Decision Tree Basics

Kazi Aminul Islam
Department of Computer Science
Kennesaw State University

Overview

- Widely used in practice
- Strengths include
  - Fast and simple to implement
  - Can convert to rules
  - Handles noisy data
- Weaknesses include
  - Univariate splits/partitioning using only one attribute at a time --- limits types of possible trees
  - Large decision trees may be hard to understand
  - Requires fixed-length feature vectors
  - Non-incremental (i.e., batch method)

Tennis Played?

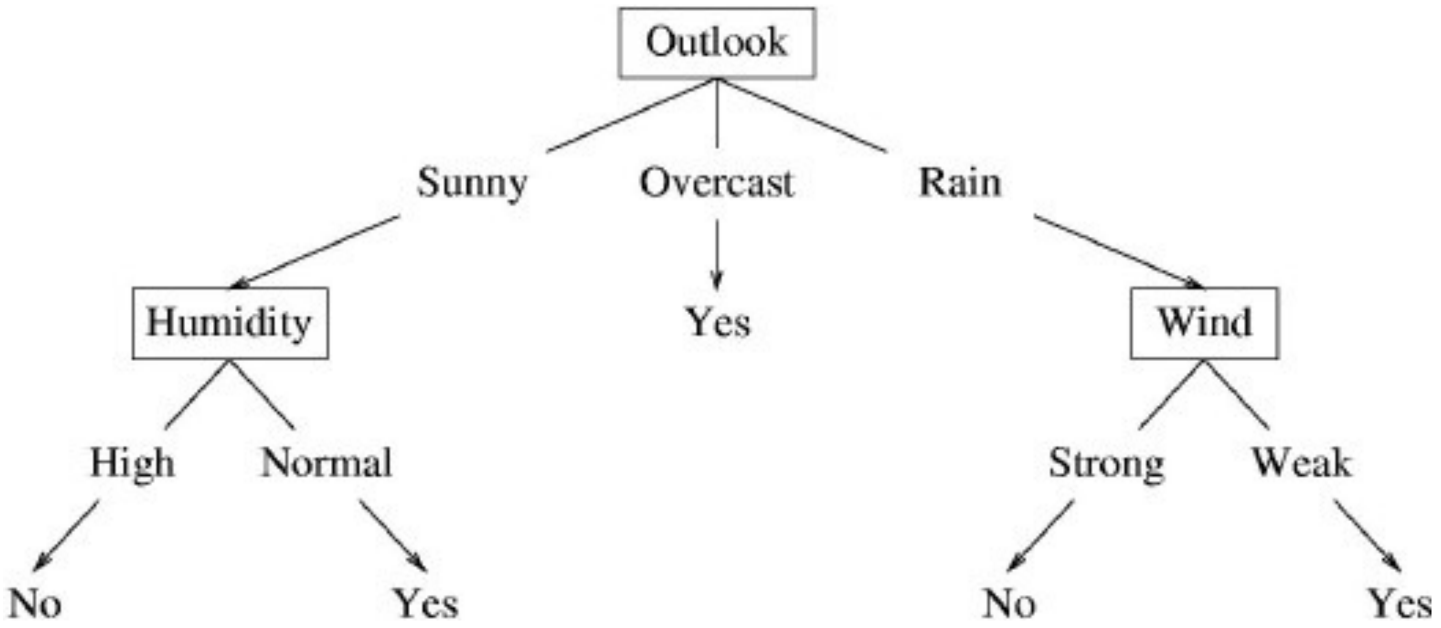
- Columns denote features Xi
- Rows denote labeled instances $\langle x_i, y_i \rangle$
- Class label denotes whether a tennis game was played

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

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

Decision Tree

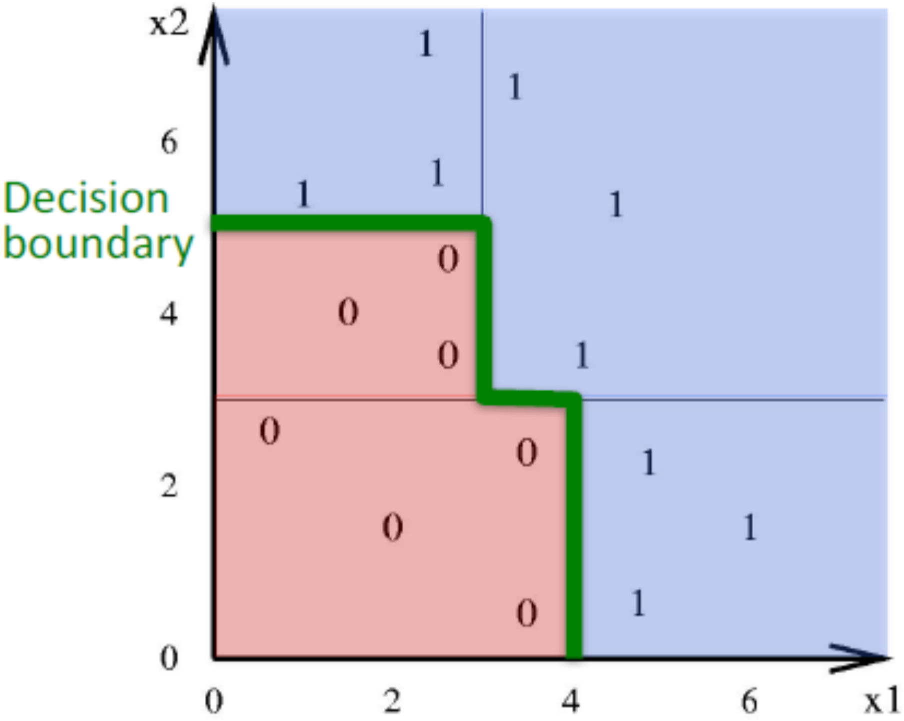
- A possible decision tree for the data:

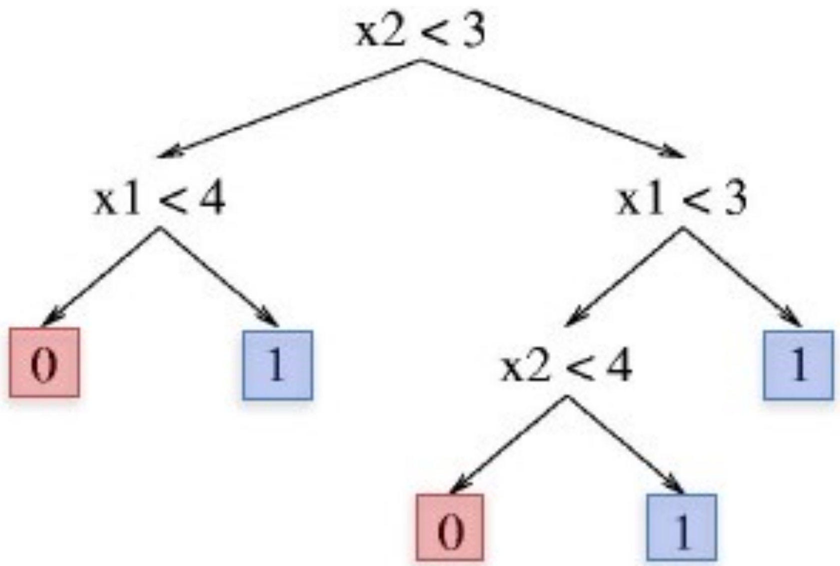


- Each internal node: test one attribute Xi
- Each branch from a node: selects one value for Xi
- Each leaf node: predict Y

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





Decision Tree – Is milk spoiled?

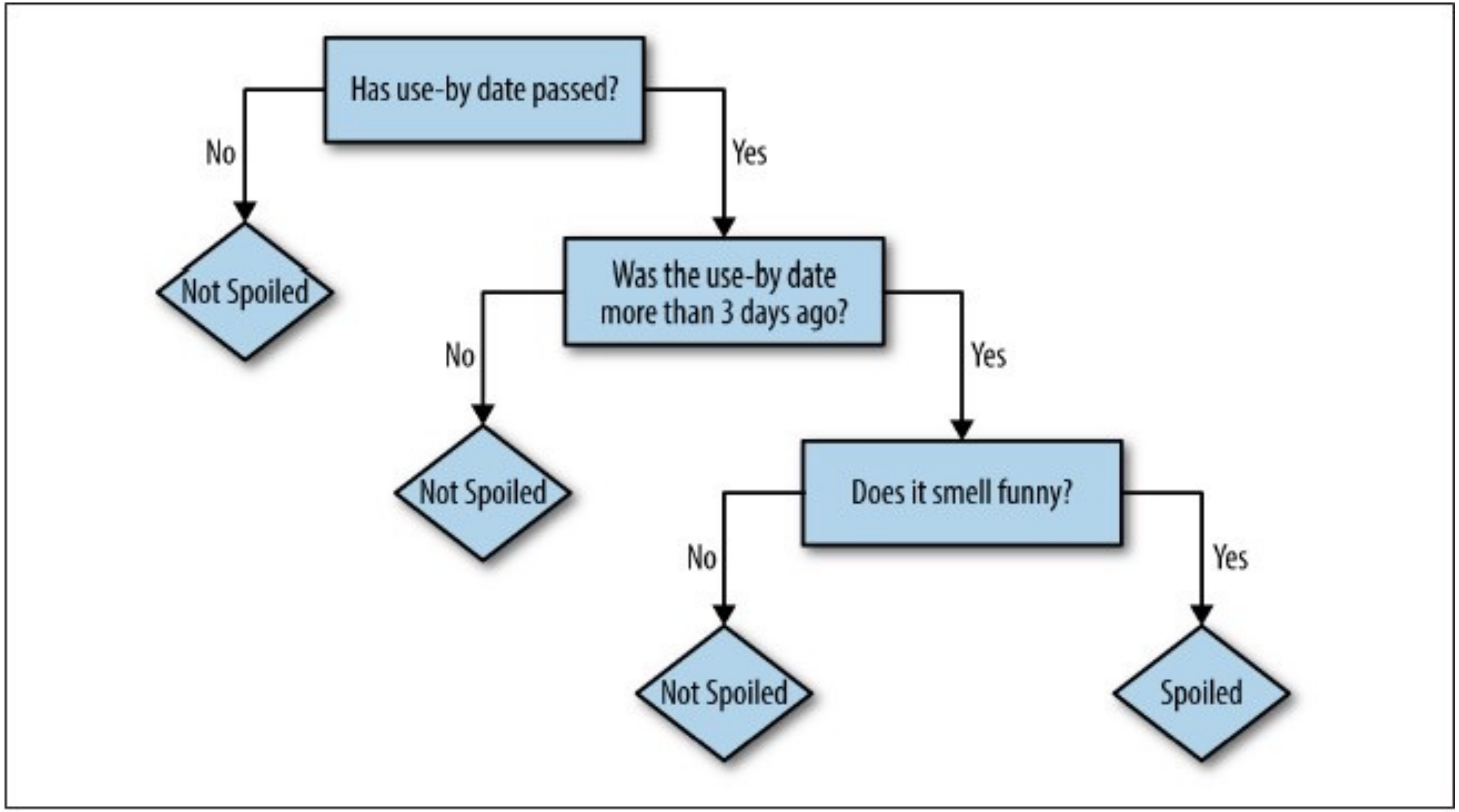


Figure 4-1. Decision tree: Is it spoiled?

Another Example

- A robot wants to decide which animals in the shop would make a good pet for a child?

Table 4-1. Exotic pet store "feature vectors"

Name	Weight (kg)	# Legs	Color	Good pet?
Fido	20.5	4	Brown	Yes
Mr. Slither	3.1	0	Green	No
Nemo	0.2	0	Tan	Yes
Dumbo	1390.8	4	Grey	No
Kitty	12.1	4	Grey	Yes
Jim	150.9	2	Tan	No

Name	Weight (kg)	# Legs	Color	Good pet?
Millie	0.1	100	Brown	No
McPigeon	1.0	2	Grey	No
Spot	10.0	4	Brown	Yes

First Decision Tree

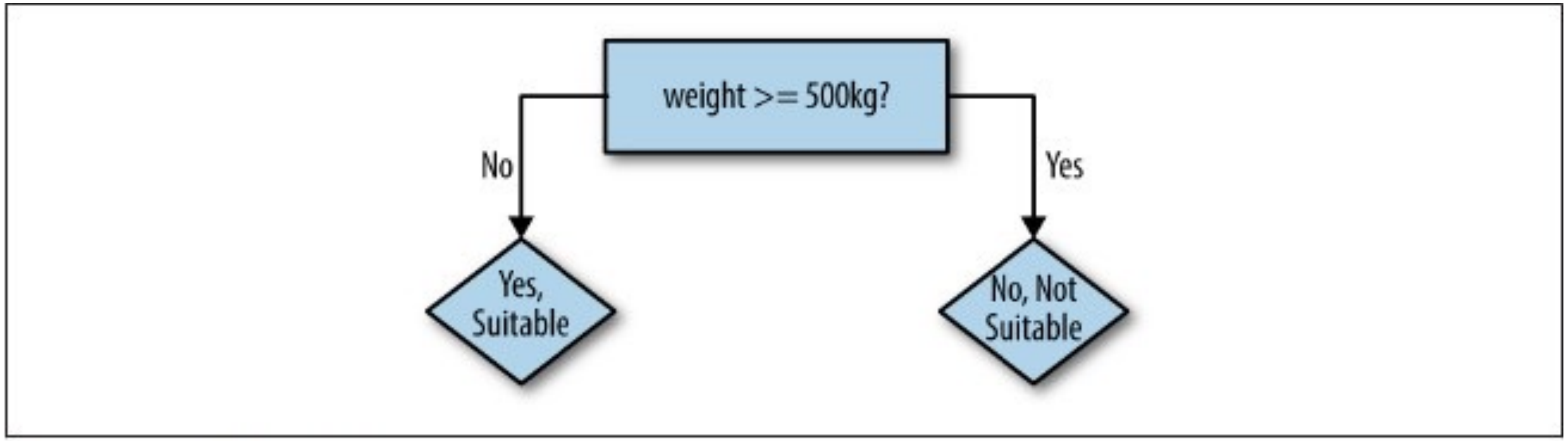


Figure 4-2. Robot's first decision tree

- This decision tree predicts 5 out of 9 correct cases.
- The threshold could be lowered to 100 kg to get 6 out of 9.
- We need to build a second decision tree for lighter cases.

The Second Decision Tree

- One direction is to pick a feature that changes some of the incorrect Yes to No, e.g., snake by color green.

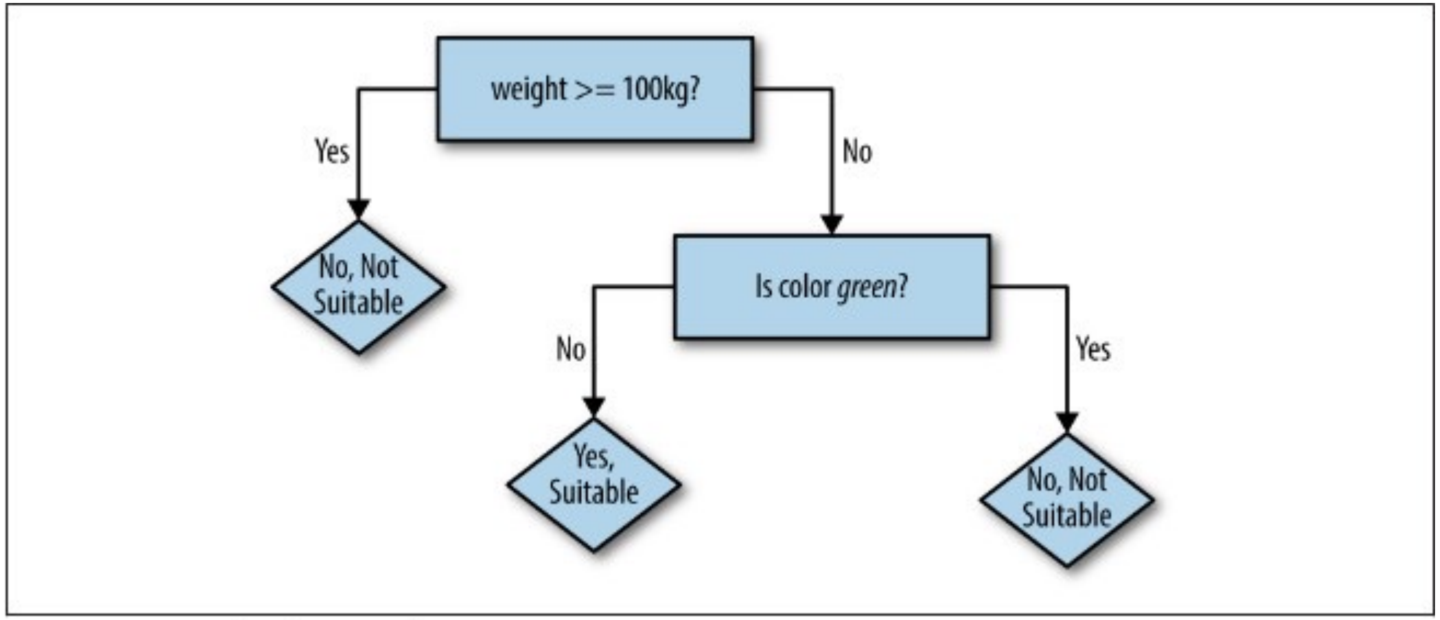
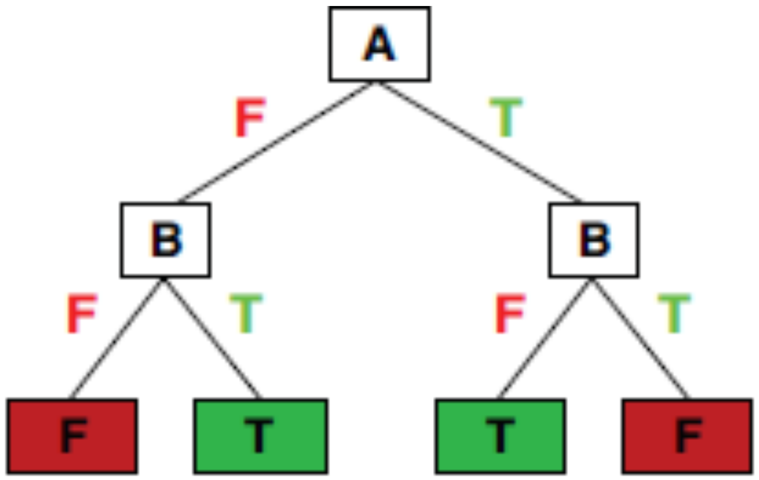


Figure 4-3. Robot's next decision tree

What Functions Can be Represented?

- Decision trees can represent any function of input attributes.
- For Boolean functions, path to leaf gives truth table row.
- However, could have exponentially many nodes.

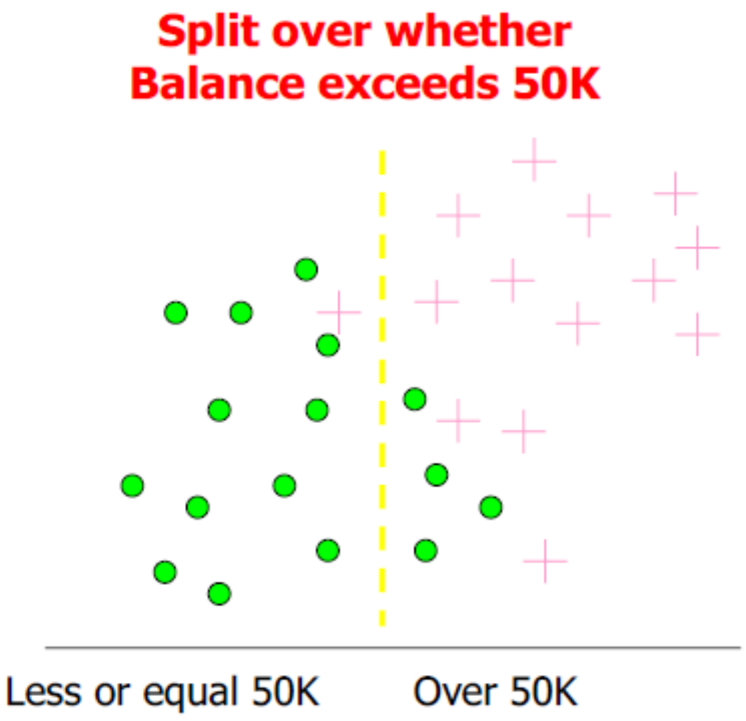
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

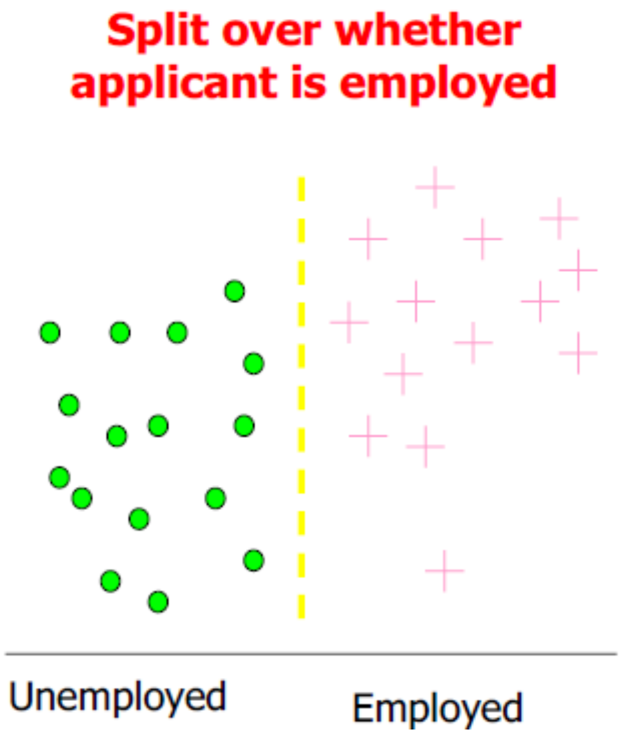


(Figure from Stuart Russell)

Information Gain

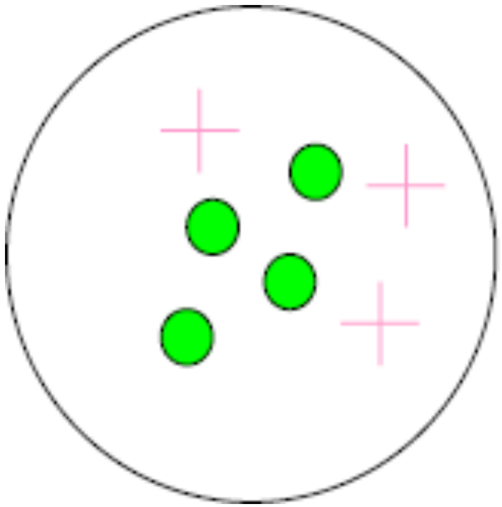
- Which test is more informative?

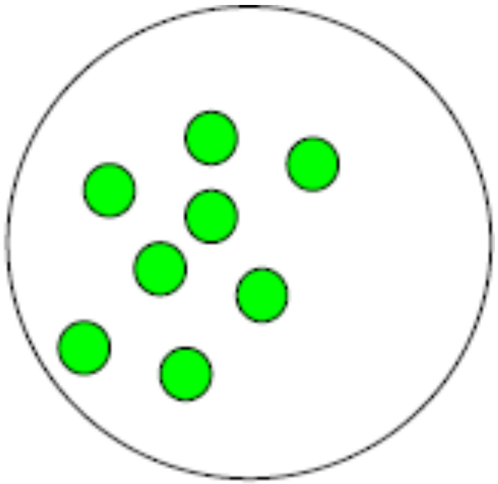




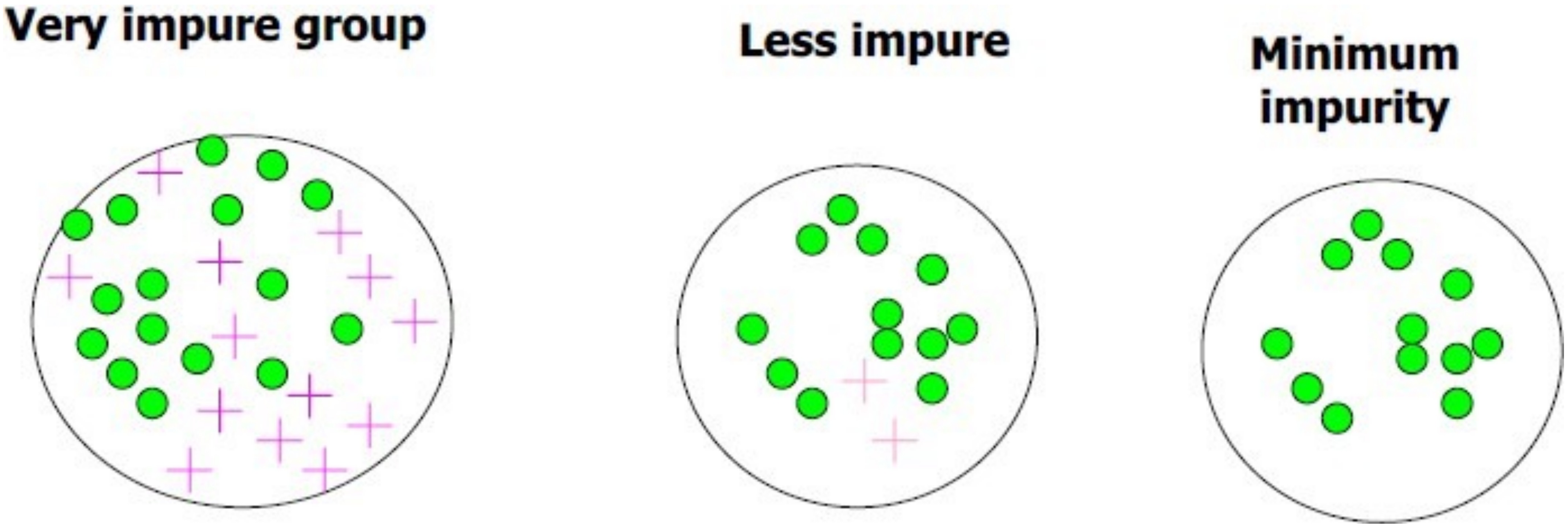
Impurity/Entropy

- Measures the level of **impurity** in a group of examples





Impurity



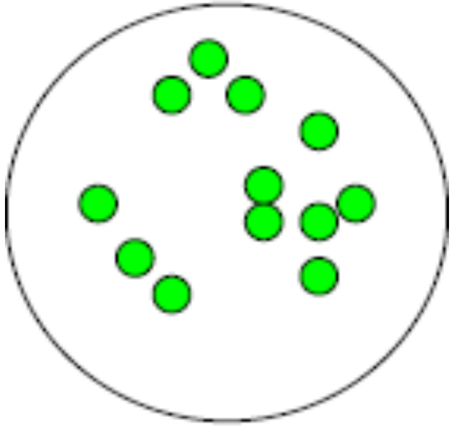
Entropy – a Common Way to Measure Impurity

- $Entropy = \Sigma_i - p_i \lg p_i$ where
  p_i is the probability of class i in a node.
- Entropy comes from information theory. The higher the entropy, the more the information content.
- Another measurement: Gini impurity $Gini = 1 - \Sigma_i p_i^2$

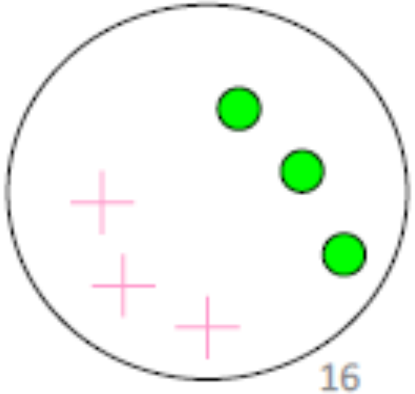
2-Class Case

- $Entropy(x) = -\Sigma_{i=1}^{2} p(x=i) \lg p(x=i)$
- What is the entropy of a group in which all examples belong to the same class?
  - $Entropy = -1 \lg 1 = 0$
  - Not a good training set for learning
- What is the entropy of a group with 50% of either class?
  - $Entropy = -0.5 \lg 0.5 - 0.5 \lg 0.5 = 1$
  - Good training set for learning

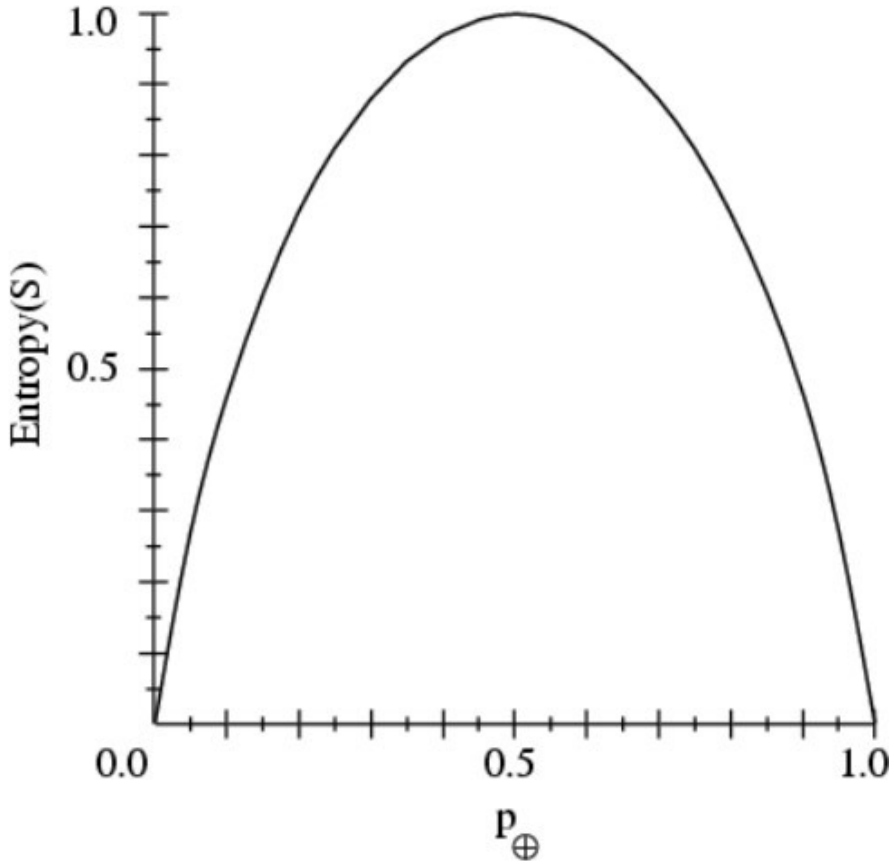
Minimum impurity



Maximum impurity



Sample Entropy



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

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.
- IG = Entropy(parent) – Weighted Sum of Entropy(children)

Basic Algorithm for Top-Down Learning of Decision Trees

ID3 (Iterative Dichotomiser 3, Ross Quinlan, 1986)

node = root of decision tree

Main loop:

1. A <- 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.

Question: How do we choose which attribute is best?

Choosing the Best Attribute

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

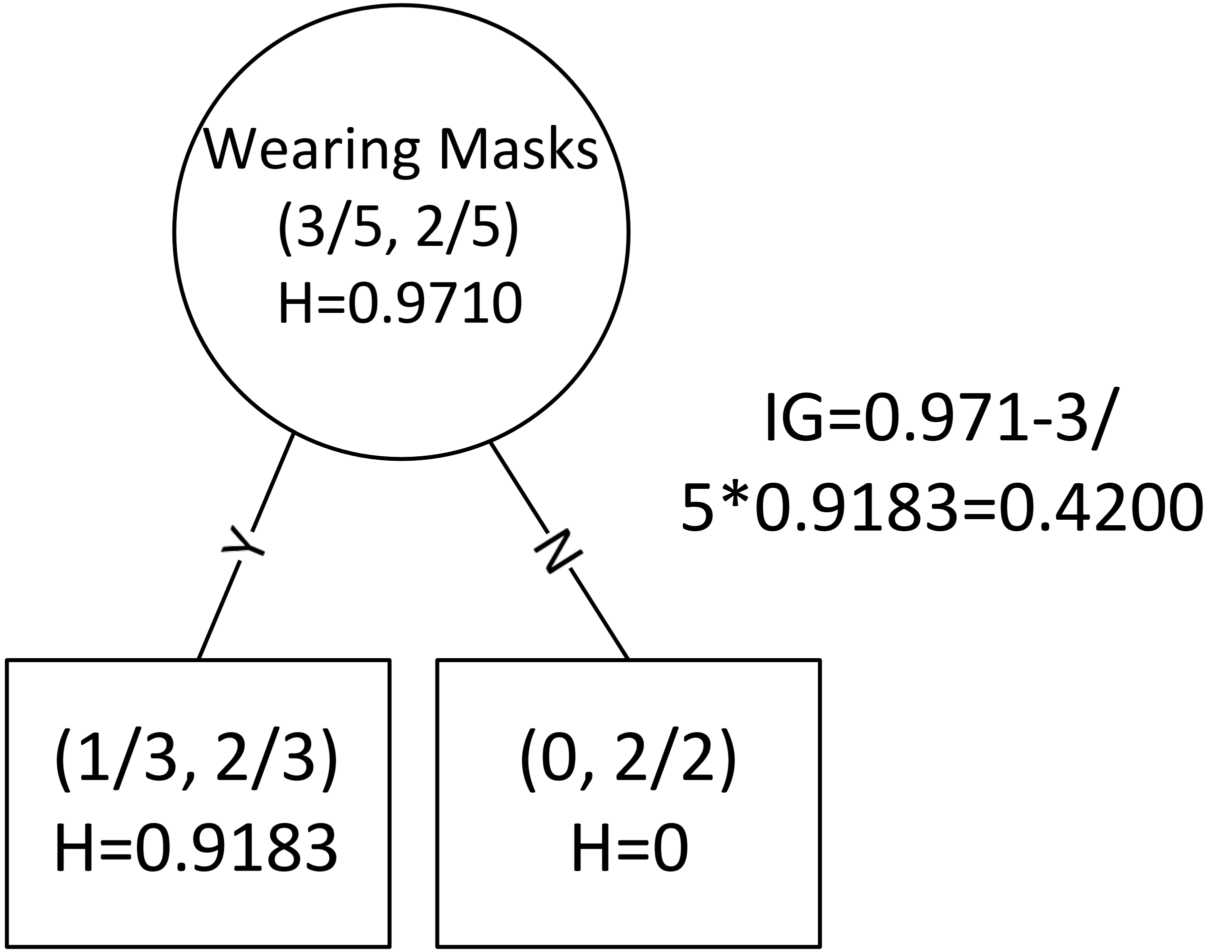
- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest 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

COVID-19 Example

Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N

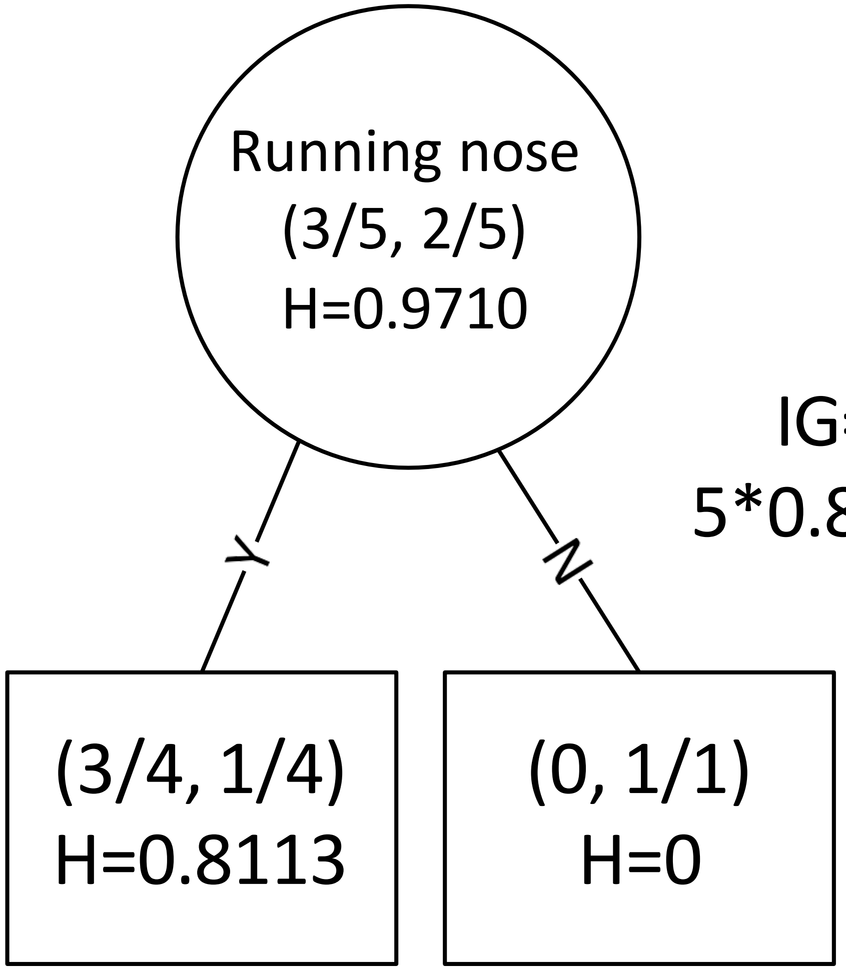


COVID-19 Example (Cont.)

Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N

COVID-19 Example (Cont.)

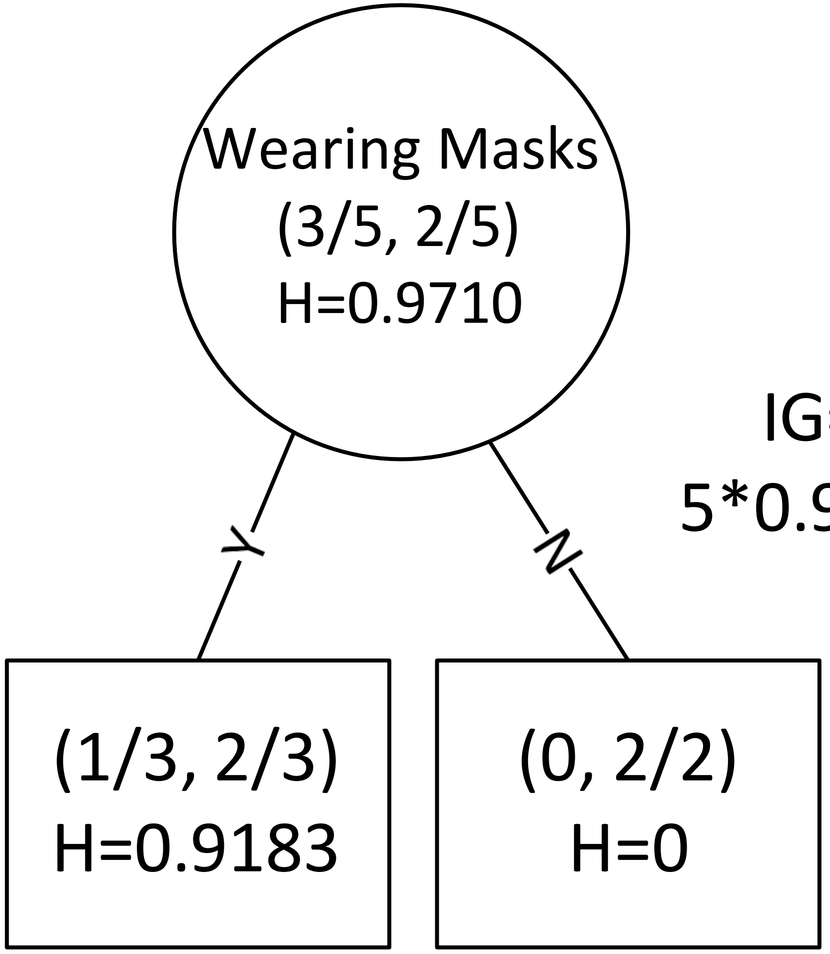
Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N



IG=0.971-4/5*0.8113=0.3220

COVID-19 Example (Pick Highest IG)

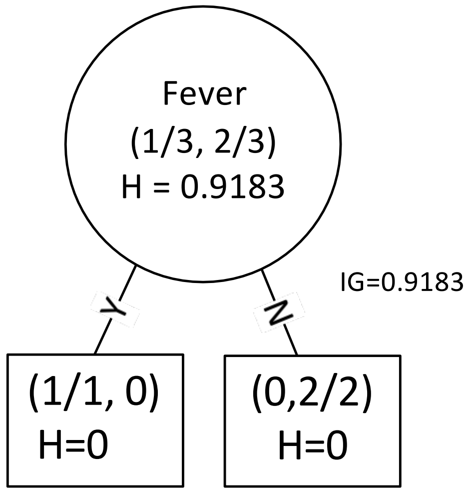
Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N

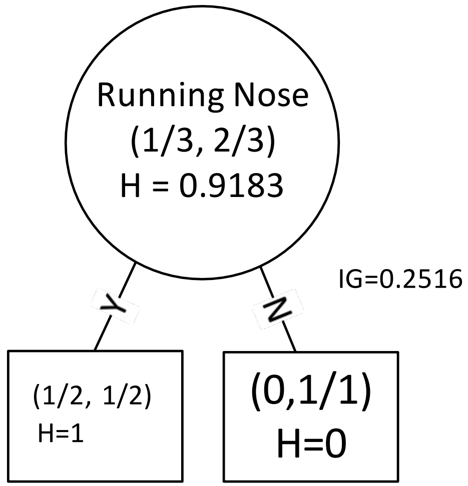


IG=0.971-3/5*0.9183=0.4200

COVID-19 Example (Expand Left Tree)

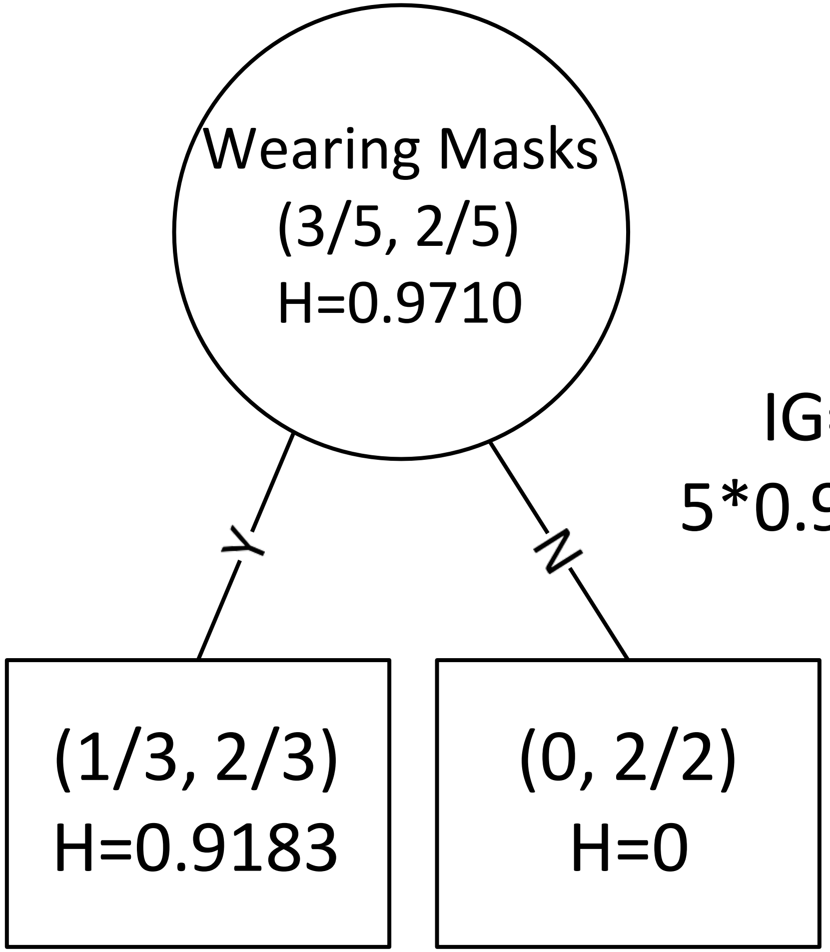
Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N





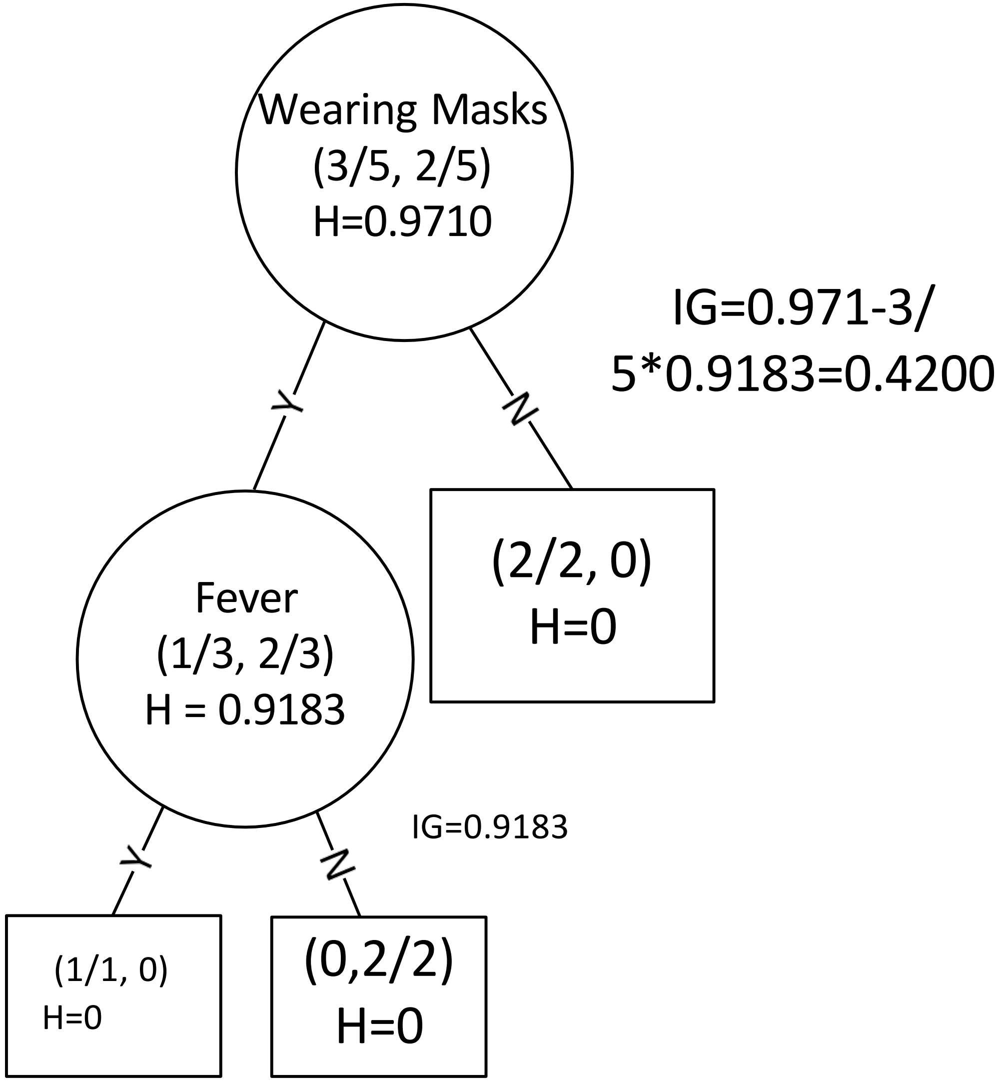
COVID-19 Example (Expand Right Tree?)

Wearing Masks	Fever	Running Nose	COVID-19
N	Y	Y	Y
N	N	Y	Y
Y	N	N	N
Y	Y	Y	Y
Y	N	Y	N



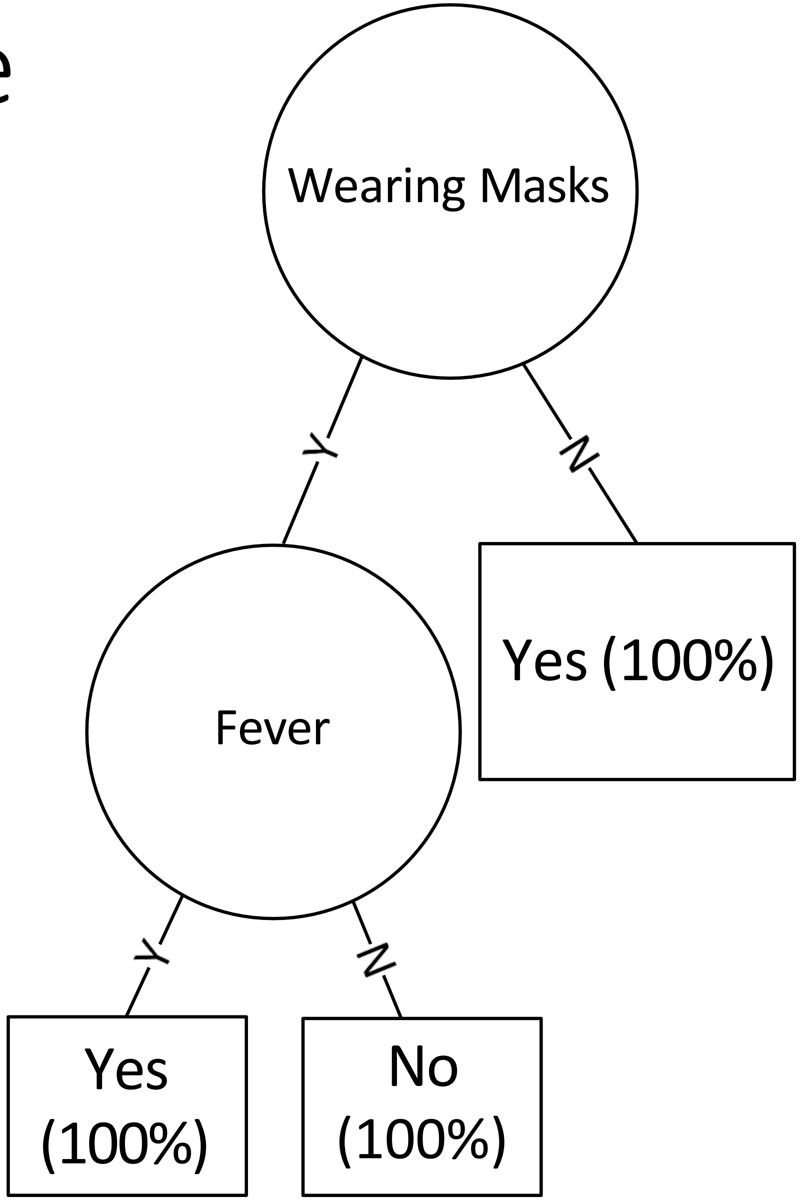
IG=0.971-3/5*0.9183=0.4200

COVID-19 Example (Decision Tree)



How to Use Decision Tree

- Fill in answers in leaf nodes
- Run test sample from root
- <wearing masks, fever, running nose>
  <N, Y, Y> ➔ Yes
  <Y, Y, N> ➔ Yes
  <Y, N, Y> ➔ No
- Not all attributes are used!



What if IG is negative?

- If IG is negative, that means children’s entropy is larger than their parent.
- I.e., adding children nodes do not get better classification.
- So stop growing nodes at that branch.
- This is one way of true pruning.

Pruning Tree

- Decision may grow fast, which we don’t like!
- It may cause overfitting by noise including incorrect attributes or class membership.
- Large decision trees requires lots of memory and may not be deployed in resource limited devices.
- Decision tree may not capture features in the training set.
- It is hard to tell if a single extra node will increase accuracy, so called the horizon effect.
- One way to prune trees is set an IG threshold to keep subtrees.
- i.e., IG has to be greater than the threshold to grow the tree;
- Another way is simply set the tree depth or set the max bin count.

How About Numeric Attributes

- IN the COVID-19 example, we only have Yes/No attributes, what if we have a person's weight?
- We could sort the weight. Find the average of two adjacent values. Calculate entropy of each $W < w_i$. Pick the one with lowest entropy.
- For ranked data, like rank 1-4 for a question. Or categorical data, like low, medium, and high. We may simply encode them as ordinals. Calculate entropies for each $\mathrm{R} < r_i$. Pick the one with lowest entropy.
- For non-sequential numeric data, like red, green, and blue. We may enumerate all possible combinations and calculate their entropies such as {C=red},{C=green}, {C=blue},{C=red, green}, {C=red, blue}, {C=green, blue}.
- Remember our goal is to split data. So we don't consider any split criteria that do not separate data like {C=red, green, blue}