# Machine Learning & Data Mining

## Chapter 7

## Decision Trees

# Top 10 Algorithms in DM

MIMA

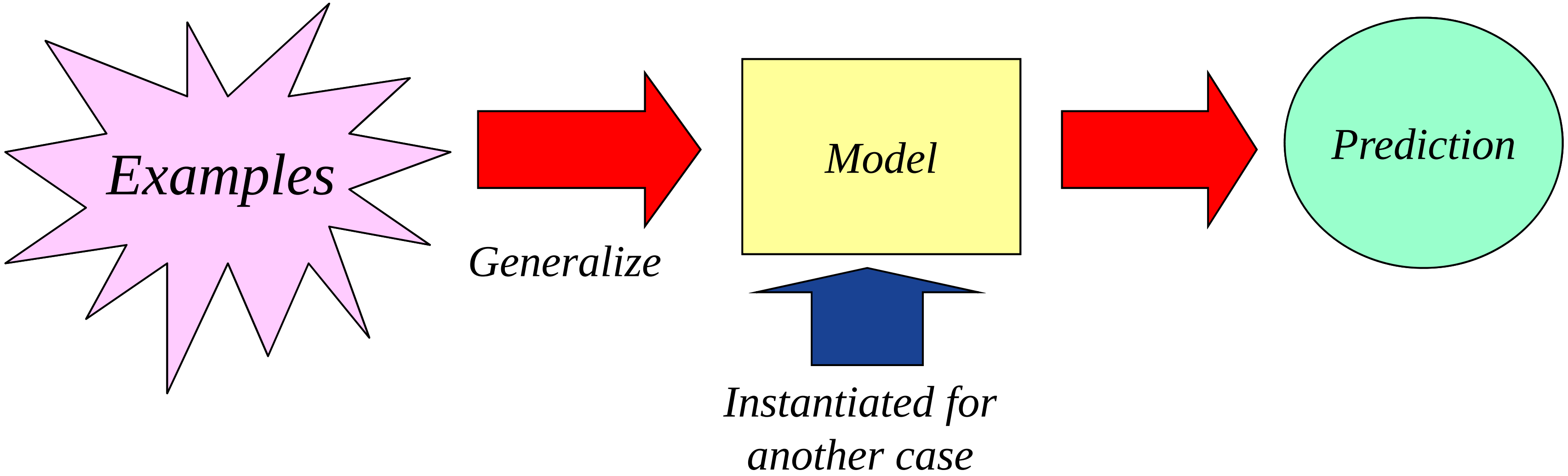
- #1: C4.5
- #2: *K*-Means
- #3: SVM
- #4: Apriori
- #5: EM
- #6: PageRank
- #7: AdaBoost
- #7: *k*NN
- #7: Naive Bayes
- #10: CART

# Content

- Introduction
- CLS
- ID3
- C4.5
- CART

# Inductive Learning

MIMA




*The general conclusion should apply to unseen examples.*

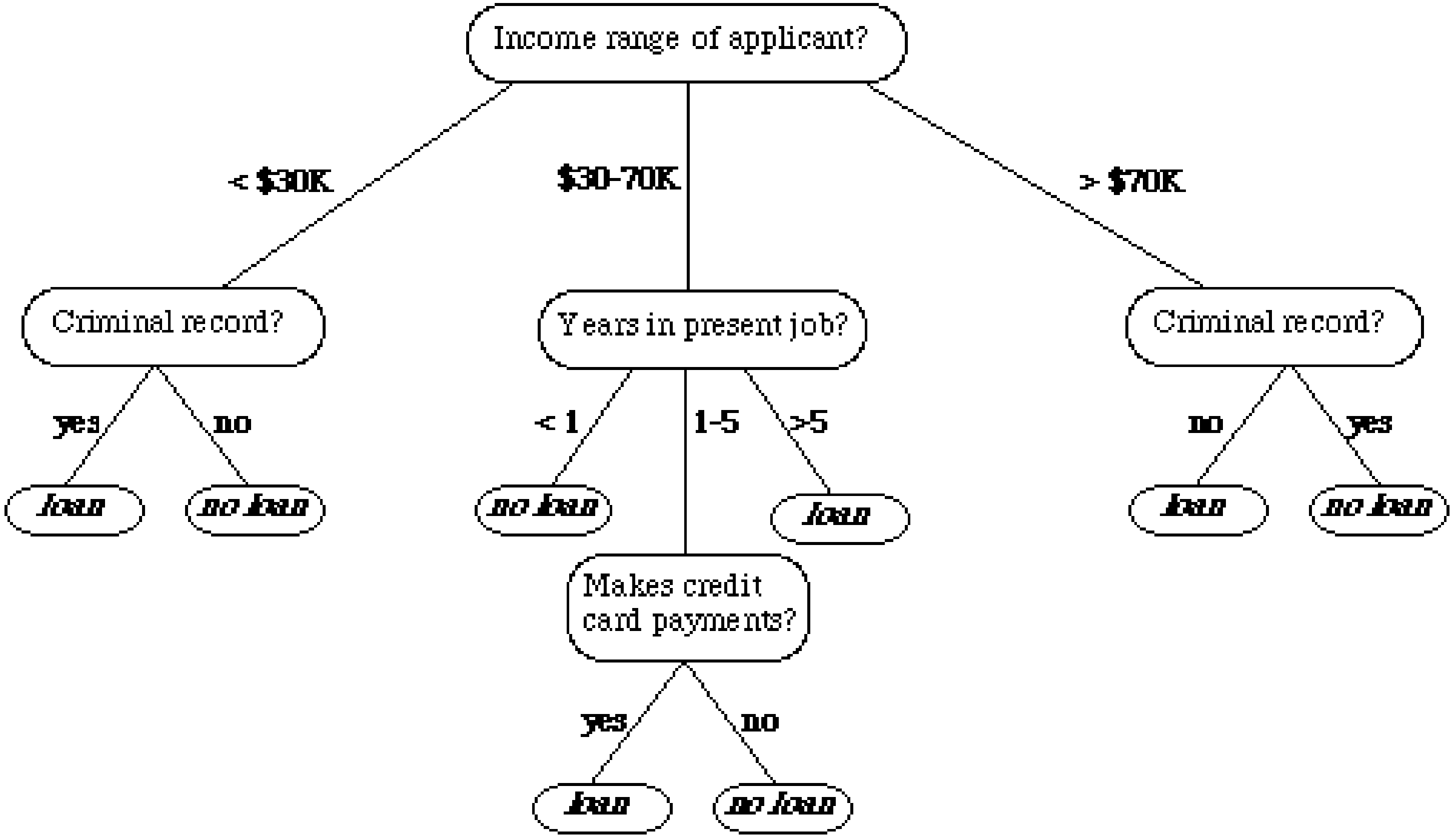
# Decision Tree

MIMA

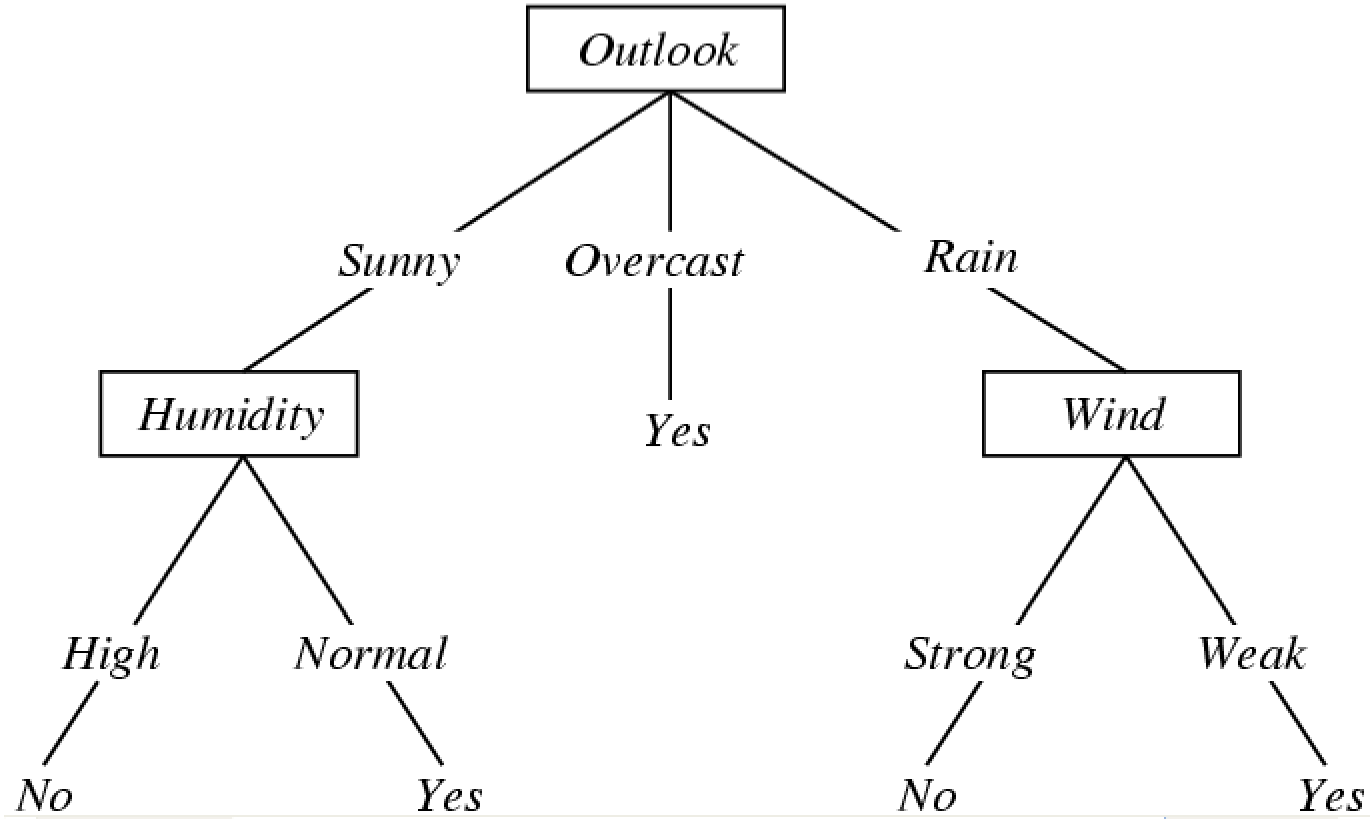
- A decision tree is a tree in which
  - each *branch node* represents a choice between a number of alternatives
  - each *leaf node* represents a classification or decision

# Example I

MIMA



# Example II

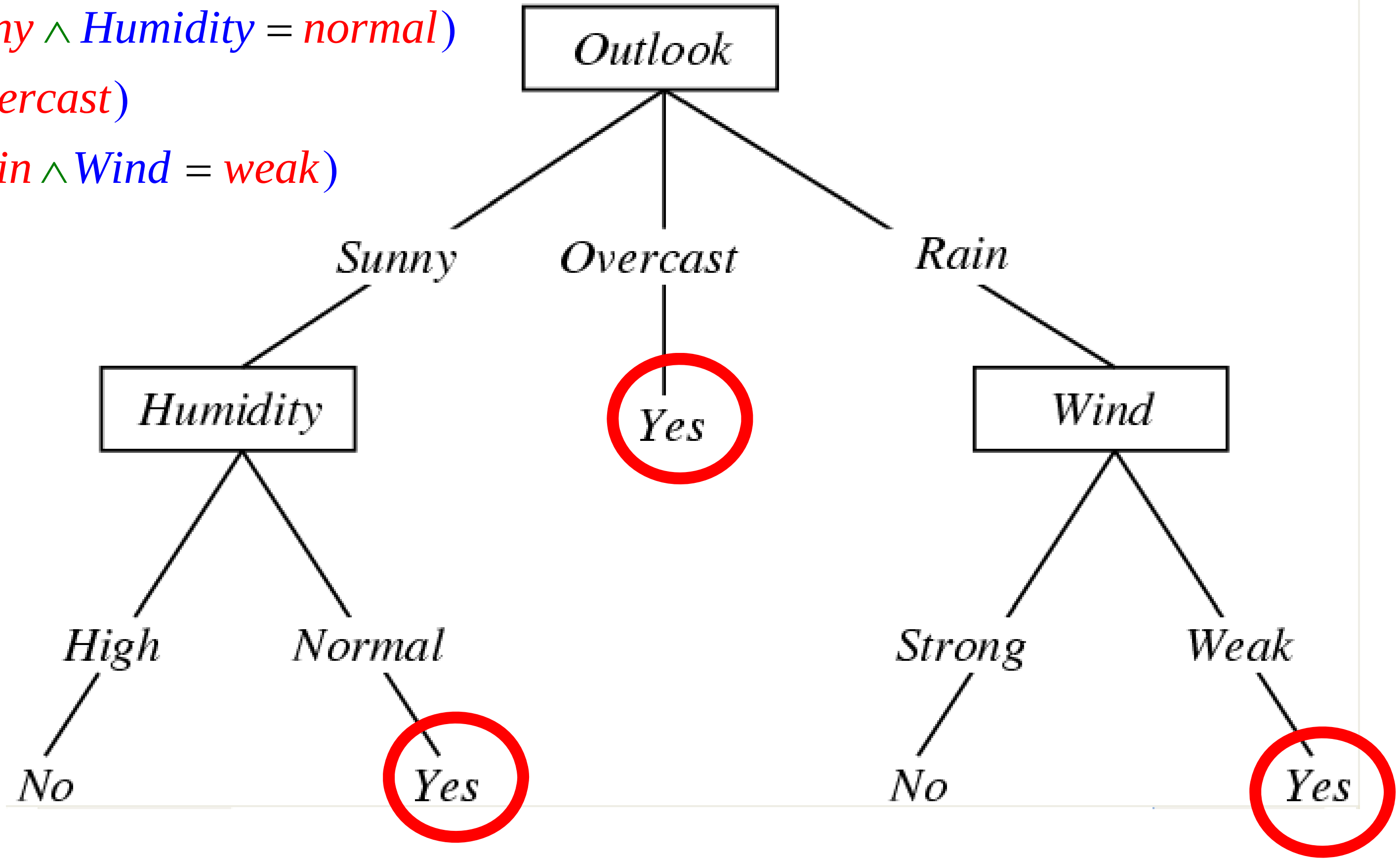


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

# Decision Rules



$$(Outlook = sunny \wedge Humidity = normal)$$
$$\vee (Outlook = overcast)$$
$$\vee (Outlook = rain \wedge Wind = weak)$$



# Decision Tree Learning

MIMA

- We wish to be able to *induce a decision tree from a set of data* about instances together with the decisions or classifications for those instances.

- Learning Algorithms:
  - CLS (Concept Learning System)
  - ID3 → C4 → C4.5 →C5

# Appropriate Problems for Decision Tree Learning

MIMA

- Instances are represented by *attribute-value pairs*.
- The target function has *discrete output values*.
- *Disjunctive descriptions* may be required.
- The training data may *contain errors*.
- The training data may contain *missing attribute* values.

# CLS Algorithm

MIMA

1. $T \leftarrow$ the whole training set. Create a $T$ node.
2. If all examples in $T$ are positive, create a 'P' node with $T$ as its parent and stop.
3. If all examples in $T$ are negative, create an 'N' node with $T$ as its parent and stop.
4. Select an attribute $X$ with values $v_1, v_2, \ldots, v_N$ and partition $T$ into subsets $T_1, T_2, \ldots, T_N$ according their values on $X$. Create $N$ nodes $T_i$ ($i$ = 1,..., $N$) with $T$ as their parent and $X = v_i$ as the label of the branch from $T$ to $T_i$.
5. For each $T_i$ do: $T \leftarrow T_i$ and goto step 2.

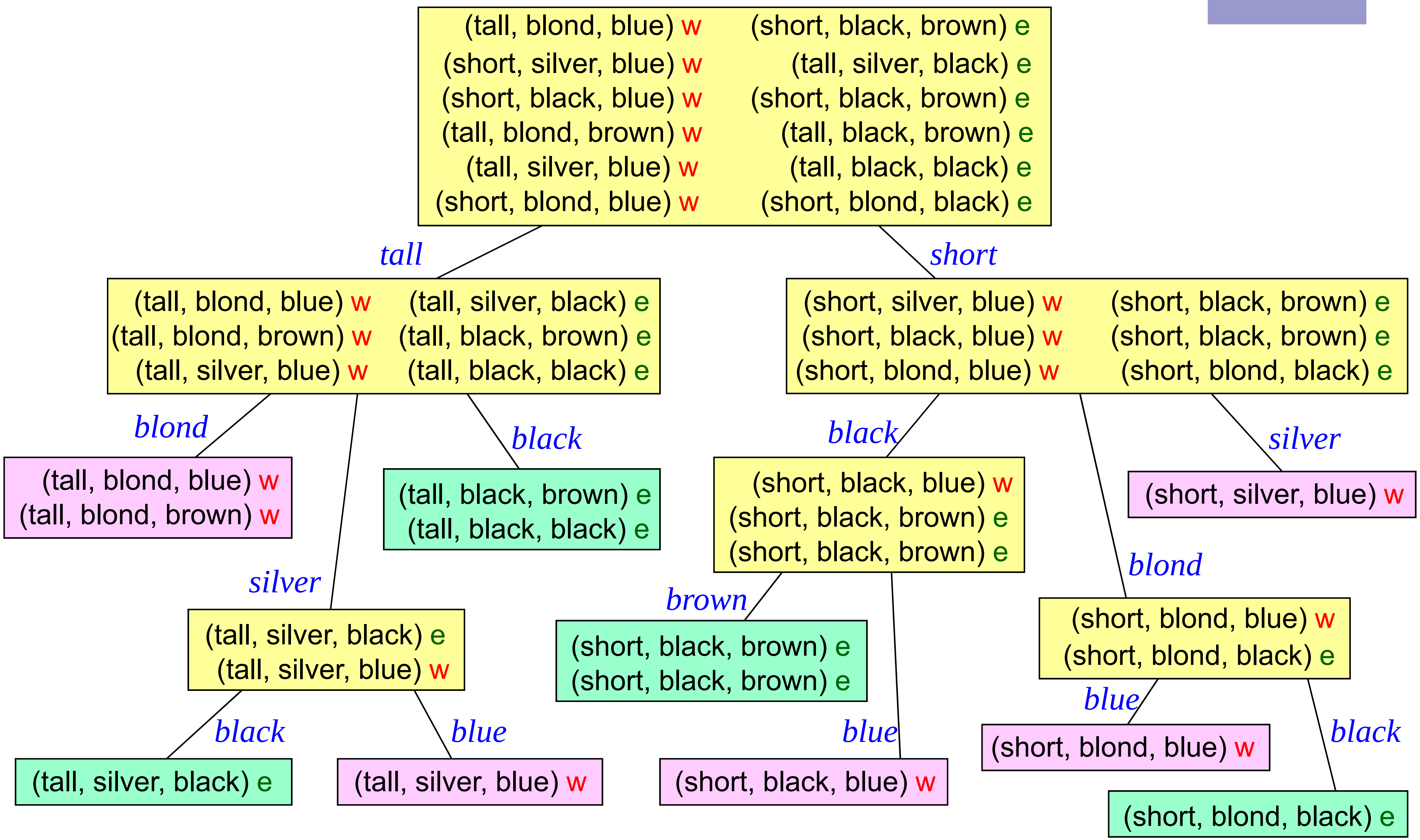
# Example

MIMA

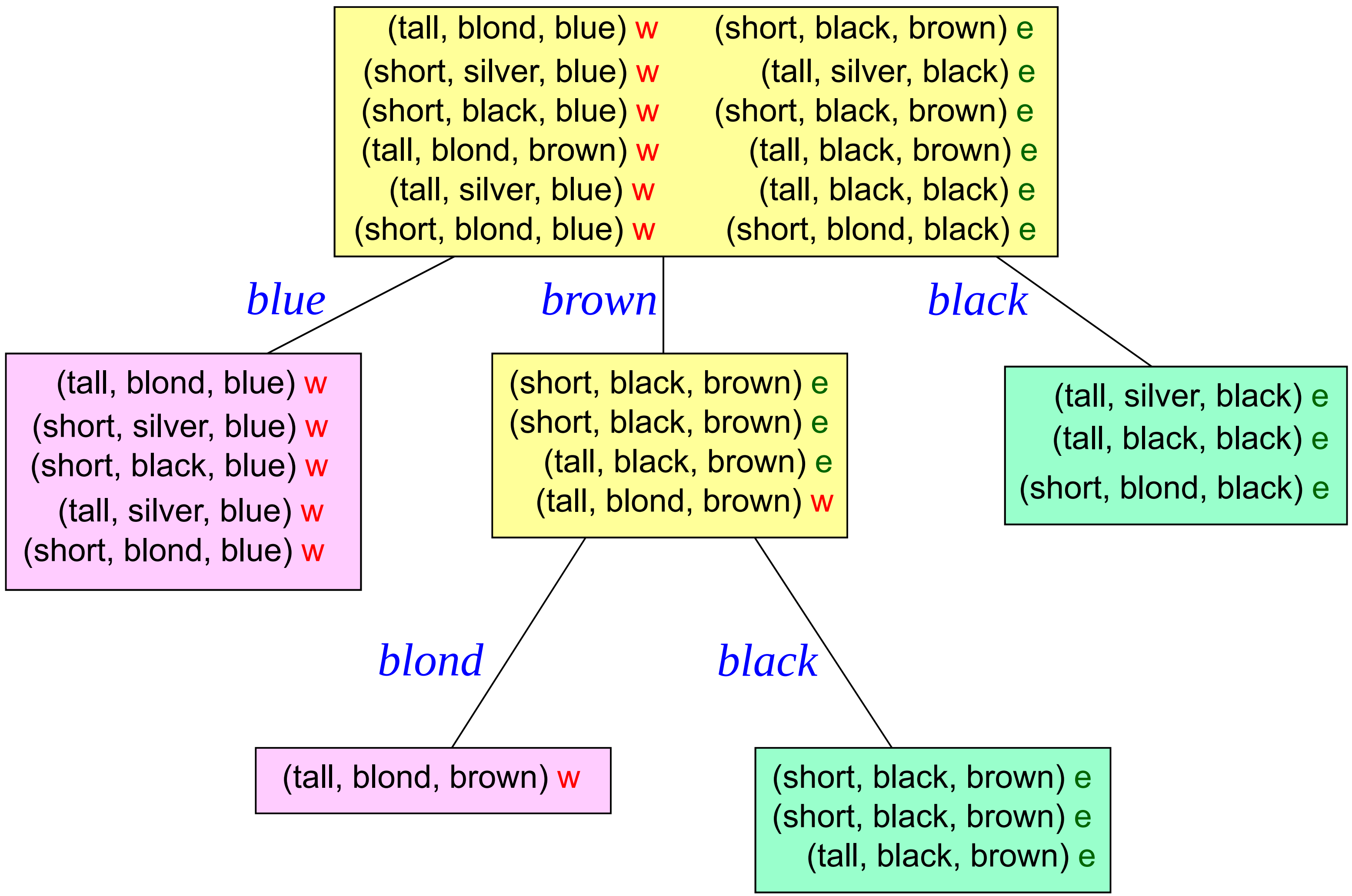
(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(short, blond, blue) w

(short, black, brown) e
(tall, silver, black) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e
(short, blond, black) e

# Example


(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(short, blond, blue) w
(short, black, brown) e
(tall, silver, black) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e
(short, blond, black) e
tall
short
(tall, blond, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(tall, silver, black) e
(tall, black, brown) e
(tall, black, black) e
(short, silver, blue) w
(short, black, blue) w
(short, blond, blue) w
(short, black, brown) e
(short, black, brown) e
(short, blond, black) e
blond
(tall, blond, blue) w
(tall, blond, brown) w
black
(tall, black, brown) e
(tall, black, black) e
silver
(tall, silver, black) e
(tall, silver, blue) w
black
(tall, silver, black) e
blue
(tall, silver, blue) w
black
(short, black, blue) w
(short, black, brown) e
(short, black, brown) e
brown
(short, black, brown) e
(short, black, brown) e
blue
(short, black, blue) w
silver
(short, silver, blue) w
blond
(short, blond, blue) w
(short, blond, black) e
blue
(short, blond, blue) w
black
(short, blond, black) e

# Example


(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(short, blond, blue) w
(short, black, brown) e
(tall, silver, black) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e
(short, blond, black) e
blue
brown
black
(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, silver, blue) w
(short, blond, blue) w
(short, black, brown) e
(short, black, brown) e
(tall, black, brown) e
(tall, blond, brown) w
(tall, silver, black) e
(tall, black, black) e
(short, blond, black) e
blond
black
(tall, blond, brown) w
(short, black, brown) e
(short, black, brown) e
(tall, black, brown) e

# ID3

MIMA

- Iterative Dichotomizer (version) 3
  - developed by Ross Quinlan

- Select decision sequence of the tree based on information gain.

# ID3

MIMA

- Entropy (Binary Classification)

$S =$

| | |
|---|---|
| (tall, blond, blue) w | (short, black, brown) e |
| (short, silver, blue) w | (tall, silver, black) e |
| (short, black, blue) w | (short, black, brown) e |
| (tall, blond, brown) w | (tall, black, brown) e |
| (tall, silver, blue) w | (tall, black, black) e |
| (short, blond, blue) w | (short, blond, black) e |

$C_1$ : Class 1  $C_2$ : Class 2

$$p_1 = P(s \in C_1)$$

$$p_2 = P(s \in C_2)$$

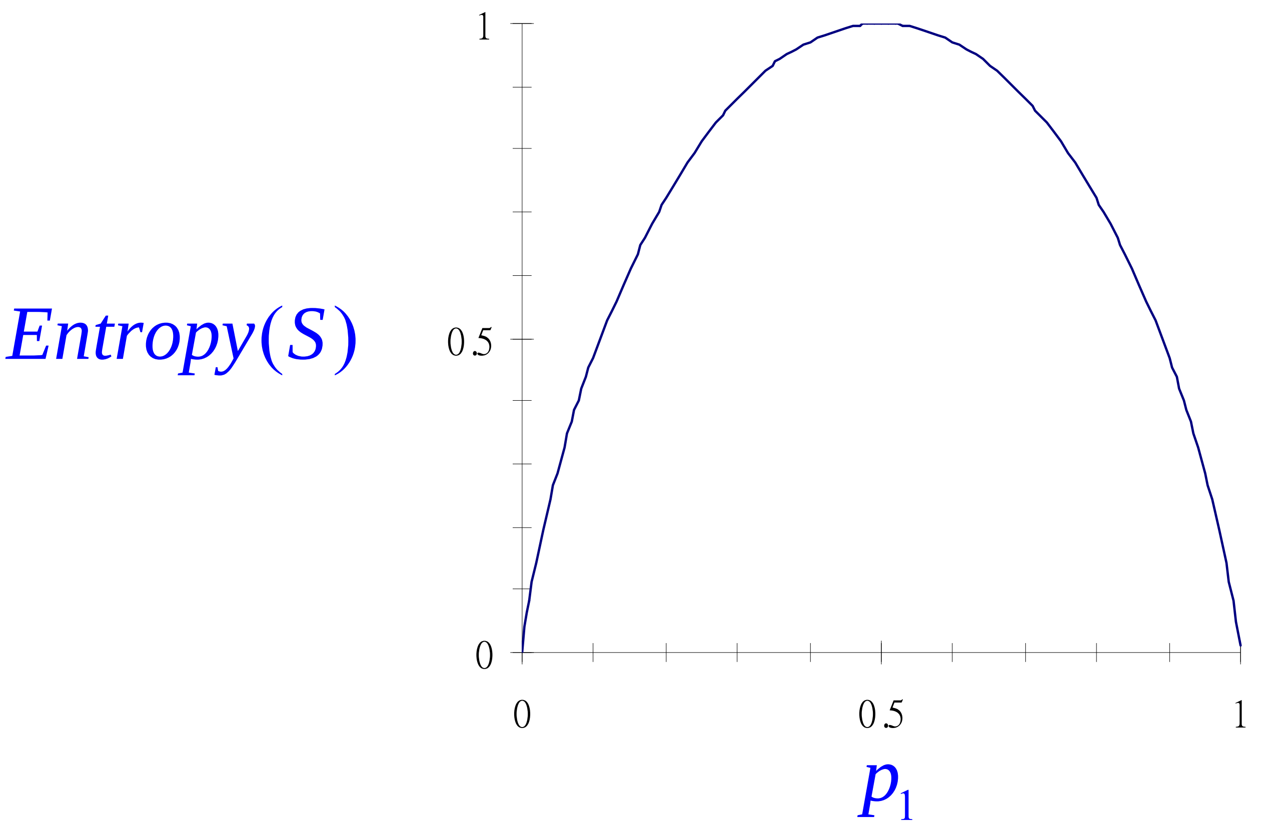
$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

# ID3

MIMA

- Entropy (Binary Classification)

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

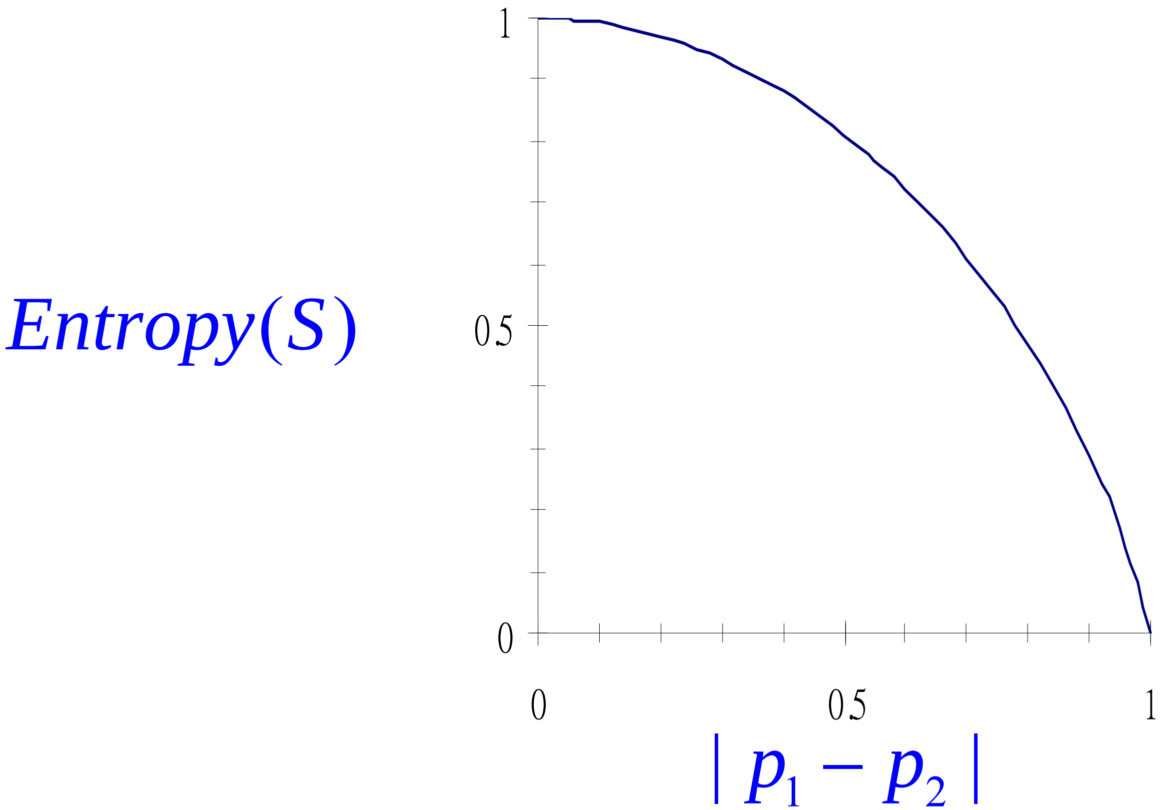


# ID3



- Entropy (Binary Classification)

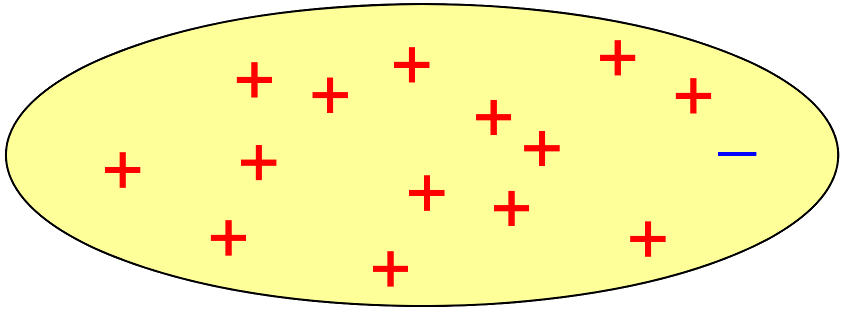
$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$



# ID3

- Entropy (Binary Classification)

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$



# + = 14

# – = 1

$$p_+ = 14/15$$

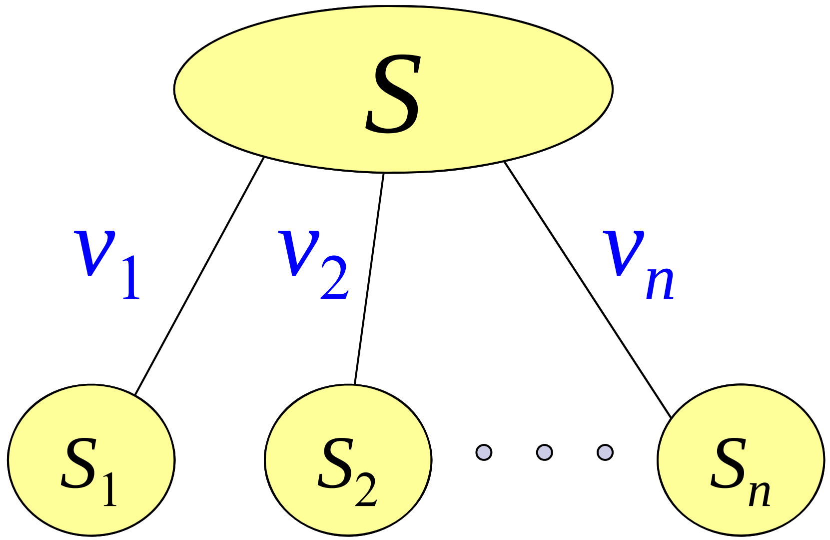
$$p_- = 1/15$$

$$Entropy = 0.353359$$

# Information Gain



Attribute $A = \{v_1, \ldots, v_n\}$




$$Gain(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

# Information Gain

MIMA

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

- (tall, blond, blue) w
- (short, silver, blue) w
- (short, black, blue) w
- (tall, blond, brown) w
- (tall, silver, blue) w
- (short, blond, blue) w
- (short, black, brown) e
- (tall, silver, black) e
- (short, black, brown) e
- (tall, black, brown) e
- (tall, black, black) e
- (short, blond, black) e

$$Entropy(S) = 1$$

*tall*

- (tall, blond, blue) w
- (tall, blond, brown) w
- (tall, silver, blue) w
- (tall, silver, black) e
- (tall, black, brown) e
- (tall, black, black) e

$$Entropy(S_{tall}) = 1$$

*short*

- (short, silver, blue) w
- (short, black, blue) w
- (short, blond, blue) w
- (short, black, brown) e
- (short, black, brown) e
- (short, blond, black) e

$$Entropy(S_{short}) = 1$$

$$Gain(S, Height) = 0$$

# Information Gain

MIMA

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(short, blond, blue) w
(short, black, brown) e
(tall, silver, black) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e
(short, blond, black) e

$$Entropy(S) = 1$$

*blond*

(tall, blond, blue) w
(tall, blond, brown) w
(short, blond, blue) w
(short, blond, black) e

$$Entropy(S_{blond}) = 0.811278$$

*silver*

(short, silver, blue) w
(tall, silver, blue) w
(tall, silver, black) e

$$Entropy(S_{silver}) = 0.918296$$

*black*

(short, black, blue) w
(short, black, brown) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e

$$Entropy(S_{black}) = 0.721928$$

$$Gain(S, Hair) = 0.199197$$

# Information Gain

MIMA

$$Gain(S,A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, blond, brown) w
(tall, silver, blue) w
(short, blond, blue) w
(short, black, brown) e
(tall, silver, black) e
(short, black, brown) e
(tall, black, brown) e
(tall, black, black) e
(short, blond, black) e

$$Entropy(S) = 1$$

*blue*

(tall, blond, blue) w
(short, silver, blue) w
(short, black, blue) w
(tall, silver, blue) w
(short, blond, blue) w

$$Entropy(S_{blue}) = 0$$

*brown*

(short, black, brown) e
(short, black, brown) e
(tall, black, brown) e
(tall, blond, brown) w

$$Entropy(S_{brown}) = 0.811278$$

*black*

(tall, silver, black) e
(tall, black, black) e
(short, blond, black) e

$$Entropy(S_{black}) = 0$$

$$Gain(S, Eye) = 0.829574$$

# Information Gain

MIMA

| | |
|---|---|
| (tall, blond, blue) w | (short, black, brown) e |
| (short, silver, blue) w | (tall, silver, black) e |
| (short, black, blue) w | (short, black, brown) e |
| (tall, blond, brown) w | (tall, black, brown) e |
| (tall, silver, blue) w | (tall, black, black) e |
| (short, blond, blue) w | (short, blond, black) e |

$$Gain(S, Hair) = 0.199197$$

$$Gain(S, Height) = 0$$

$$Gain(S, Eye) = 0.829574$$

# ID3 (modify of CLS)

MIMA

1. $T \leftarrow$ the whole training set. Create a $T$ node.
2. If all examples in $T$ are positive, create a 'P' node with $T$ as its parent and stop.
3. If all examples in $T$ are negative, create a 'N' node with $T$ as its parent and stop.
4. Select an attribute $X$ with values $v_1, v_2, \ldots, v_N$ and partition $T$ into subsets $T_1, T_2, \ldots, T_N$ according their values on $X$. Create $N$ nodes $T_i$ ($i = 1, \ldots, N$) with $T$ as their parent and $X = v_i$ as the label of the branch from $T$ to $T_i$.
5. For each $T_i$ do: $T \leftarrow T_i$ and goto step 2.

# ID3 (modify of CLS)

MIMA

1 $T \leftarrow$ the whole training set. Create a $T$ node.

2 If all examples i… with $T$ as it…

3 If all exampl… …node with $T$ as its parent and stop.

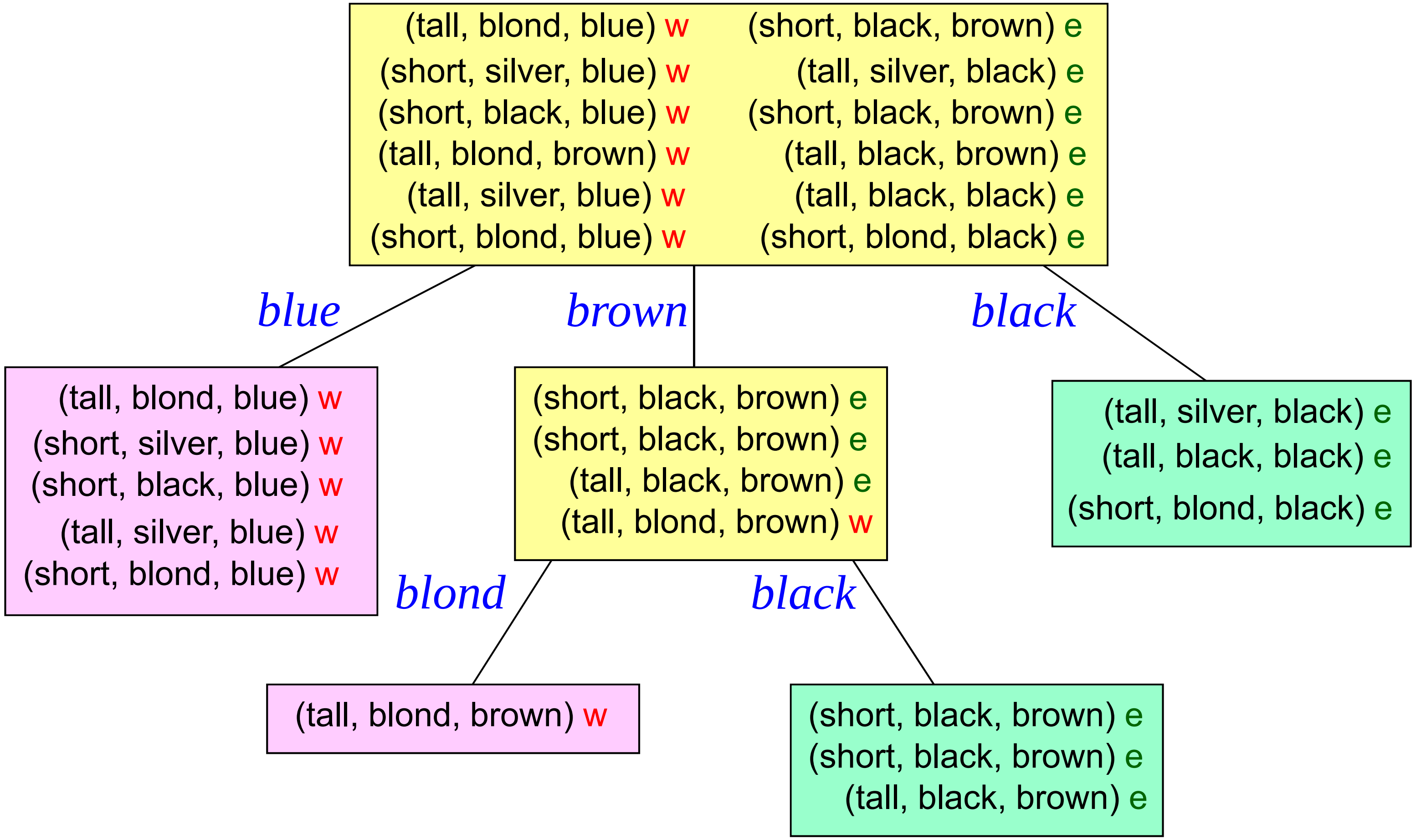
*By maximizing the information gain.*

4 Select an attribute $X$ with values $v_1, v_2, \ldots, v_N$ and partition $T$ into subsets $T_1, T_2, \ldots, T_N$ according their values on $X$. Create $N$ nodes $T_i$ ($i$ = 1,..., $N$) with $T$ as their parent and $X = v_i$ as the label of the branch from $T$ to $T_i$.

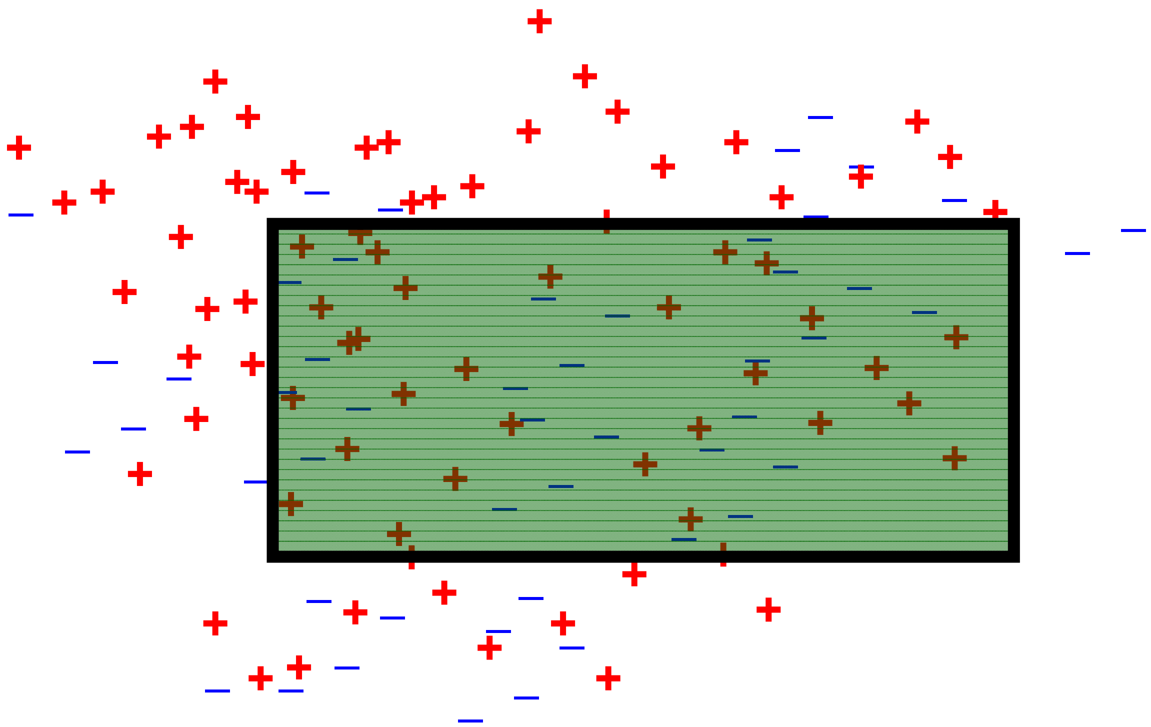
5 For each $T_i$ do: $T \leftarrow T_i$ and goto step 2.

# Example

MIMA



# Windowing



# Windowing

MIMA

- ID3 can deal with very large data sets by performing induction on subsets or *windows* onto the data.
  1. Select a random subset of the whole set of training instances.
  2. Use the induction algorithm to form a rule to explain the current window.
  3. Scan through all of the training instances looking for exceptions to the rule.
  4. Add the exceptions to the window
- Repeat steps 2 to 4 until there are no exceptions left.

# Inductive Biases



- Shorter trees are preferred.
- Attributes with higher information gain are selected first in tree construction.
  - Greedy Search
- Preference bias (relative to restriction bias as in the VS approach)
- Why prefer short hypotheses?
  - Occam's razor Generalization

# Overfitting to the Training Data

MIMA

- The training error is statistically smaller than the test error for a given hypothesis.
- Solutions:
  - Early stopping
  - Validation sets
  - Statistical criterion for continuation (of the tree)
  - Post-pruning
  - Minimal description length

    cost-function = error + complexity

# Pruning Techniques

MIMA

- Reduced error pruning (of nodes)
  - Used by ID3
- Rule post-pruning
  - Used by C4.5

# Reduced Error Pruning

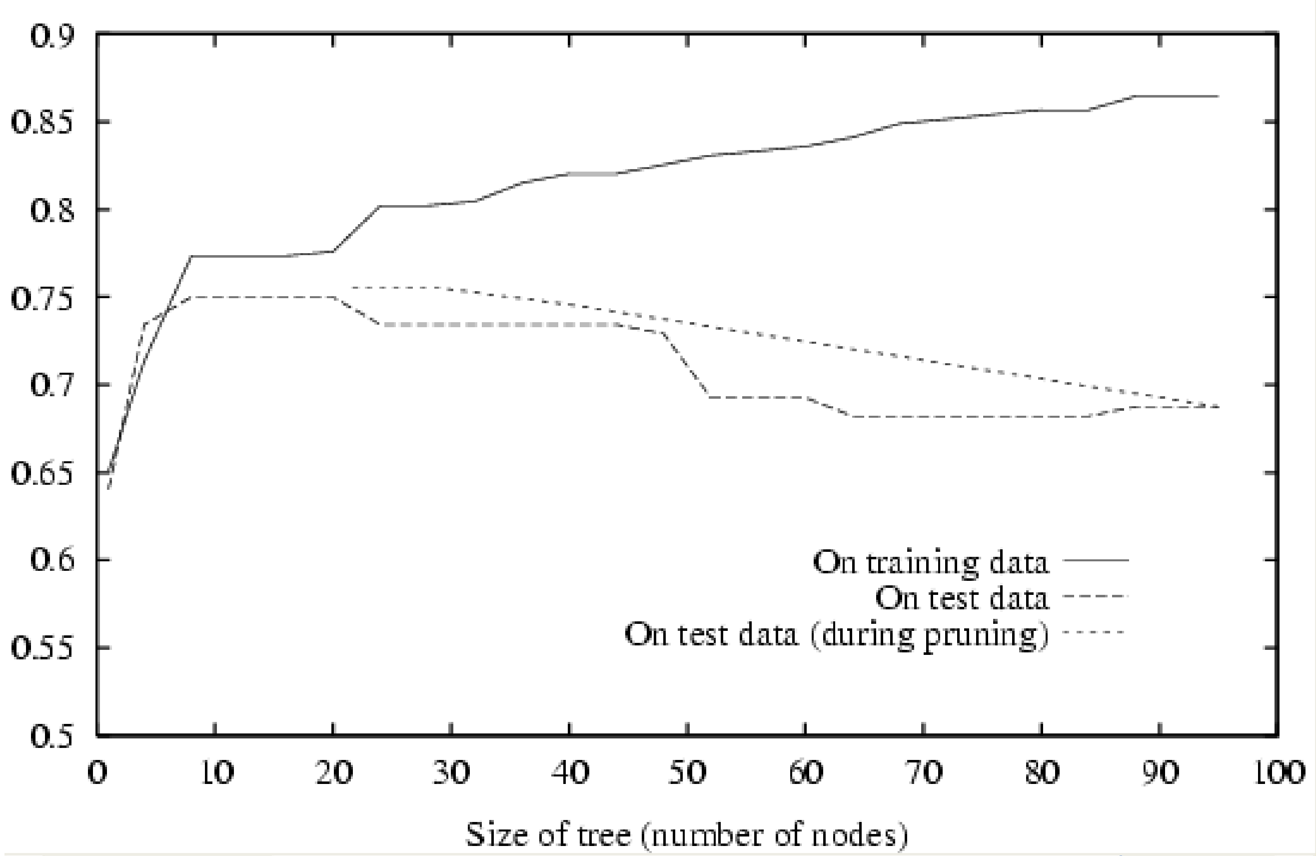
MIMA

- Use a separated validation set
- Tree accuracy:
  - percentage of correct classifications on validation set
- Method:

  Do until further pruning is harmful
  - Evaluate the impact on validation set of pruning each possible node.
  - Greedily remove the one that most improves the validation set accuracy.

# Reduced Error Pruning

MIMA



# C4.5:An Extension of ID3

MIMA

- Some additional features of C4.5 are:
  - Incorporation of numerical (continuous) attributes.
  - Nominal (discrete) values of a single attribute may be grouped together, to support more complex tests.
  - Post-pruning after induction of trees, e.g. based on test sets, in order to increase accuracy.
  - C4.5 can deal with incomplete information (missing attribute values).
  - Use gain ratio instead of information gain

# Rule Post-Pruning

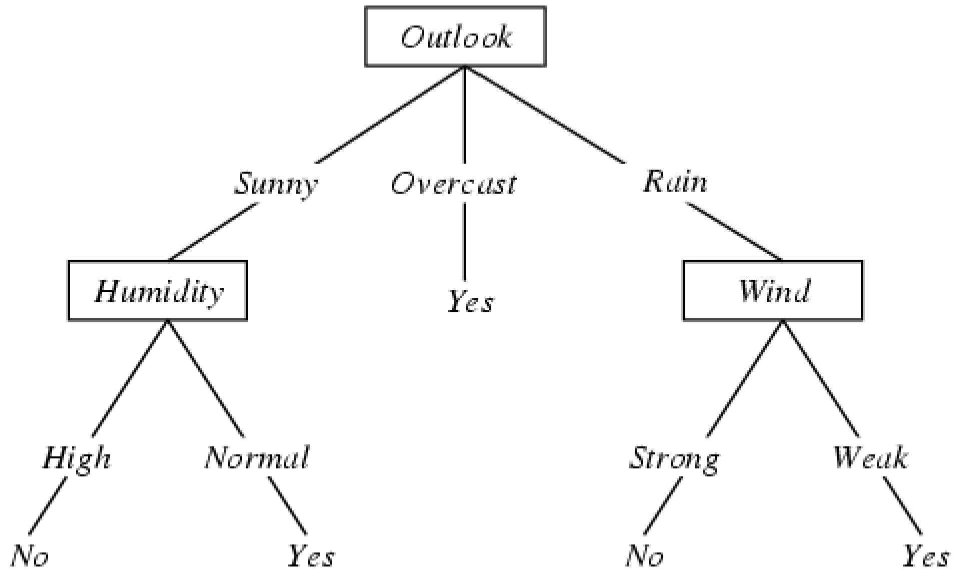
MIMA

- Fully induce the decision tree from the training set (allowing overfitting)
- Convert the learned tree to rules
  - one rule for each path from the root node to a leaf node
- Prune each rule by removing any preconditions that result in improving its estimated accuracy
- Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances

# Converting to Rules

IF $(Outlook = Sunny) \wedge (Humidity = High)$
THEN $PlayTennis = No$

IF $(Outlook = Sunny) \wedge (Humidity = Normal)$
THEN $PlayTennis = Yes$

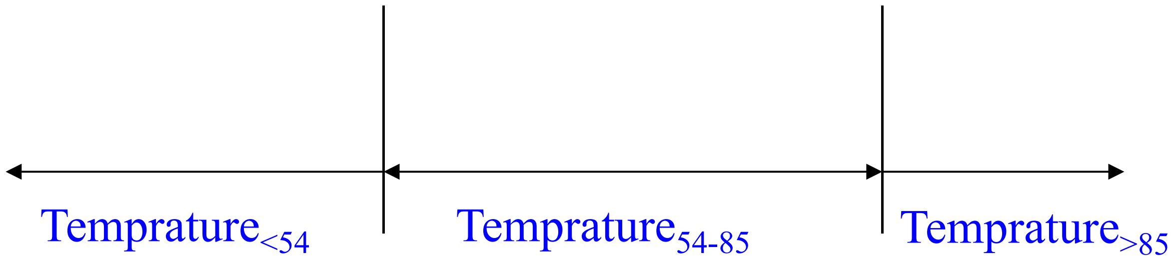


# handling numeric attributes

- Continuous attribute ➔ discrete attribute
- Example
  - Original attribute: Temperature = 82.5
  - New attribute: (temperature > 72.3) = t, f

| | | | | | | |
|---|---|---|---|---|---|---|
| *Temperature:* | 40 | 48 | 60 | 72 | 80 | 90 |
| *PlayTennis:* | No | No | Yes | Yes | Yes | No |

Example:




*How to choose split points?*

# handling numeric attributes



- Choosing split points for a continuous attribute
  - Sort the examples according to the values of the continuous attribute.
  - Identify adjacent examples that differ in their target labels and attribute values → a set of candidate split points
  - Calculate the gain for each split point and choose the one with the highest gain.

# Attributes with Many Values

MIMA

- Information Gain – biases to attribute with many values
  - e.g., date
- One approach – use *GainRatio* instead of information gain.

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

# Associate Attributes of Cost

MIMA

- The availability of attributes may vary significantly in costs, e.g., medical diagnosis.

- Example: Medical disease classification
  - *Temperature*
  - *BiopsyResult* High
  - *Pulse*
  - *BloodTestResult* High

*How to learn a consistent tree with low expected cost?*

# Associate Attributes of Cost

MIMA

- Tan and Schlimmer (1990)

$$\frac{Gain^2(S, A)}{Cost(S, A)}$$

- Nunez (1988)

$$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$

$w \in [0, 1]$ determines the importance of cost.

# Unknown Attribute Values

MIMA

$$S = \begin{Bmatrix} (\cdots, x, \cdots)+ \\ (\cdots, x, \cdots)- \\ (\cdots, x, \cdots)+ \\ (\cdots, y, \cdots)- \\ (\cdots, y, \cdots)+ \\ (\cdots, y, \cdots)+ \\ (\cdots, y, \cdots)+ \\ (\cdots, z, \cdots)+ \\ (\cdots, z, \cdots)+ \\ (\cdots, z, \cdots)- \\ (\cdots, ?, \cdots)- \\ (\cdots, ?, \cdots)+ \end{Bmatrix}$$

Attribute $A=\{x, y, z\}$

$$Gain(S, A) = ?$$

Possible Approaches:

- Assign most common value of *A* to the unknown one.
- Assign most common value of *A* with the same target value to the unknown one.
- Assign probability to each possible value.

# CART

MIMA

- Classification And Regression Trees
  - Generates binary decision tree: only 2 children created at each node (whereas ID3 creates a child for each subcategory).
  - Each split makes the subset more pure than that before splitting.
  - In ID3, Entropy is used to measure the splitting; in CART, impurity is used.

# CART



- Node impurity is 0 when all patterns at the node are of the same category; it becomes maximum when all the classes at the node are equally likely.
- Entropy Impurity

$$i(N) = -\sum_j P(\omega_j)\log_2 P(\omega_j)$$

- Gini Impurity

$$i(N) = \sum_{i \neq j} P(\omega_i)P(\omega_j)$$

- Misclassification impurity

$$i(N) = 1 - \max_j P(\omega_j)$$

# 【 Thank You ! 】

**Any Question?**