Decision Tree

Sudipta Bhattacharya

DECISION TREE CLASSIFICATION

- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

HOW DO WE CONSTRUCT THE DECISION TREE?

- Basic algorithm
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they can be discretized in advance)
  - Examples are partitioned recursively based on selected attributes.
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

INFORMATION GAIN AS A SPLITTING CRITERIA

- Select the attribute with the highest information gain (information gain is the expected reduction in entropy).
- Assume there are two classes, P and N
  - Let the set of examples $\boldsymbol{S}$ contain p elements of class P and n elements of class N
  - The amount of information, needed to decide if an arbitrary example in S belongs to P or N is defined as

$$E(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

0 log(0) is defined as 0

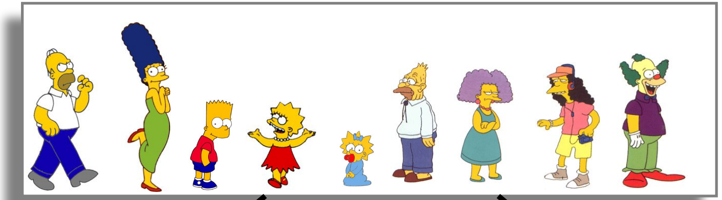
INFORMATION GAIN IN DECISION TREE INDUCTION

- Assume that using attribute A, a current set will be partitioned into some number of child sets
- The encoding information that would be gained by branching on *A*

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Note: entropy is at its minimum if the collection of objects is completely uniform

Person		Hair Length	Weight	Age	Class
	Homer	0”	250	36	**M**
	Marge	10”	150	34	**F**
	Bart	2”	90	10	**M**
	Lisa	6”	78	8	**F**
	Maggie	4”	20	1	**F**
	Abe	1”	170	70	**M**
	Selma	8”	160	41	**F**
	Otto	10”	180	38	**M**
	Krusty	6”	200	45	**M**
	Comic	8”	290	38	**?**



$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

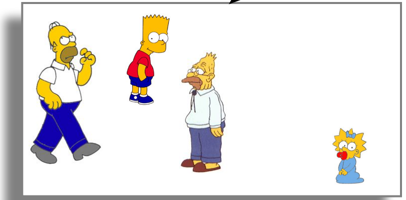
Entropy(4**F**,5**M**) = $-(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$

= **0.9911**

yes

no

Hair Length <= 5?



Let us try splitting on *Hair length*

Entropy(1**F**,3**M**) = $-(1/4)\log_2(1/4) - (3/4)\log_2(3/4)$

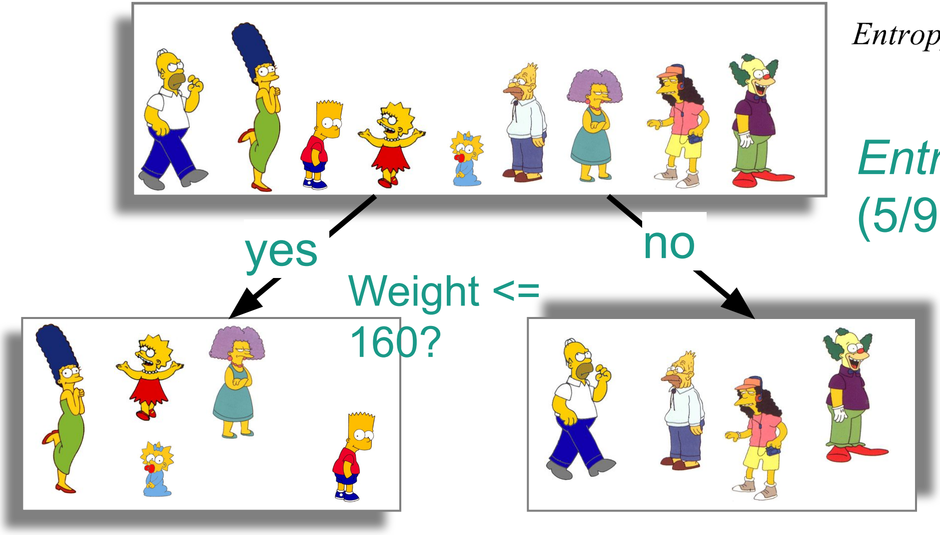
= ***0.8113***

Entropy(3**F**,2**M**) = $-(3/5)\log_2(3/5) - (2/5)\log_2(2/5)$

= ***0.9710***

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Gain(Hair Length <= 5) = **0.9911** – (4/9 * **0.8113** + 5/9 * **0.9710**) = **0.0911**



$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

Entropy(4**F**,5**M**) = -(4/9)log$_2$(4/9) - (5/9)log$_2$(5/9)

= **0.9911**

Let us try splitting on *Weight*

Entropy(4**F**,1**M**) = -(4/5)log$_2$(4/5) - (1/5)log$_2$(1/5)

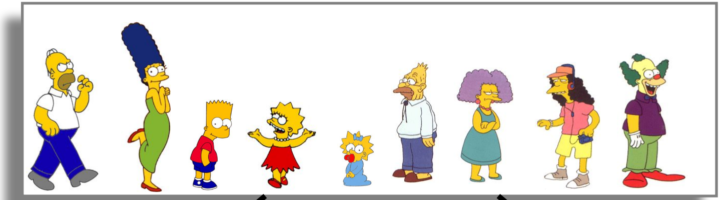
= **0.7219**

Entropy(0**F**,4**M**) = -(0/4)log$_2$(0/4) - (4/4)log$_2$(4/4)

= ***0***

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Gain(Weight <= 160) = **0.9911** – (5/9 * **0.7219** + 4/9 * **0**) = **0.5900**



$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

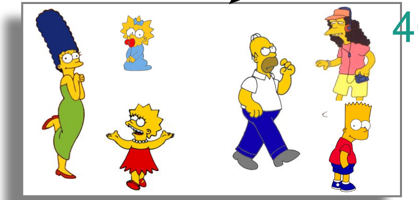
$Entropy(4\mathbf{F},5\mathbf{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$

$= \mathbf{0.9911}$

yes

age <= 40?

no



Let us try splitting on *Age*

$Entropy(3\mathbf{F},3\mathbf{M}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6)$

$= \mathbf{1}$

$Entropy(1\mathbf{F},2\mathbf{M}) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3)$

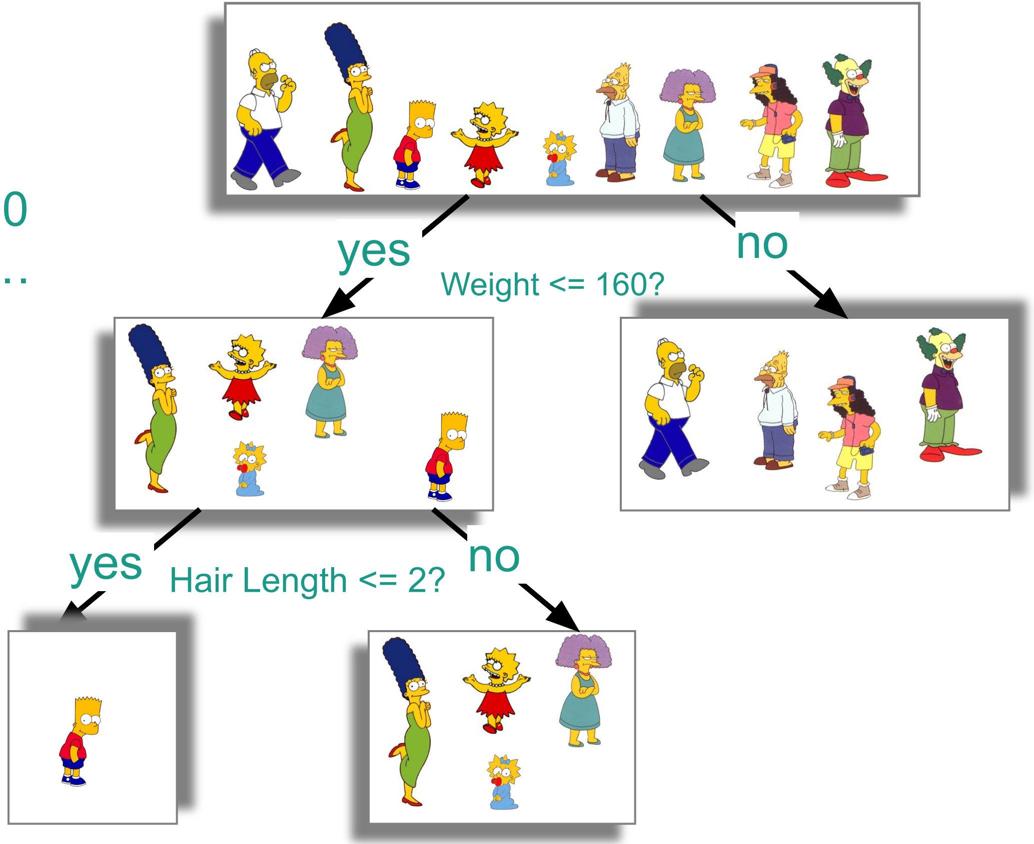
$= \mathbf{0.9183}$

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

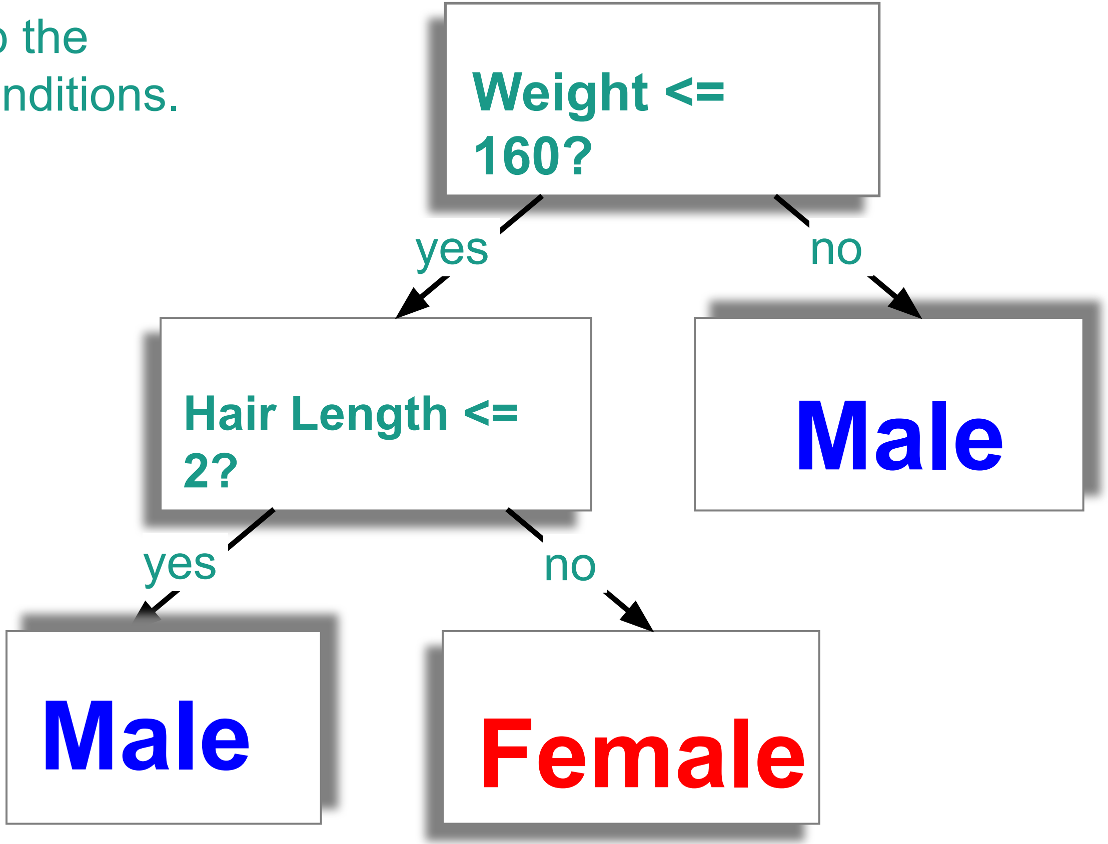
$Gain(Age <= 40) = \mathbf{0.9911} - (6/9 * \mathbf{1} + 3/9 * \mathbf{0.9183}) = \mathbf{0.0183}$

Of the 3 features we had, *Weight* was best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified… So we simply recurse!

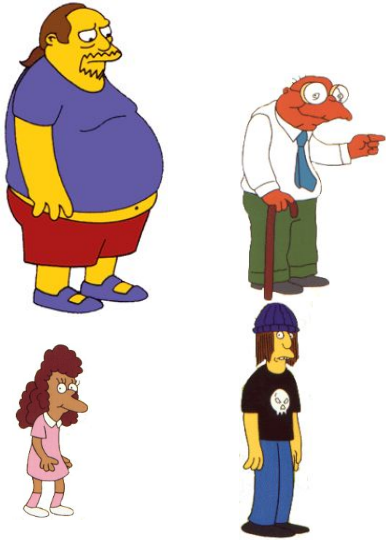
This time we find that we can split on *Hair length,* and we are done!



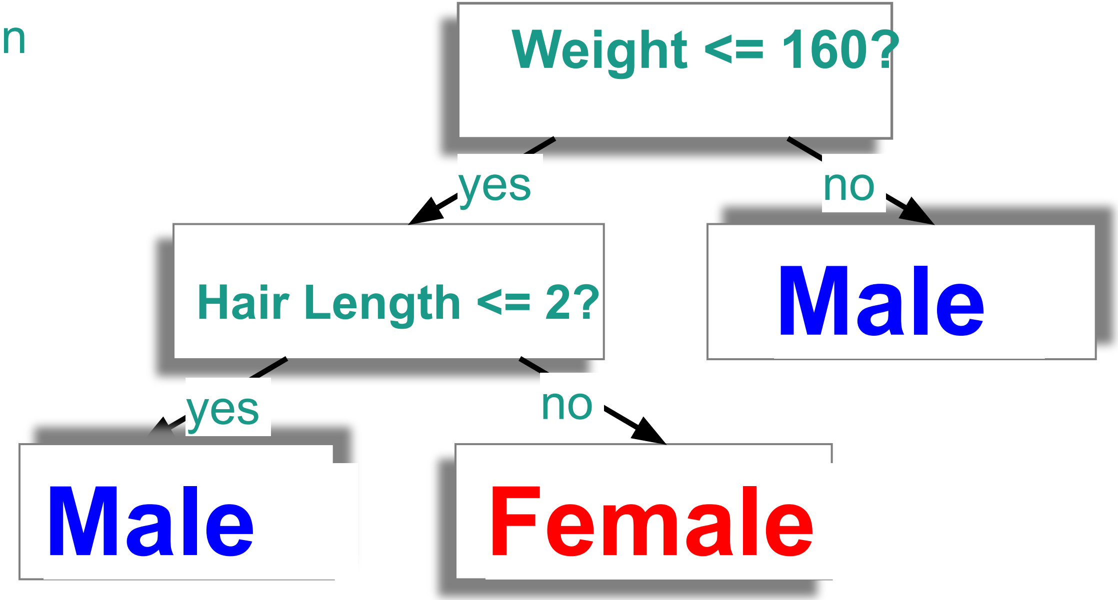
We need don’t need to keep the data around, just the test conditions.



How would these people be classified?



It is trivial to convert Decision Trees to rules…



Rules to Classify Males/Females

If *Weight* **greater than** 160, classify as **Male**
Elseif *Hair Length* **less than or equal** to 2, classify as **Male**
Else classify as **Female**