ID3 Algorithm

Allan Neymark

CS157B – Spring 2007

Agenda

- **Decision Trees**
- **What is ID3?**
- **Entropy**
- **Calculating Entropy with Code**
- **Information Gain**
- **Advantages and Disadvantages**
- **Example**

Decision Trees

- Rules for classifying data using attributes.
- The tree consists of decision nodes and leaf nodes.
- A decision node has two or more branches, each representing values for the attribute tested.
- A leaf node attribute produces a homogeneous result (all in one class), which does not require additional classification testing.

Decision Tree Example

Outlook

sunny

overcast

rain

Humidity

Yes

Windy

high

normal

true

false

No

Yes

No

Yes
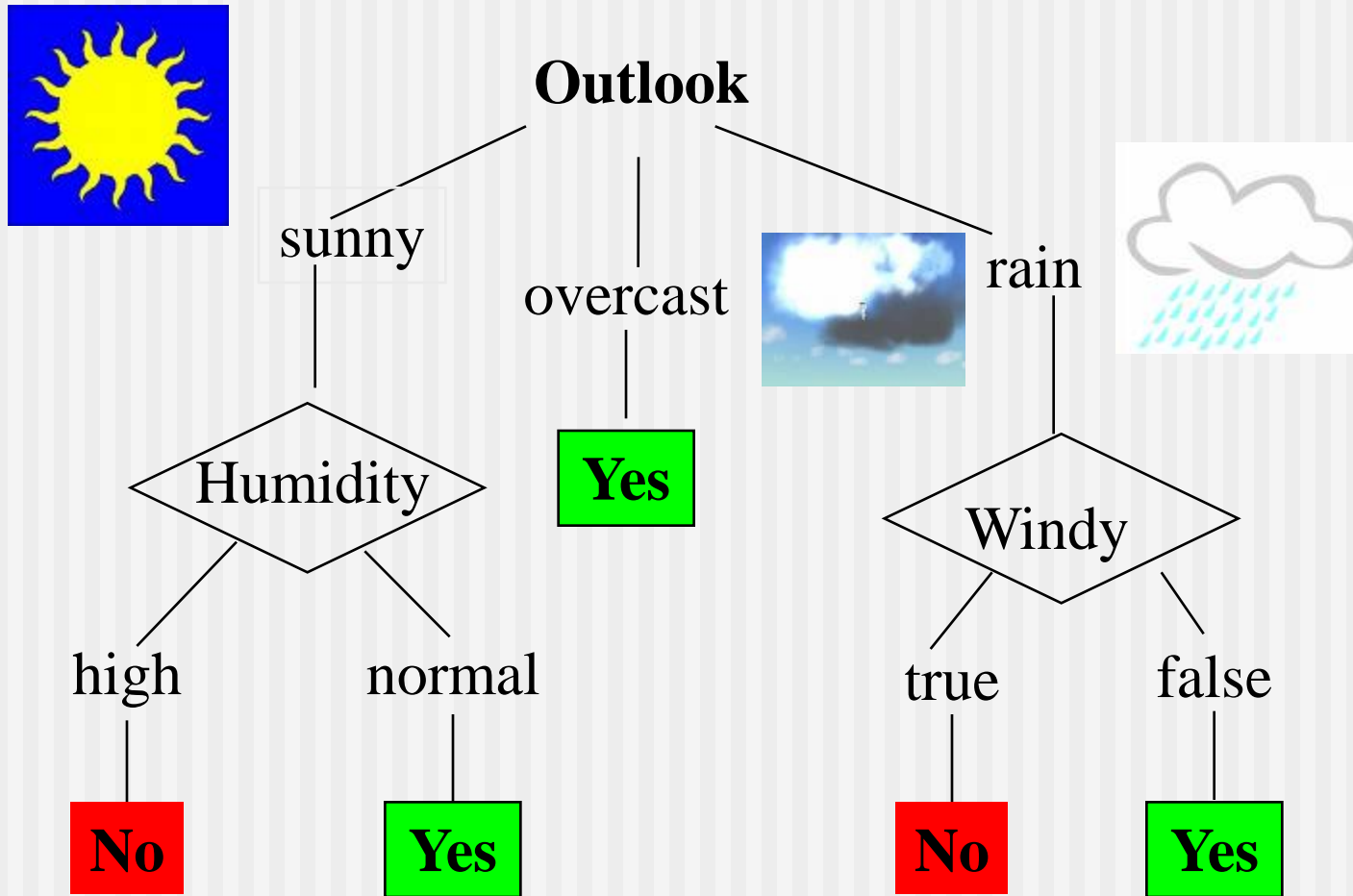
What is ID3?

- A mathematical algorithm for building the decision tree.
- Invented by J. Ross Quinlan in 1979.
- Uses Information Theory invented by Shannon in 1948.
- Builds the tree from the top down, with no backtracking.
- Information Gain is used to select the most useful attribute for classification.

Entropy

- A formula to calculate the homogeneity of a sample.
- A completely homogeneous sample has entropy of 0.
- An equally divided sample has entropy of 1.
- Entropy(s) = - p+log2 (p+) -p-log2 (p-) for a sample of negative and positive elements.
- The formula for entropy is:

$$Entropy(S) = \sum_{i=1}^{c} p_i \log_2 p_i$$

Entropy Example

Entropy(S) =

    - (9/14) Log2 (9/14) - (5/14) Log2 (5/14)

    = 0.940

Calculating Entropy with Code

- Most programming languages and calculators do not have a log2 function.

- Use a conversion factor

- Take log function of 2, and divide by it.

- Example: log10(2) = .301

- Then divide to get log2(n):

- log10(3/5) / .301 = log2(3/5)

Calculating Entropy with Code (cont'd)

- Taking log10(0) produces an error.
- Substitute 0 for (0/3)log10(0/3)
- Do not try to calculate log10(0/3)

Information Gain (IG)

- The information gain is based on the decrease in entropy after a dataset is split on an attribute.
- Which attribute creates the most homogeneous branches?
- First the entropy of the total dataset is calculated.
- The dataset is then split on the different attributes.
- The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split.
- The resulting entropy is subtracted from the entropy before the split.
- The result is the Information Gain, or decrease in entropy.
- The attribute that yields the largest IG is chosen for the decision node.

Information Gain (cont'd)

- A branch set with entropy of 0 is a leaf node.
- Otherwise, the branch needs further splitting to classify its dataset.
- The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Advantages of using ID3

- Understandable prediction rules are created from the training data.
- Builds the fastest tree.
- Builds a short tree.
- Only need to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing number of tests.
- Whole dataset is searched to create tree.

Disadvantages of using ID3

- Data may be over-fitted or over-classified, if a small sample is tested.

- Only one attribute at a time is tested for making a decision.

- Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

Example: The Simpsons

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**

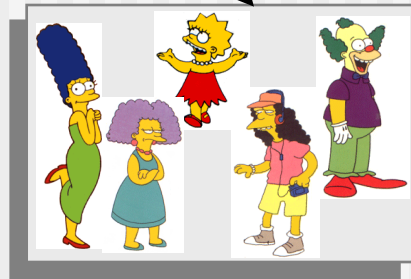|	Comic	8"	290	38	**?**
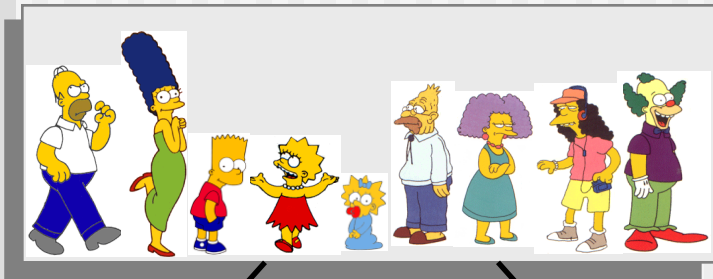
$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

$$Entropy(4\textbf{F},5\textbf{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$$
$$= \textbf{0.9911}$$
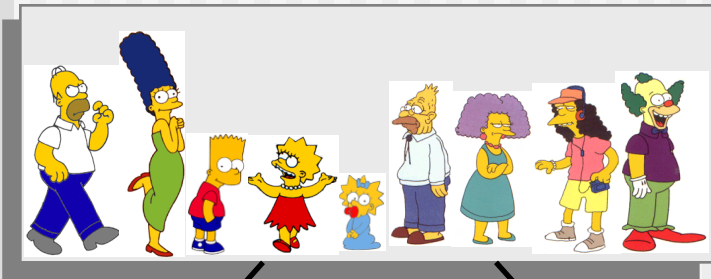
yes   no

Hair Length <= 5?

Let us try splitting on *Hair length*

$$Entropy(1\textbf{F},3\textbf{M}) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4)$$
$$= \textbf{0.8113}$$

$$Entropy(3\textbf{F},2\textbf{M}) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5)$$
$$= \textbf{0.9710}$$

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

$$Gain(\text{Hair Length} <= 5) = \textbf{0.9911} - (4/9 * \textbf{0.8113} + 5/9 * \textbf{0.9710}) = \textbf{0.0911}$$
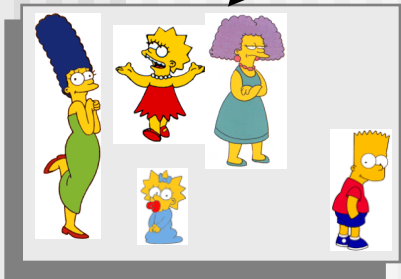
$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

$$Entropy(4F,5M) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$$
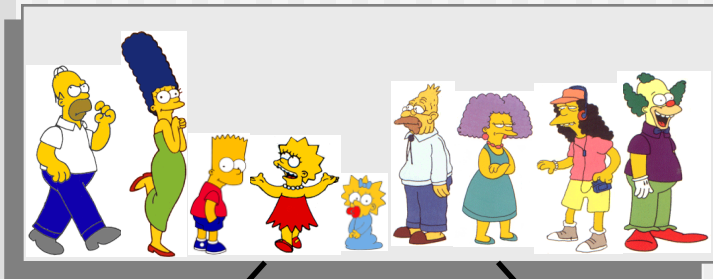$$= 0.9911$$

yes

no

Weight <= 160?

Let us try splitting on *Weight*

$$Entropy(4F,1M) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5)$$
$$= 0.7219$$

$$Entropy(0F,4M) = -(0/4)\log_2(0/4) - (4/4)\log_2(4/4)$$
$$= 0$$

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

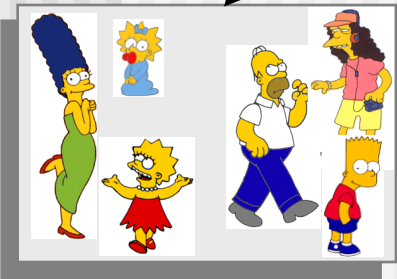$$Gain(Weight <= 160) = 0.9911 - (5/9 * 0.7219 + 4/9 * 0) = 0.5900$$

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

$$Entropy(4\textbf{F},5\textbf{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$$
$$= \textbf{0.9911}$$

yes   no

age <= 40?

Let us try splitting on *Age*

$$Entropy(3\textbf{F},3\textbf{M}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6)$$
$$= \textbf{1}$$

$$Entropy(1\textbf{F},2\textbf{M}) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3)$$
$$= \textbf{0.9183}$$

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

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

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!

yes

no

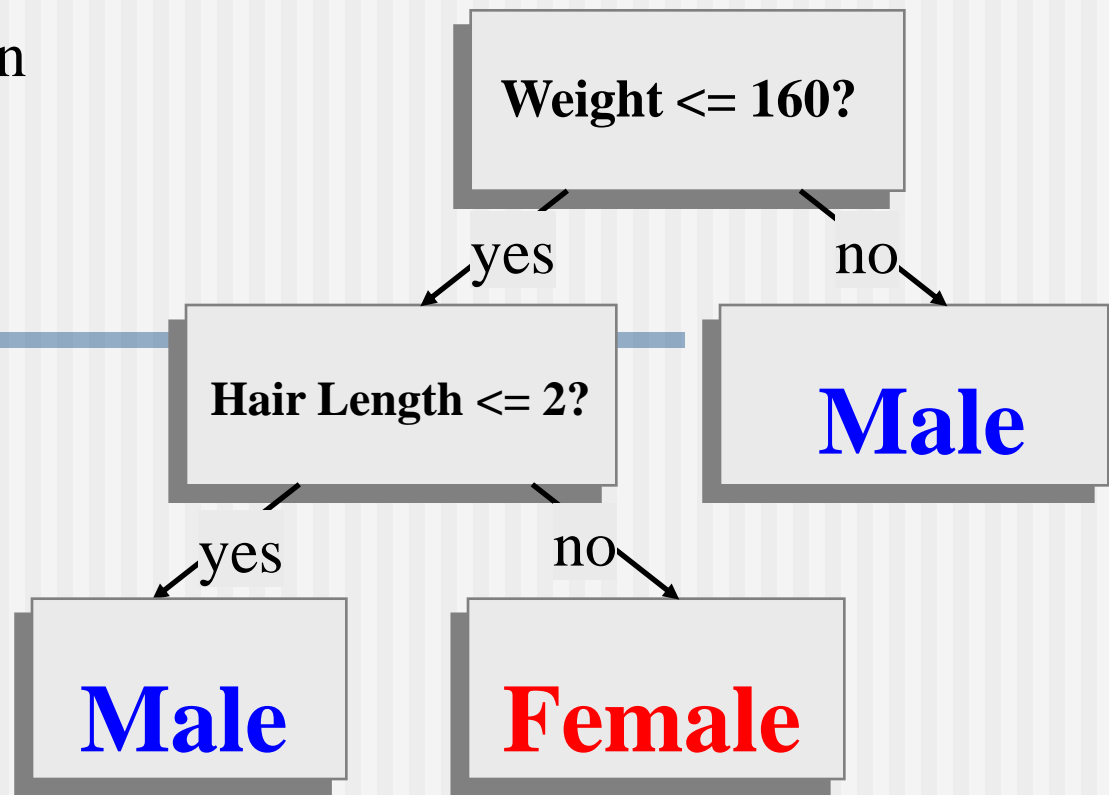Weight <= 160?

yes

no

Hair Length <= 2?
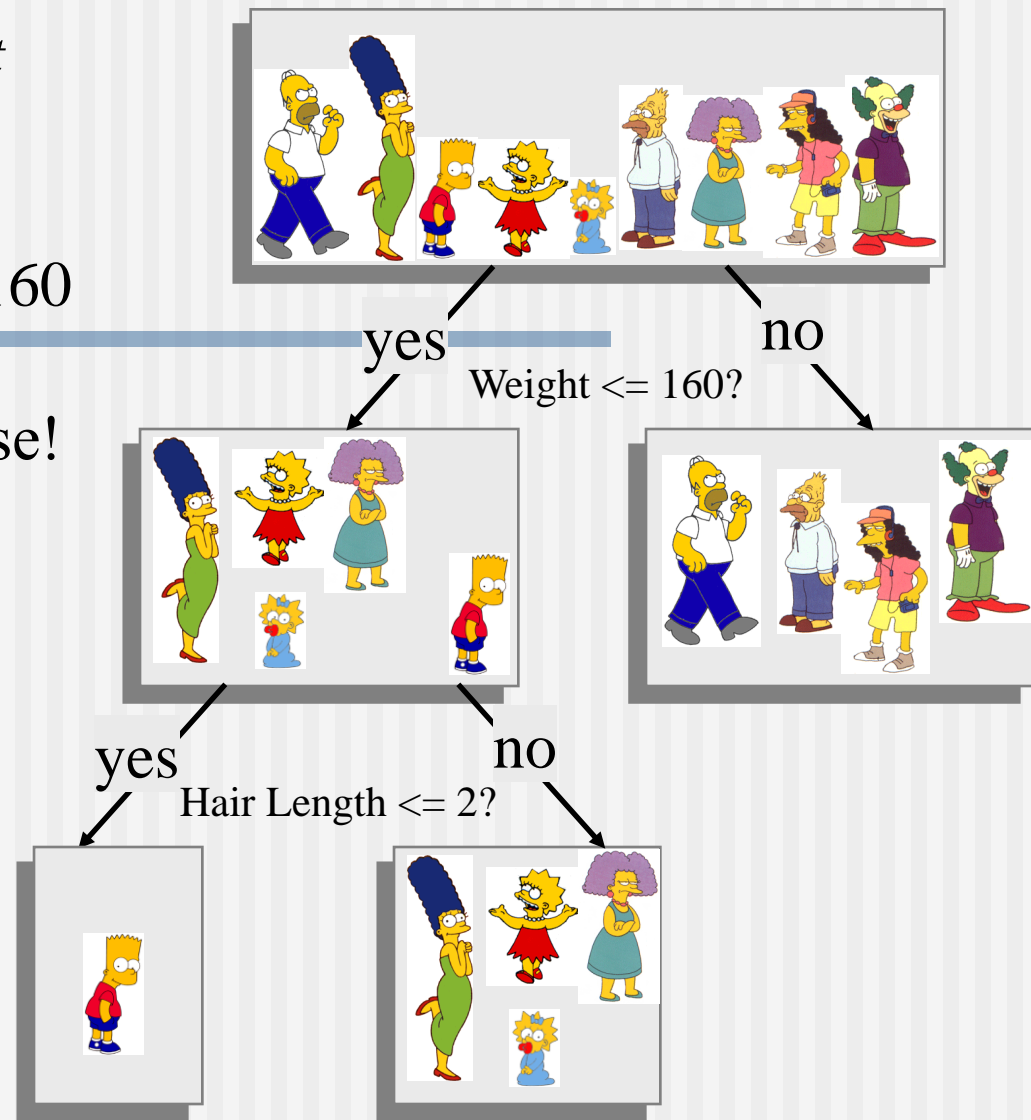
We need don't need to keep the data around, just the test conditions.

How would these people be classified?

Weight <= 160?

yes — no

Hair Length <= 2?

Male

yes — no

Male   **Female**
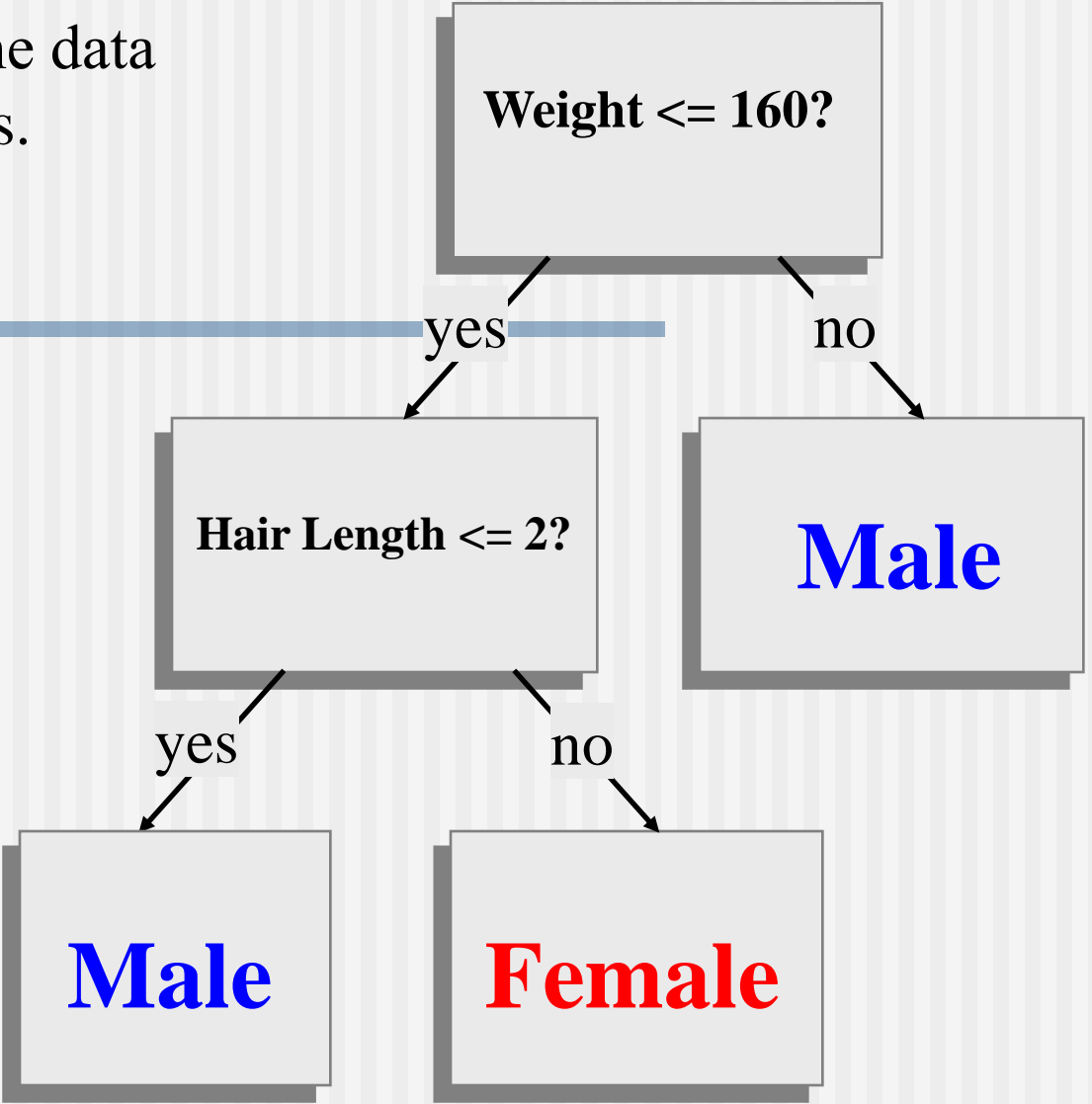
It is trivial to convert Decision Trees to rules…

Weight <= 160?

yes — no

Hair Length <= 2?

Male

yes — no

Male

Female

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**

References

- Quinlan, J.R. 1986, Machine Learning, 1, 81

- http://dms.irb.hr/tutorial/tut_dtrees.php

- http://www.dcs.napier.ac.uk/~peter/vldb/dm/node11.html

- http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/4_dtrees2.html

- Professor Sin-Min Lee, SJSU.
  http://cs.sjsu.edu/~lee/cs157b/cs157b.html