Classification -Part2
[Attribute selection measures]

Shubham kumar

Dept. of CS&IT

MGCUB

Attribute Selection Measures

- It is a heuristic approach to select the best splitting criterion that separates a given data partition, D, of class-labeled training tuples into individual classes.

- Splitting criterion is called the best when after splitting, each partition will be pure.

- A partition is called pure when all the tuples that fall into the partition belongs to the same class.

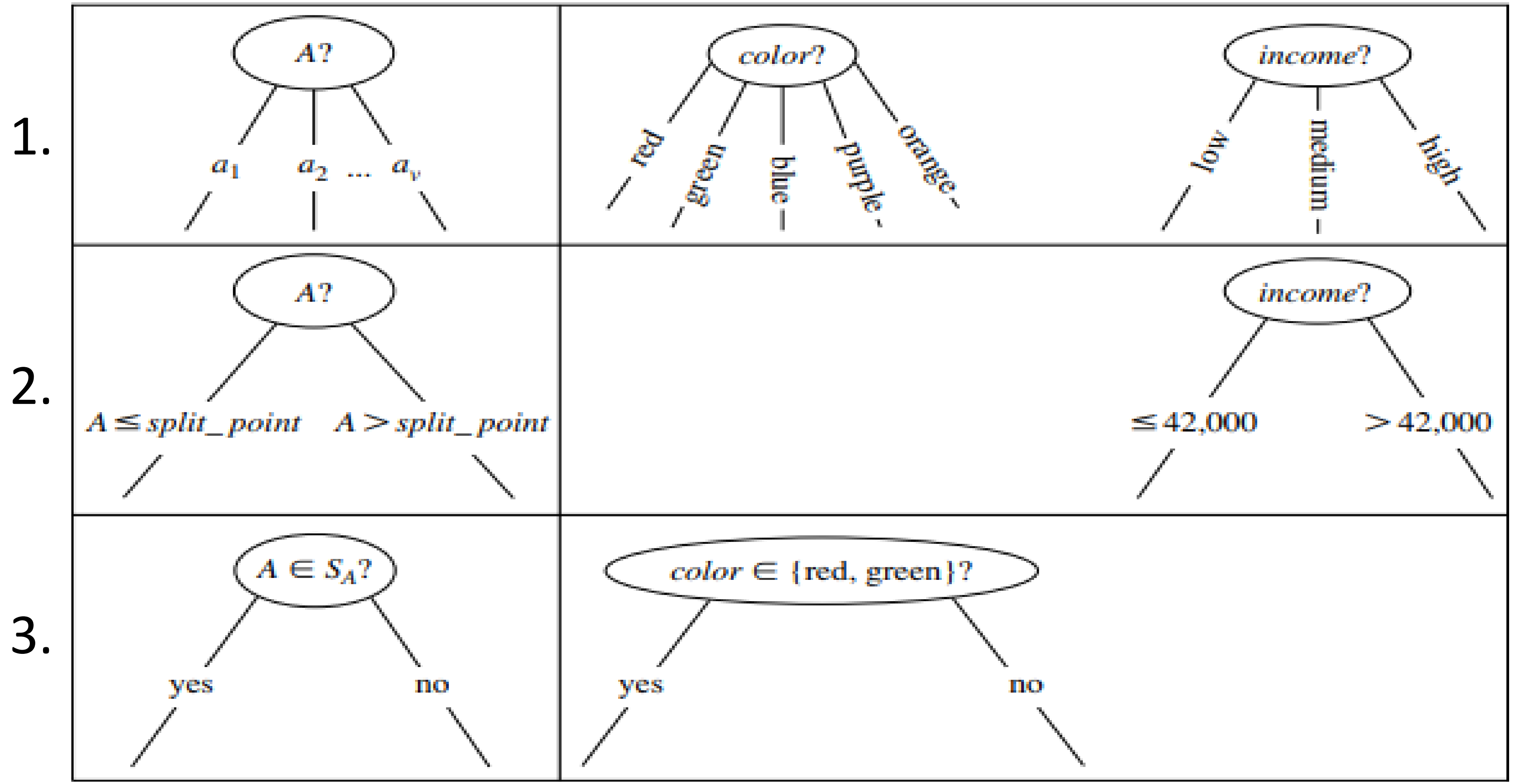
- Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split.

- First, a rank is provided for each attribute that describes the training tuples. And the attribute having the best score for the measure is chosen as the splitting attribute for the given tuples.

- If the splitting attribute is continuous-valued or if we are restricted to binary trees, then respectively either a split point or a splitting subset must also be determined as part of the splitting criterion.

partition scenarios

Examples



1. **A is discrete-valued**: In this case, the outcomes of the test at node N correspond directly to the known values of A. A branch is created for each known value, a_j, of A and labeled with that value (as in the figure). Partition D_j is the subset of class-labeled tuples in D having value a_j of A.

2. **A is continuous-valued:** In this case, the test at node N has two possible outcomes, corresponding to the conditions **A ≤ split point and A > split point**, respectively, where split point is the split-point returned by Attribute selection method as part of the splitting criterion.

3. If A is discrete-valued and a binary tree must be produced, then the test is of the form $A \in S_A$, where S_A is the splitting subset for A.

- According to the algorithm the tree node created for partition D is labeled with the splitting criterion, and the tuples are partitioned accordingly. [Also Shown in the figure].
- There are three popular attribute selection measures: Information Gain, Gain ratio, and, Gini index.
- **Information gain:**

The attribute with the highest information gain is chosen as the splitting attribute.

This attribute minimizes the information needed to classify the tuples in the resulting partitions.

Let D, the data partition, be a training set of class-labeled tuples.

let class label attribute has m distinct values defining m distinct classes, C_i (for i = 1,..., m). Let $C_{i,D}$ be the set of tuples of class C_i in D. Let $|D|$ and $|C_{i,D}|$ denote the number of tuples in D and $C_{i,D}$, respectively.

- Then the expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

- where p_i is the nonzero probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. Info(D) is the average amount of information needed to identify the class label of a tuple in D. Info(D) is also known as the entropy of D.
- Now, suppose we have to partition the tuples in D on some attribute A having v distinct values, $\{a_1, a_2, \ldots, a_v\}$.

Then the expected information required to classify the tuple from D based on attribute A is:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- The term $|D_j| / |D|$ acts as the weight of the j th partition. Info A (D) is the expected information required to classify a tuple from D based on the partitioning by A.

- Information gain is defined as the difference between the original information requirement and the new requirement (i.e. obtained after portioning on A).

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

Now, the attribute A with the highest information gain is chosen as the splitting attribute.

Example:

This is a training set D, of class-labeled tuples randomly selected from the AllElectronics customer database.

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	Mdium	No	Excellent	no

- Here, the class label attribute, buys computer, has two distinct values: yes & no.
- Therefore, there are two distinct classes (i.e., m = 2). Let class C1 correspond to yes and class C2 correspond to no.
- There are nine tuples of class yes and five tuples of class no.
- A (root) node N is created for the tuples in D.
- To find the splitting criterion for these tuples, we must compute the information gain of each attribute.

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

- Next, we need to compute the expected information requirement for each attribute.

(1) Age: We need to look at the distribution of yes and no tuples for each category of age. For category “youth,” there are two yes tuples and three no tuples. For the category “middle_aged,” there are four yes tuples and zero no tuples. For the category “senior,” there are three yes tuples and two no tuples.

Therefore, the expected information needed to classify a tuple in D if the tuples are partitioned according to age is:

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} \right)$$

$$
\begin{aligned}
= \quad & + \frac{4}{14} \times \left(-\frac{4}{4}\log_2 \frac{4}{4}\right) \\
& + \frac{5}{14} \times \left(-\frac{3}{5}\log_2 \frac{3}{5} - \frac{2}{5}\log_2 \frac{2}{5}\right) \\
& = 0.694 \text{ bits.}
\end{aligned}
$$

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

Gain(age) = Info(D) - Info_{Age} (D)

= 0.940 – 0.694 = 0.246 bits

Similarly,

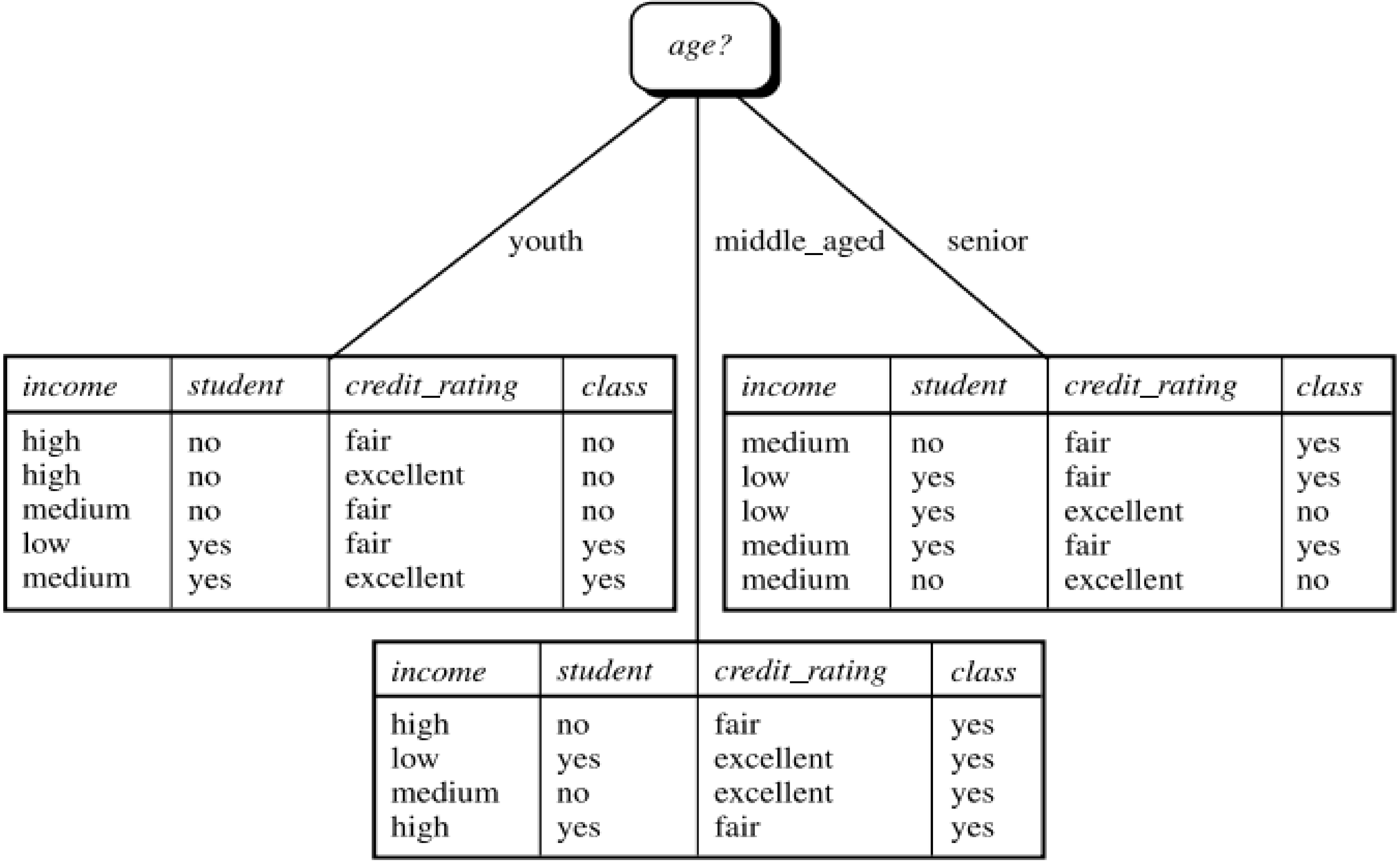
Gain(income) = 0.029 bits

Gain (student)= 0.151 bits, and Gain(credit_rating) = 0.048 bits.

- Since Age has the highest information gain among the attributes, therefore it is selected as the splitting attribute.
- According to the decision tree algorithm, Node N is labeled with age, and branches are grown for each of the attribute’s values and the tuples are then partitioned accordingly.

[Shown in the next figure]

- Here, the tuples falling into the partition for age = middle_aged, 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 “yes.”



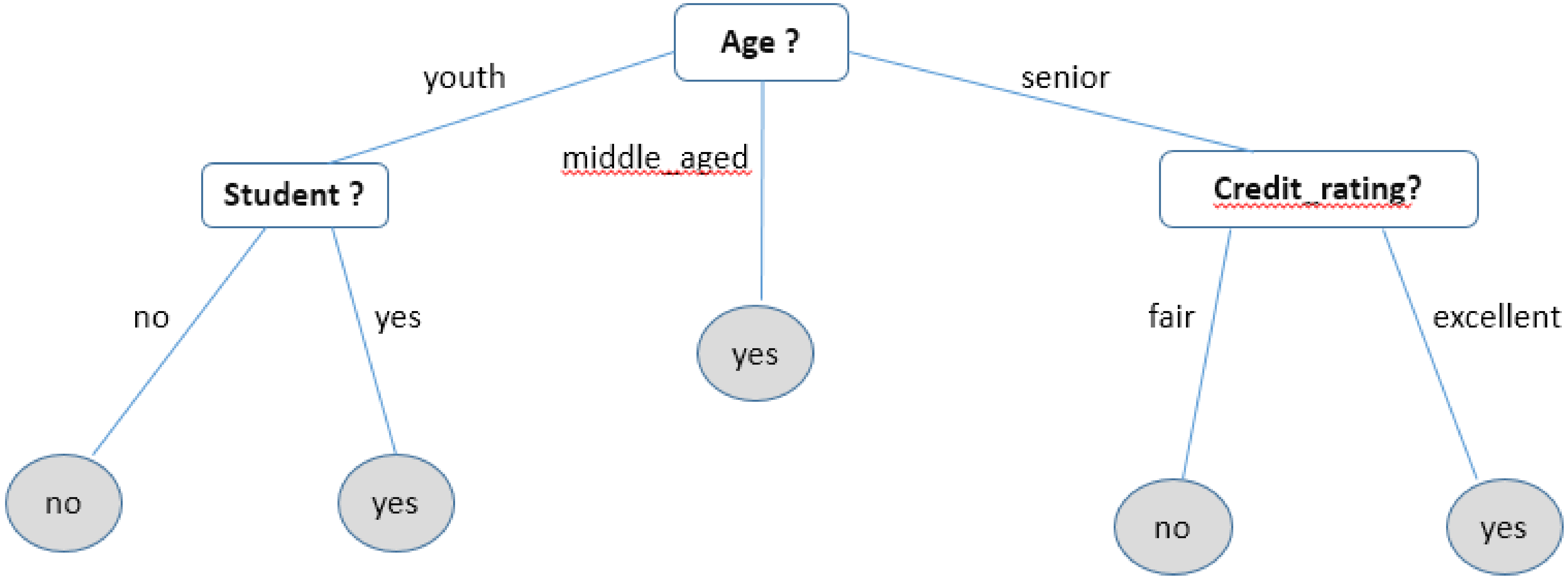
income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

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

But for the other resulting partition where classes are not same, decision tree algorithm uses the same splitting process recursively to form a decision tree.

Final decision tree would be: (Using Information gain as attribute selection measure.)



Reference

- Jiawei Han, Micheline kamber and Jian pei. “DATA MINING concepts and Techniques” 3/e, Elsevier, 2012