# Decision Trees – part 2


"I know you're a new analyst but there's no need to hide from our clients."

"I'm not hiding! I'm trying a new technique I learned on the web: The Decision Tree".

# Announcements

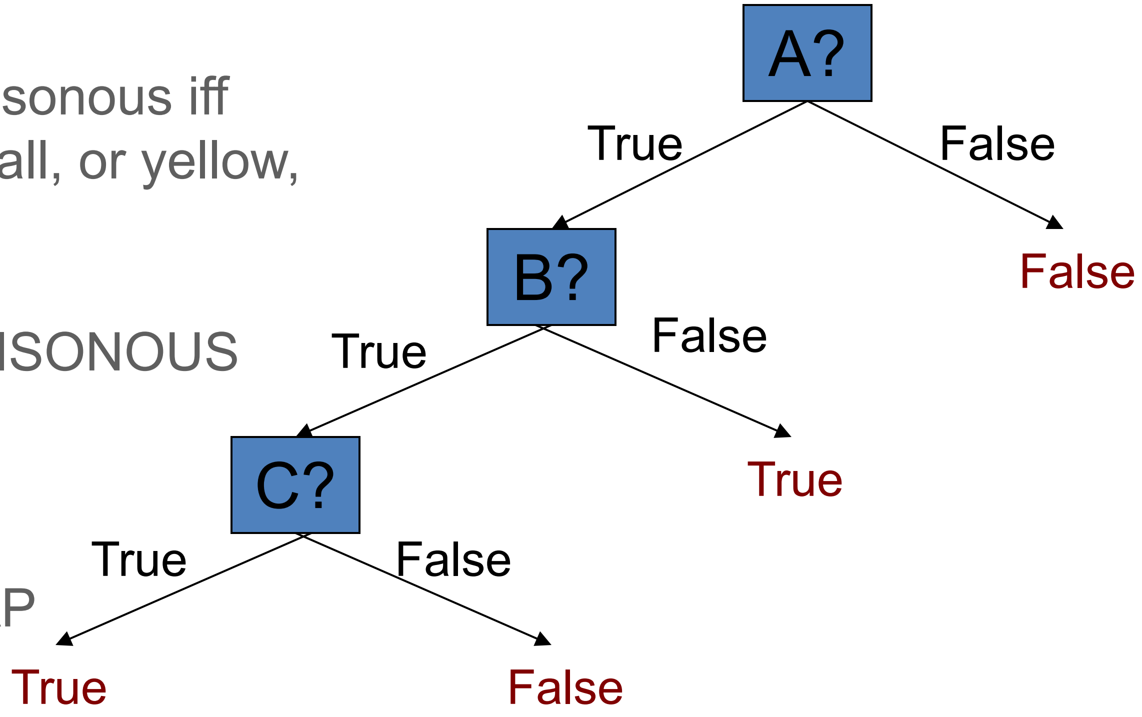
- A2 due November 7 (make sure you are using the latest pdf – there was a change in the first paragraph of part 1, updated on October 25)
- A3 and optional A4 to follow (A4 will have a deadline during the last week of classes)
- Regrade requests are being processed – if it’s been more than a week since you submitted your request, you can follow up on Q&A

# Predicate as a Decision Tree

The predicate CONCEPT(x) $\Leftrightarrow$ A(x) $\wedge$ ($\neg$B(x) v C(x)) can be represented by the following decision tree:

Example:
A mushroom is poisonous iff it is yellow and small, or yellow, big and spotted

- x is a mushroom
- CONCEPT = POISONOUS
- A = YELLOW
- B = BIG
- C = SPOTTED
- D = FUNNEL-CAP
- E = BULKY



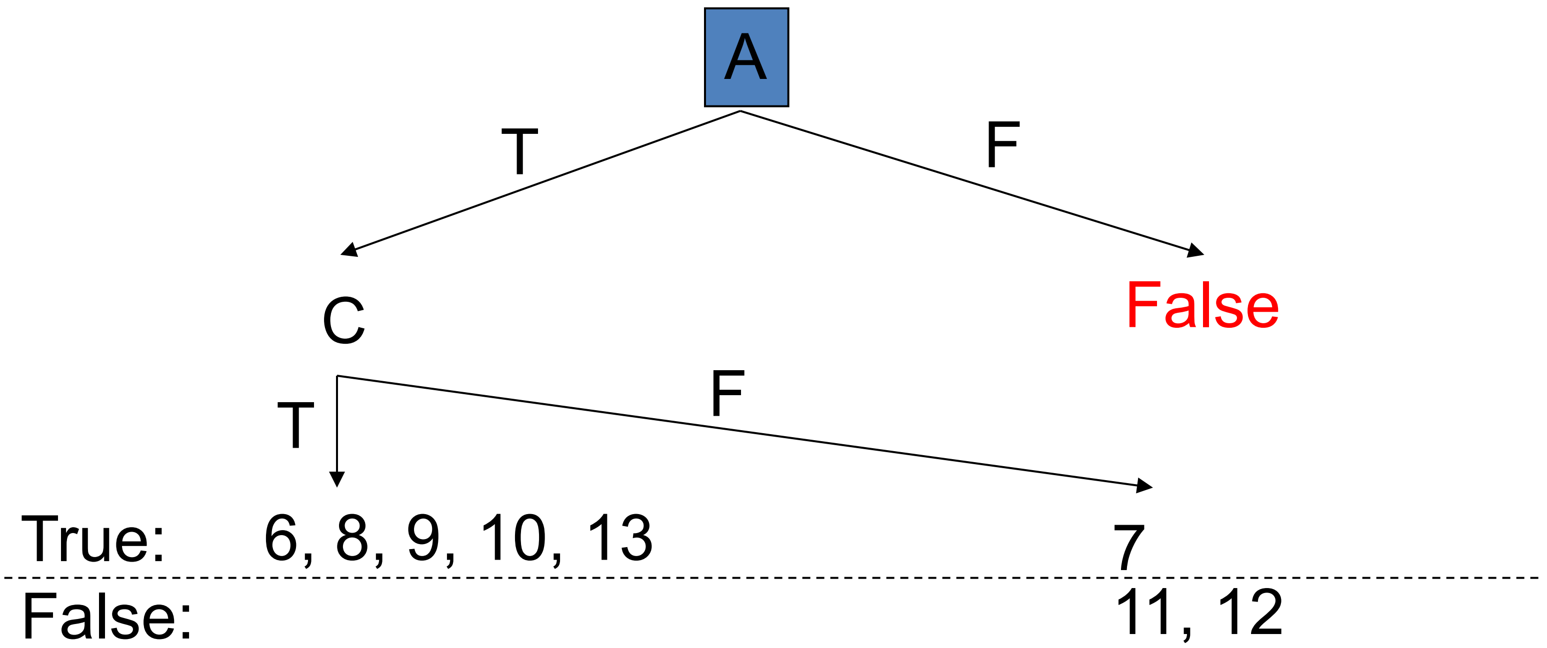
# Top-down induction of decision tree

| Ex. # | A | B | C | D | E | CONCEPT |
|---|---|---|---|---|---|---|
| 1 | False | False | True | False | True | False |
| 2 | False | True | False | False | False | False |
| 3 | False | True | True | True | True | False |
| 4 | False | False | True | False | False | False |
| 5 | False | False | False | True | True | False |
| 6 | True | False | True | False | False | True |
| 7 | True | False | False | True | False | True |
| 8 | True | False | True | False | True | True |
| 9 | True | True | True | False | True | True |
| 10 | True | True | True | True | True | True |
| 11 | True | True | False | False | False | False |
| 12 | True | True | False | False | True | False |
| 13 | True | False | True | True | True | True |

DTL(D, Predicates)

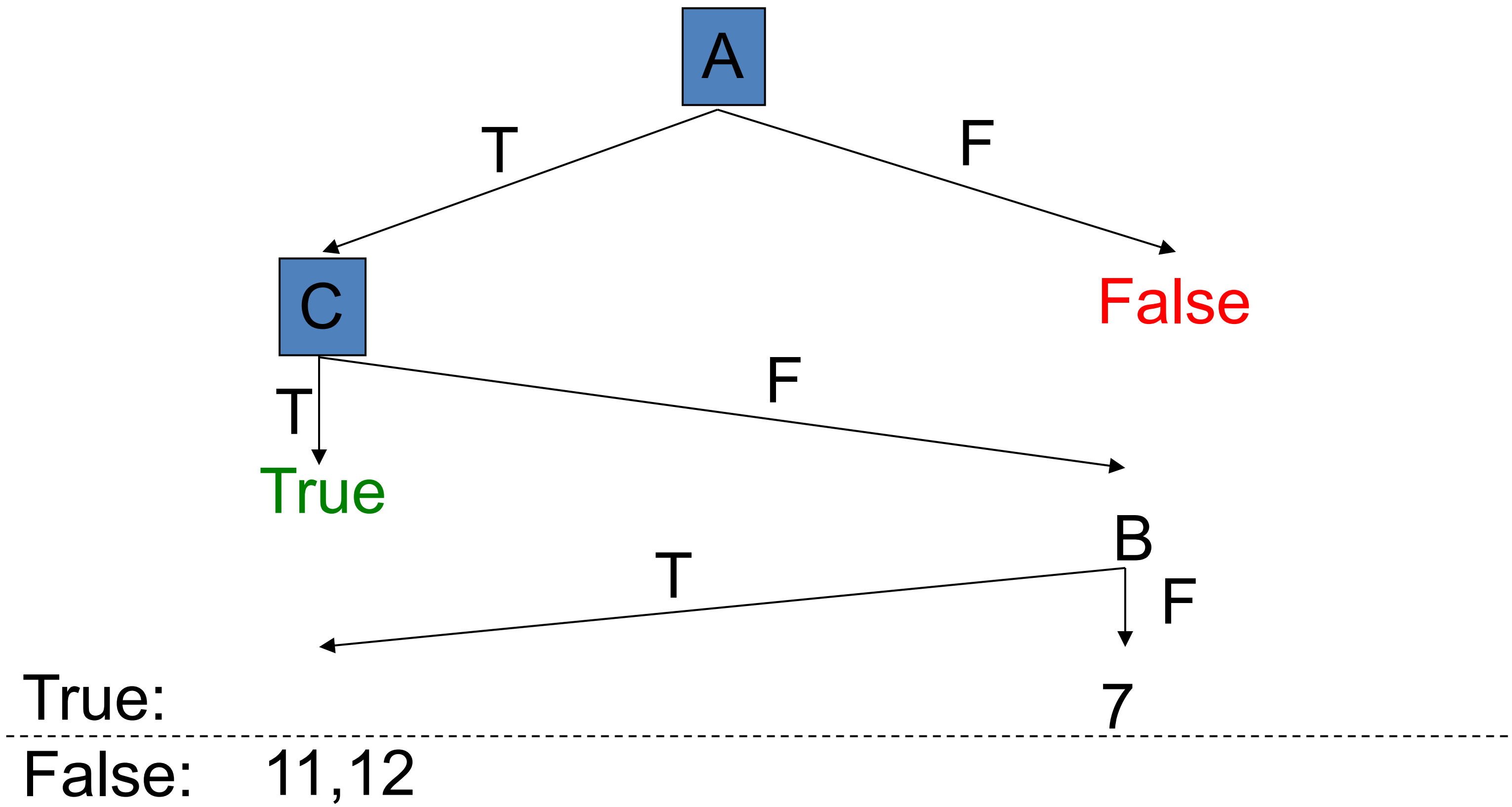
1. If all examples in D are positive then return True
2. If all examples in D are negative then return False
3. If Predicates is empty then return *failure*
4. A ← **error-minimizing** predicate in Predicates
5. Return the tree whose:
   - root is A,
   - left branch is DTL($D^{+A}$,Predicates-A),
   - right branch is DTL($D^{-A}$,Predicates-A)

# Choice of Second Predicate

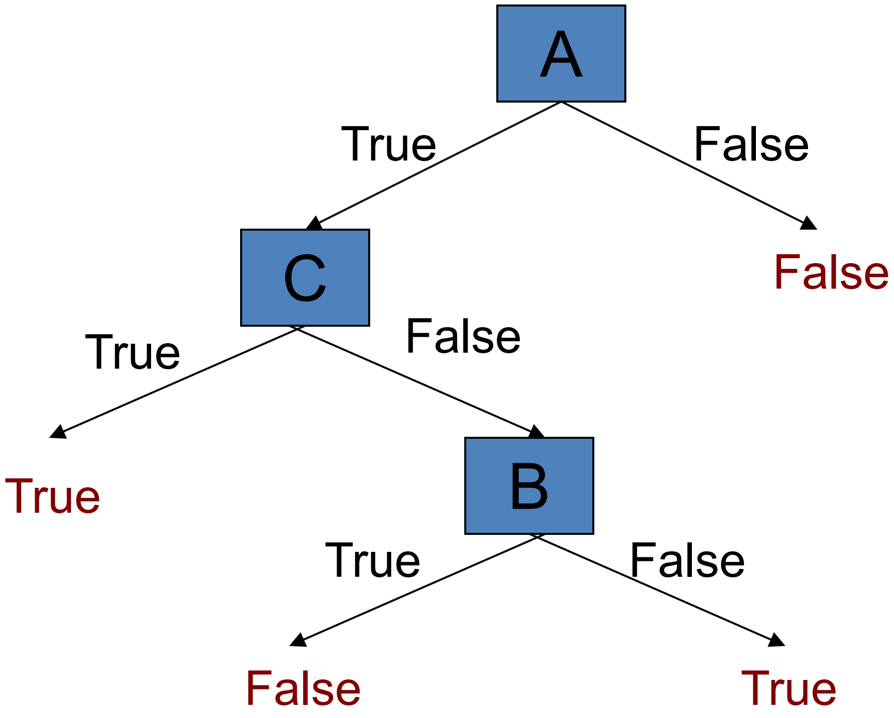


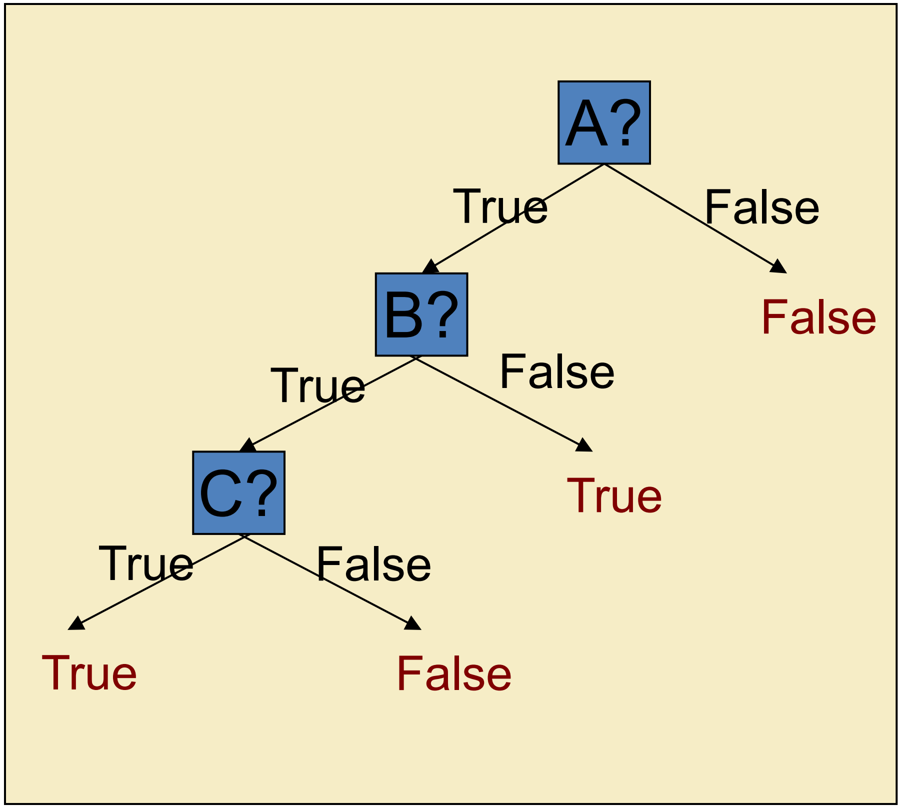

→ The number of misclassified examples from the training set is 1

# Choice of Third Predicate



# Final Tree




CONCEPT $\Leftrightarrow$ A $\wedge$ (C v $\neg$B)

CONCEPT $\Leftrightarrow$ A $\wedge$ ($\neg$B v C)

# Generalizing decision trees

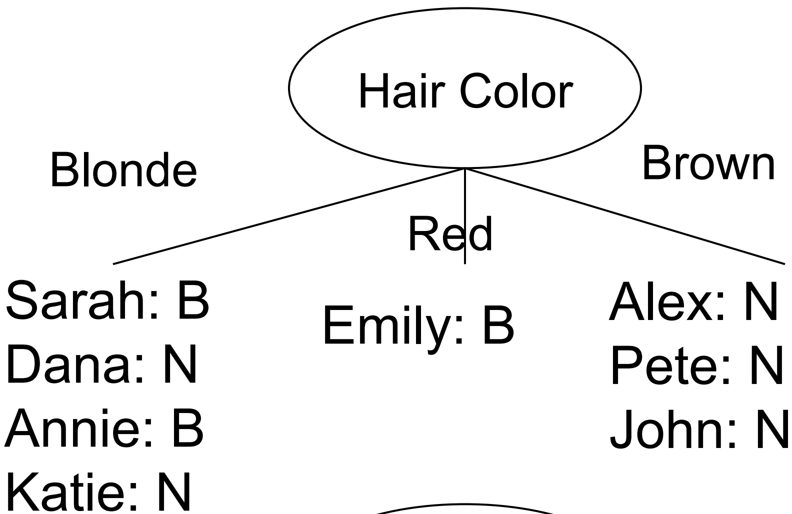
| Name | Hair | Height | Weight | Lotion | Result |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |

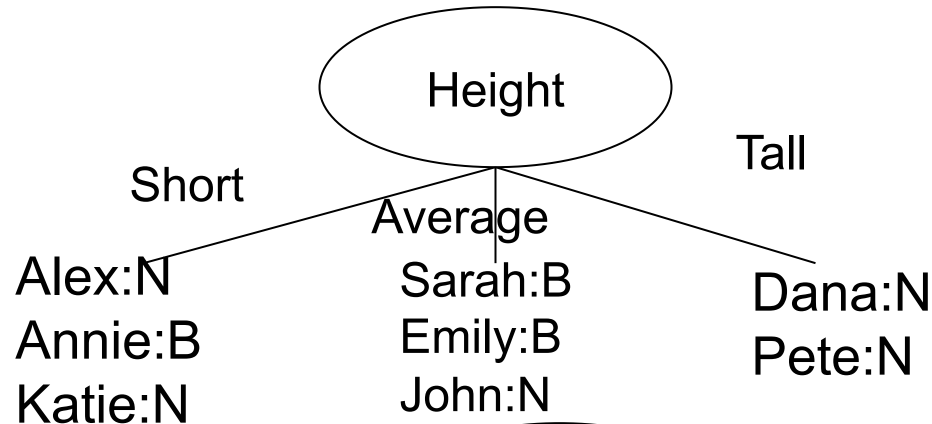
# Building Decision Trees

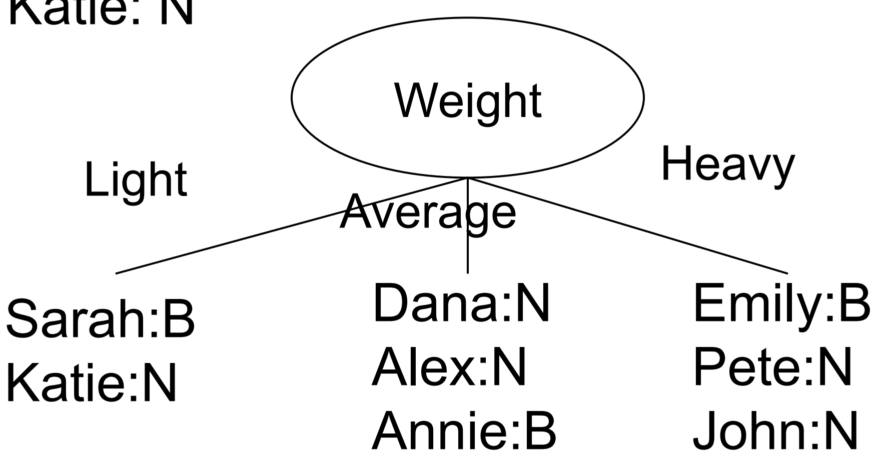
- Goal: Build a small tree such that all samples at leaves have same class
- Greedy solution:
  - At each node, pick test such that branches are closest to having same class
    - Split into subsets with least “disorder”
      - (Disorder ~ Entropy)
  - Find test that minimizes disorder

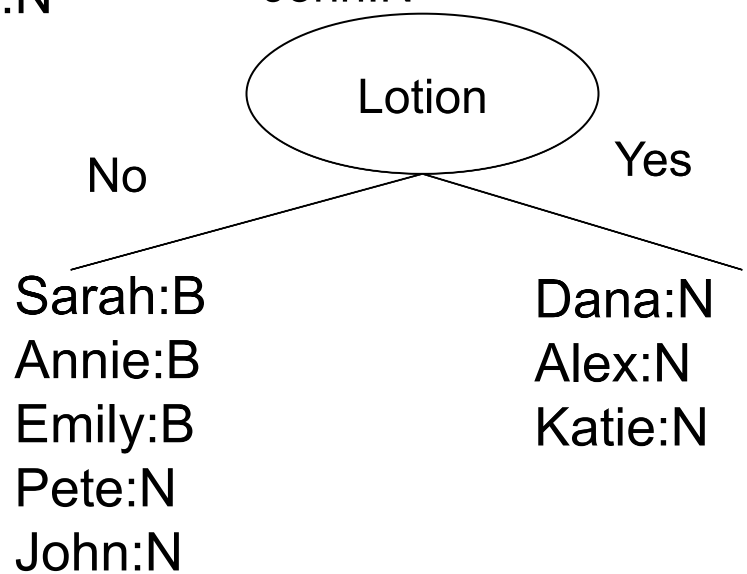
# Minimizing Disorder

| Name | Hair | Height | Weight | Lotion | Result |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |



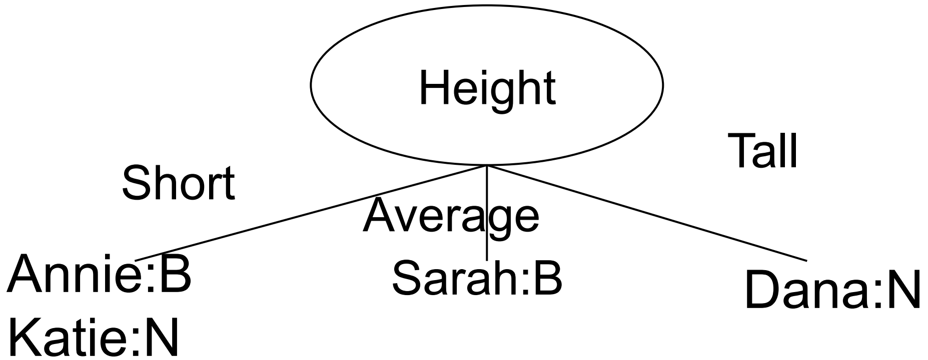



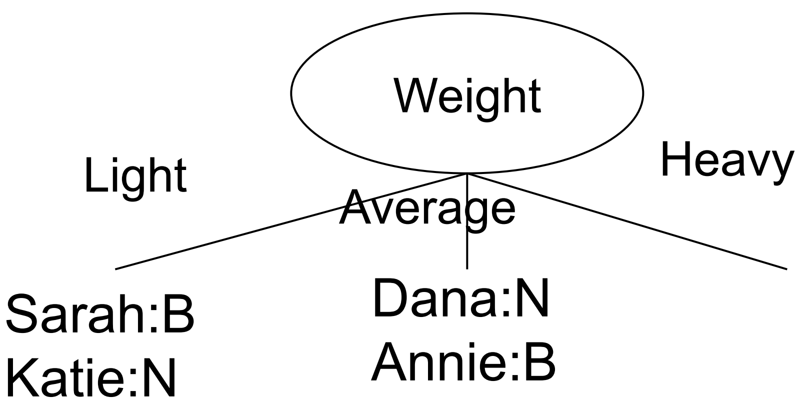



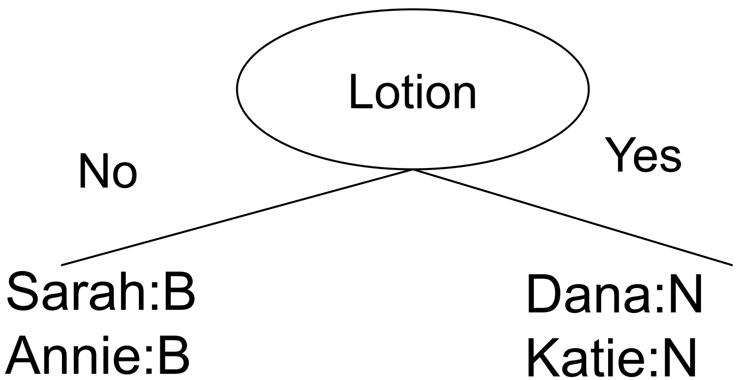
# Minimizing Disorder

| Name | Hair | Height | Weight | Lotion | Result |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |

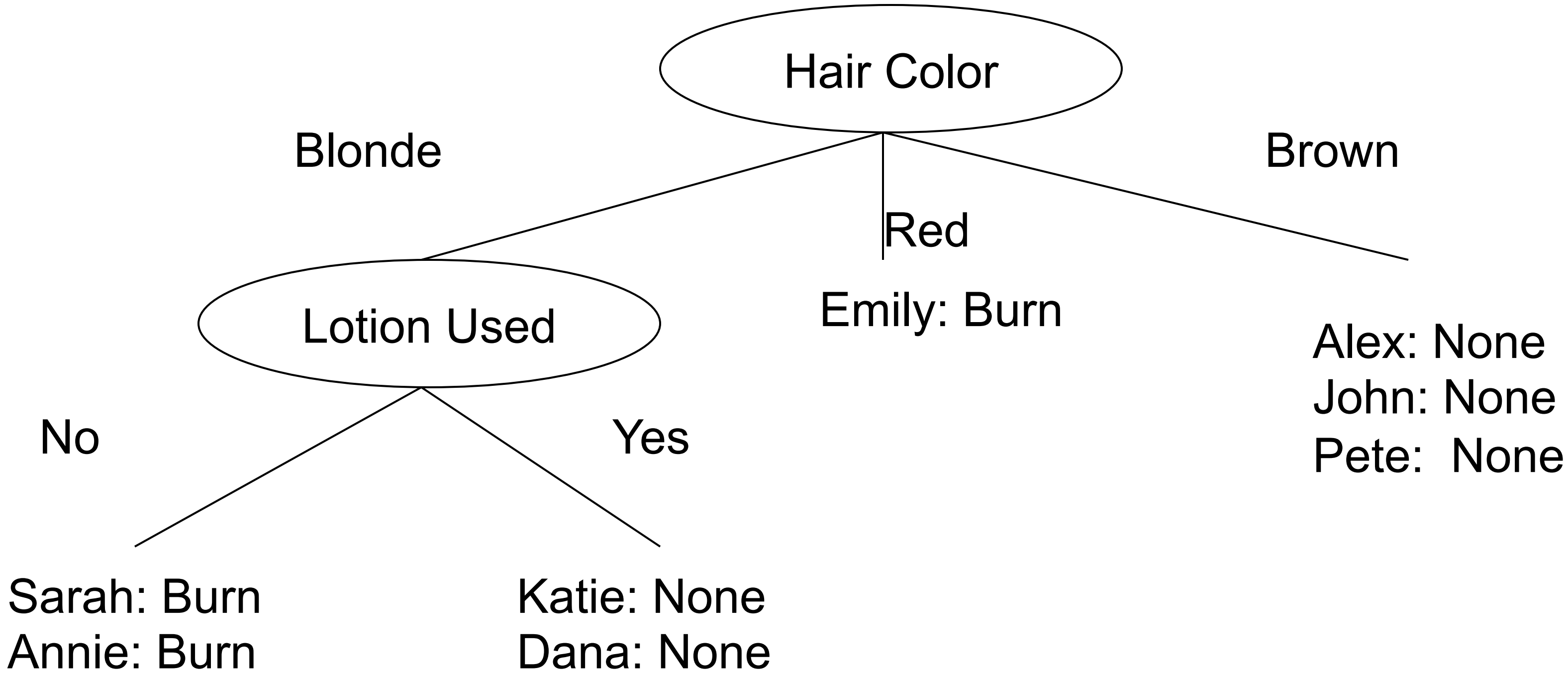
Splitting Blonde!



# Sunburn Identification Tree



# Measuring Disorder

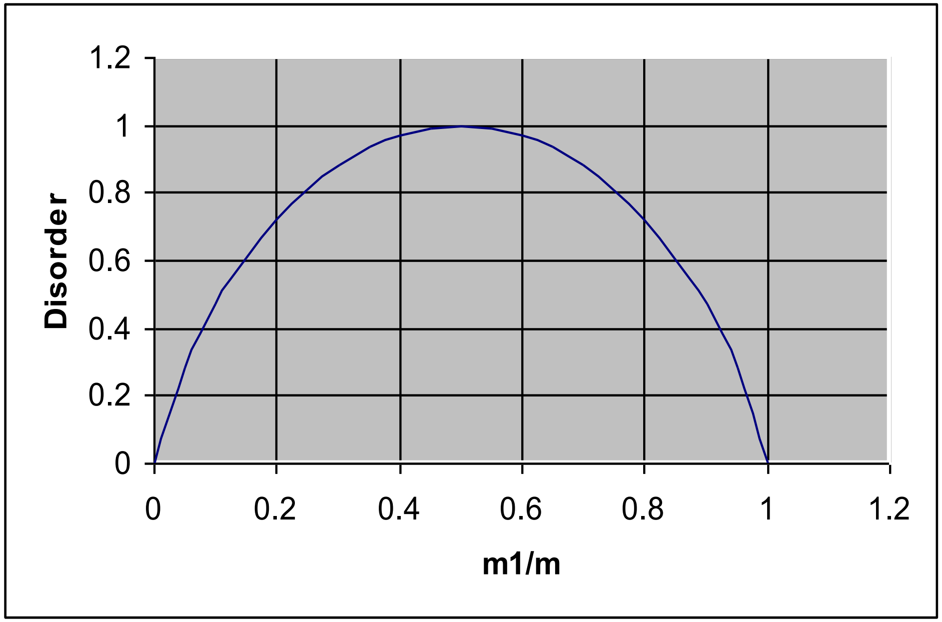
- Problem:
  - In general, tests on large DB's don't yield homogeneous subsets
- Solution:
  - General information theoretic measure of disorder
  - Desired features:
    - Homogeneous set: least disorder = 0
    - Even split: most disorder = 1

# Measuring Entropy

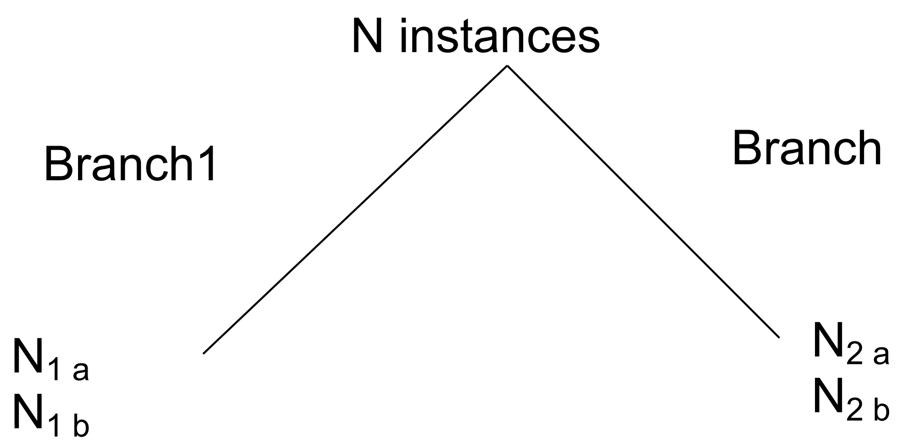
- If split we m objects into 2 bins of size $m_1$ and $m_2$, what is the entropy?

$$\sum_i -\frac{m_i}{m}\log_2\frac{m_i}{m} =$$

$$-\frac{m_1}{m}\log_2\frac{m_1}{m}-\frac{m_2}{m}\log_2\frac{m_2}{m}$$



# Computing Disorder




$$AvgDisorder = \sum_{i=1}^{k} \left( \boxed{\frac{n_i}{n_t}} \boxed{\sum_{c \in class} -\frac{n_{i,c}}{n_i} \log_2 \frac{n_{i,c}}{n_i}} \right)$$

Fraction of samples down branch i

Disorder of class distribution on branch i

# Entropy in Sunburn Example

$$AvgDisorder = \sum_{i=1}^{k} \left( \frac{n_i}{n_t} \sum_{c \in class} -\frac{n_{i,c}}{n_i} \log_2 \frac{n_{i,c}}{n_i} \right)$$

| **Name** | **Hair** | **Height** | **Weight** | **Lotion** | **Result** |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |

# Entropy in Sunburn Example

$$AvgDisorder = \sum_{i=1}^{k}\left(\frac{n_i}{n_t}\sum_{c\in class} -\frac{n_{i,c}}{n_i}\log_2\frac{n_{i,c}}{n_i}\right)$$

| **Name** | **Hair** | **Height** | **Weight** | **Lotion** | **Result** |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |

Hair color = 4/8(-2/4 log 2/4 - 2/4log2/4) + 1/8*0 + 3/8 *0
= 0.5
Height = 0.69
Weight = 0.94
Lotion = 0.61

# Entropy in Sunburn Example

$$AvgDisorder = \sum_{i=1}^{k}\left(\frac{n_i}{n_t}\sum_{c\in class} -\frac{n_{i,c}}{n_i}\log_2\frac{n_{i,c}}{n_i}\right)$$

| Name | Hair | Height | Weight | Lotion | Result |
|---|---|---|---|---|---|
| Sarah | Blonde | Average | Light | No | Burn |
| Dana | Blonde | Tall | Average | Yes | None |
| Alex | Brown | Short | Average | Yes | None |
| Annie | Blonde | Short | Average | No | Burn |
| Emily | Red | Average | Heavy | No | Burn |
| Pete | Brown | Tall | Heavy | No | None |
| John | Brown | Average | Heavy | No | None |
| Katie | Blonde | Short | Light | Yes | None |

Hair color already happened!

Height = 2/4(-1/2log1/2-1/2log1/2) + 1/4*0+1/4*0 = 0.5

Weight = 2/4(-1/2log1/2-1/2log1/2) +2/4(-1/2log1/2-1/2log1/2) = 1

Lotion = 0

So after hair color, we should expand lotion

# ID3 Algorithm

- Iterative Dichotomiser 3 -- Ross Quinlan (1986)

**ID3 (Examples, Target_Attribute, Attributes)**

- Create a root node for the tree
- If all examples are positive, Return the single-node tree Root, with label = +.
- If all examples are negative, Return the single-node tree Root, with label = -.
- If # of attributes is empty, return Root, with label = most common value of target.

- Otherwise:
- A ← Attribute that best classifies examples, use this attribute as Root
- For each possible value, $v_i$, of A,
- Add a new tree branch below Root, corresponding to the test A = $v_i$.
- Let Examples($v_i$) be the subset of examples that have the value $v_i$ for A
- If Examples($v_i$) empty, add leaf with label = most common value of target, otherwise add subtree ID3 (Examples($v_i$), Target_Attribute, Attributes – {A})

- Return Root

# Decision tree properties

- Widely used
- Greedy
- Robust to noise (incorrect examples)
- Easily understood by humans; this is important in medical, financial, military applications