Decision Tree

Artificial Intelligence

Ko, Youngjoong

Index

1. What is Decision Tree?

2. Entropy

3. Information Gain

Decision Tree

❖ Function Approximation

Problem Setting

- Set of possible instances $\mathcal{X}$
- Set of possible labels $\mathcal{Y}$
- Unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \rightarrow \mathcal{Y}\}$

Input: Training examples of unknown target function f

$$\{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n} = \{\langle \boldsymbol{x}_1, y_1 \rangle, \ldots, \langle \boldsymbol{x}_n, y_n \rangle\}$$

Output: Hypothesis $h \in H$ that best approximates f

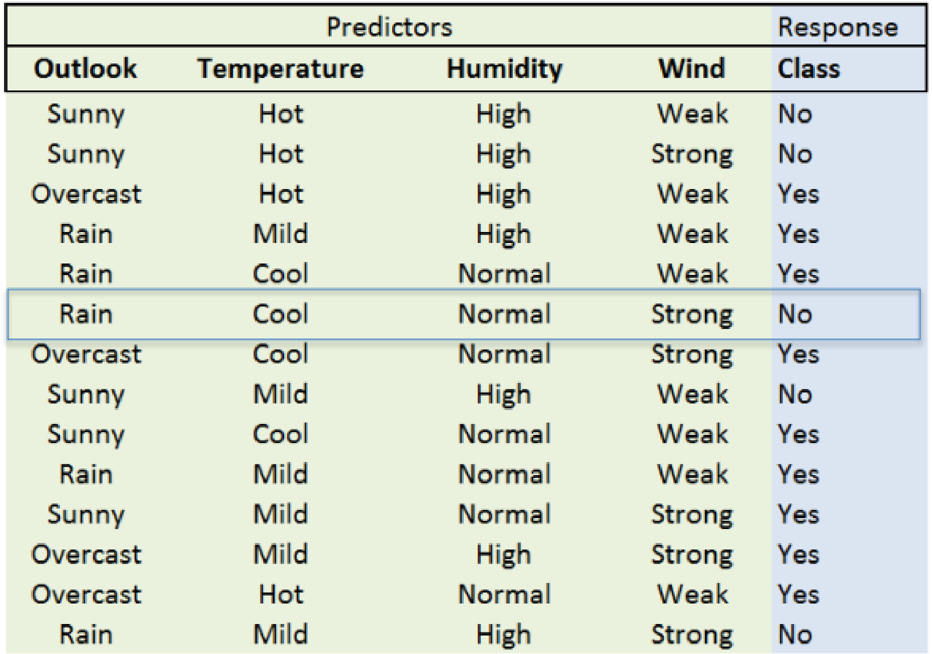
Based on slide by Tom Mitchell

Decision Tree

Sample Dataset

- Columns denote features X_i
- Rows denote labeled instances $\langle \boldsymbol{x}_i, y_i \rangle$
- Class label denotes whether a tennis game was played

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

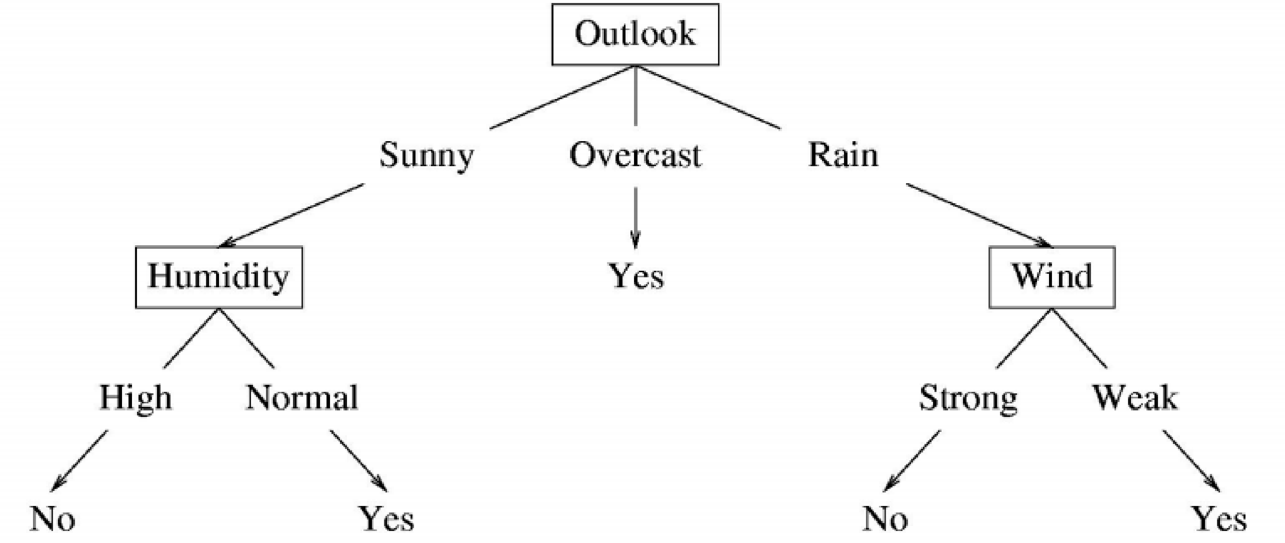


Predictors				Response
Outlook	**Temperature**	**Humidity**	**Wind**	**Class**
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal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

Slide Credit: Eric Eaton

Decision Tree

- A possible decision tree for the data:

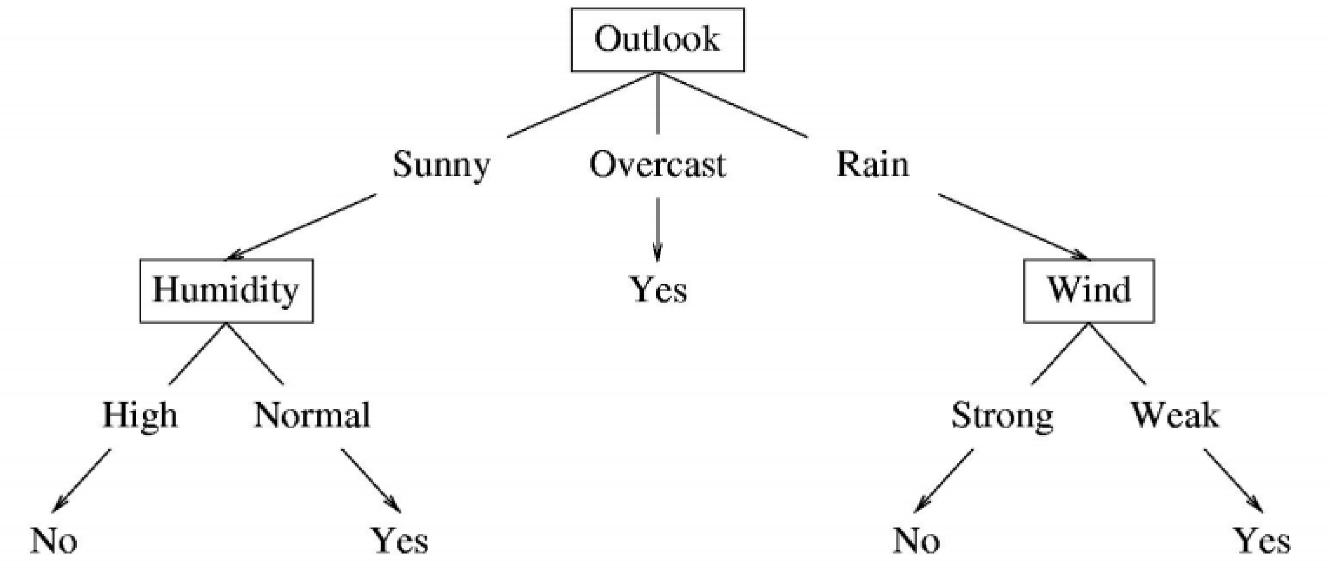


- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid \boldsymbol{x} \in \text{leaf})$)

Based on slide by Tom Mitchell

Decision Tree

- A possible decision tree for the data:



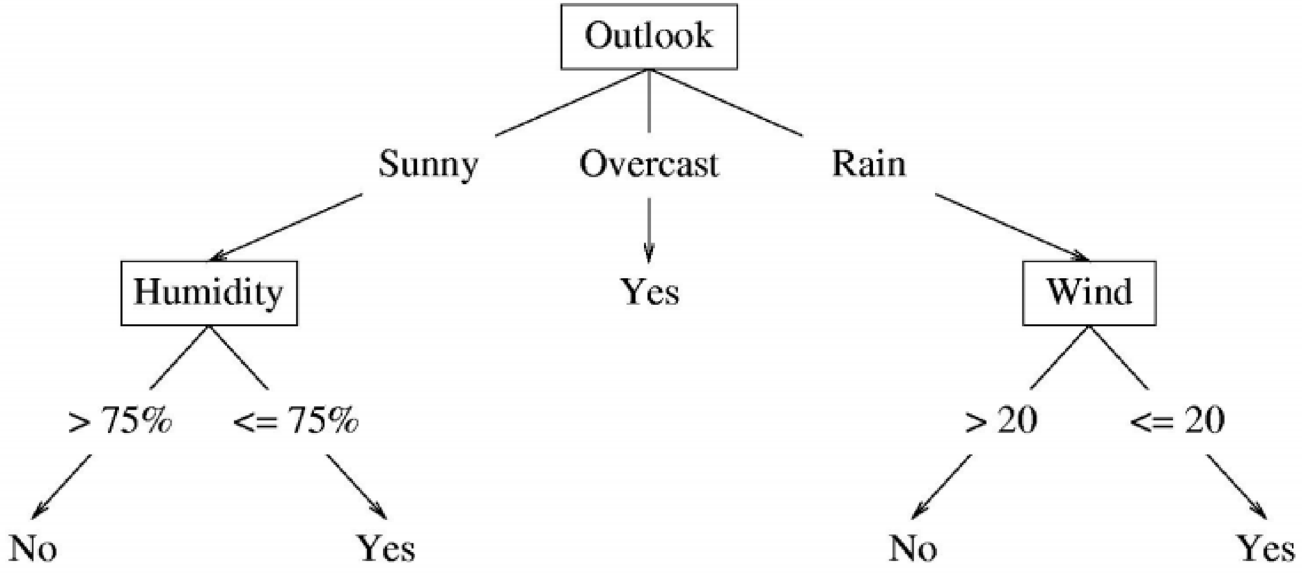
- What prediction would we make for

  <outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

Based on slide by Tom Mitchell

Decision Tree

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

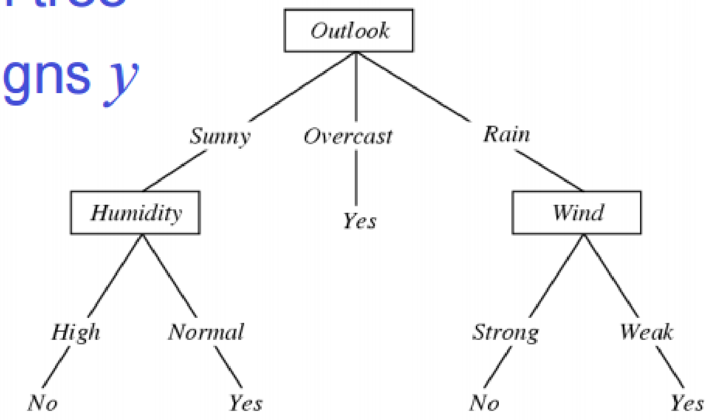


Slide Credit: Eric Eaton

Decision Tree

❖ Problem Setting:

- Set of possible instances X
  - each instance x in X is a feature vector
  - e.g., <*Humidity=low, Wind=weak, Outlook=rain, Temp=hot*>
- Unknown target function $f : X \rightarrow Y$
  - Y is discrete valued
- Set of function hypotheses $H=\{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree
  - trees sorts x to leaf, which assigns y



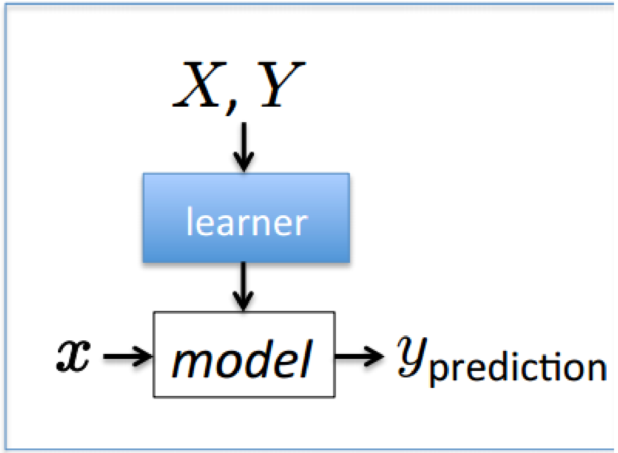
Slide by Tom Mitchell

Decision Tree

- Stages of (Batch) Machine Learning

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n}$

- Assumes each $\boldsymbol{x}_i \sim \mathcal{D}(\mathcal{X})$ with $y_i = f_{target}(\boldsymbol{x}_i)$



Train the model:

model ← *classifier*.train(X, Y)

Apply the model to new data:

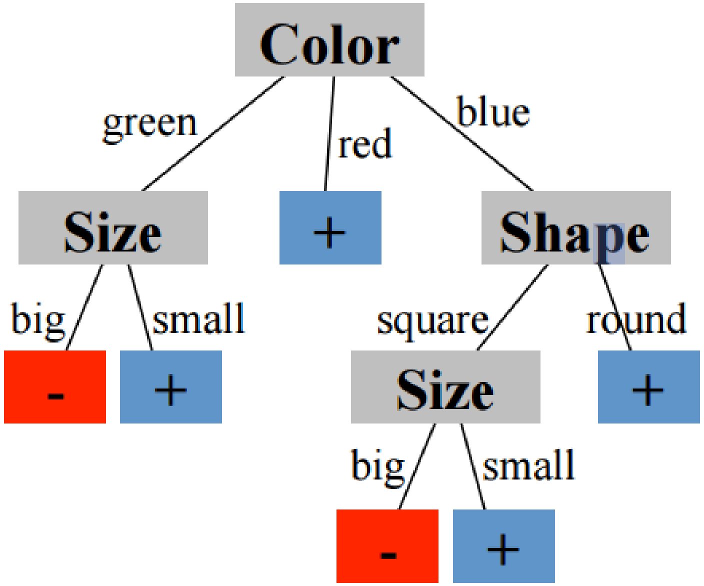
- Given: new unlabeled instance $\boldsymbol{x} \sim \mathcal{D}(\mathcal{X})$

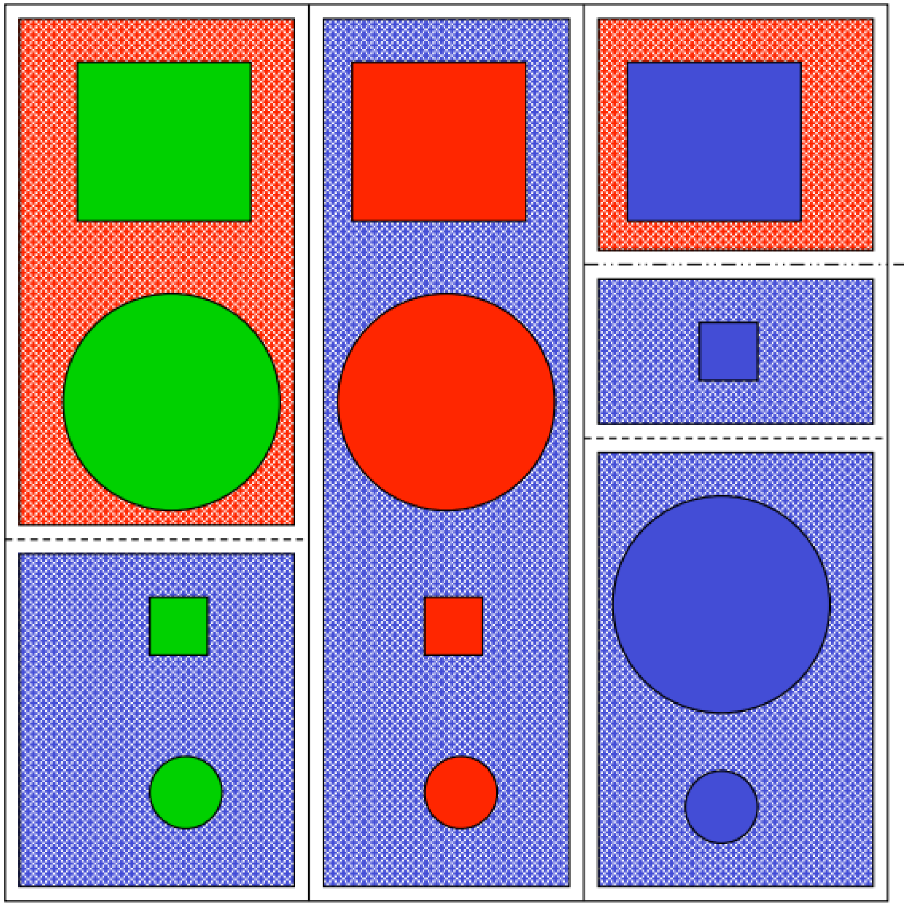
$y_{\text{prediction}}$ ← *model*.predict($\boldsymbol{x}$)

Slide Credit: Eric Eaton

Decision Tree

- Decision Tree Induced Partition



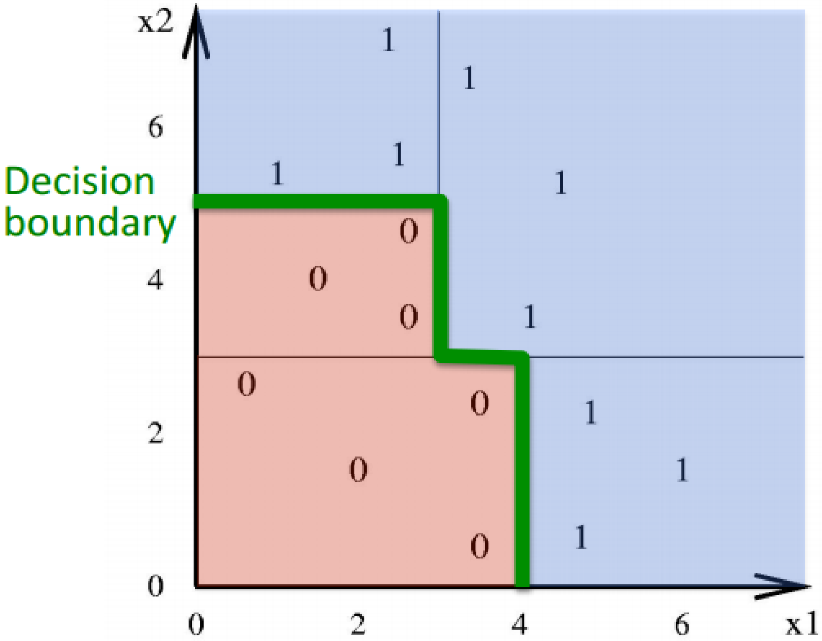


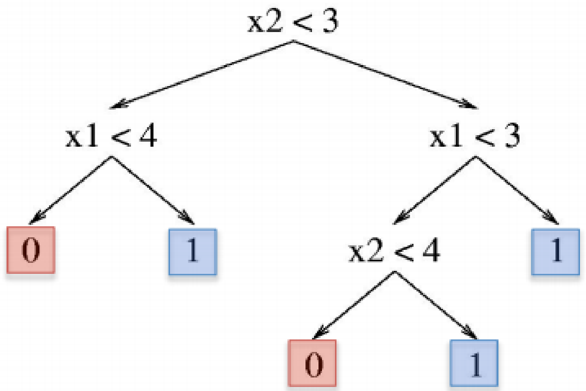
Slide Credit: Eric Eaton

Decision Tree

❖ Decision Tree – Decision Boundary

- 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



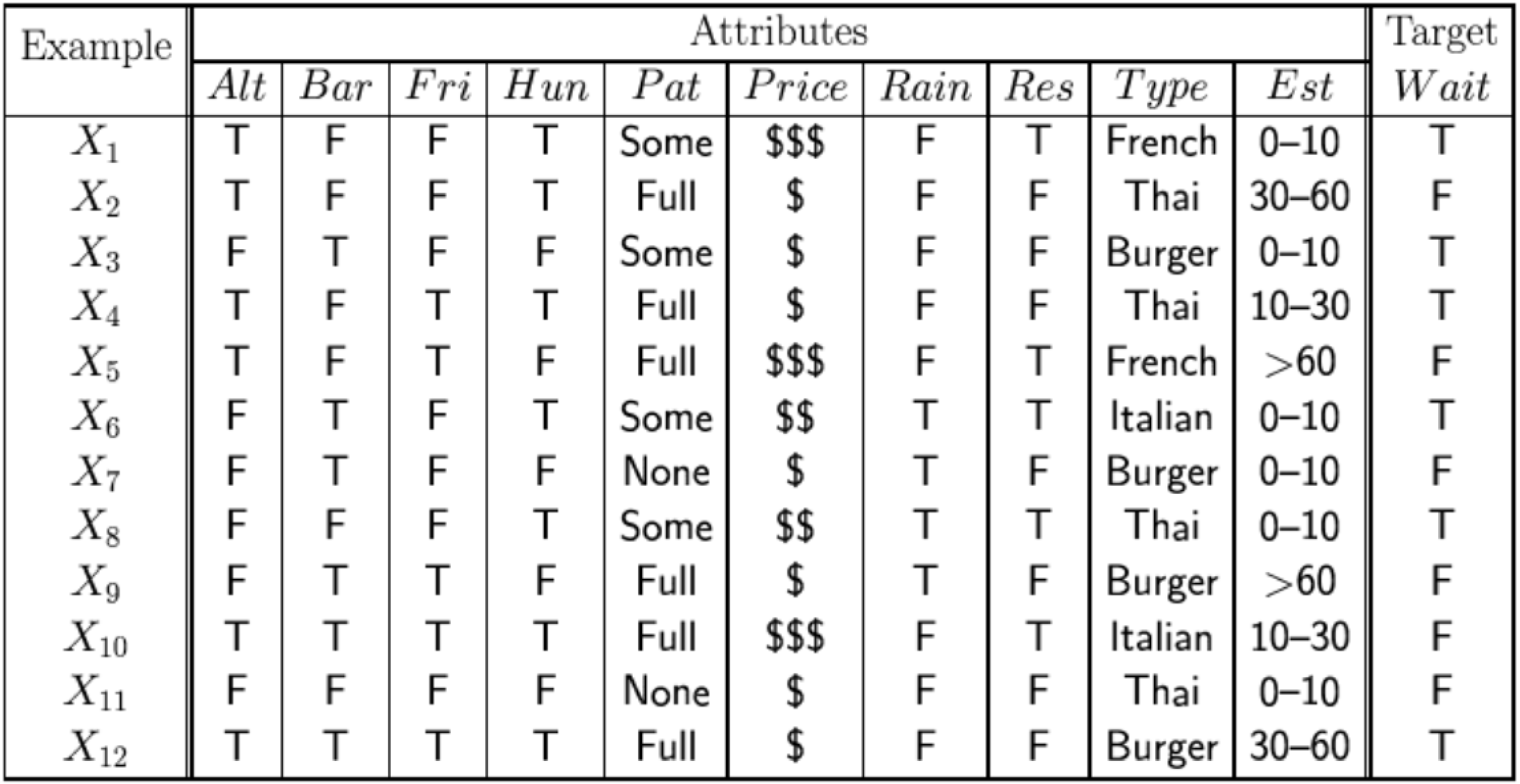


Slide Credit: Eric Eaton

Decision Tree

- Restaurant Domain (Russell & Norvig)

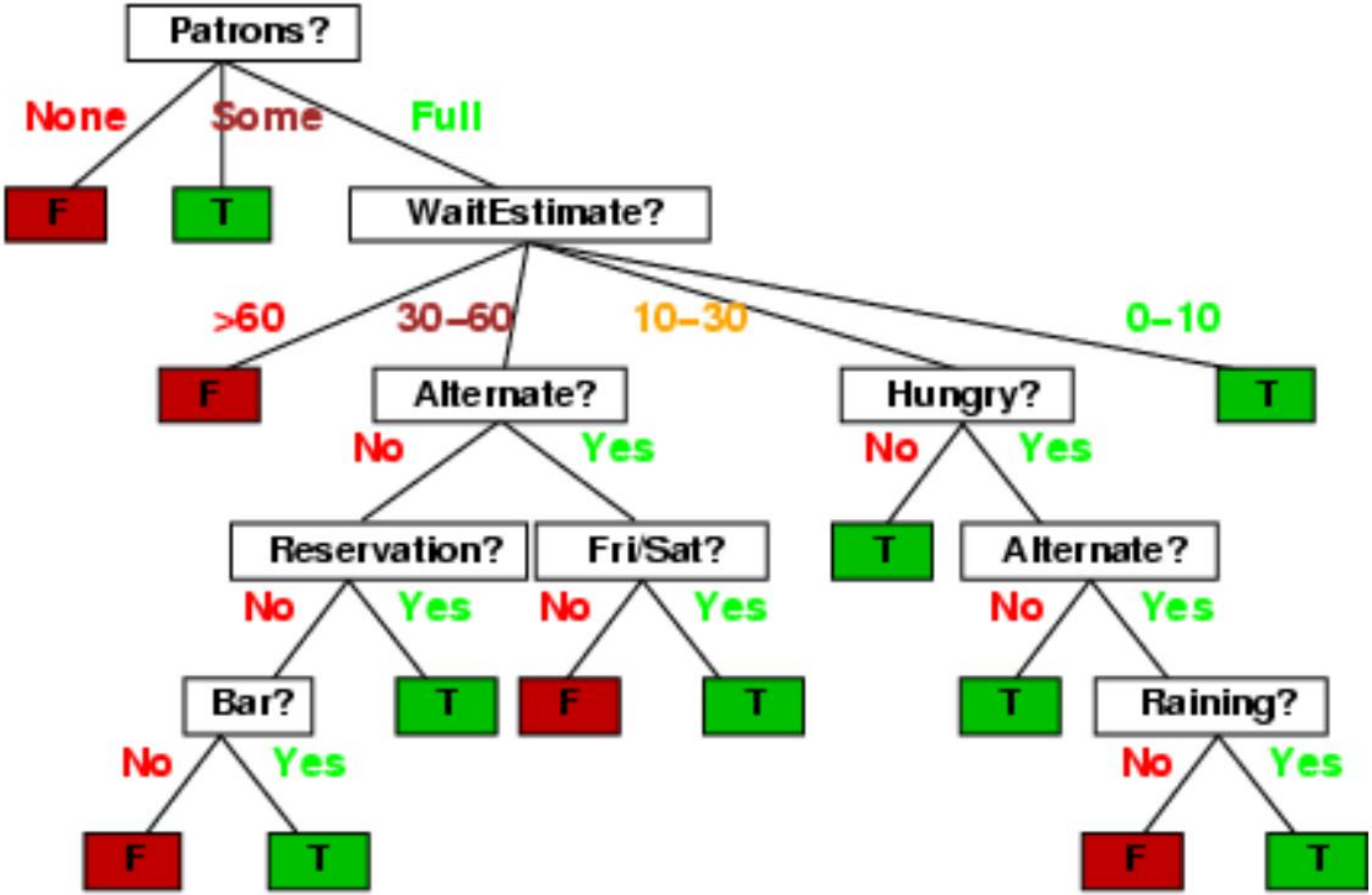
Model a patron’s decision of whether to wait for a table at a restaurant



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

~7,000 possible cases

Decision Tree



Is this the best decision tree?

Decision Tree

- Ockham’s Razor (1285-1347)

Idea: The simplest consistent explanation is the best

- Therefore, the smallest decision tree that correctly classifies all of the training examples is best.

- Finding the provably smallest decision tree is NP-hard
- So instead of constructing the absolute smallest tree consistent with the training examples, construct one that is pretty small

Basic Algorithm for Top-Down Induction of Decision Trees

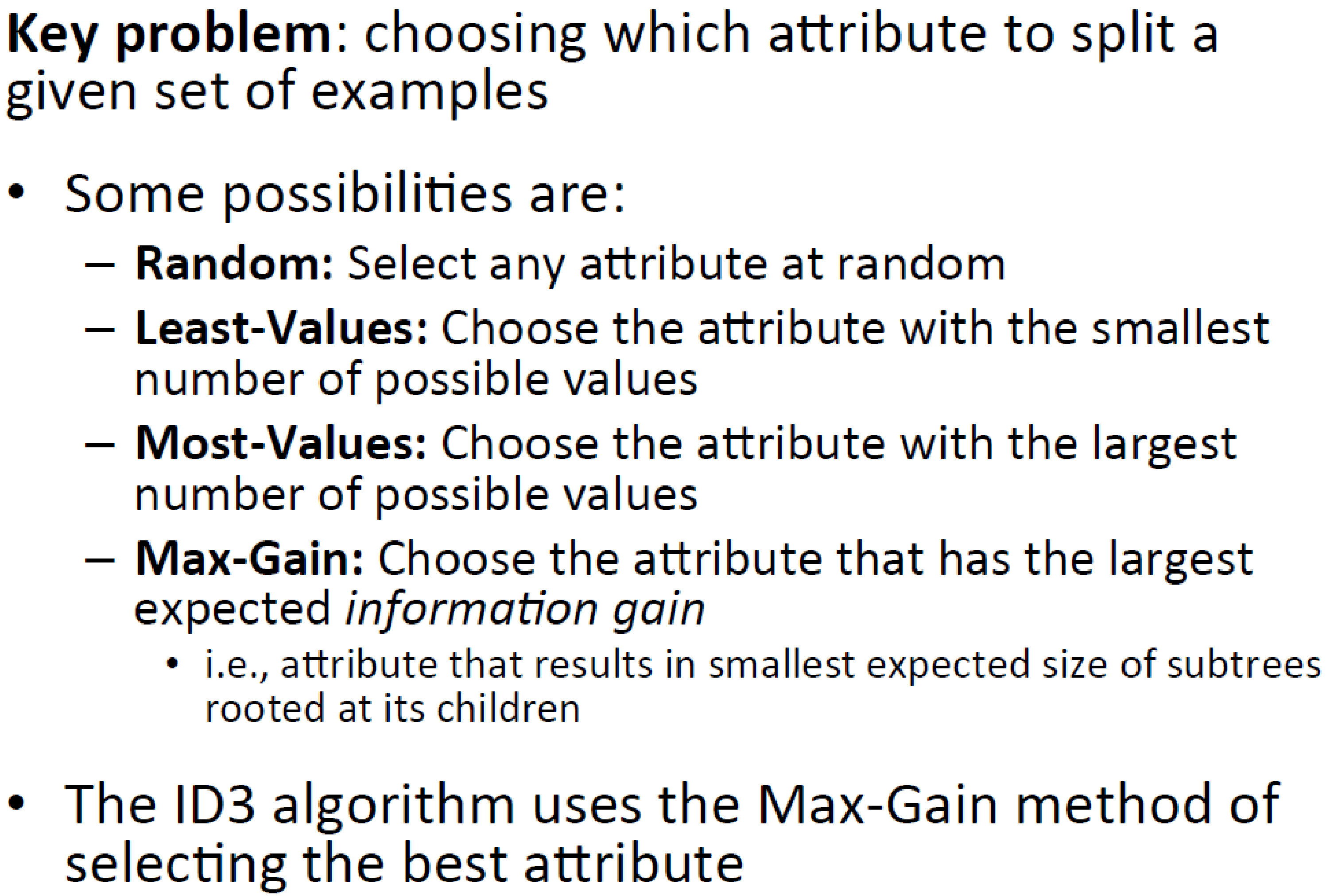
[ID3, C4.5 by Quinlan]

```
node = root of decision tree
Main loop:
1.  $A \leftarrow$ the “best” decision attribute for the next node.
2.  Assign $A$ as decision attribute for node.
3.  For each value of $A$, create a new descendant of node.
4.  Sort training examples to leaf nodes.
5.  If training examples are perfectly classified, stop.
    Else, recurse over new leaf nodes.
```

How do we choose which attribute is best?

Decision Tree

- Choosing the best Attribute



Key problem: choosing which attribute to split a given set of examples

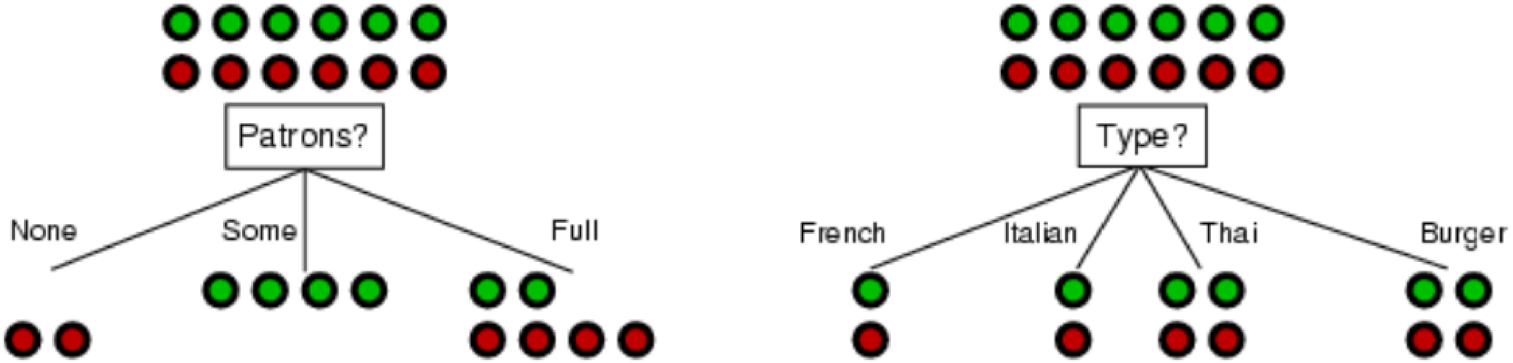
- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest expected *information gain*
    - i.e., attribute that results in smallest expected size of subtrees rooted at its children
- The ID3 algorithm uses the Max-Gain method of selecting the best attribute

Slide Credit: Eric Eaton

Decision Tree

- Choosing an Attribute

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

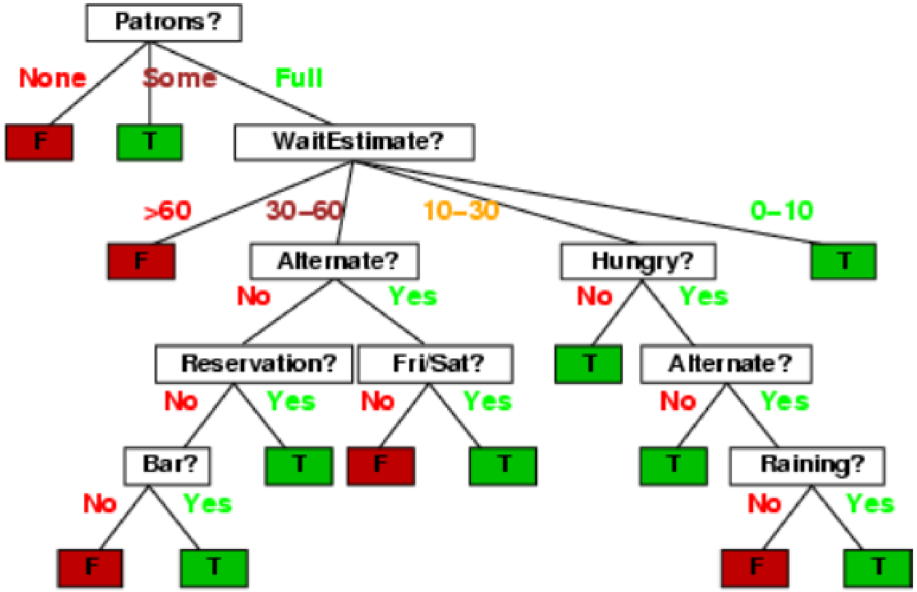


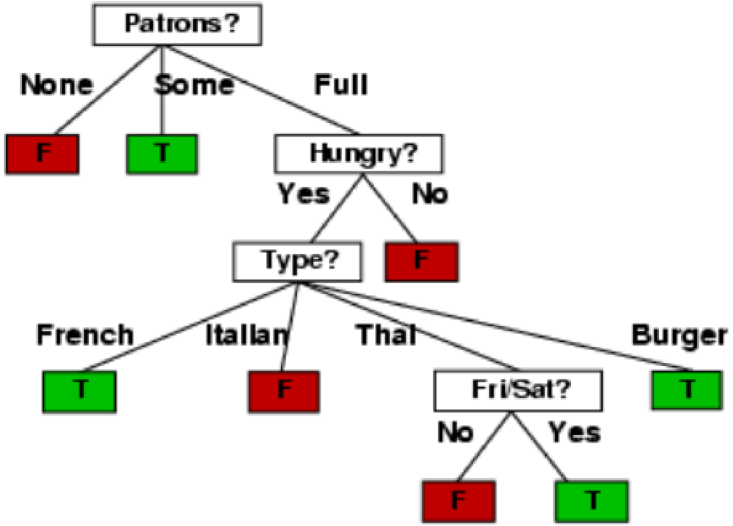
Which split is more informative: *Patrons?* or *Type?*

Based on Slide from M. desJardins & T. Finin

Decision Tree

- Compare the Two Decision Trees

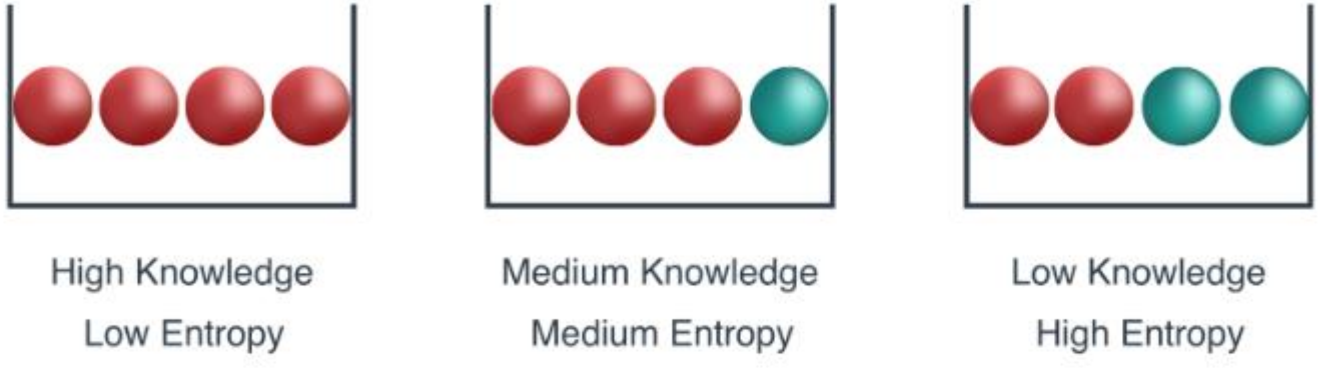




Based on Slide from M. desJardins & T. Finin

Entropy

- Entropy and Knowledge
  - 3 buckets with 4 balls each
    - Bucket 1: 4 red balls
    - Bucket 2: 3 red balls and 1 blue ball
    - Bucket 3: 2 red balls and 2 blue balls
  - Entropy is in some way, the opposite of knowledge

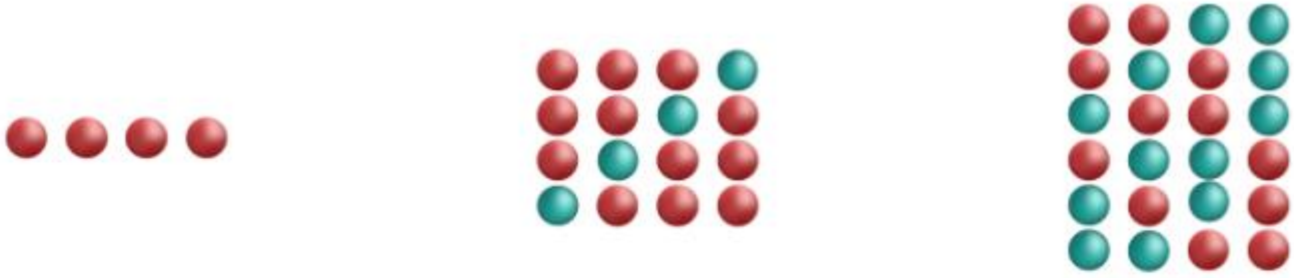


Entropy and Information are opposites

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- Entropy and Probability
  - The number of rearrangements of balls
    - 1 possible rearrangement for bucket 1
    - 4 possible rearrangement for bucket 2
    - 6 possible rearrangement for bucket 3



Number of rearrangements for the balls in each bucket

  - If there are many arrangements, then entropy is large, and if there are very few arrangements, then entropy is low.

Based on Web Site by Luis Serrano: https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4

Entropy

Entropy and Game

Game rules:

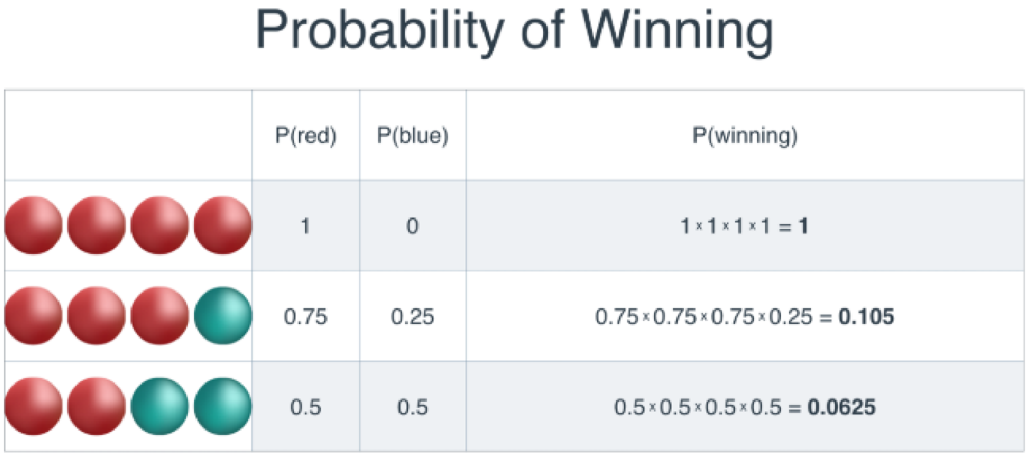
- We choose one of the three buckets.
- We are shown the balls in the bucket, in some order. Then, the balls go back in the bucket.
- We then pick one ball out of the bucket, at a time, record the color, and return the ball back to the bucket.
- If the colors recorded make the same sequence than the sequence of balls that we were shown at the beginning, then we win 1,000,000 dollars. If not, then we lose.

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

Entropy and Game

- Opposite Results:



Probability of Winning

	P(red)	P(blue)	P(winning)
🔴🔴🔴🔴	1	0	1 × 1 × 1 × 1 = **1**
🔴🔴🔴🔵	0.75	0.25	0.75 × 0.75 × 0.75 × 0.25 = **0.105**
🔴🔴🔵🔵	0.5	0.5	0.5 × 0.5 × 0.5 × 0.5 = **0.0625**

- Turning Products into Sums
  - Products are never very good
  - How would the product of a million small probabilities (between 0 and 1) would look? It would be a ridiculously tiny number.

$$\log(ab) \;=\; \log(a) + \log(b)$$

Logarithm identity

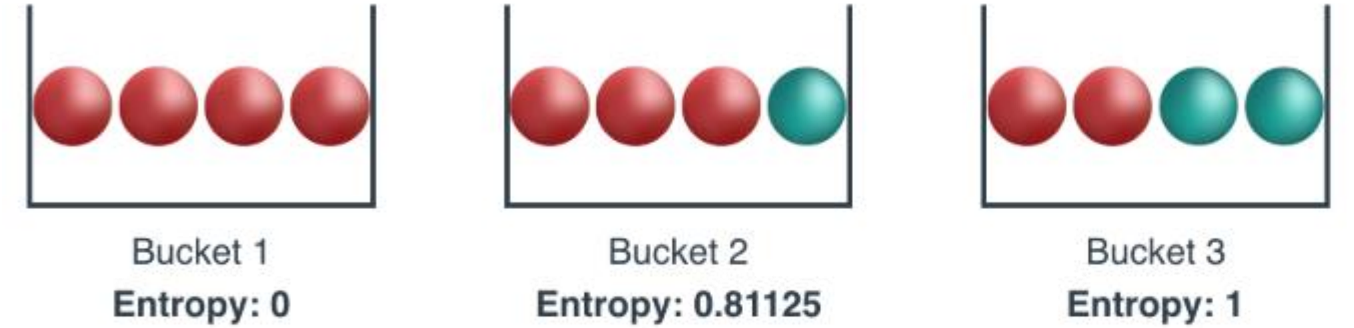
Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- Entropy and Game
  - Taking the logarithm

$$0.75 * 0.75 * 0.75 * 0.25 \;=\; 0.10546875$$

$$-\log_2(0.75) - \log_2(0.75) - \log_2(0.75) - \log_2(0.25) = 3.245$$



$$\frac{1}{4}(-\log_2(1) - \log_2(1) - \log_2(1) - \log_2(1)) = 0$$

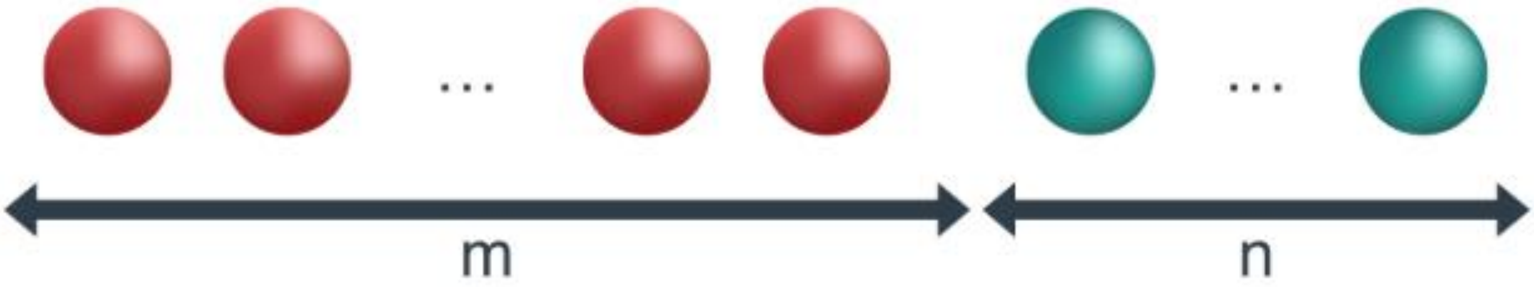
$$\frac{1}{4}(-\log_2(0.75) - \log_2(0.75) - \log_2(0.75) - \log_2(0.25)) = 0.81125$$

$$\frac{1}{4}(-\log_2\ 0.5 - \log_2\ 0.5 - \log_2\ 0.5 - \log_2\ 0.5) \;=\; 1$$

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- General Formula for Entropy

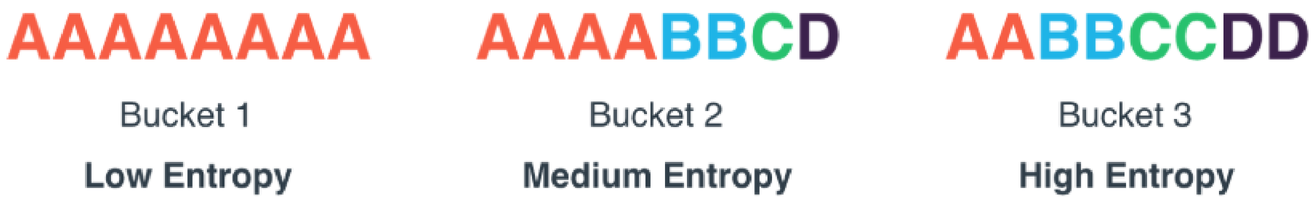


$$\text{Entropy} = \frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right) + \frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$$

Based on Web Site by Luis Serrano: https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4

Entropy

- Multi-class Entropy
  - Entropy with several classes



AAAAAAAA

Bucket 1

Low Entropy

AAAABBCD

Bucket 2

Medium Entropy

AABBCCDD

Bucket 3

High Entropy

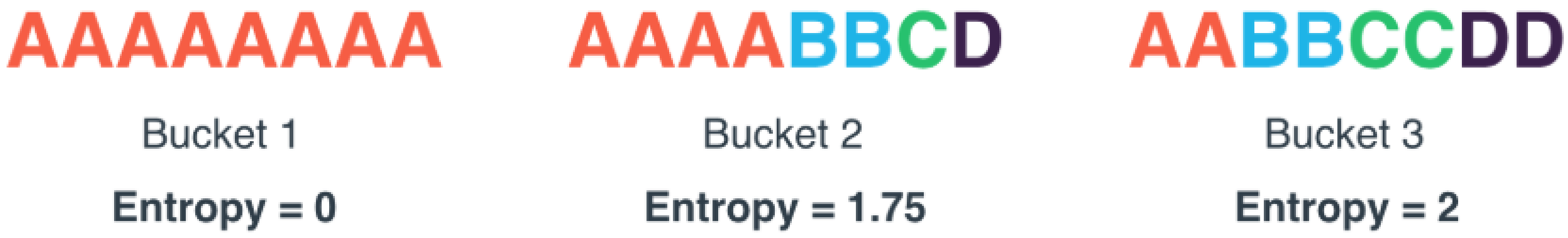
  - General formula for Multi-class entropy

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \, \log_2 \, p_i$$

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- Multi-class Entropy
  - Entropy for the three buckets



AAAAAAAA

Bucket 1

Entropy = 0

AAAABBCD

Bucket 2

Entropy = 1.75

AABBCCDD

Bucket 3

Entropy = 2

- ✓ **Bucket 1:** $\text{Entropy} = -1\log_2(1) = 0$

- ✓ **Bucket 2:** $\text{Entropy} = -\frac{4}{8}\log_2\left(\frac{4}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) = 1.75$

- ✓ **Bucket 3:** $\text{Entropy} = -\frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) = 2$

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- Information Theory
  - Another way to see entropy
  - Draw a random letter from one of the buckets.
  - *On average, how many questions do we need to ask to find out what letter it is?*

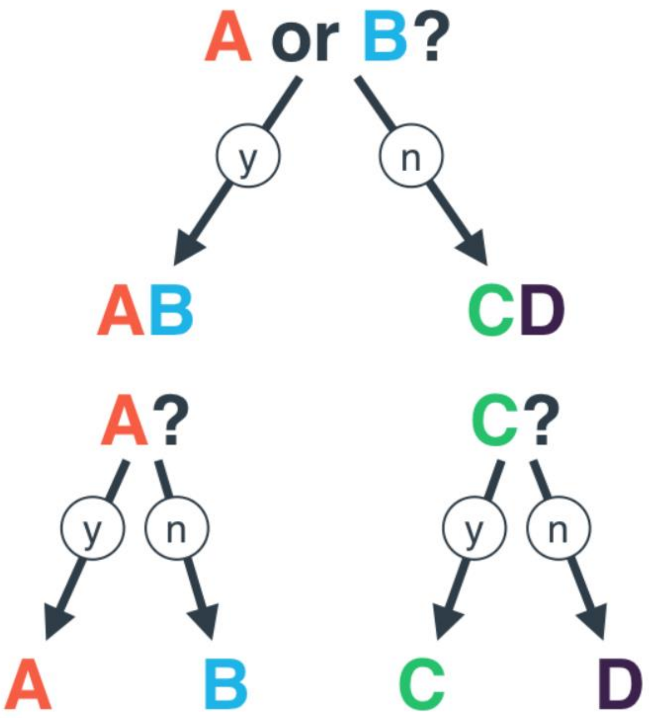
  - **The case of Bucket 1**: Average number of questions to find out the letter drawn out of Bucket 1

$$\text{Average Number of Questions} \;=\; 0$$

Based on Web Site by Luis Serrano: https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4

Entropy

- Information Theory
  - **The case of Bucket 3**: Average number of questions to find out the letter drawn out of Bucket 3



1. "Yes" and "Yes": Letter is A

2. "Yes" and "No": Letter is B

3. "No" and "Yes": Letter is C

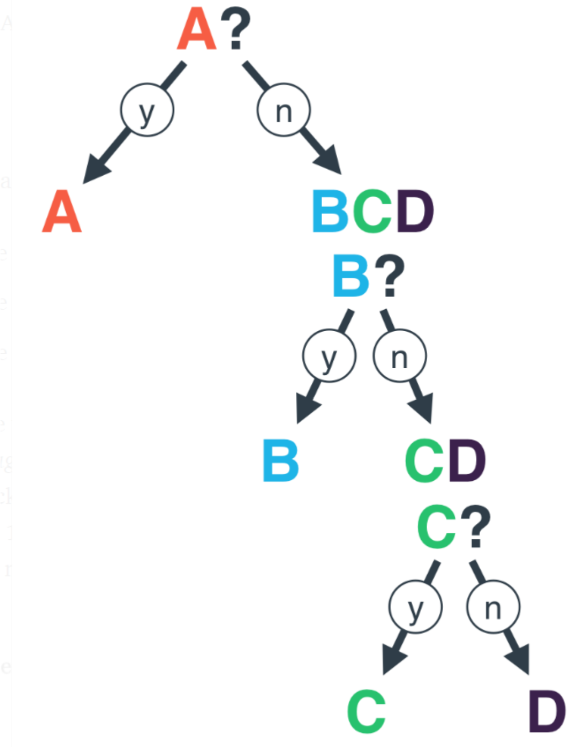
4. "No" and "No": Letter is D

$$\text{Average Number of Questions} = \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = 2$$

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

- Information Theory
  - **The case of Bucket 2**: Average number of questions to find out the letter drawn out of Bucket 2



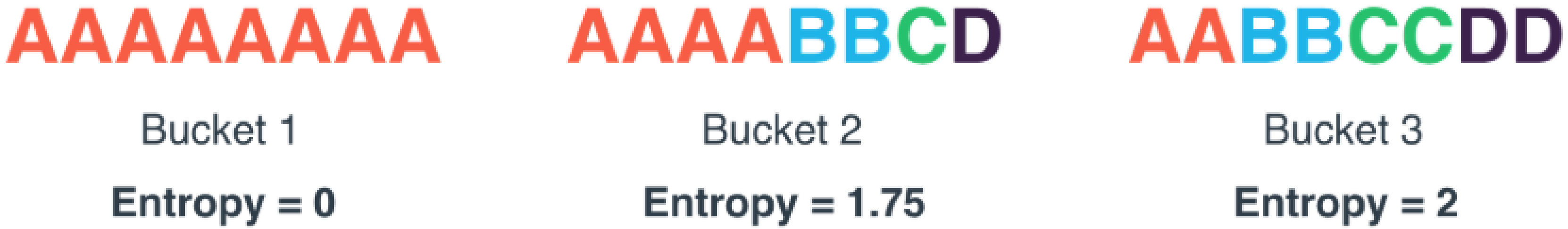
1. If the letter is A, we found out in 1 question.
2. If the letter is B, we found out in 2 questions.
3. If the letter is C or D, we found out in 3 questions.

$$\text{Average Number of Questions} = \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 = 1.75$$

Based on Web Site by Luis Serrano: https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4

Entropy

- Multi-class Entropy
  - Entropy *vs.* Average Number of Questions

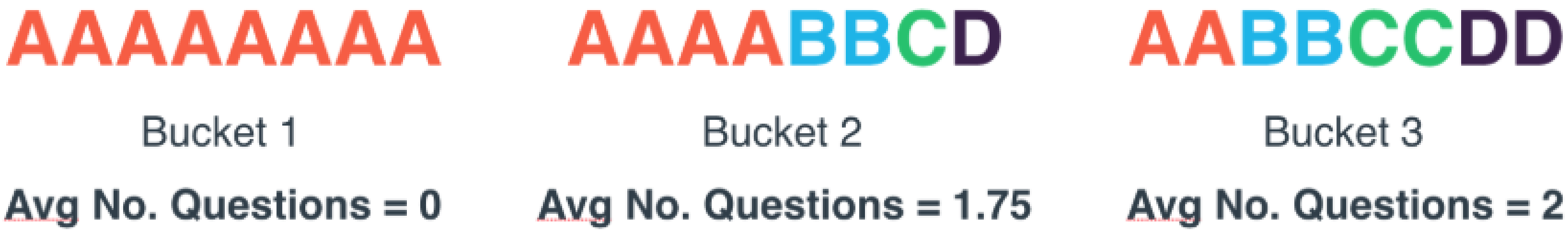


AAAAAAAA — Bucket 1 — **Entropy = 0**

AAAABBCD — Bucket 2 — **Entropy = 1.75**

AABBCCDD — Bucket 3 — **Entropy = 2**

=



AAAAAAAA — Bucket 1 — **Avg No. Questions = 0**

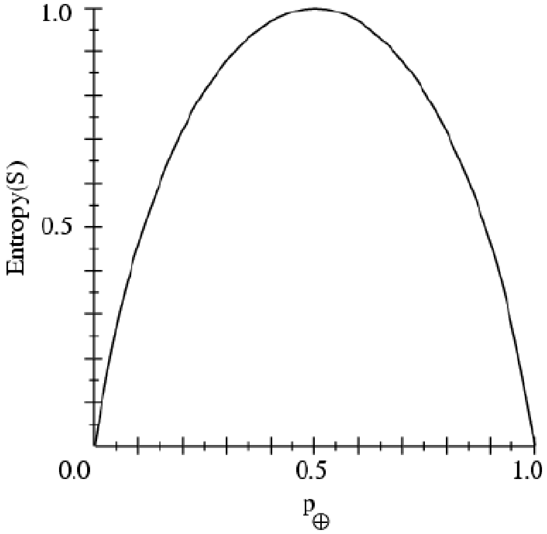
AAAABBCD — Bucket 2 — **Avg No. Questions = 1.75**

AABBCCDD — Bucket 3 — **Avg No. Questions = 2**

Based on Web Site by Luis Serrano: *https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4*

Entropy

Sample Entropy

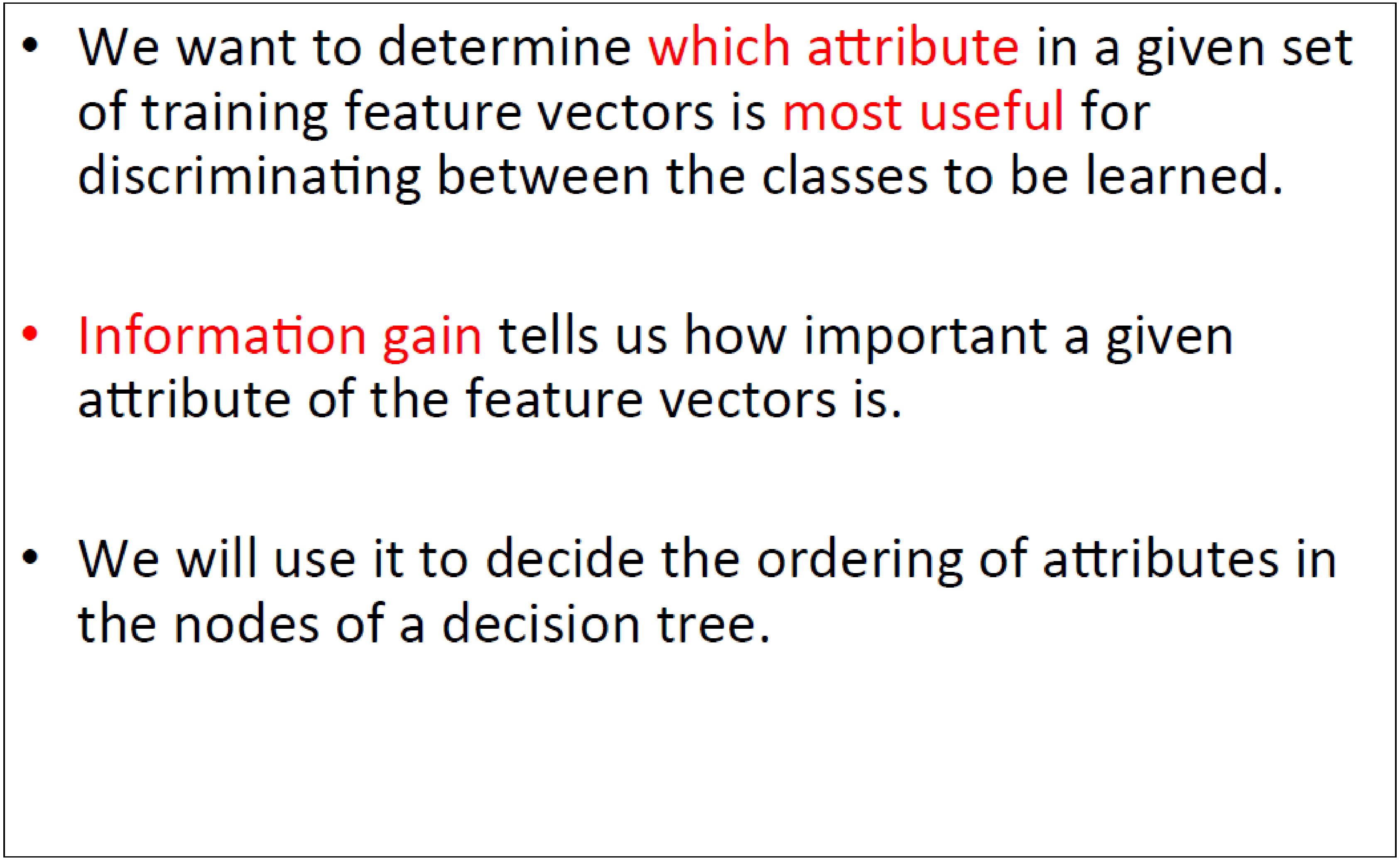


- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

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

Slide by Tom Mitchell

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.

Based on Slide by Pedro Domingos

Information Gain

- From Entropy to Information Gain

Entropy *H(X)* of a random variable *X*

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

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

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

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

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

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

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

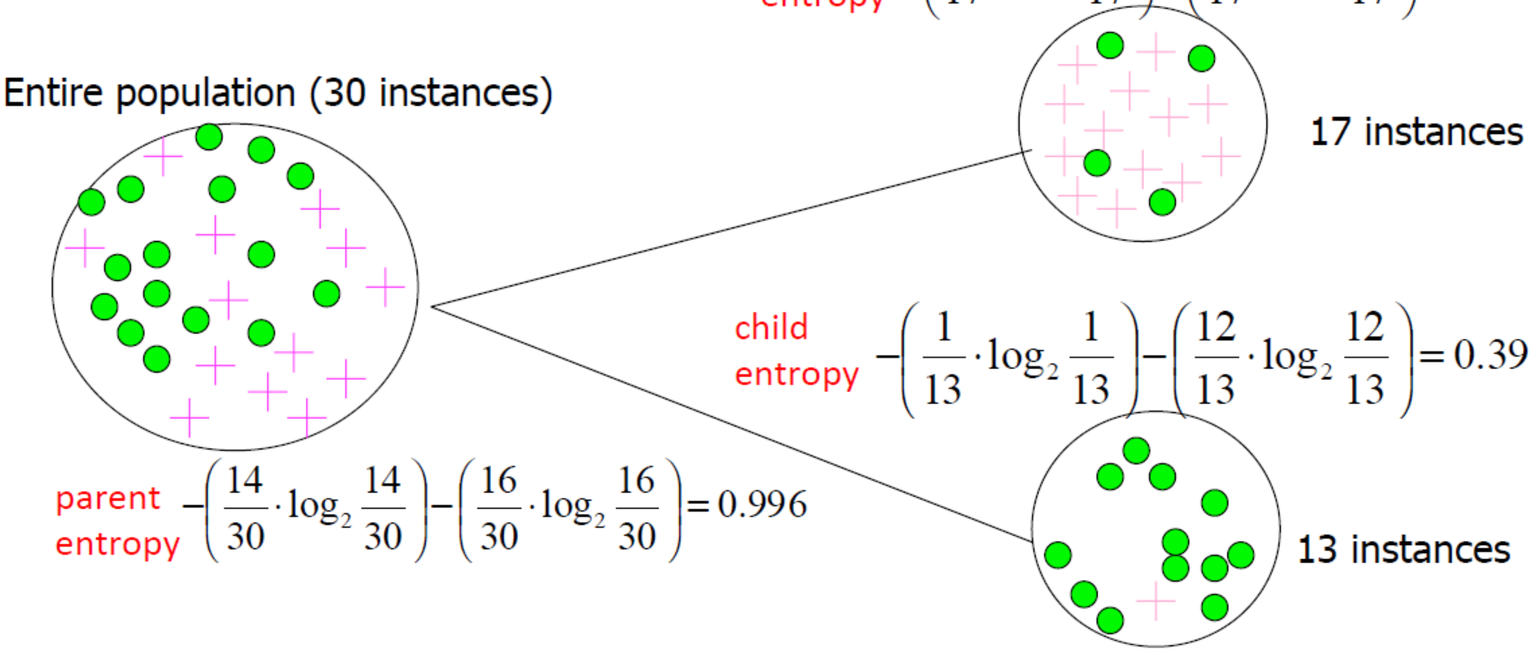
Slide by Tom Mitchell

Information Gain

- Calculating Information Gain

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

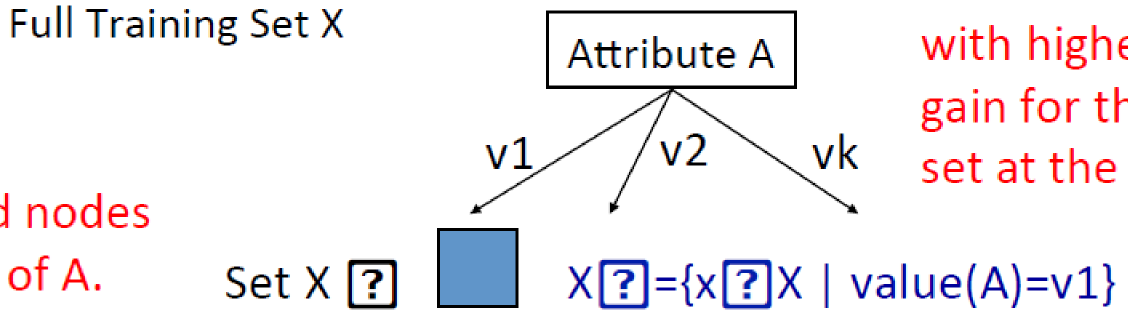
(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38

Based on slide by Pedro Domingos

Information Gain

- Using Information Gain to construct a Decision Tree



Full Training Set X

Attribute A

v1 v2 vk

Set X ?

X?={x?X | value(A)=v1}

Choose the attribute A with highest information gain for the full training set at the root of the tree.

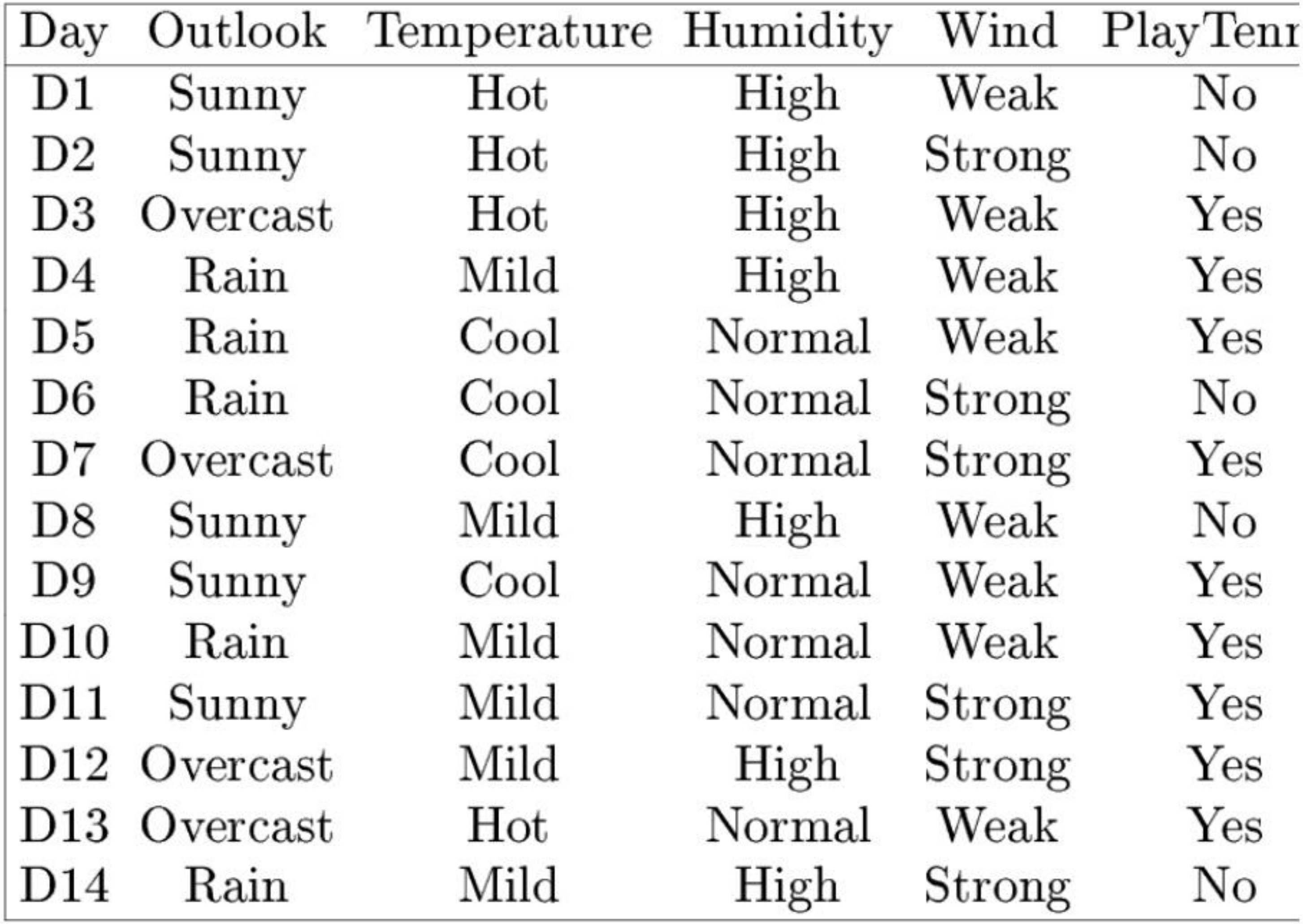
Construct child nodes for each value of A. Each has an associated subset of vectors in which A has a particular value.

repeat recursively till when?

Based on Slide by Pedro Domingos

Information Gain

- Sample Examples

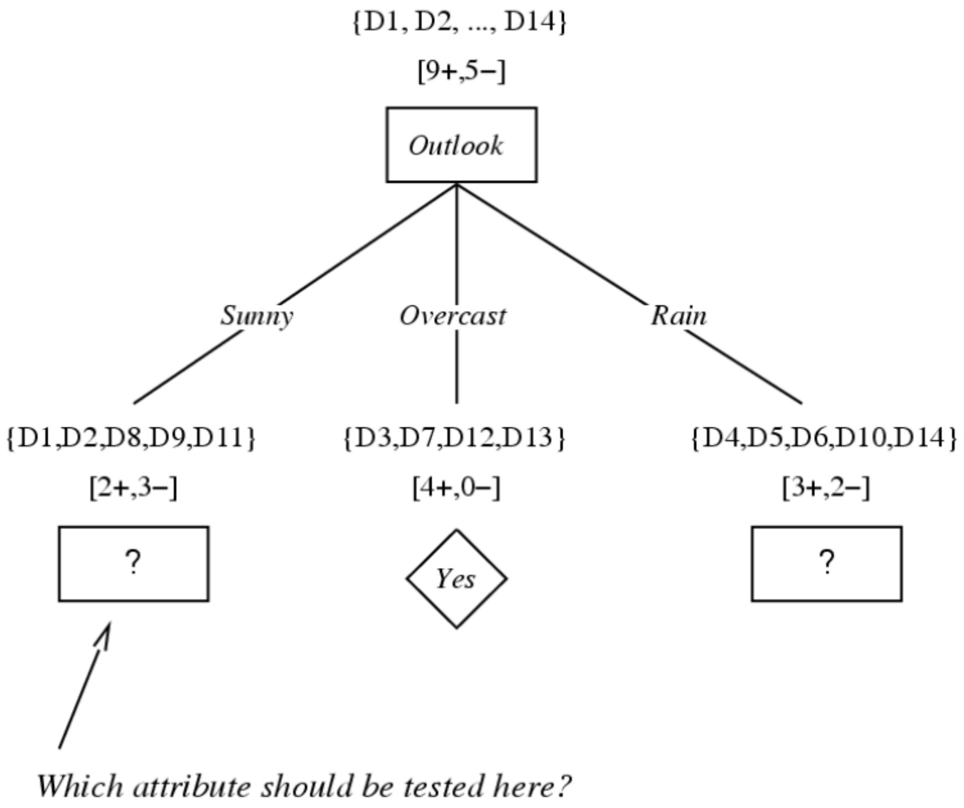


Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Slide Credit: Tom Mitchell

Information Gain

- Select the Next Attribute



{D1, D2, ..., D14}

[9+,5−]

Outlook

Sunny *Overcast* *Rain*

{D1,D2,D8,D9,D11}

[2+,3−]

?

{D3,D7,D12,D13}

[4+,0−]

Yes

{D4,D5,D6,D10,D14}

[3+,2−]

?

Which attribute should be tested here?

$$S_{sunny} = \{D1,D2,D8,D9,D11\}$$

$$Gain\ (S_{sunny}\ ,\ Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$$

$$Gain\ (S_{sunny}\ ,\ Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$$

$$Gain\ (S_{sunny},\ Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$$

Slide by Tom Mitchell