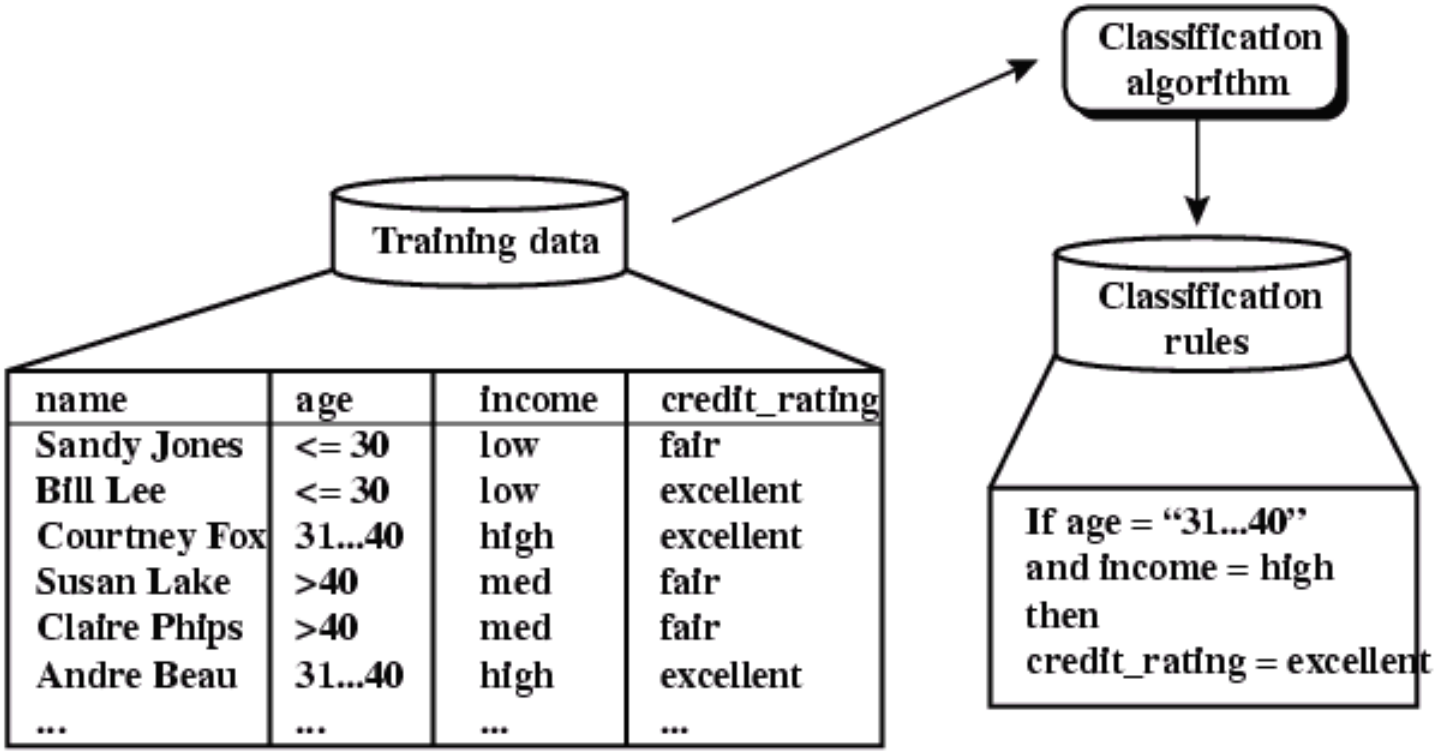
資料分類技術

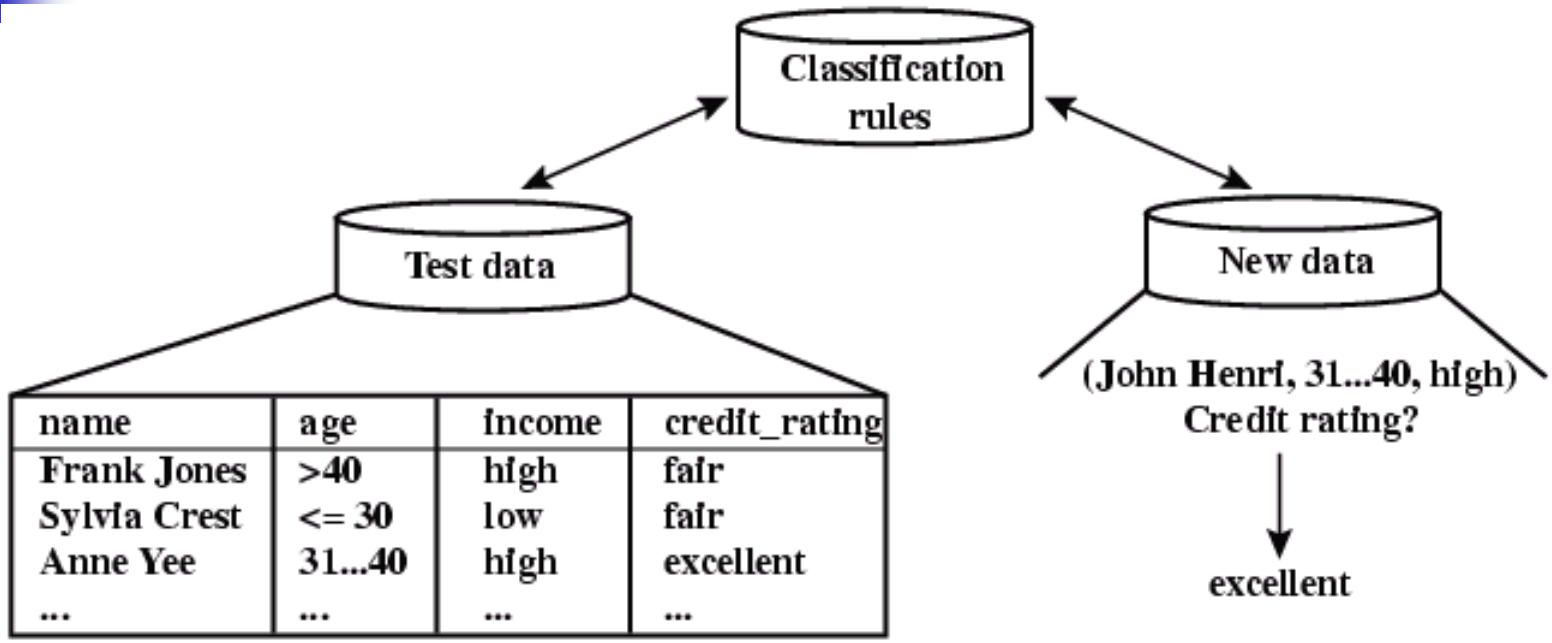
國立宜蘭大學資訊工程系
吳政瑋 助理教授
wucw@niu.edu.tw

Learning Phase



- Learning
  - The class label attribute is credit_rating
  - Training data are analyzed by a classification algorithm
  - The classifier is represented in the form of classification rules

Testing Phase

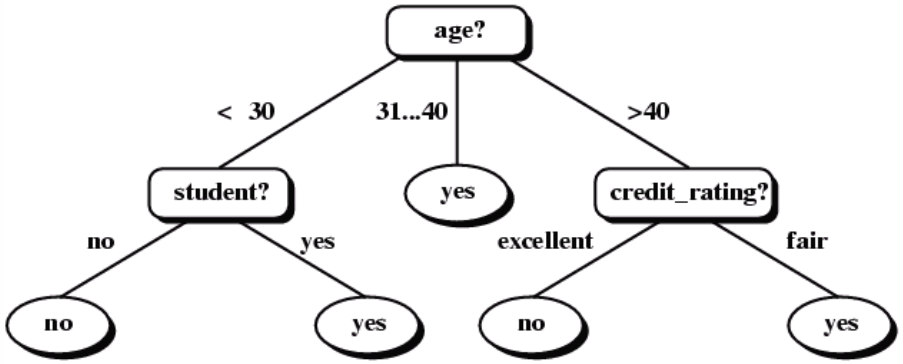


- Testing (Classification)
  - Test data are used to estimate the accuracy of the classification rules
  - If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples

Classification by ID3 Decision Tree

Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

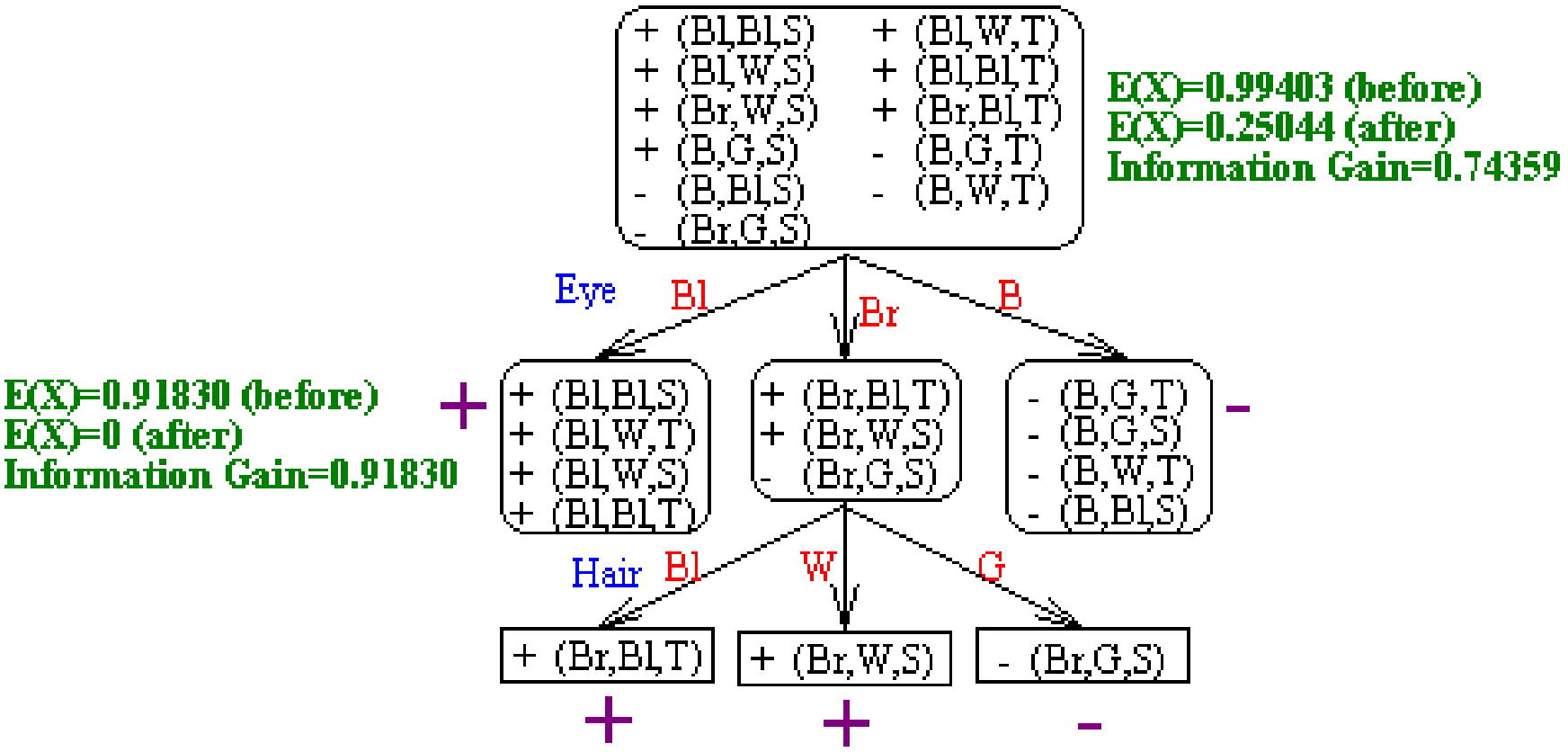


A top-down decision tree generation algorithm: ID-3

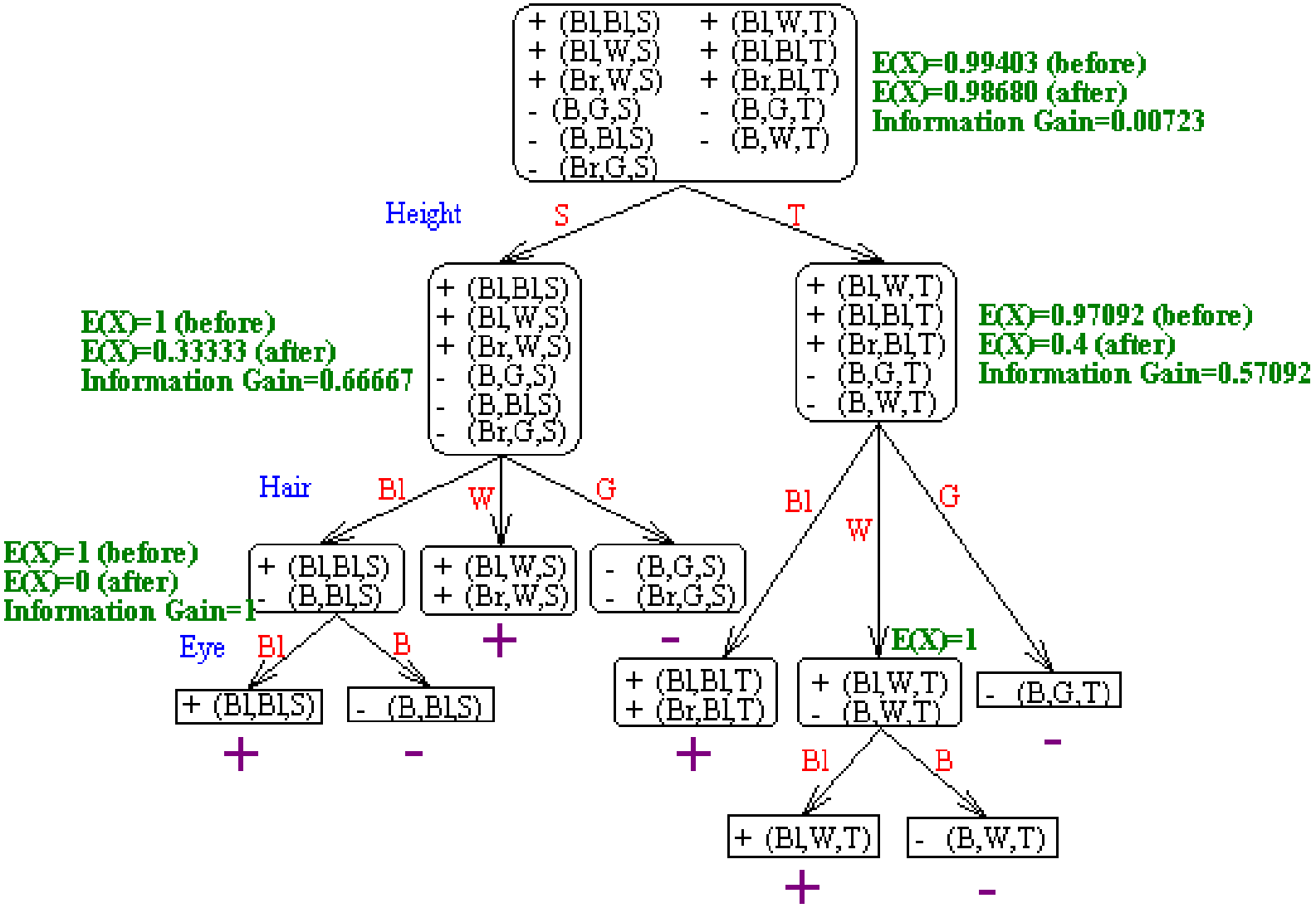
Example Data 1 (東方人西方人資料集)

	Eye	Hair	Height	Oriental
1	Black	Black	Short	Yes
2	Black	White	Tall	Yes
3	Black	White	Short	Yes
4	Black	Black	Tall	Yes
5	Brown	Black	Tall	Yes
6	Brown	White	Short	Yes
7	Blue	Gold	Tall	No
8	Blue	Gold	Short	No
9	Blue	White	Tall	No
10	Blue	Black	Short	No
11	Brown	Gold	Short	No

A Good Decision Tree



A Bad Decision Tree



Example Data 2
(全國電子買電腦資料集)

Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Decision Tree Generation Algorithm: ID3 (Cont. 1/5)

ID: Iterative Dichotomiser

Let S be a set consisting of s data samples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1, \ldots, m$). Let s_i be the number of samples of S in class C_i. The expected information needed to classify a given sample is given by

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i), \qquad (7.1)$$

◀ **Entropy**

where p_i is the probability that an arbitrary sample belongs to class C_i and is estimated by s_i/s. Note that a log function to the base 2 is used since the information is encoded in bits.

Decision Tree Generation Algorithm: ID3 (Cont. 2/5)

Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

The class label attribute, *buys_computer*, has two distinct values (namely, {*yes*, *no*}); therefore, there are two distinct classes ($m = 2$). Let class C_1 correspond to *yes* and class C_2 correspond to *no*. There are 9 samples of class *yes* and 5 samples of class *no*. To compute the information gain of each attribute, we first use Equation (7.1) to compute the expected information needed to classify a given sample:

$$I(s_1, s_2) = I(9, 5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940.$$

Decision Tree Generation Algorithm: ID3 (Cont. 3/5)

Next, we need to compute the entropy of each attribute. Let's start with the attribute *age*. We need to look at the distribution of *yes* and *no* samples for each value of *age*. We compute the expected information for each of these distributions.

For *age* = "<=30":

$s_{11} = 2 \quad s_{21} = 3 \quad I(s_{11}, s_{21}) = 0.971$

For *age* = "31 . . . 40":

$s_{12} = 4 \quad s_{22} = 0 \quad I(s_{12}, s_{22}) = 0$

For age = ">40":

$s_{13} = 3 \quad s_{23} = 2 \quad I(s_{13}, s_{23}) = 0.971$

the expected information needed to classify a given sample

$$E(age) = \frac{5}{14} I(s_{11}, s_{21}) + \frac{4}{14} I(s_{12}, s_{22}) + \frac{5}{14} I(s_{13}, s_{23}) = 0.694$$

Decision Tree Generation Algorithm: ID3 (Cont. 4/5)

Hence, the gain in information from such a partitioning would be

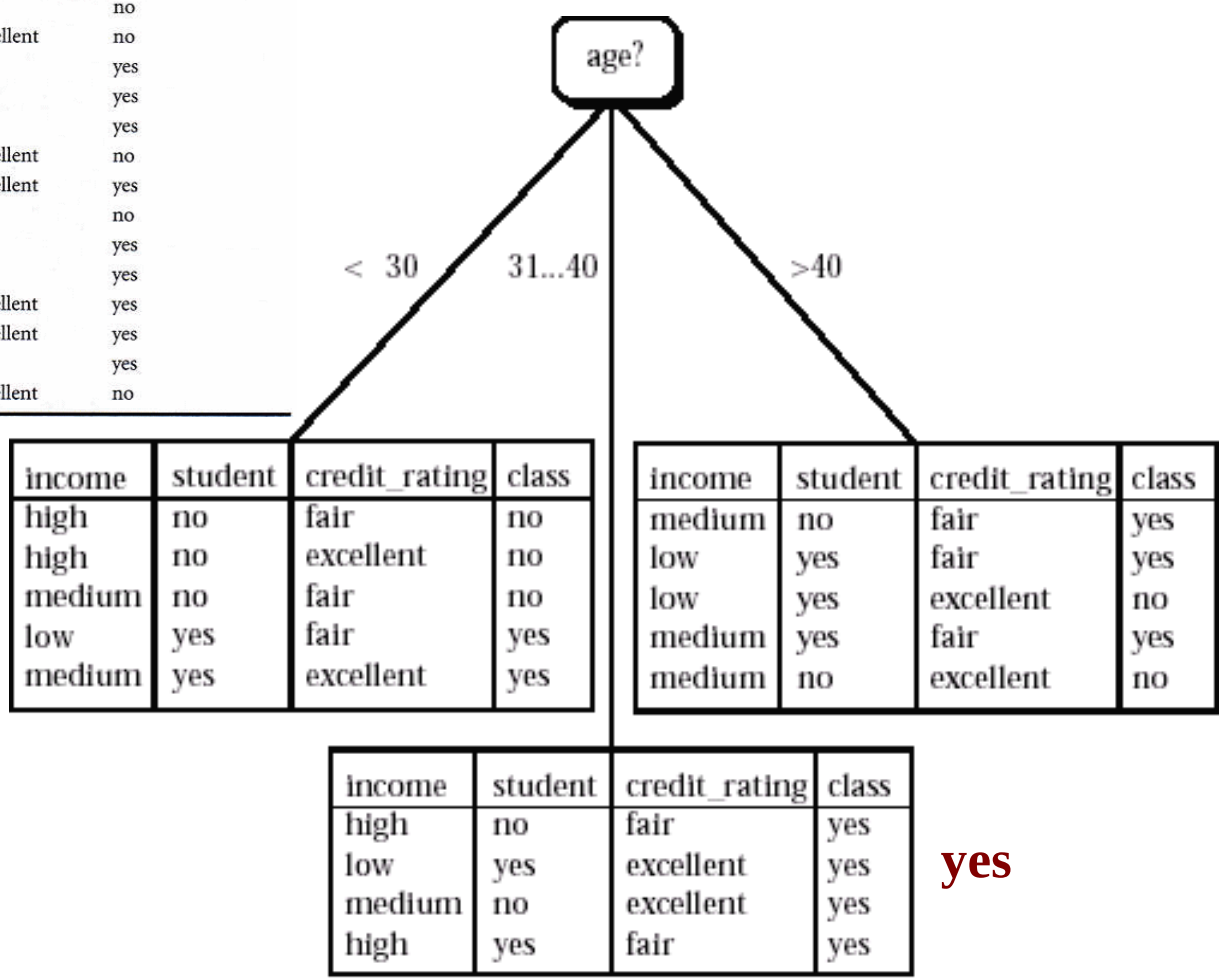
$$Gain(age) = I(s_1, s_2) - E(age) = 0.246.$$

Similarly, we can compute *Gain(income)* = 0.029, *Gain(student)* = 0.151, and *Gain(credit_rating)* = 0.048. Since *age* has the highest information gain among the attributes, it is selected as the test attribute. A node is created and labeled with *age*, and branches are grown for each of the attribute's values. The samples falling into the partition for *age* = "*31 . . . 40*" all belong to the same class. Since they all belong to class *yes*, a leaf should therefore be created at the end of this branch and labeled with *yes*.

Decision Tree Generation Algorithm: ID3 (Cont. 5/6)

Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no



Decision Tree Generation Algorithm: ID3 (Cont. 6/6)