Decision Tree Algorithm

Week 4

Team Homework Assignment #5

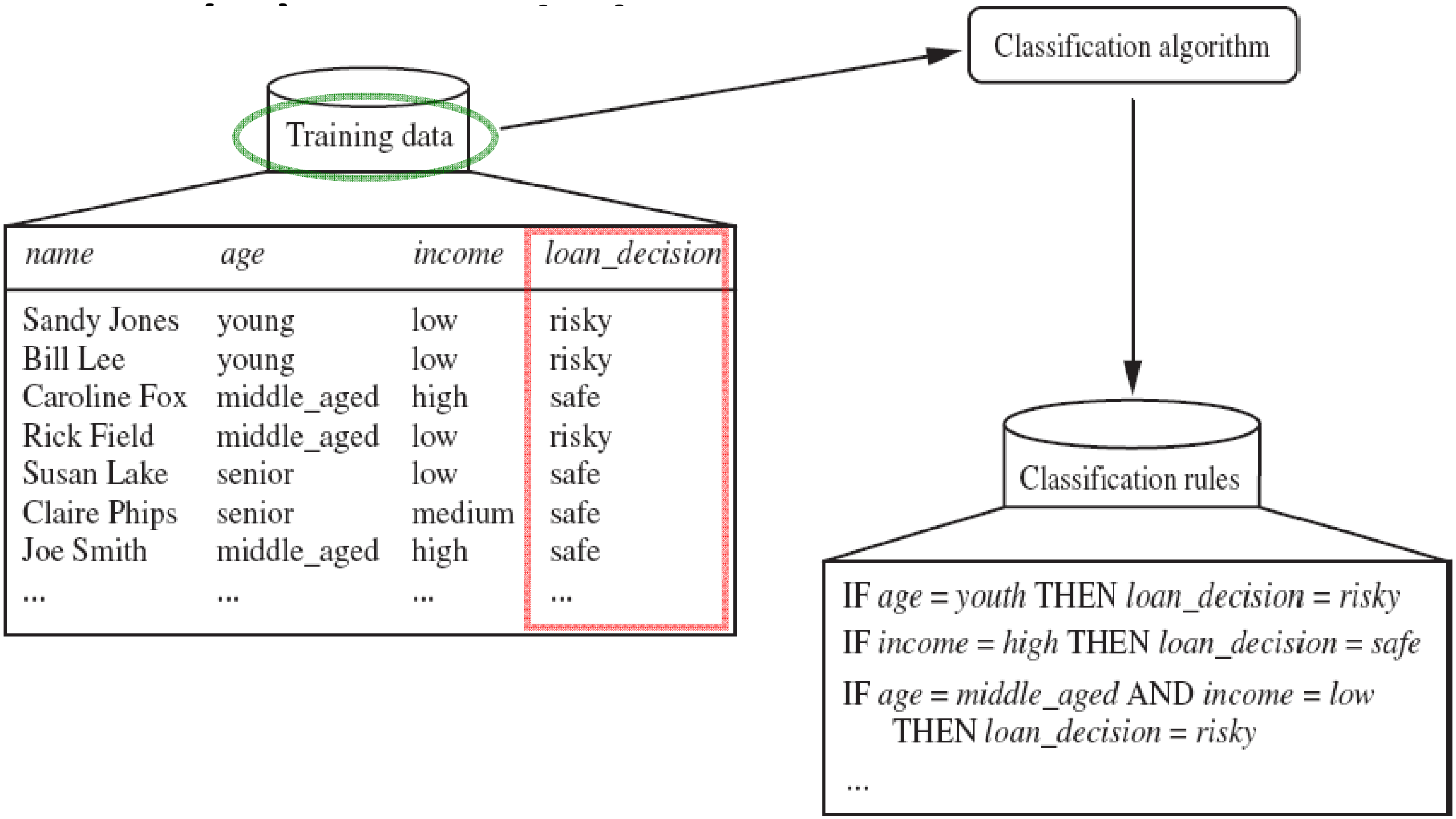
- Read pp. 105 – 117 of the text book.
- Do Examples 3.1, 3.2, 3.3 and Exercise 3.4 (a). Prepare for the results of the homework assignment.
- Due date
  - beginning of the lecture on Friday February 25th.

Team Homework Assignment #6

- Decide a data warehousing tool for your future homework assignments
- Play the data warehousing tool
- Due date
  - beginning of the lecture on Friday February 25th.

Classification - A Two-Step Process

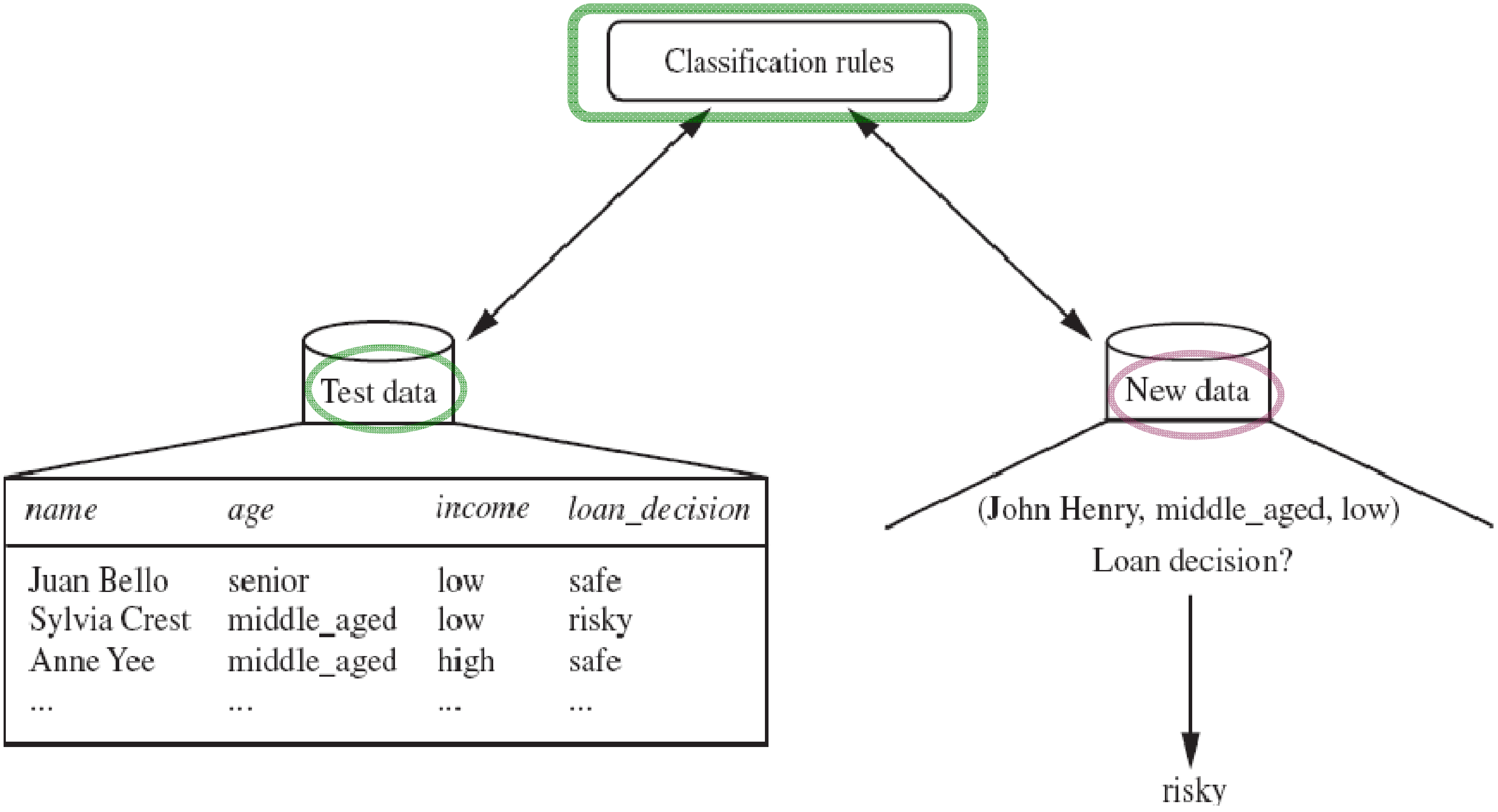
- **Model usage**: classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test data is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known



name	*age*	*income*	*loan_decision*
Sandy Jones	young	low	risky
Bill Lee	young	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

IF *age* = *youth* THEN *loan_decision* = *risky*
IF *income* = *high* THEN *loan_decision* = *safe*
IF *age* = *middle_aged* AND *income* = *low*
THEN *loan_decision* = *risky*
...

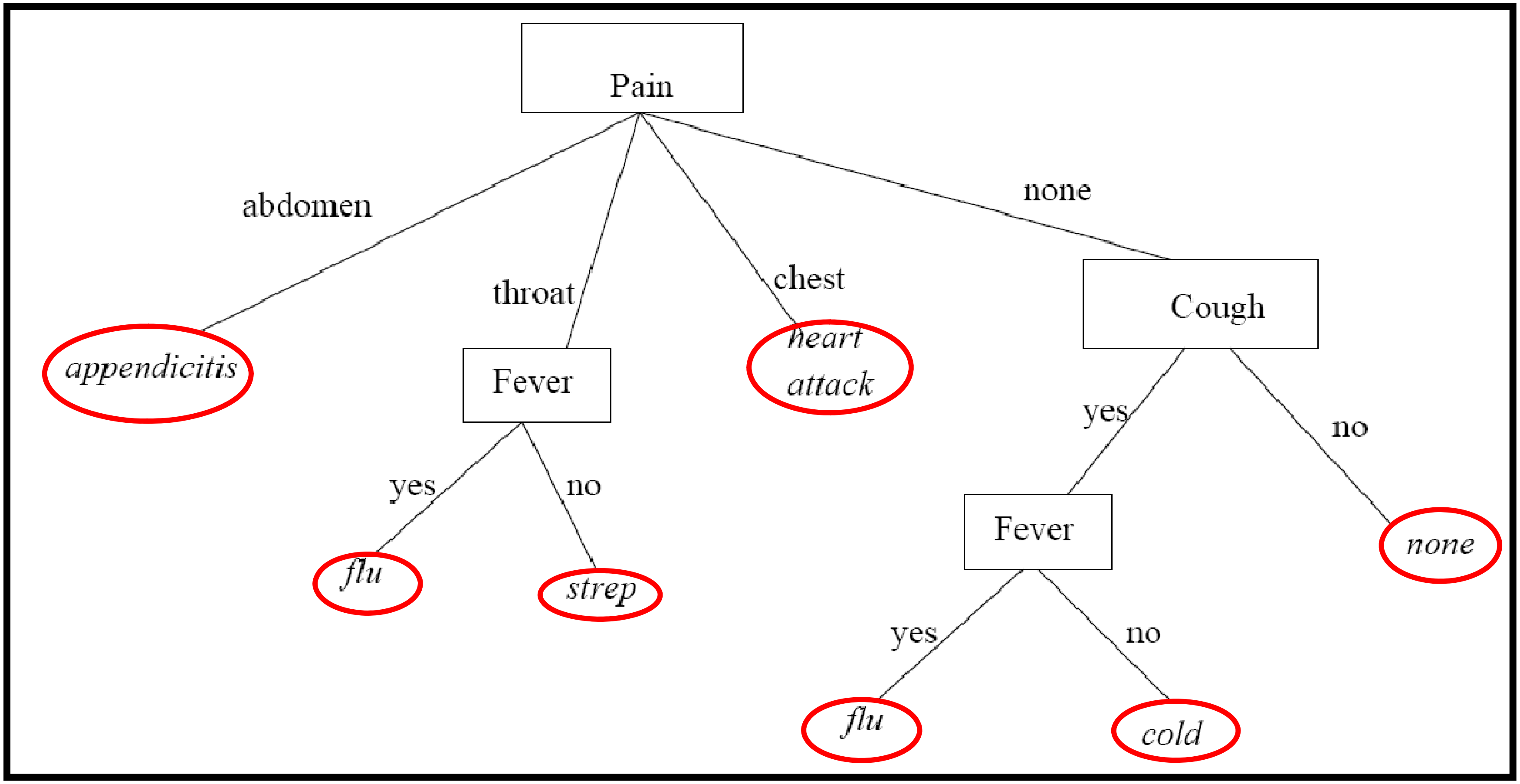
Figure 6.1 The data classification process: (a) Learning: Training data are analyzed by a classification algorithm. Here, the class label attribute is ***loan_decision***, and the learned model or classifier is represented in the form of classification rules.



name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

Figure 6.1 The data classification process: (b) Classification: Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

Decision Tree Classification Example

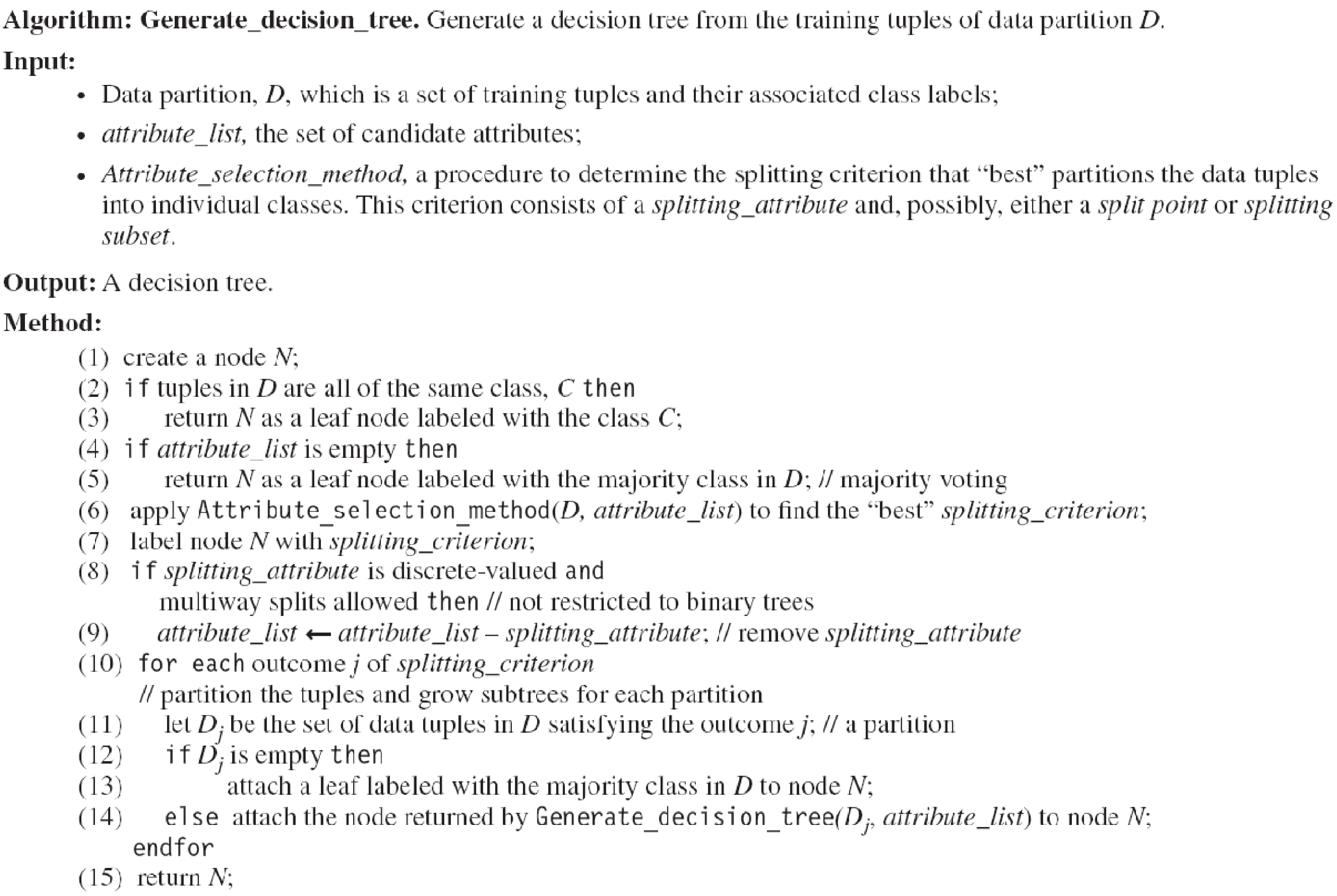


Decision Tree Learning Overview

- Decision Tree learning is one of the most widely used and practical methods for inductive inference over supervised data.
- A decision tree represents a procedure for classifying categorical data based on their attributes.
- It is also efficient for processing large amount of data, so is often used in data mining application.
- The construction of decision tree does not require any domain knowledge or parameter setting, and therefore appropriate for exploratory knowledge discovery.
- Their representation of acquired knowledge in tree form is intuitive and easy to assimilate by humans

Decision Tree Algorithm – ID3

- Decide which attribute (splitting-point) to test at node ***N*** by determining the "best" way to separate or partition the tuples in ***D*** into individual classes
- The splitting criteria is determined so that, ideally, the resulting partitions at each branch are as "pure" as possible.
  - A partition is pure if all of the tuples in it belong to the same class



Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.

Input:

- Data partition, D, which is a set of training tuples and their associated class labels;
- *attribute_list,* the set of candidate attributes;
- *Attribute_selection_method,* a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

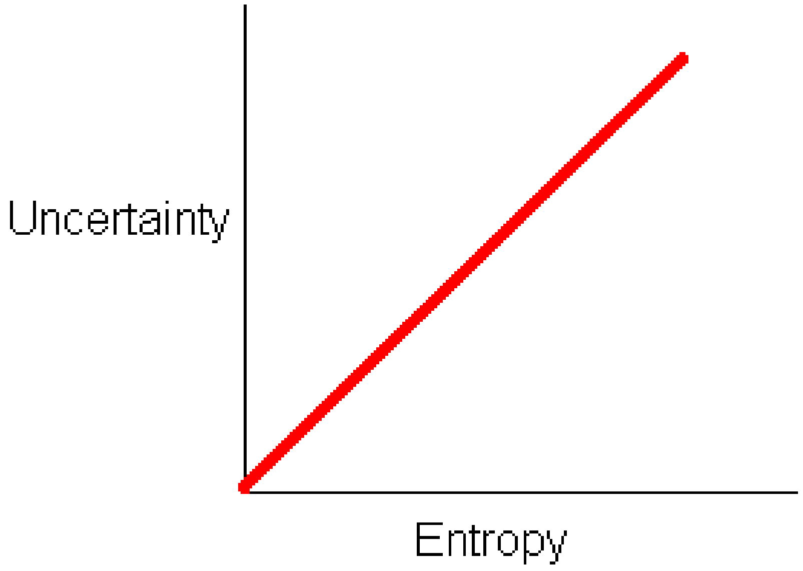
Method:

```
create a node $N$;
if tuples in $D$ are all of the same class, $C$ then
    return $N$ as a leaf node labeled with the class $C$;
if attribute_list is empty then
    return $N$ as a leaf node labeled with the majority class in $D$; // majority voting
apply Attribute_selection_method($D$, attribute_list) to find the "best" splitting_criterion;
label node $N$ with splitting_criterion;
if splitting_attribute is discrete-valued and
    multiway splits allowed then // not restricted to binary trees
    attribute_list ← attribute_list – splitting_attribute; // remove splitting_attribute
for each outcome $j$ of splitting_criterion
  // partition the tuples and grow subtrees for each partition
    let $D_j$ be the set of data tuples in $D$ satisfying the outcome $j$; // a partition
    if $D_j$ is empty then
        attach a leaf labeled with the majority class in $D$ to node $N$;
    else attach the node returned by Generate_decision_tree($D_j$, attribute_list) to node $N$;
endfor
return $N$;
```

Figure 6.3 Basic algorithm for inducing a decision tree from training examples.

What is Entropy?

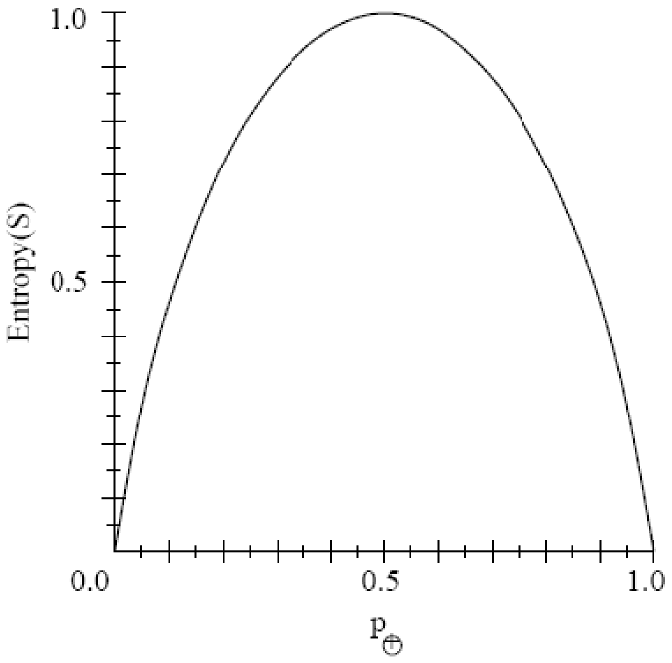
- The entropy is a measure of the uncertainty associated with a random variable
- As uncertainty and or randomness increases for a result set so does the entropy
- Values range from 0 – 1 to represent the entropy of information



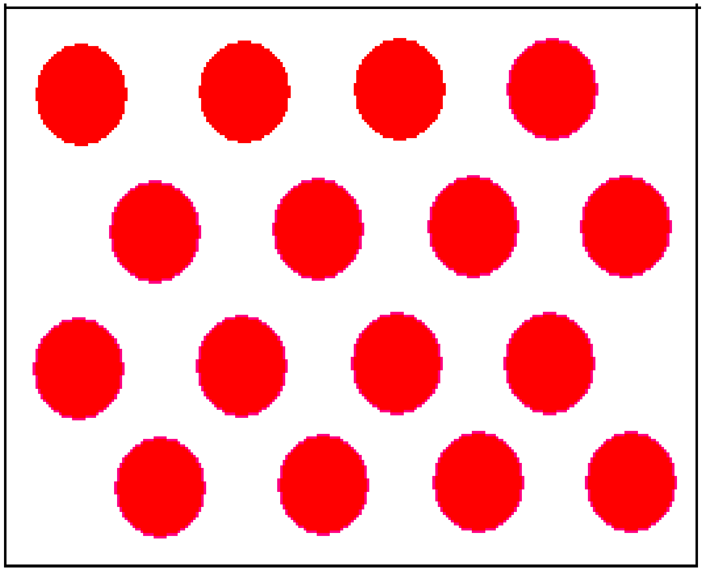
$$Entropy\,(D) \equiv \sum_{i=1}^{c} - p_i \log_2(p_i)$$

Entropy Example (1)

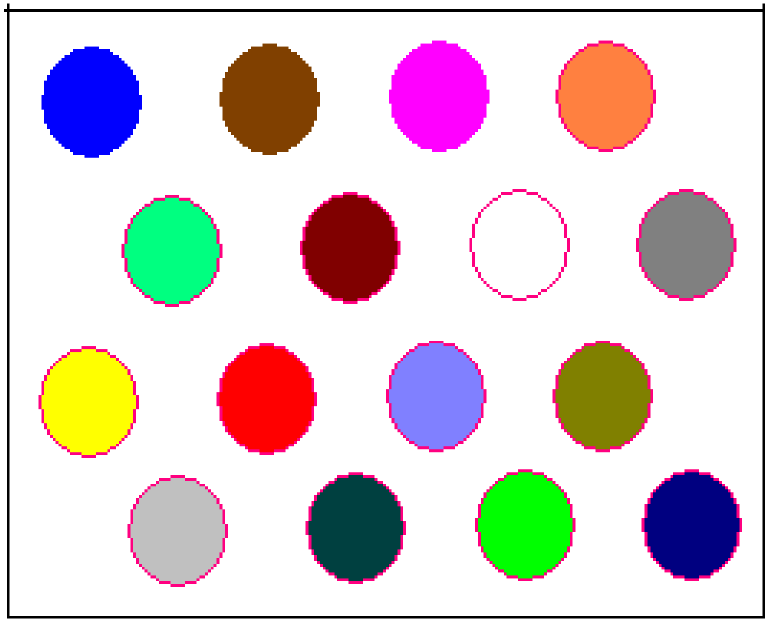
$$Entropy(D) \equiv -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$$



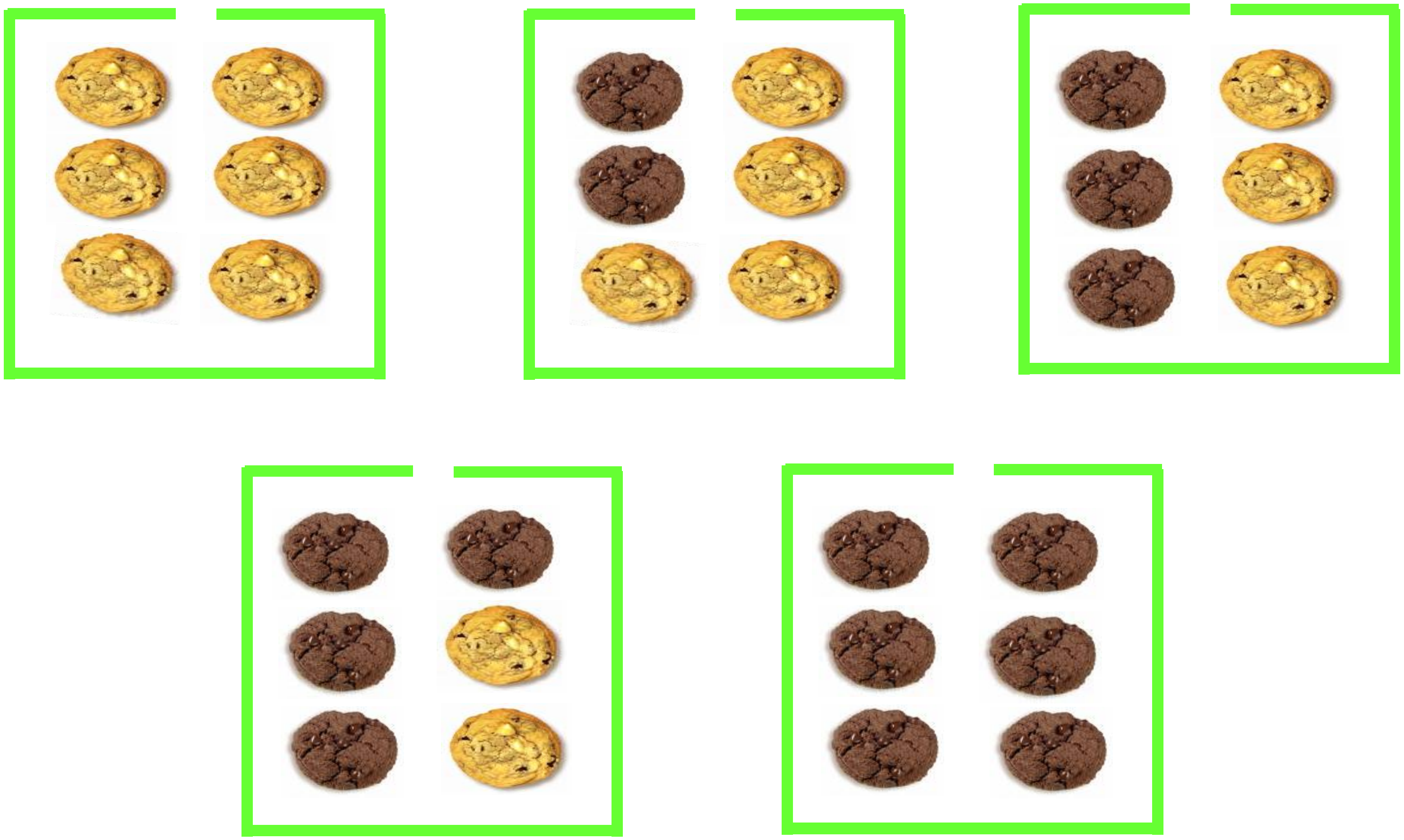
Entropy Example (2)



Entropy Example (3)



Entropy Example (4)



Information Gain

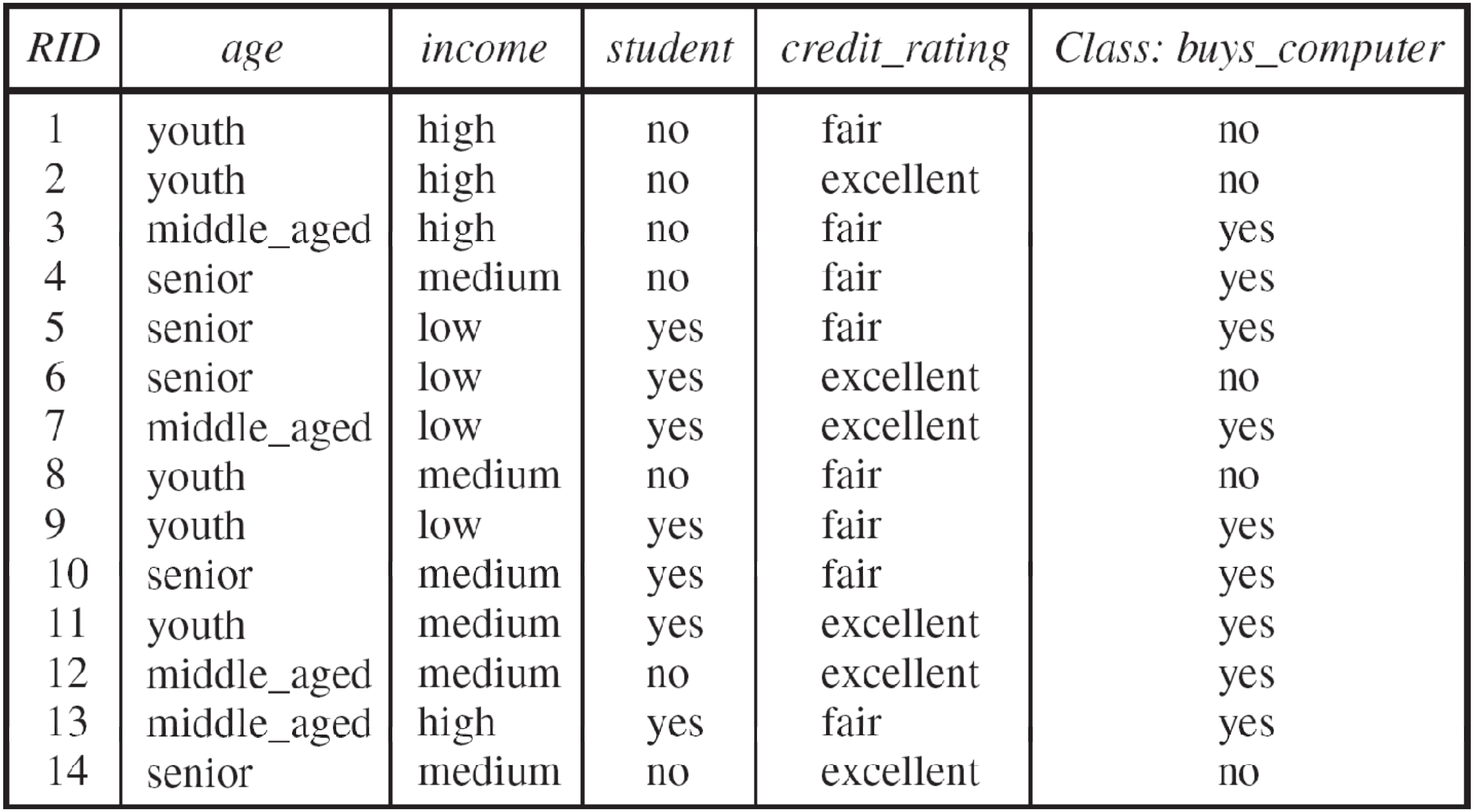
- Information gain is used as an attribute selection measure
- Pick the attribute that has the highest Information Gain

$$Gain(D,A) = Entropy(D) - \sum_{j=1}^{v} \frac{|D_j|}{|D|} Entropy(D_j)$$

D: A given data partition

A: Attribute

v: Suppose we were partition the tuples in ***D*** on some attribute ***A*** having ***v*** distinct values ***D*** is split into v partition or subsets, {***D_1, D2, … Dj***}, where ***Dj*** contains those tupes in ***D*** that have outcome ***a_j*** of ***A***.



RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Table 6.1 Class-labeled training tuples from AllElectronics customer database.

- Class P: *buys_computer* = “yes”
- Class N: *buys_computer* = “no”

$$Entropy(D) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

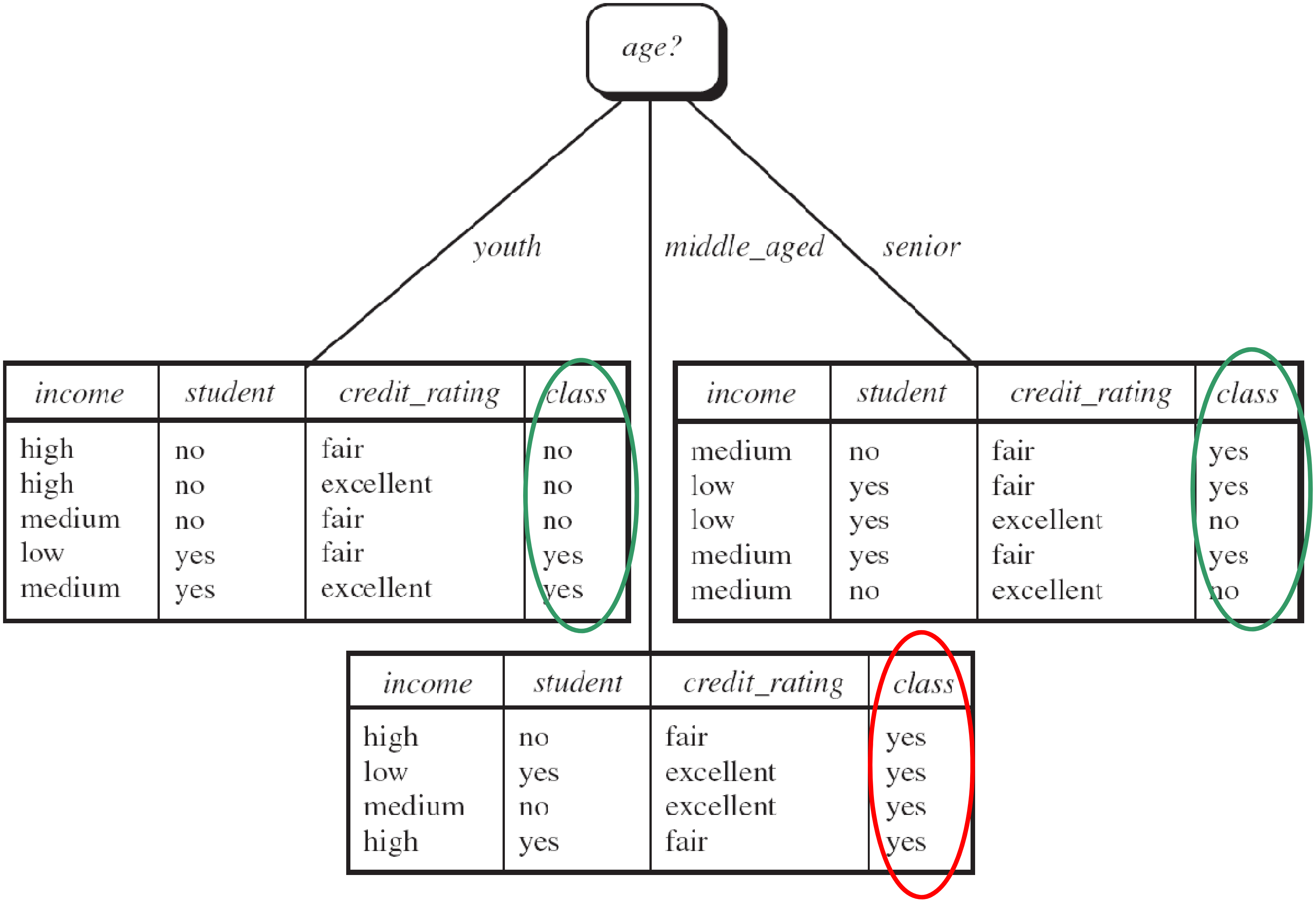
- Compute the expected information requirement for each attribute: start with the attribute *age*

$$\begin{aligned}
&Gain(age, D) \\
&= Entropy(D) - \sum_{v \in \{Youth, Middle-aged, Senior\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(D) - \frac{5}{14} Entropy(S_{youth}) - \frac{4}{14} Entropy(S_{middle\_aged}) - \frac{5}{14} Entropy(S_{senior}) \\
&= 0.246
\end{aligned}$$

$$Gain\,(income, D) = 0.029$$

$$Gain\,(student, D) = 0.151$$

$$Gain\,(credit\_rating, D) = 0.048$$



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Figure 6.5 The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of age. The tuples are shown partitioned accordingly.

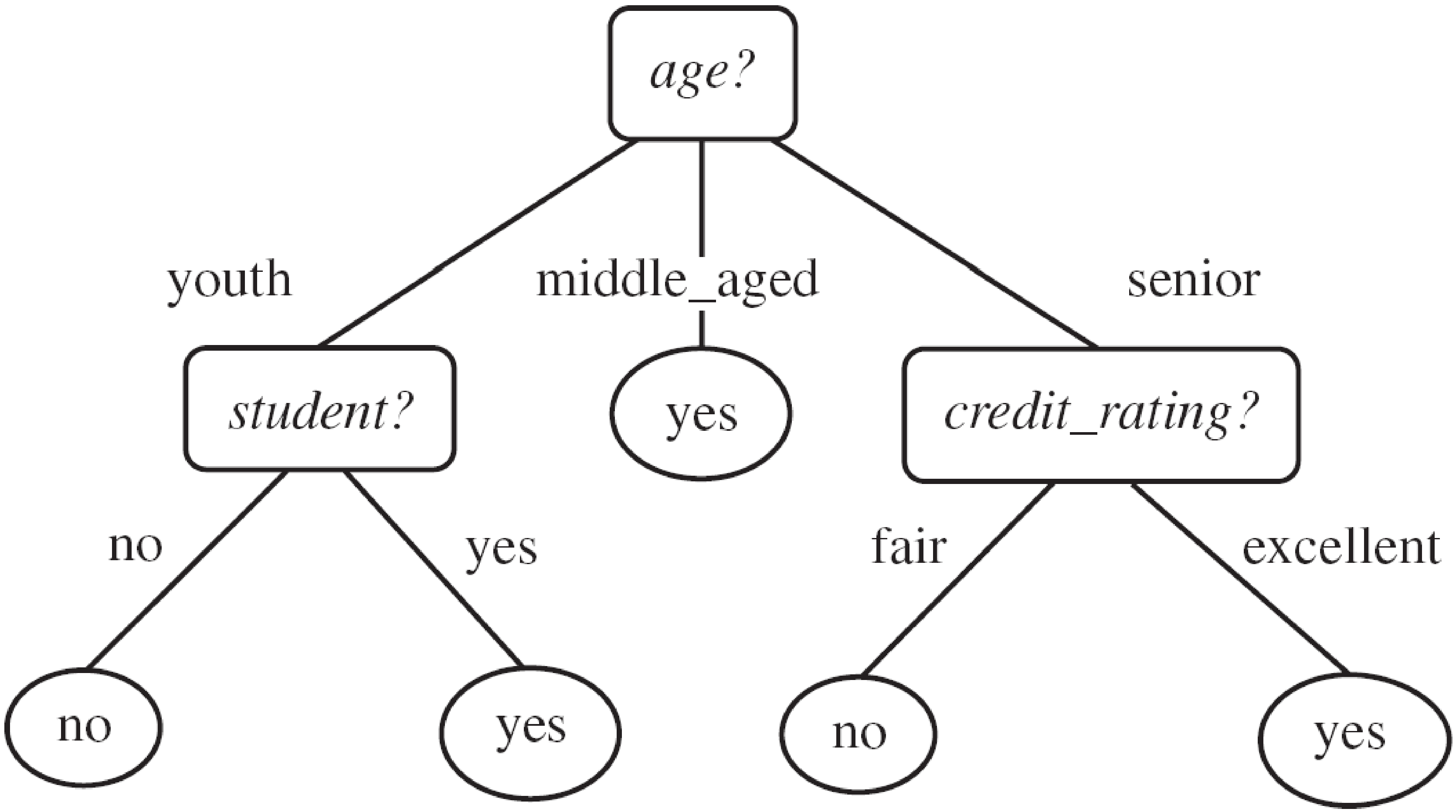
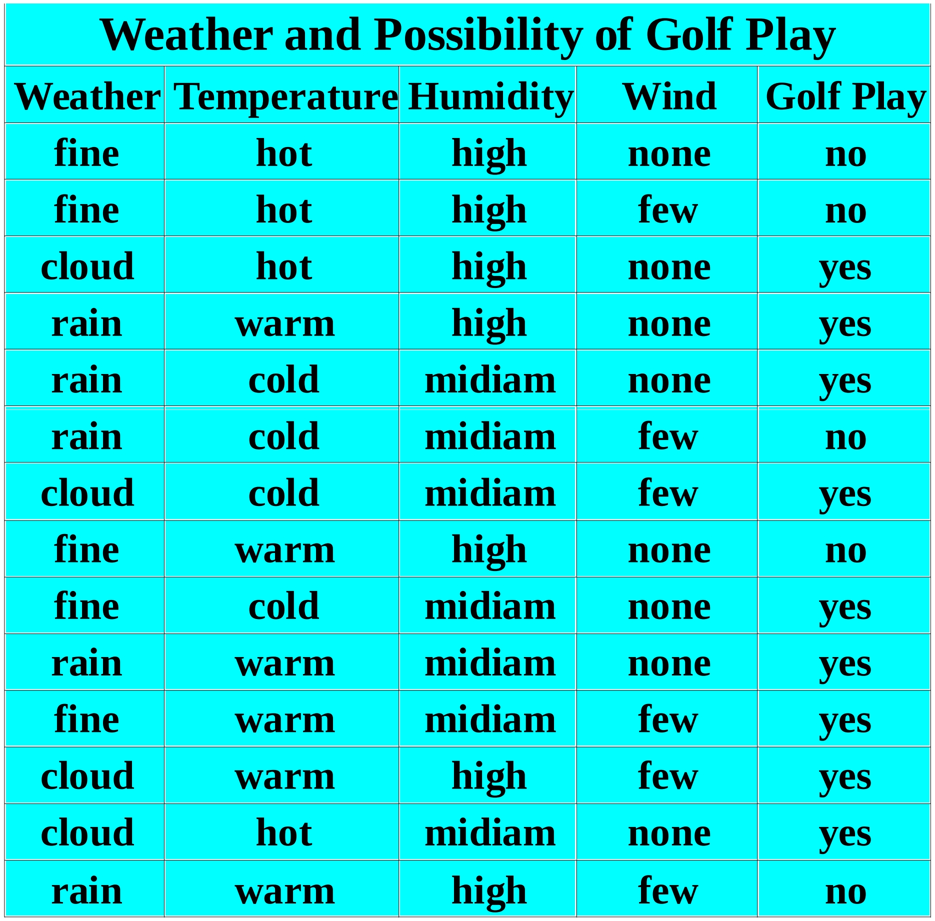


Figure 6.2 A decision tree for the concept *buys_computer*, indicating whether a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer* = *yes* or *buy_computers* = *no*.

Exercise

Construct a decision tree to classify “golf play.”



Weather and Possibility of Golf Play				
Weather	Temperature	Humidity	Wind	Golf Play
fine	hot	high	none	no
fine	hot	high	few	no
cloud	hot	high	none	yes
rain	warm	high	none	yes
rain	cold	midiam	none	yes
rain	cold	midiam	few	no
cloud	cold	midiam	few	yes
fine	warm	high	none	no
fine	cold	midiam	none	yes
rain	warm	midiam	none	yes
fine	warm	midiam	few	yes
cloud	warm	high	few	yes
cloud	hot	midiam	none	yes
rain	warm	high	few	no