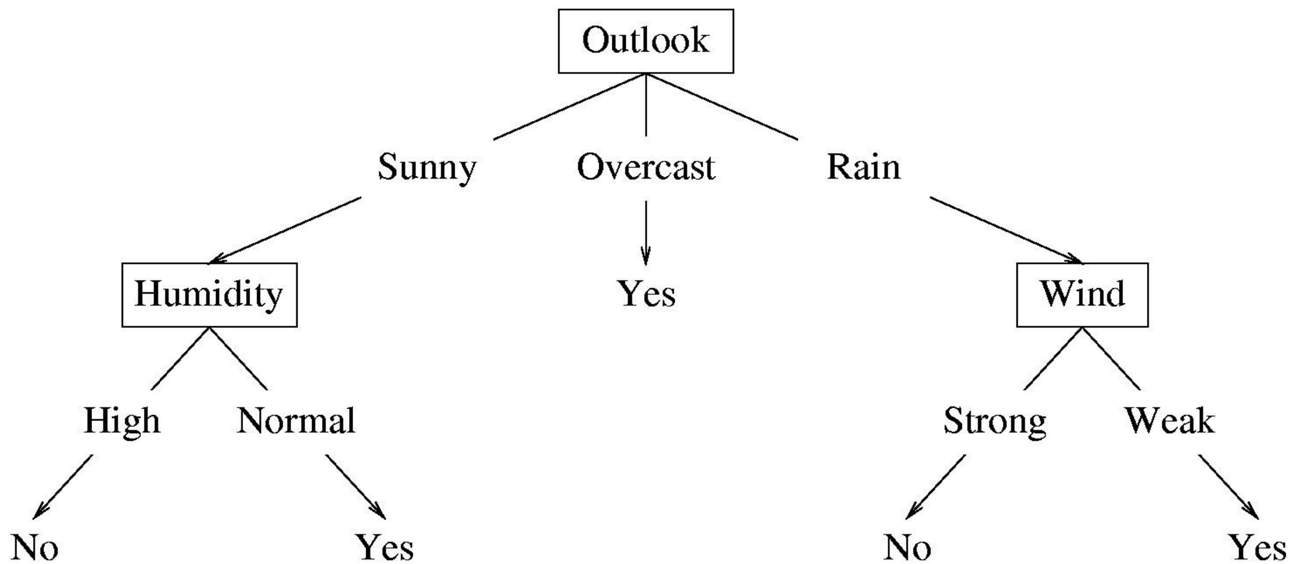
Decision Trees

A possible decision tree for the data:



- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid \mathbf{x} \in \text{leaf})$)

Decision Trees as a Logical Representation

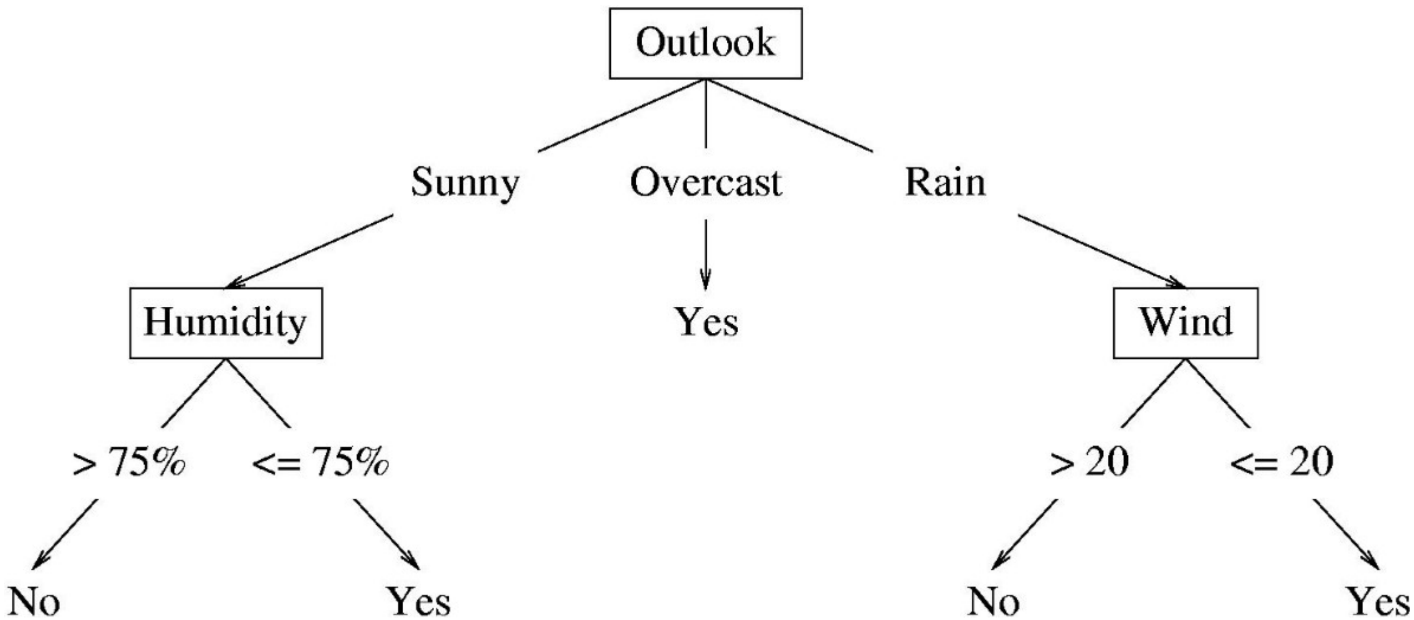


$f(x) = yes$ iff...

```
(Outlook == Sunny ∧ Humidity == Normal) ∨
(Outlook == Overcast) ∨
(Outlook == Rain ∧ Wind == Weak)
```

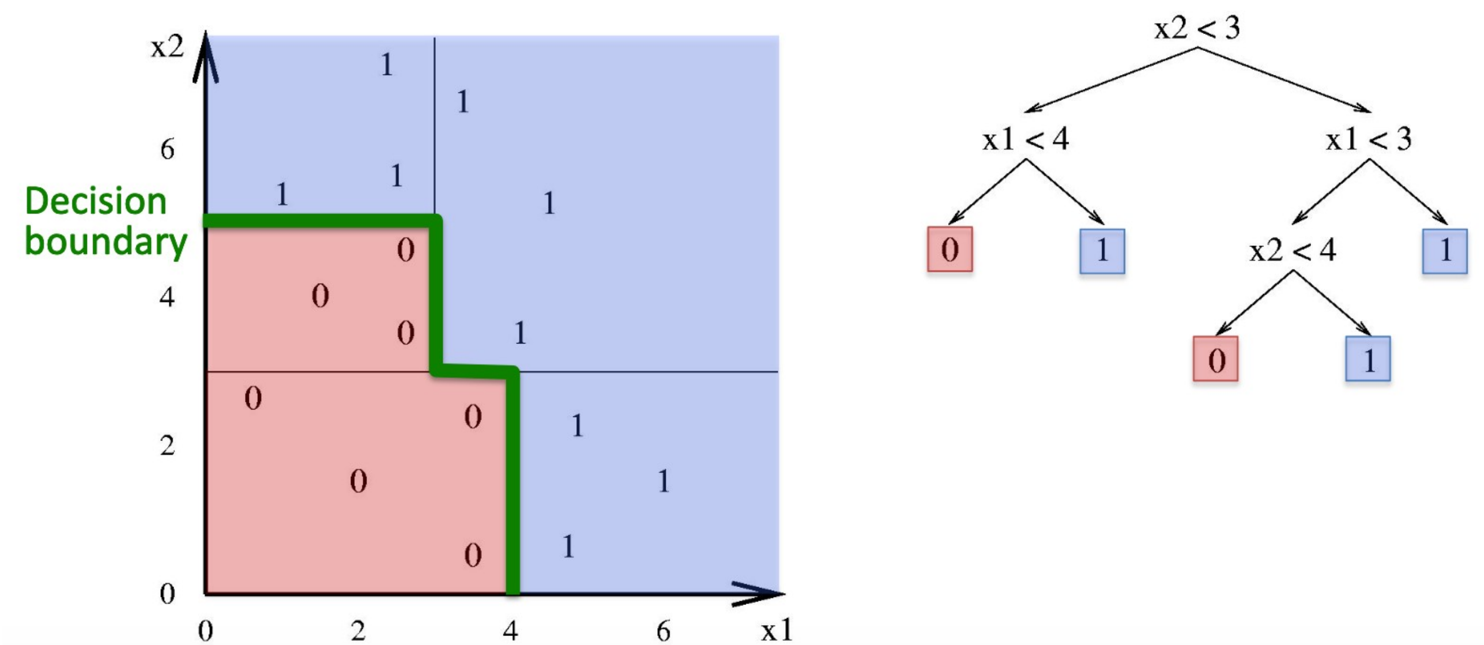
Decision Tree

If features are continuous, internal nodes can test the value of a feature against a threshold



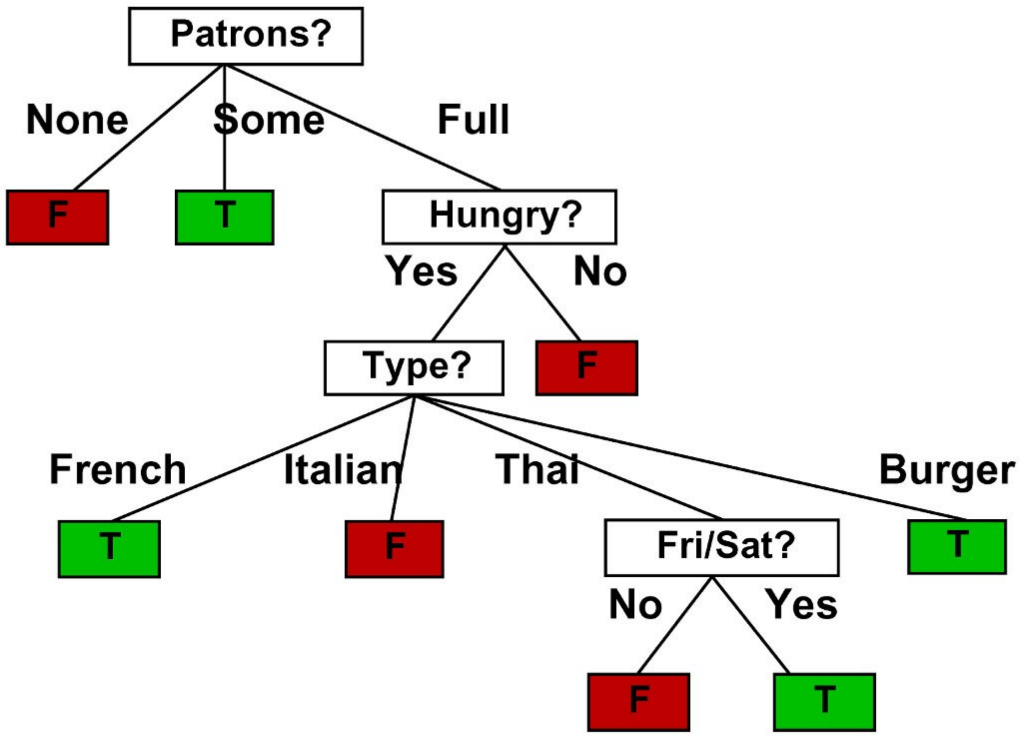
Decision Tree Boundaries

- Decision trees divide the feature space into axis-parallel (hyper-)rectangles
- Each rectangular region is labeled with one label
  - or a probability distribution over labels



A learned Decision Tree

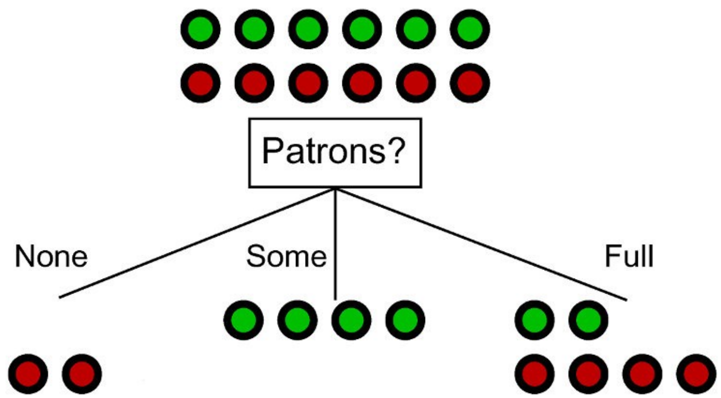
Decision tree learned from the 12 examples:

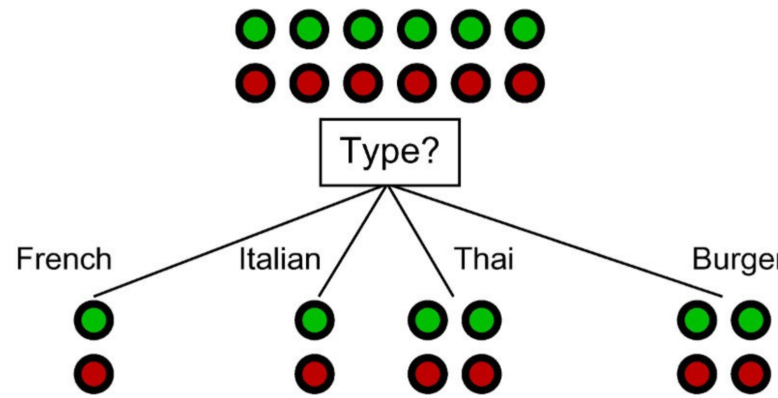


Substantially simpler than "true" tree—a more complex hypothesis isn't justified by small amount of data

How to choose
an attribute ?

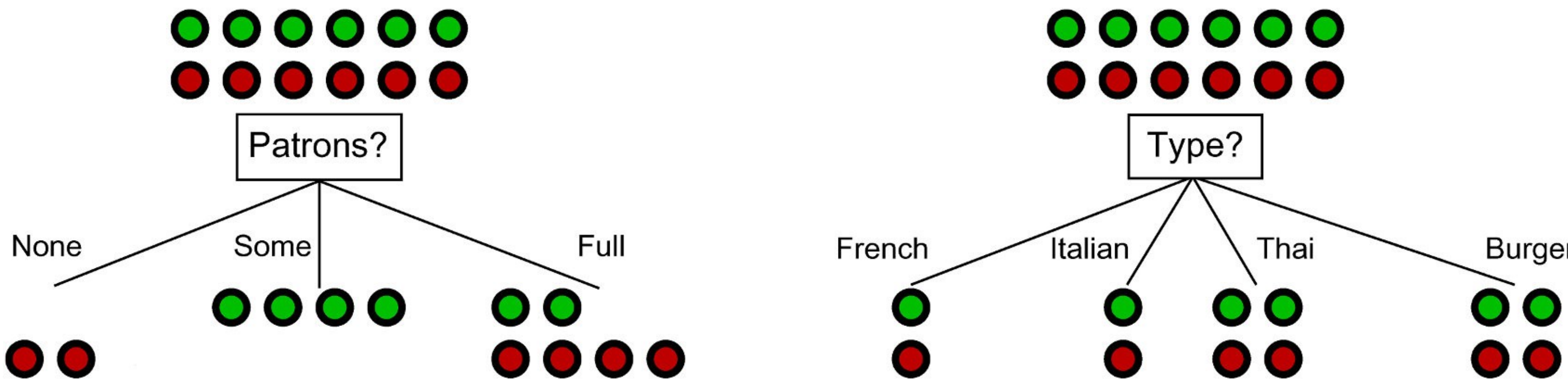
Choosing an Attribute





Choosing an Attribute

Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"



Patrons? is a better choice—gives **information** about the classification

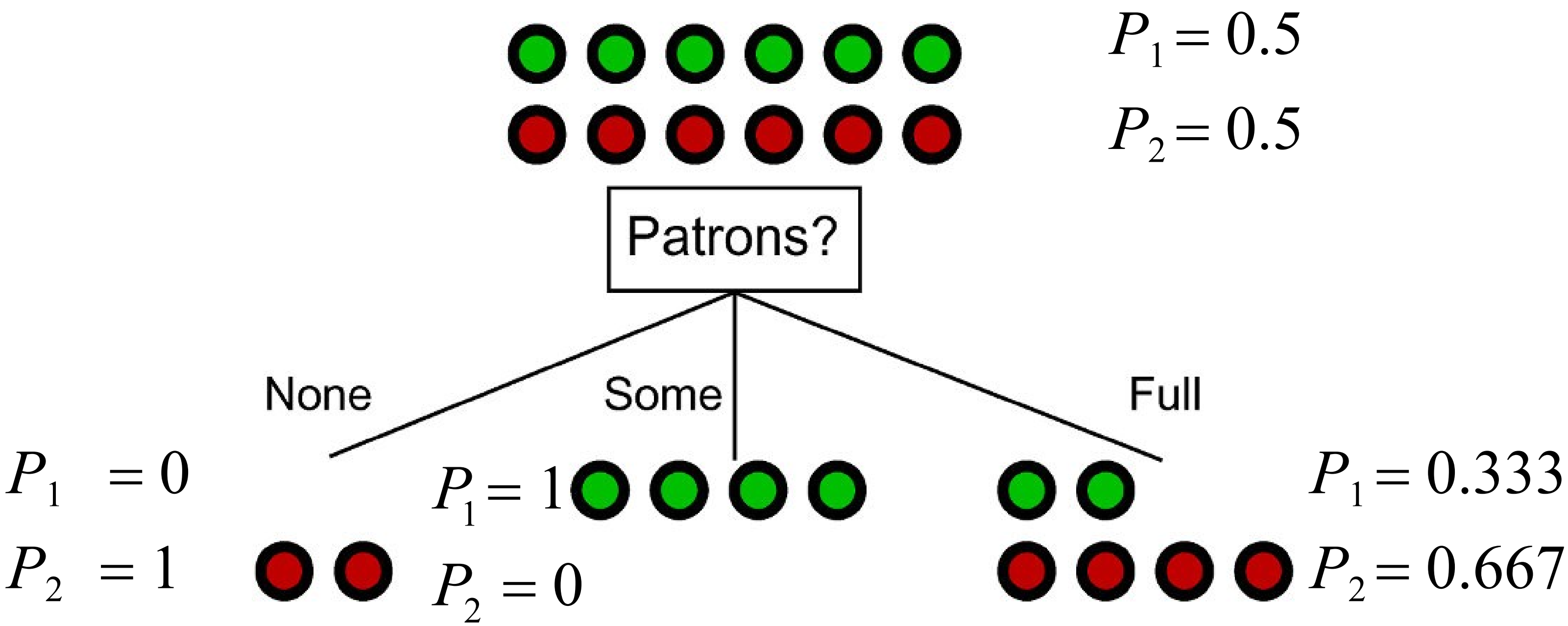
The more skewed the examples in a bin, the better.

We're going to use ENTROPY to as a measure of how skewed each bin is.

Counts as Probabilities

$$P_1 = \text{probability I will wait for a table}$$

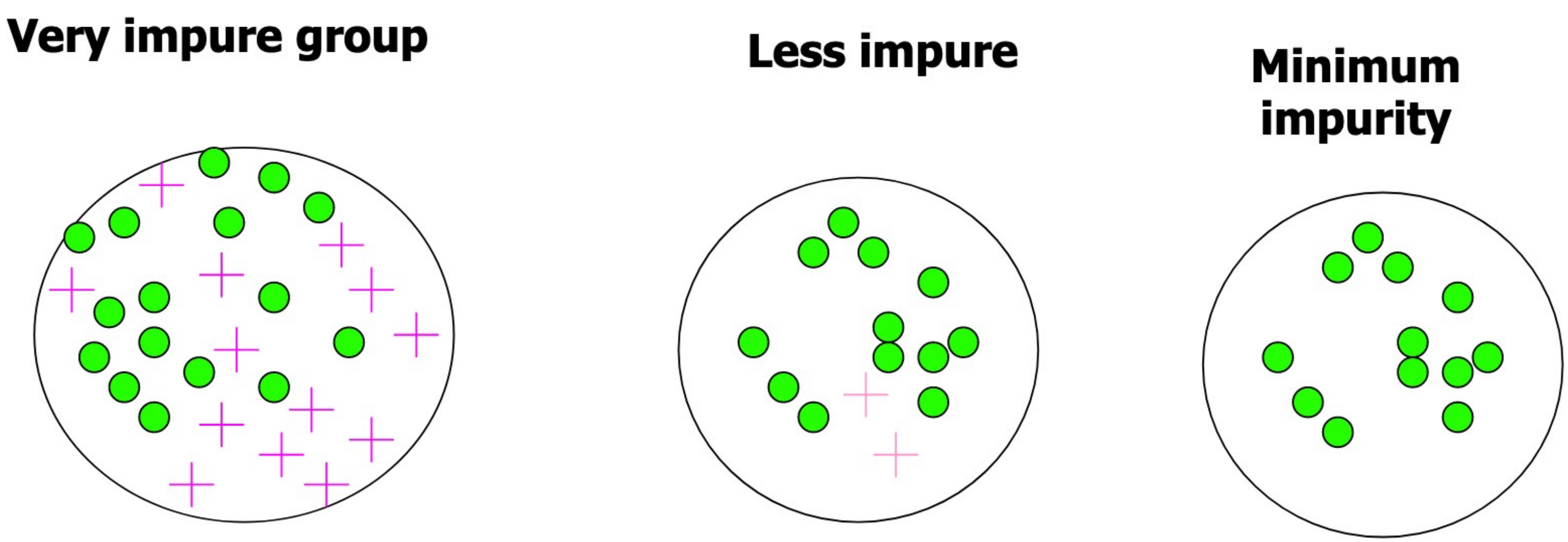
$$P_2 = \text{probability I will NOT wait for a table}$$



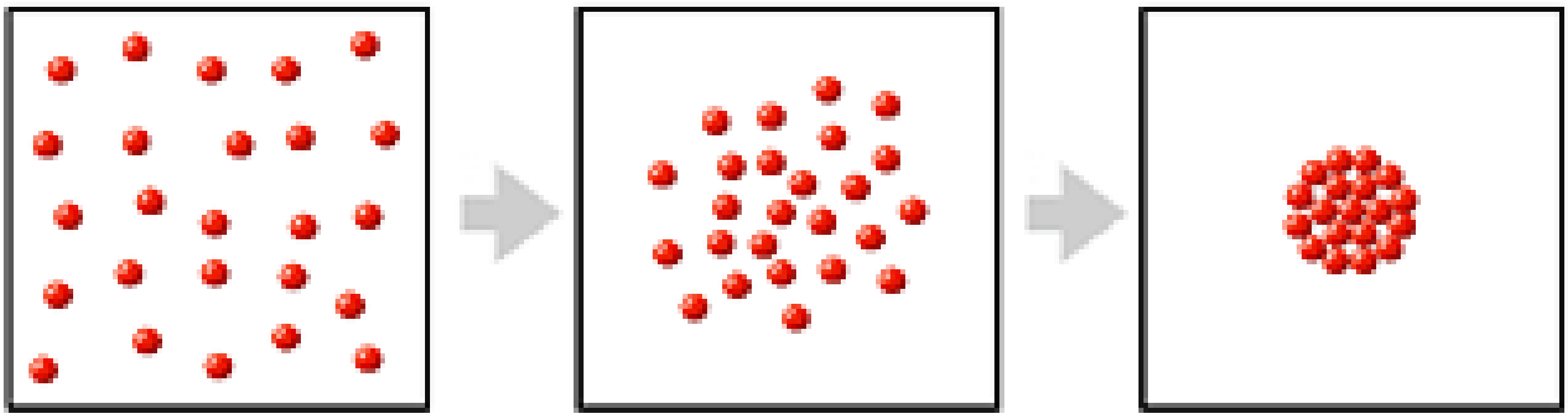
Entropy

Impurity/Entropy (informal):

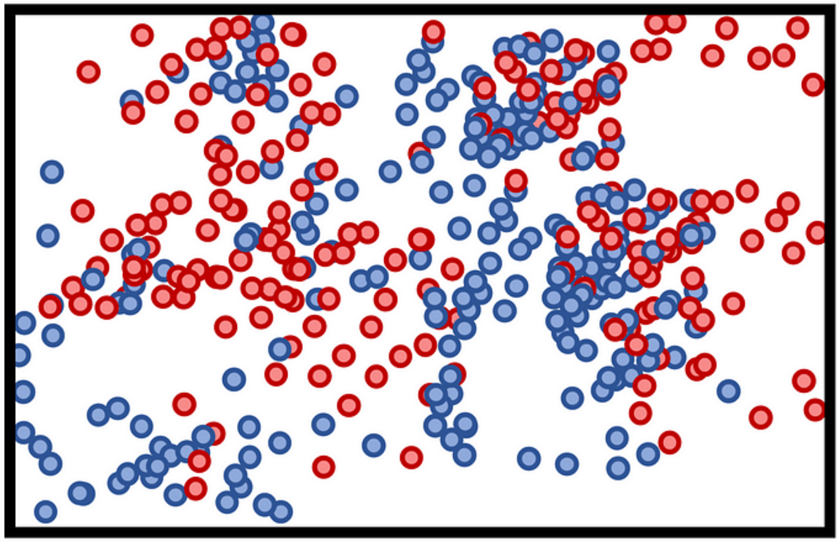
- Measures the level of impurity in a group of examples



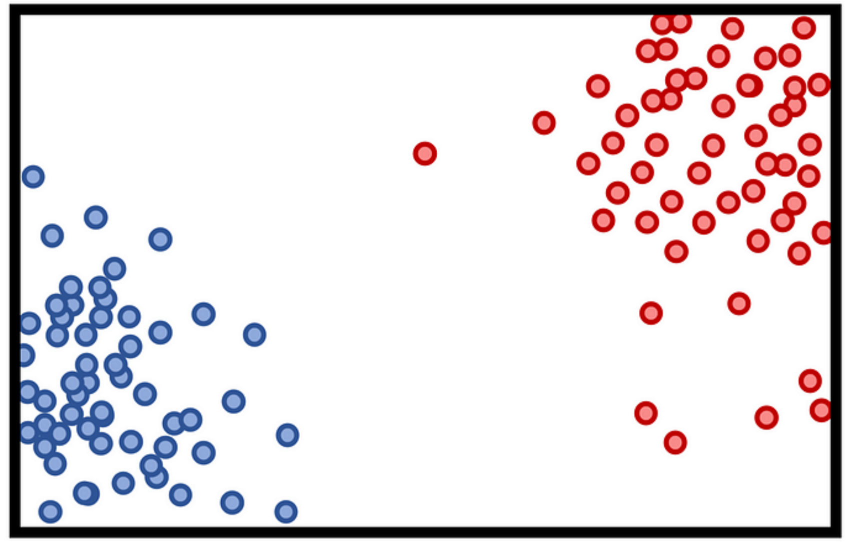
Entropy *(Cont.)*



Entropy *(Cont.)*



High Entropy



Low Entropy

Entropy (*Cont.)*

Information answers questions

The more clueless I am about the answer initially, the more information is contained in the answer

Scale: 1 bit = answer to Boolean question with prior $\langle 0.5, 0.5 \rangle$

Information in an answer when prior is $\langle P(x=1), \ldots, P(x=n) \rangle$ is

$$H(P_1, \ldots, P_n) = \sum_{i=1}^{n} -P(x=i) \cdot \log_2 P(x=i)$$

Example of Entropy

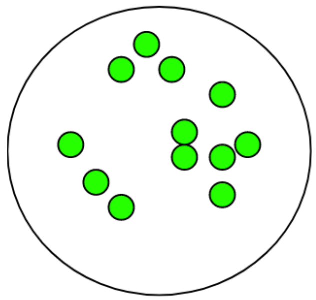
$$H(P_1, \ldots, P_n) = \sum_{i=1}^{n} -P(x = i) \cdot \log_2 P(x = i)$$

What is the entropy of a group in which all examples belong to the same class?

$$H_{min} = -1 \cdot \log_2 1 = 0$$

Not a good training set for learning

Minimum impurity

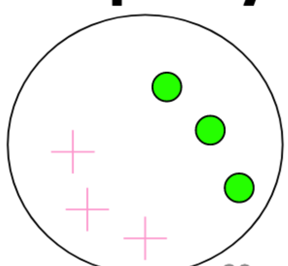


What is the entropy of a group with 50% in either class?

$$H_{max} = -0.5 \cdot \log_2 0.5 - 0.5 \cdot \log_2 0.5 = 1$$

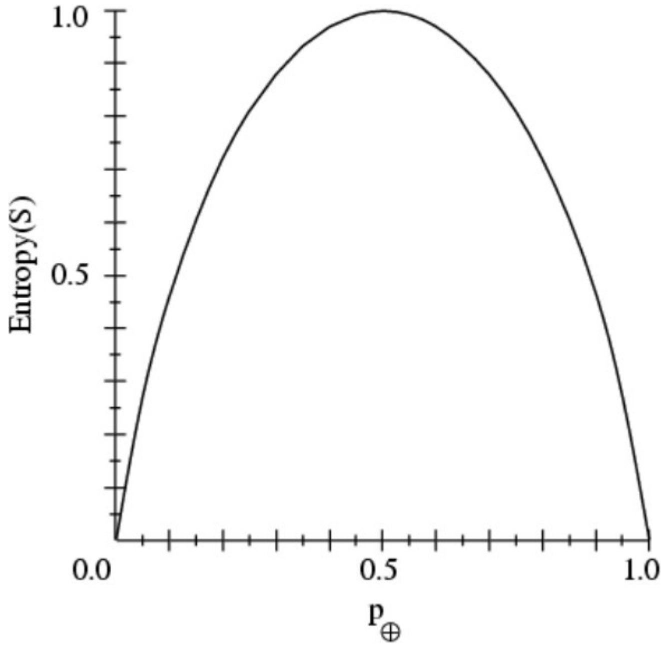
Good training set for learning

Maximum impurity



30

Sample Entropy

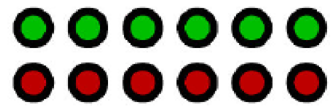


- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Entropy prior to splitting

Instances where I waited
Instances where I didn't

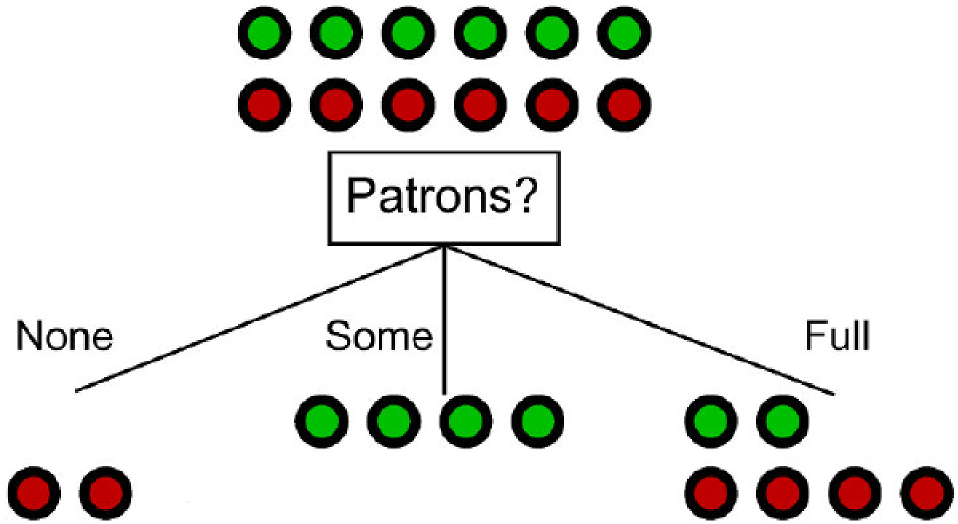


$$P_1 = \text{probability I will wait for a table}$$

$$P_2 = \text{probability I will NOT wait for a table}$$

$$\begin{aligned} H_0\langle P_1, P_2\rangle &= \sum_j -P_j \log_2 P_j \\ &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\ &= 1 \end{aligned}$$

If we split on Patrons

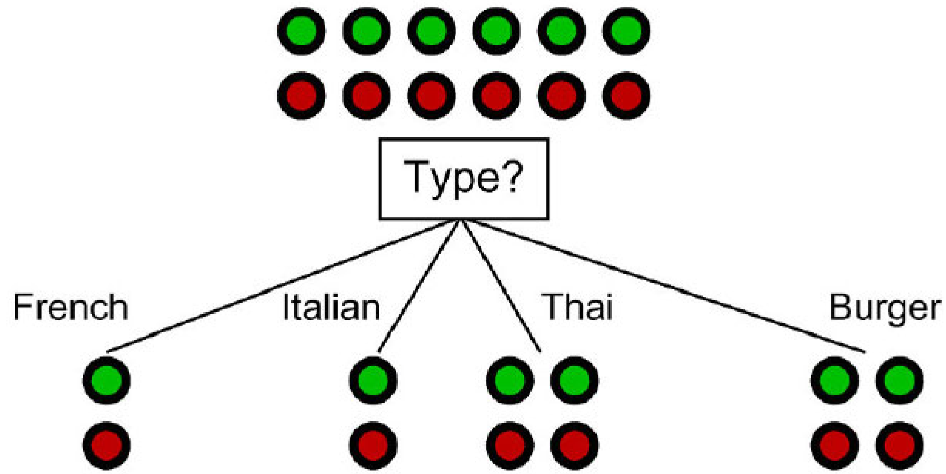


$$H_{none} \qquad H_{some} \qquad H_{full}$$

$$H_{Patrons} = W_{none}H_{none} + W_{some}H_{some} + W_{full}H_{full}$$

$$= \frac{2}{12}0 + \frac{4}{12}0 + \frac{6}{12}\left(-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}\right) = .459$$

If we split on Type



$$H_{Type} = W_{french}H_{french} + W_{italian}H_{italian} + W_{thai}H_{thai} + W_{burger}H_{burger}$$

$$= \frac{2}{12}1 + \frac{2}{12}1 + \frac{4}{12}1 + \frac{4}{12}1 = 1$$

Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.

- Information gain tells us how important a given attribute of the feature vectors is.

- We will use it to decide the ordering of attributes in the nodes of a decision tree.

Information Gain - Mathematics

Entropy H(X) of a random variable X

$$H(X) = \sum_{i=1}^{n} -P(X = i) \cdot \log_2 P(X = i)$$

Specific conditional entropy H(X|Y=v) of X given Y=v :

$$H(X|Y = v) = \sum_{i=1}^{n} -P(X = i|Y = v) \cdot \log_2 P(X = i|Y = v)$$

Conditional entropy H(X|Y) of X given Y :

$$H(X|Y) = \sum_{v \in values(Y)} P(Y = v) \cdot H(X|Y = v)$$

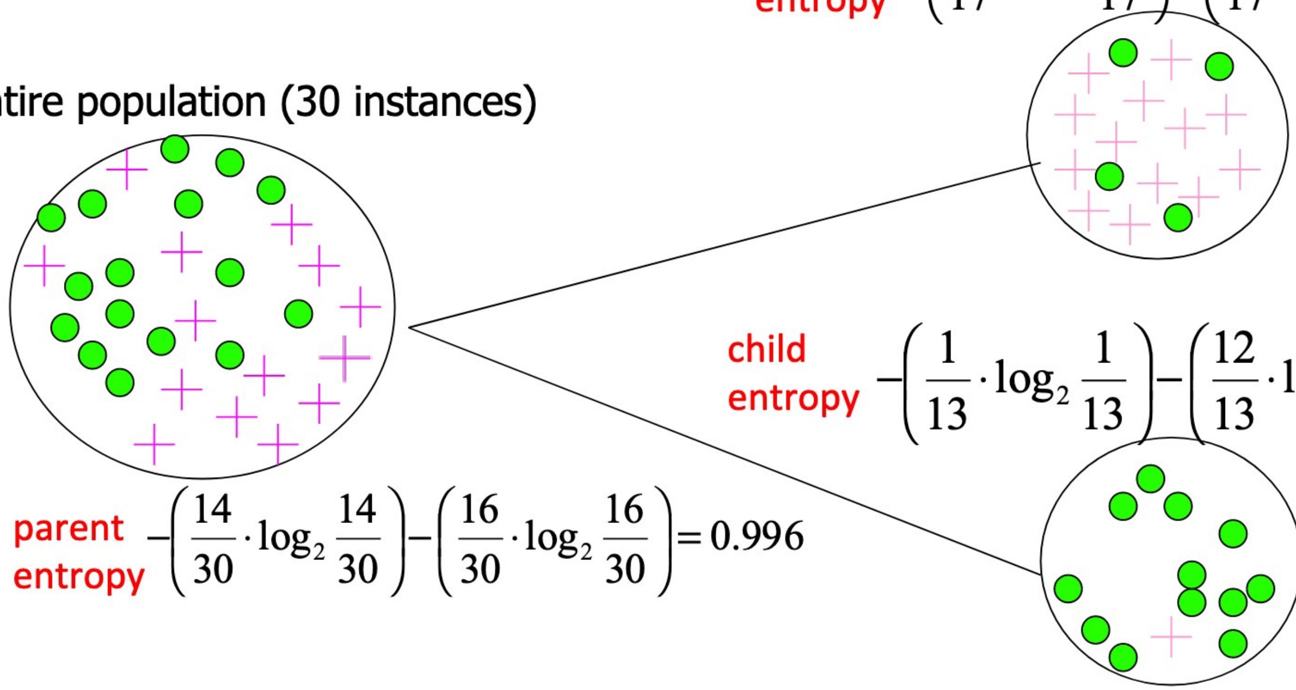
Mututal information (aka Information Gain) of X and Y :

$$I(X, Y) = H(X) - H(X|Y)$$

Information Gain Example

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$



child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38

38