Unit 4: Classification and Prediction

What is Classification?

- Is a data mining technique used to predict the category of categorical data by building a model based on some predictor variables(to classify data).
- Predicator variable/ attribute is called class label attribute(predefined class)
- Following are the **examples** of cases where the data analysis task is Classification:
- A bank loan officer wants to analyze the data in order to know which customers (loan applicant) are risky or which are safe.

What is classification?

- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.
- In both of the above examples, a model or classifier is constructed to predict the categorical labels.
- These labels are risky or safe for loan application data and yes or no for marketing data.

How does classification work?

- It is a two-step process:

  1. Model Construction (learning step or training phase)

     -build a model to explain the target concept

     -model is represented as classification rules, decision trees, or mathematical formulae.

  2. Model Usage

     -is used for classifying future or unknown cases

     -estimate the accuracy of the model

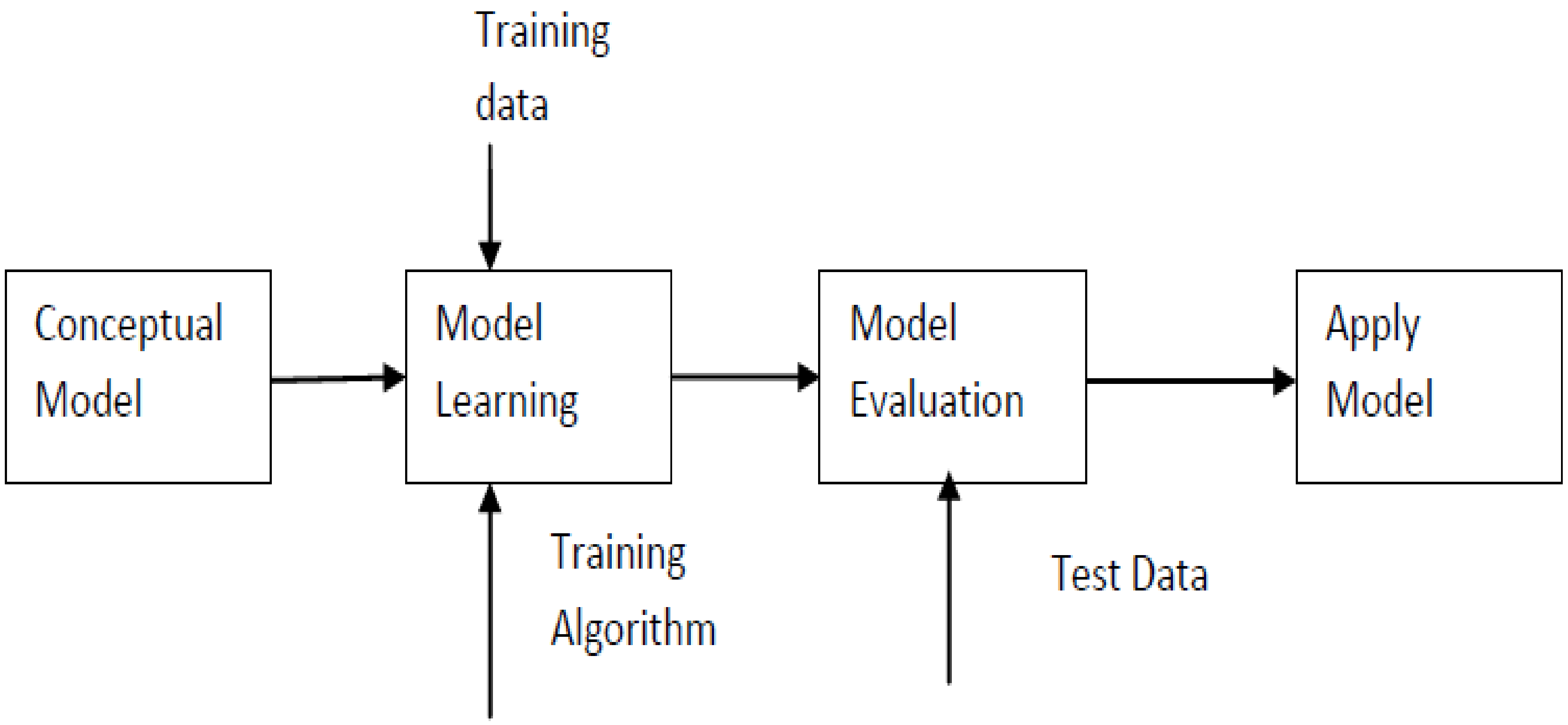
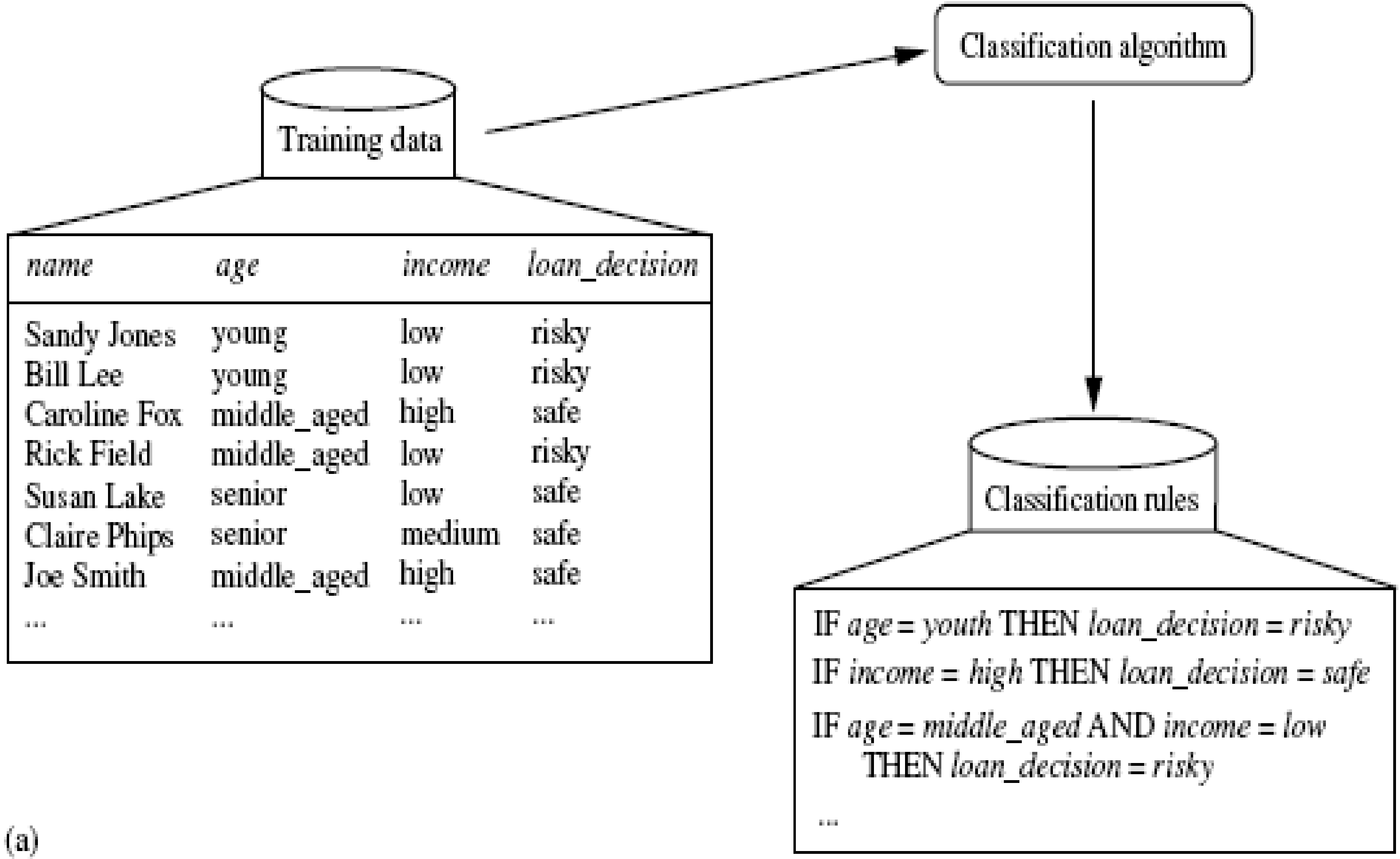


Fig: Stages in classification

Building the Classifier or Model

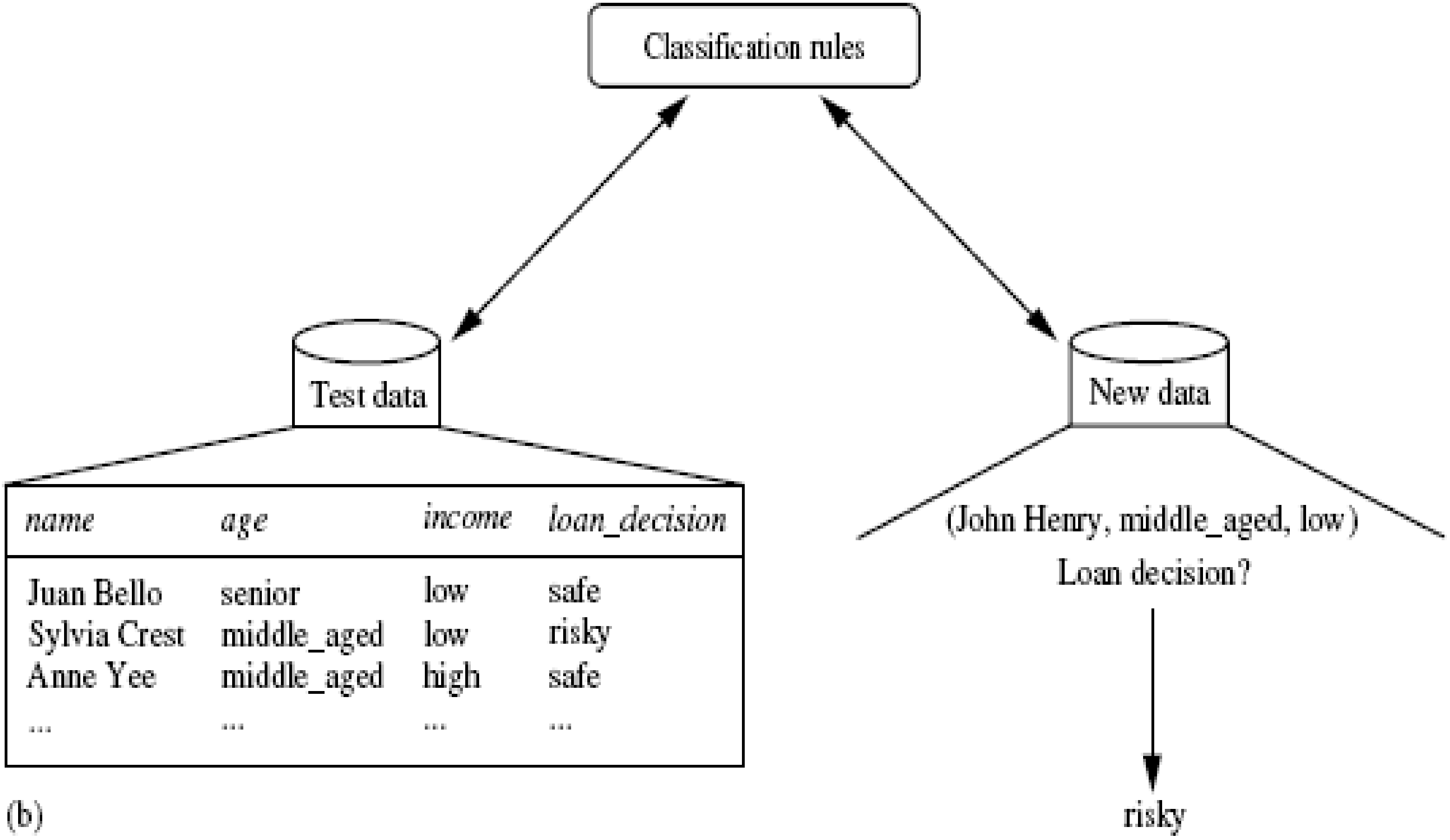
For example, Consider the following set training data:

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


Training data
Classification algorithm
Classification rules
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
...

(a)

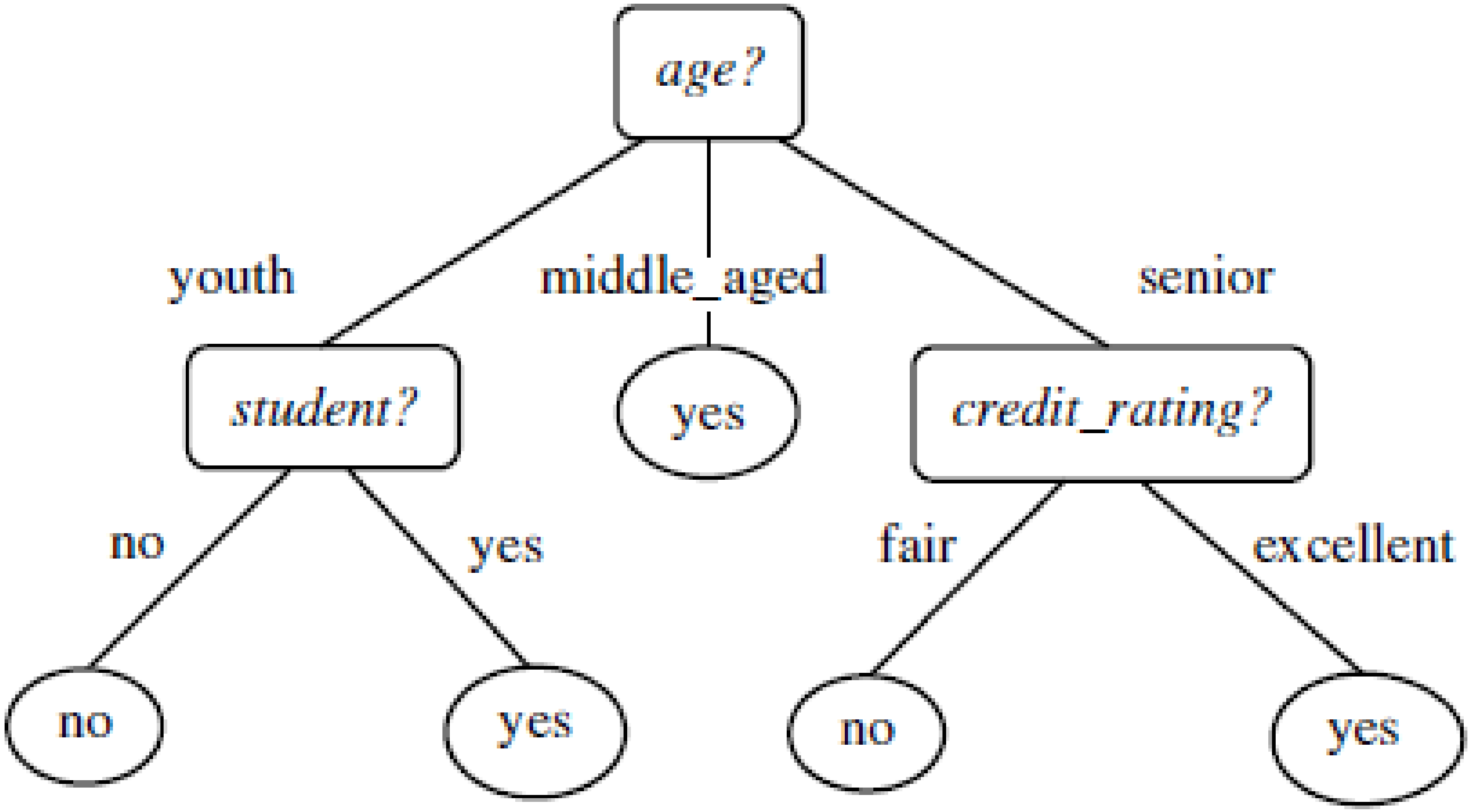
Using Classifier for Classification



(b)

Decision Tree classifier

- A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label.
- The topmost node in a tree is the root node.


age?
youth
middle_aged
senior
student?
yes
credit_rating?
no
yes
fair
excellent
no
yes
no
yes

How to Build Decision Tree?

- Generally, building a decision tree involved 2 steps:

1. Tree construction: recursively split the tree according to selected attributes (conditions)
2. Tree pruning: identify and remove the irrelevance branches – to increase classification accuracy.

How to Build Decision Tree?

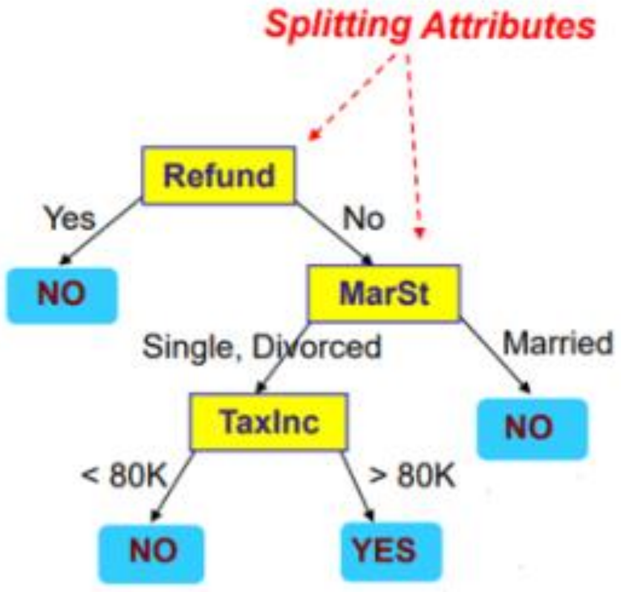
- In principle, there are exponentially many decision tree that can be construct from a given set of attributes.
- Finding the optimal tree is computationally infeasible because of the exponential size of the search space.
- Efficient algorithms has been develop to induce reasonably accurate, albeit suboptimal, decision tree in a reasonable amount of time.
- These algorithm usually employ a greedy strategy-making a series of locally optimal decisions about which attribute to use for partitioning the data.

Example of a Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

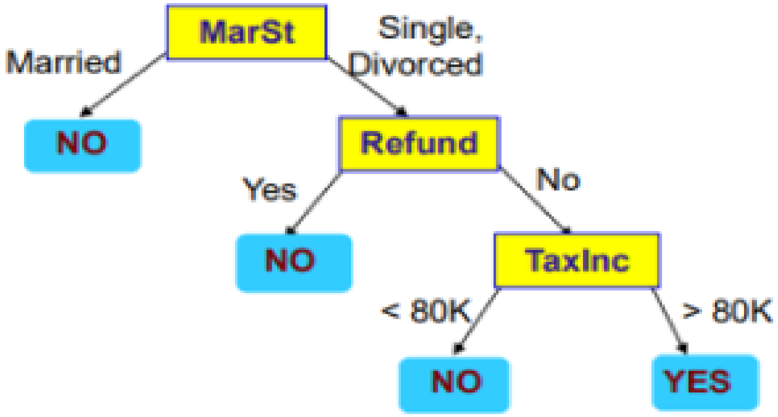
Training Data



Model: Decision Tree

Another Example of Decision Tree

Tid	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

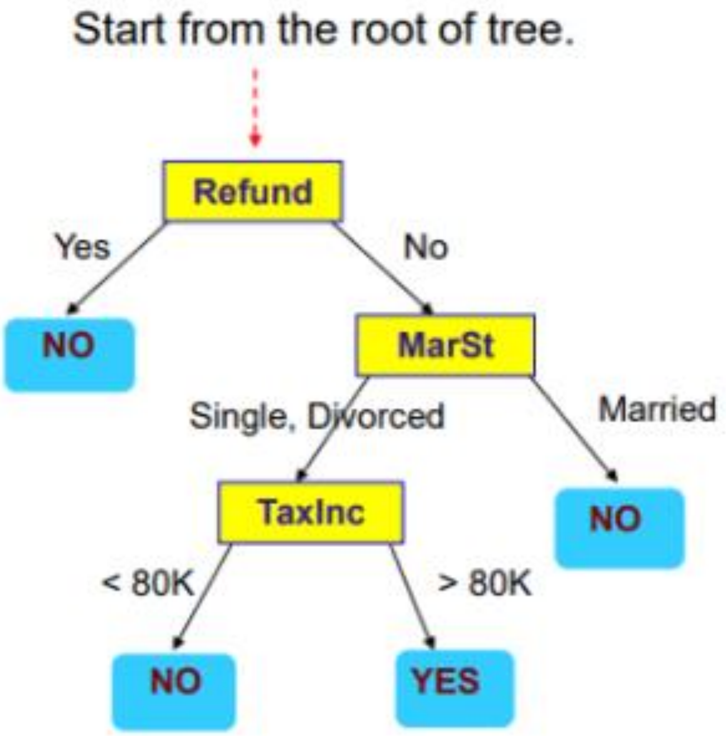


There could be more than one tree that fits the same data!

How is decision trees used for classification?

- Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree.
- A path is traced from the root to a leaf node, which holds the class prediction for that tuple.
- Decision trees can easily be converted to classification rules.

Apply Model to Test Data



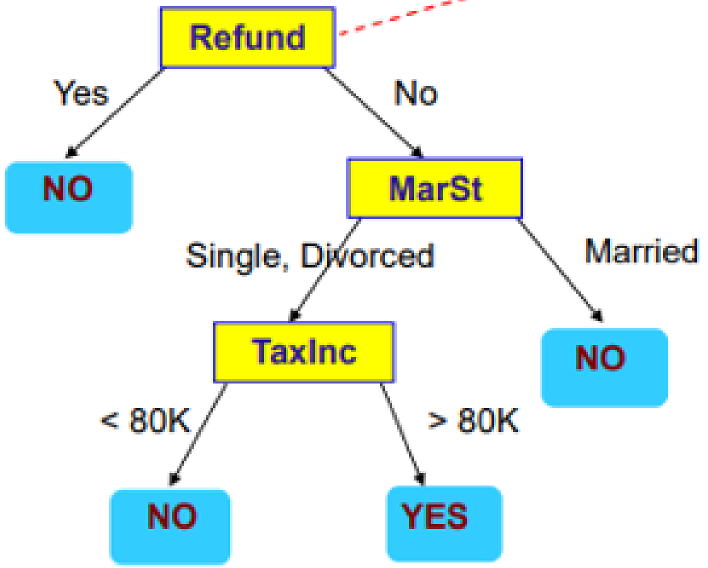
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

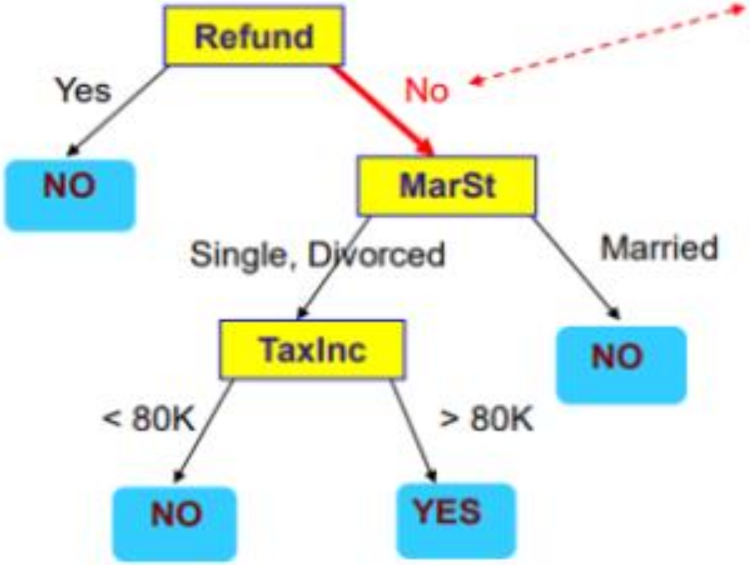
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



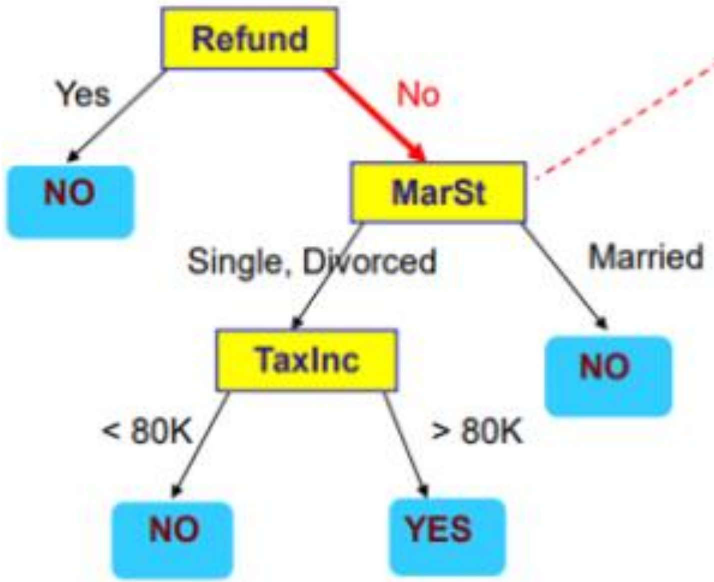
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



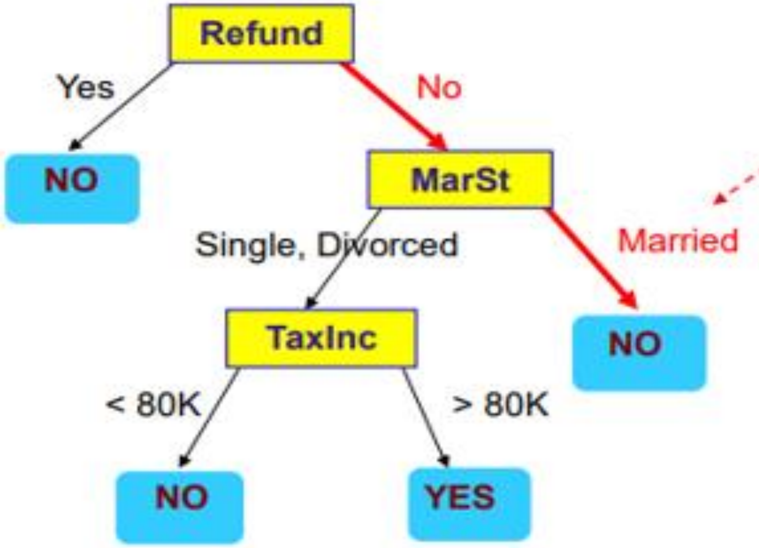
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

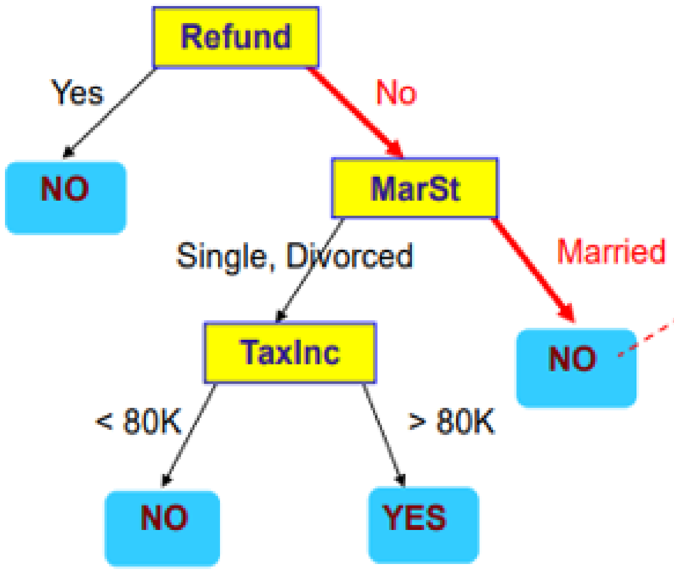
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

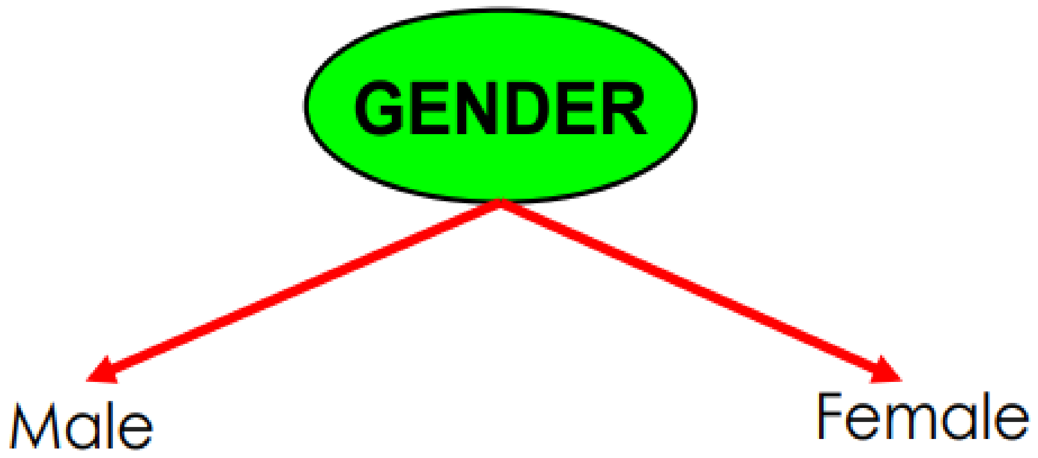
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

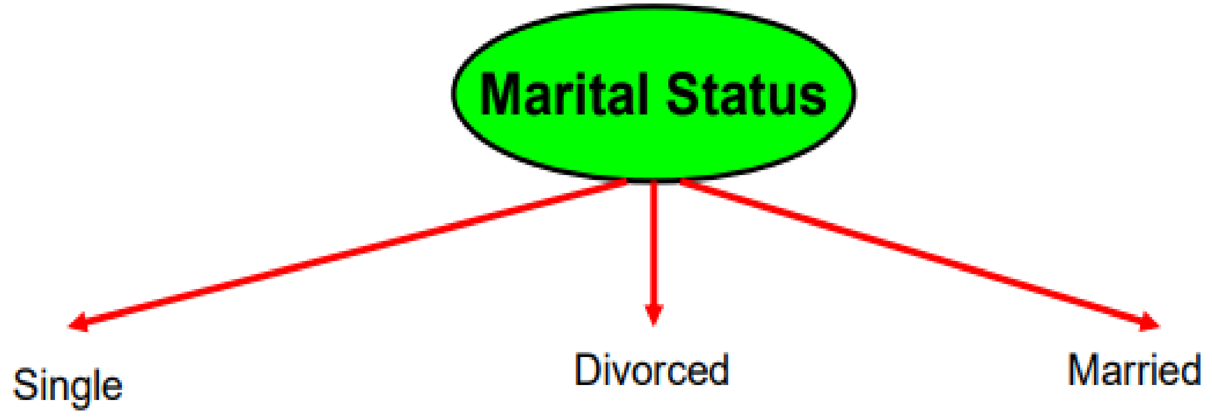
Methods for Expressing Attribute Test Condition

a) **Binary Attributes** → generates two possible outcomes (binary split)



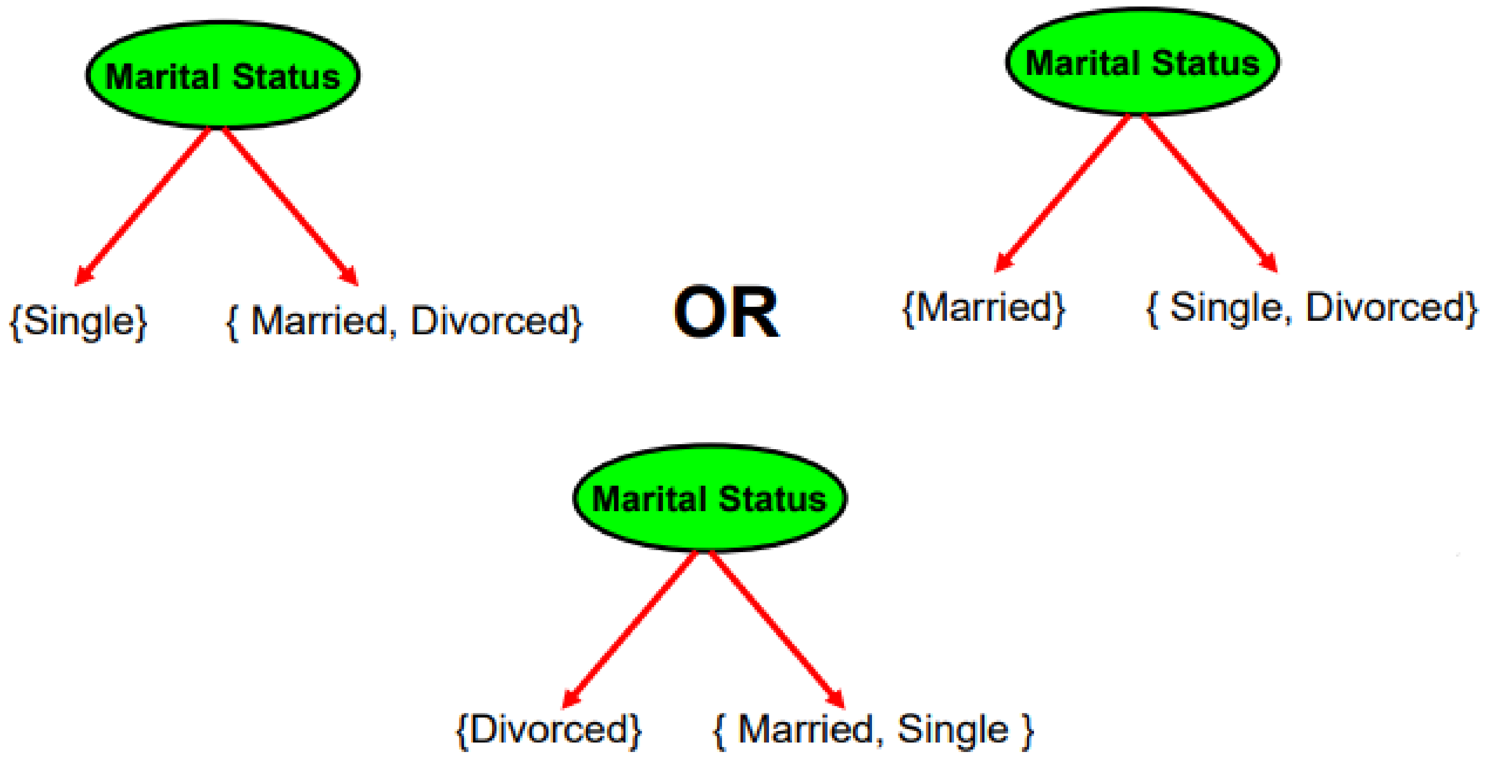
Methods for Expressing Attribute Test Condition

b) **Nominal Attributes** : Multiway split



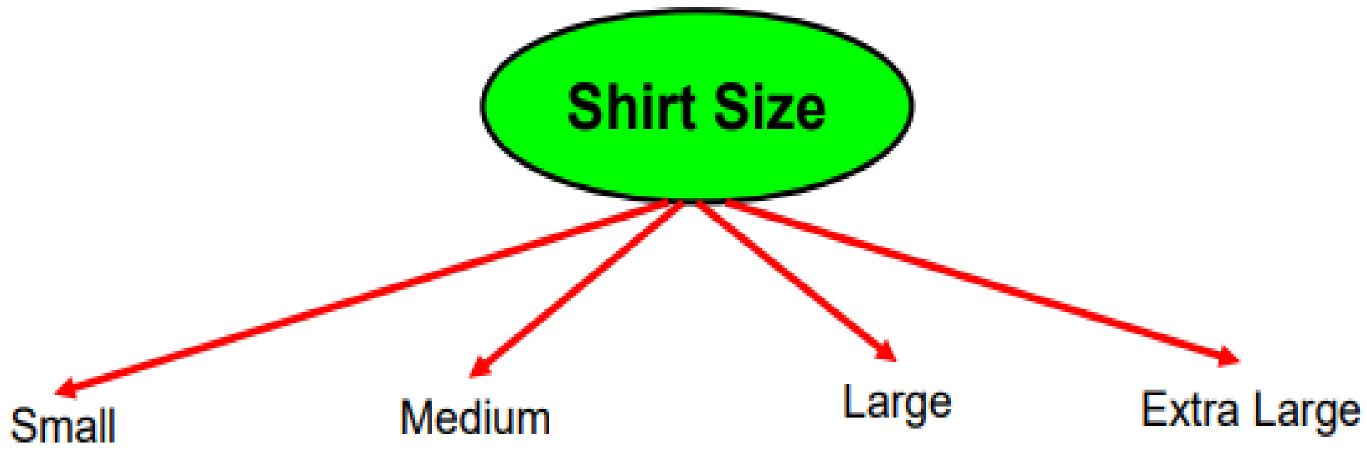
Methods for Expressing Attribute Test Condition

b) **Nominal Attributes** : Binary split (eg : in CART)



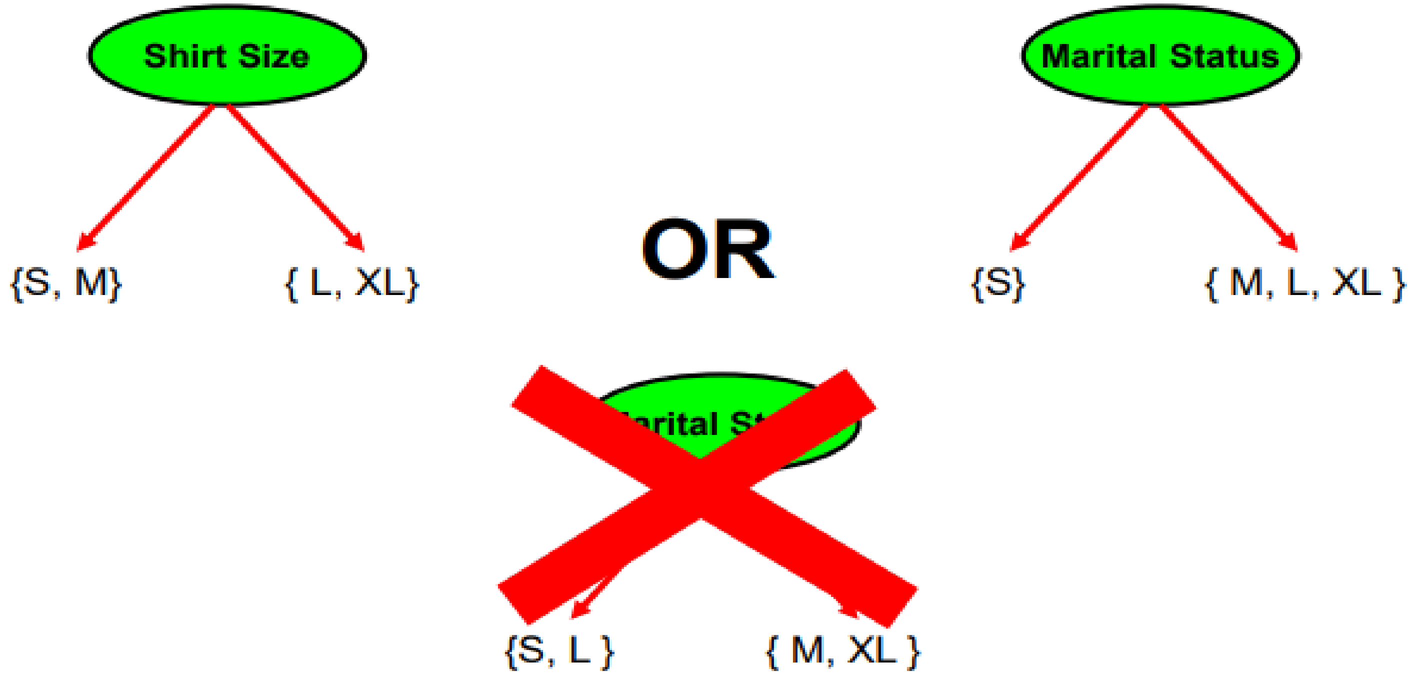
Methods for Expressing Attribute Test Condition

c) **Ordinal Attributes** : Multiway split



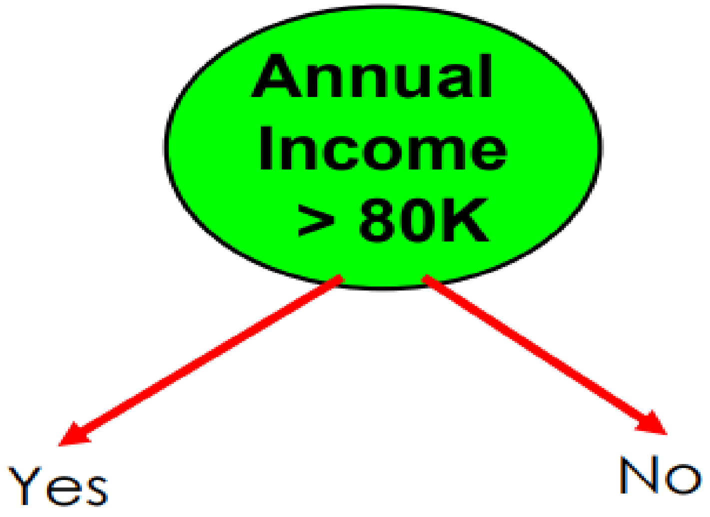
Methods for Expressing Attribute Test Condition

c) Ordinal Attributes : Binary split – as long as it does not violate the order property of the attribute values.



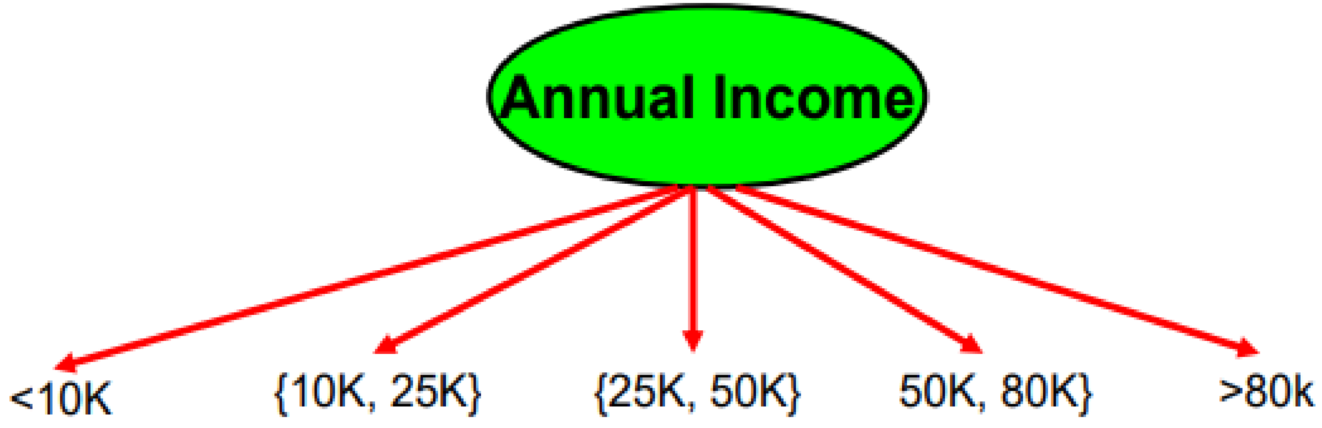
Methods for Expressing Attribute Test Condition

d) Continuous Attributes → Binary split



Methods for Expressing Attribute Test Condition

d) **Continuous Attributes** : Multiway split



Attribute Selection Measures

- An attribute selection measure is a heuristic for selecting the splitting criterion that “best” separates a given data partition, D, of class-labeled training tuples into individual classes.
- Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split.
- There are three popular attribute selection measures—information gain, gain ratio, and gini index.

Information gain

- ID3 uses information gain as its attribute selection measure.
- 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 pi is the probability that an arbitrary tuple in D belongs to class Ci and is estimated by |Ci,D|/|D|.

Info(D) is also known as the entropy of D.

- Now, suppose we were to partition the tuples in D on some attribute A having v distinct values, {a1, a2, : : : , a_V} as observed from the training data.
- Attribute A can be used to split D into v partitions or subsets, {D1, D2, : : : , Dv}, where Dj contains those tuples in D that have outcome aj of A

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

$$Gain(A) = Info(D) - Info_A(D).$$

The **attribute A with the highest information gain, (Gain(A)), is chosen as the splitting attrib**ute at node N.

Example: Decision tree using information gain.

Class-labeled training tuples 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	medium	no	excellent	no

- In this example, the class label attribute, buys computer, has two distinct values (namely, {yes, no}); therefore, there are two distinct classes (that is, $m = 2$).
- 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.

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

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

- Let’s start with the attribute age.
- We need to look at the distribution of yes and no tuples for each category of age.
- For the age 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.

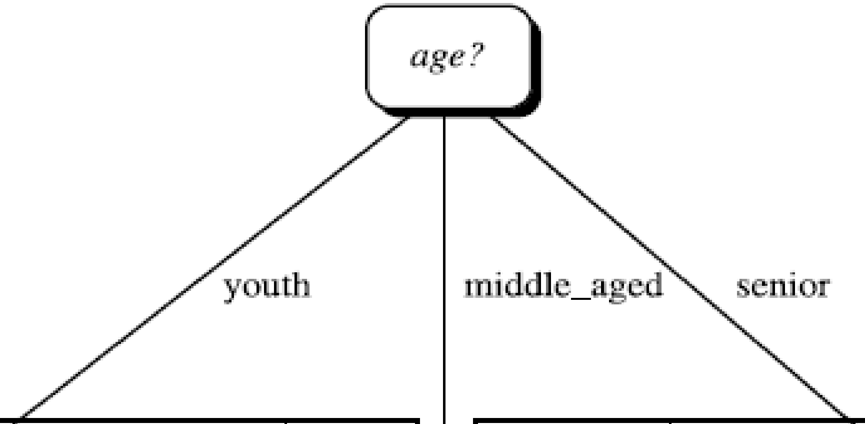
- The expected information needed to classify a tuple in D if the tuples are partitioned according to age is,

$$
\begin{aligned}
Info_{age}(D) = {} & \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\
& + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\
& + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\
= {} & 0.694 \text{ bits.}
\end{aligned}
$$

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

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

- Similarly, we can compute Gain(income) = 0.029 bits, Gain(student) = 0.151 bits, and Gain(credit rating) = 0.048 bits.
- Because age has the highest information gain among the attributes, it is selected as the splitting attribute as shown in tree below:



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*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

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