ECLT 5810/SEEM5750
Classification – Decision Trees

Prof. Wai Lam

Classification and Decision Tree

- What is classification?
- Issues regarding classification
- Classification by decision tree induction

Classification vs. Prediction

- *Classification:*
  - predicts *categorical class labels*
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
  - E.g. categorize bank loan applications as either *safe* or *risky.*
- *Prediction:*
  - models *continuous-valued functions*, i.e., predicts unknown or missing values
  - E.g. predict the *expenditures* of potential customers on computer equipment given their income and occupation.
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

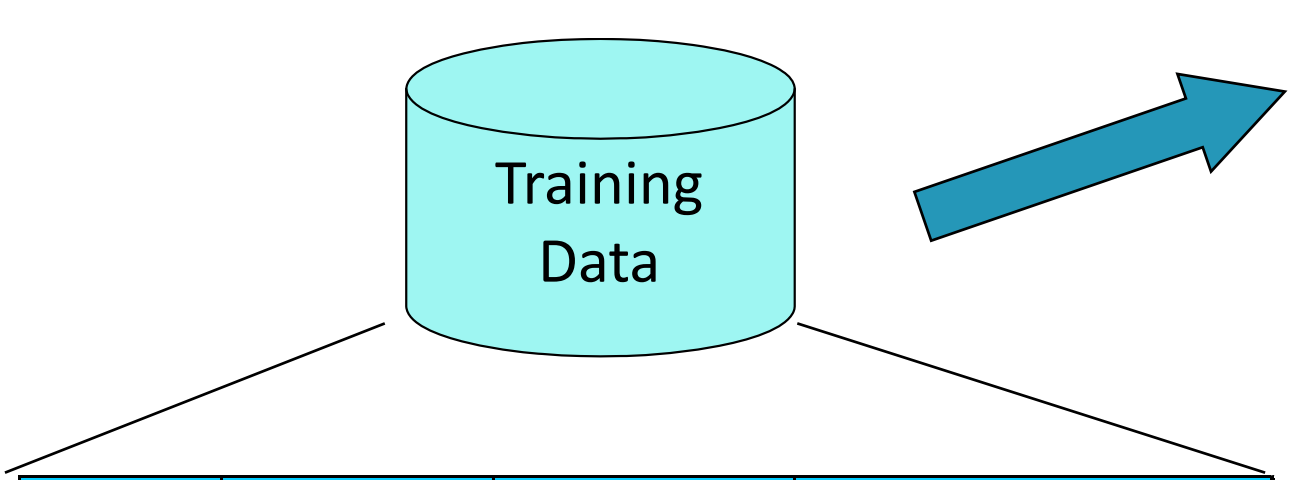
Classification - A Two-Step Process

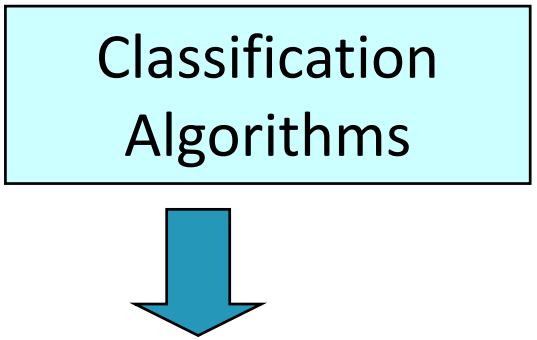
- *Step1 (Model construction):* describing a predetermined set of data classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the *class label* attribute
  - The set of tuples used for model construction: *training set*
  - The individual tuples making up the training set are referred to as *training samples*
  - *Supervised learning*: Learning of the model with a given training set.
  - The learned model is represented as
    - classification rules
    - decision trees, or
    - mathematical formulae.

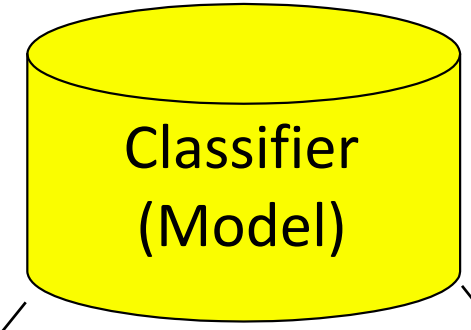
Classification - A Two-Step Process

- *Step 2 (Model usage):* the model is used for classifying future or unseen objects.
  - Estimate accuracy of the model
    - The known label of *test sample* 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.
    - Test set is independent of training set, otherwise over-fitting will occur
    - If the accuracy is acceptable, the model is used to classify future data tuples with unknown class labels.

Classification Process (1): Model Construction



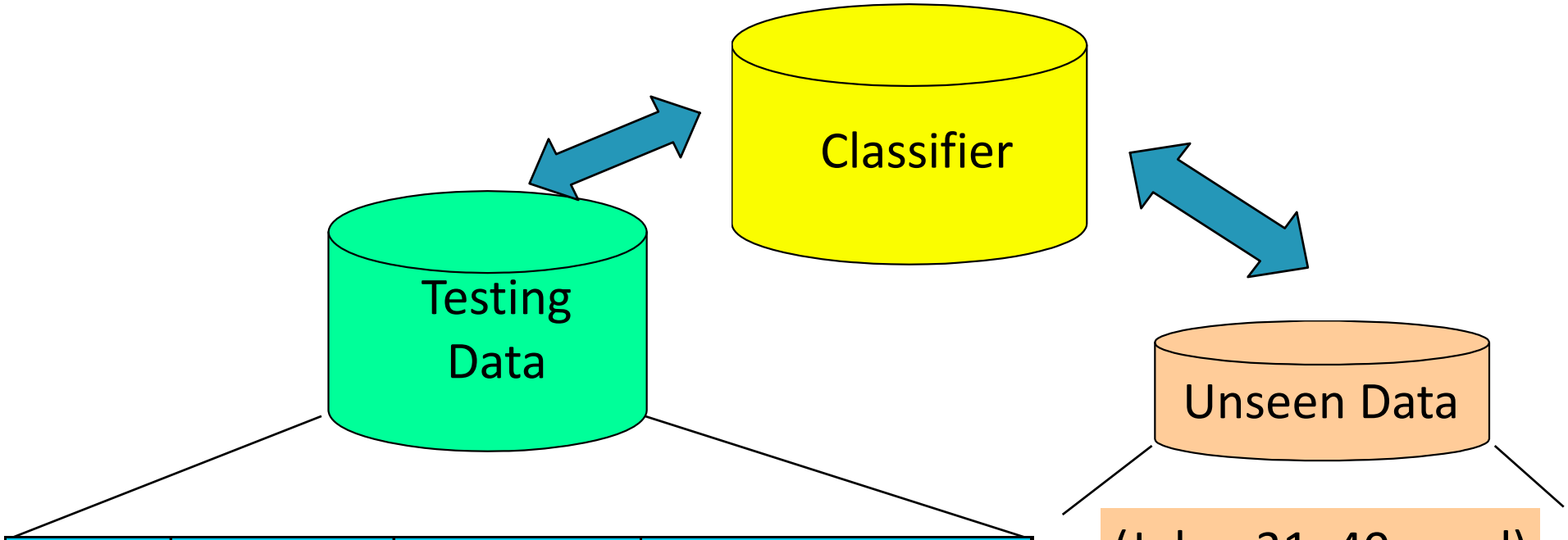




NAME	AGE	INCOME	CREDIT RATING
Mike	<= 30	low	fair
Mary	<= 30	low	poor
Bill	31..40	high	excellent
Jim	>40	med	fair
Dave	>40	med	fair
Anne	31..40	high	excellent

IF age = “31..40”
and income = high
THEN
credit rating = excellent

Classification Process (2): Use the Model in Prediction



NAME	AGE	INCOME	CREDIT RATING
May	<= 30	high	fair
Ana	31..40	low	poor
Wayne	>40	high	excellent
Jack	<=30	med	fair

(John, 31..40, med)

Credit rating?

fair

Supervised vs. Unsupervised Learning

- *Supervised learning (classification)*
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- *Unsupervised learning (clustering)*
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Issues regarding Classification and Prediction (1): Data Preparation

- Data cleaning
  - Preprocess data in order to reduce *noise* and handle *missing values*
- Relevance analysis (feature selection)
  - Remove the *irrelevant* or *redundant* attributes
  - E.g. date of a bank loan application is not relevant
  - Improve the efficiency and scalability of data mining
- Data transformation
  - Data can be *generalized* to higher level concepts (concept hierarchy)
  - Data should be *normalized* when methods involving distance measurements are used in the learning step (e.g. neural network)

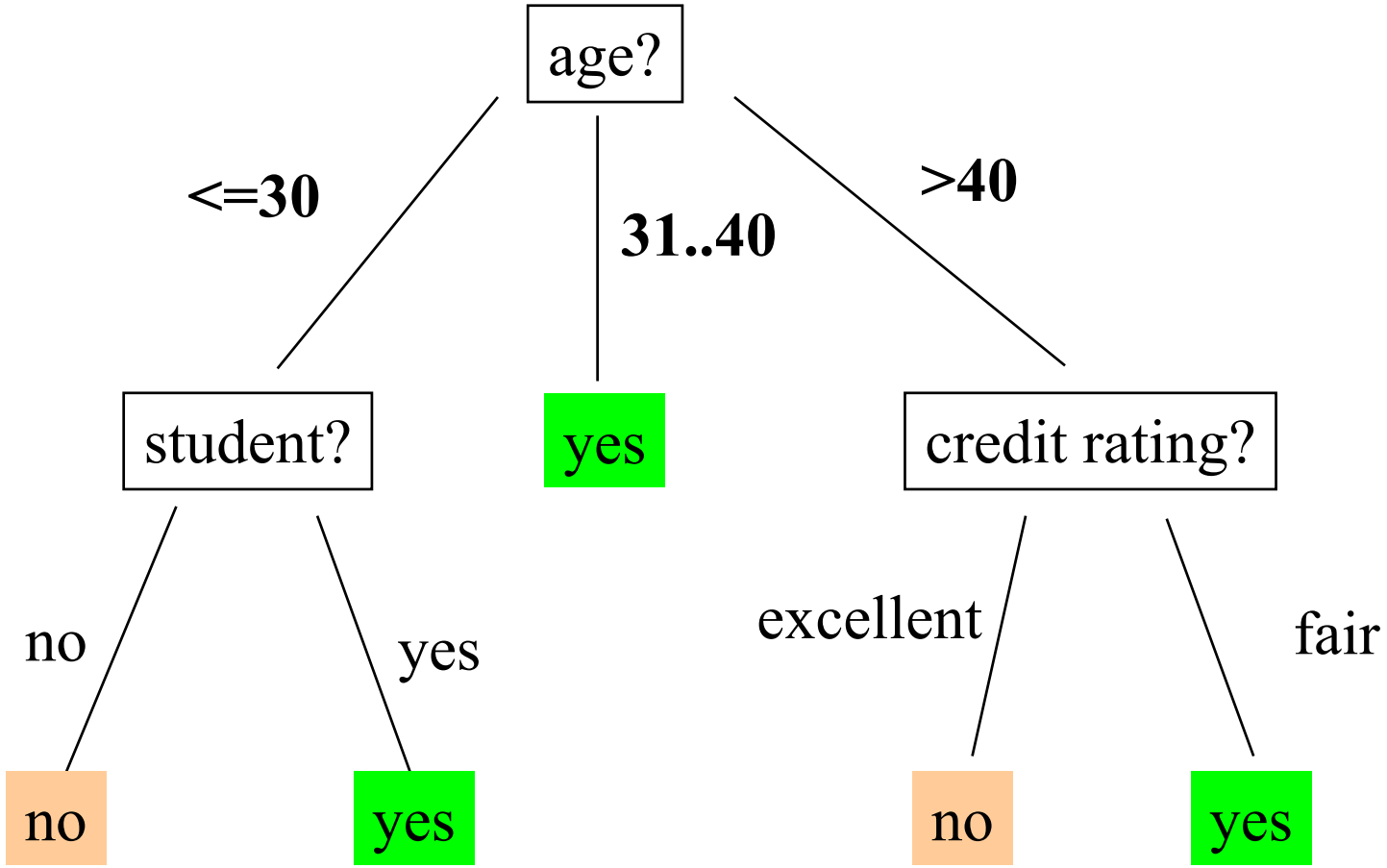
Issues regarding Classification and Prediction (2): Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
  - time to construct the model
  - time to use the model
- Robustness
  - handling noise and missing values
- Scalability
  - efficiency in disk-resident databases (large amount of data)
- Interpretability:
  - understanding and insight provided by the model
- Goodness of rules
  - decision tree size
  - compactness of classification rules

Classification by Decision Tree Induction

- Decision tree
  - A flow-chart-like tree structure
  - *Internal node* denotes a test on an attribute
  - *Branch* represents an outcome of the test
  - *Leaf nodes* represent class labels or class distribution
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

An Example of a Decision Tree For "buys computer"



How to Obtain a Decision Tree?

- *Manual construction*
- *Decision tree induction:*
  Automatically discover a decision tree from data
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers

Training Dataset

This follows an example from Quinlan's ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31..40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31..40	medium	no	excellent	yes
31..40	high	yes	fair	yes
>40	medium	no	excellent	no

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a *top-down recursive divide-and-conquer* manner
  - At start, all the training examples are at the root
  - Attributes are *categorical* (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes

Basic Algorithm for Decision Tree Induction

- If the samples are all of the same class, then the node becomes a leaf and is labeled with that class
- Otherwise, it uses a *statistical measure* (e.g., information gain) for selecting the attribute that will best separate the samples into individual classes. This attribute becomes the "test" or "decision" attribute at the node.
- A *branch* is created for *each known value of the test attribute*, and the samples are partitioned accordingly
- The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node's descendents.

Basic Algorithm for Decision Tree Induction

- The recursive partitioning stops only when any one of the following conditions is true:
  - All samples for a given node belong to the same class
  - There are no remaining attributes on which the samples may be further partitioned. In this case, *majority voting* is employed. This involves converting the given node into a leaf and labeling it with the class in majority voting among samples.
  - There are no samples for the branch *test-attribute=a_i*. In this case, a leaf is created with the majority class in samples.

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:

(1) create a node N;
(2) **if** tuples in D are all of the same class, C, **then**
(3) return N as a leaf node labeled with the class C;
(4) **if** *attribute_list* is empty **then**
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, *attribute_list*) to **find** the "best" *splitting_criterion*;
(7) label node N with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and**
multiway splits allowed **then** // not restricted to binary trees
(9) *attribute_list* ← *attribute_list* − *splitting_attribute*; // remove *splitting_attribute*
(10) **for each** outcome j of *splitting_criterion*
// partition the tuples and grow subtrees for each partition
(11) let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12) **if** D_j is empty **then**
(13) attach a leaf labeled with the majority class in D to node N;
(14) **else** attach the node returned by **Generate_decision_tree**(D_j, *attribute_list*) to node N;
endfor
(15) return N;

Attribute Selection by Information Gain Computation

attribute1	attribute2	class label
high	high	yes
high	high	yes
high	high	yes
high	low	yes
high	low	yes
high	low	yes
high	low	no
low	low	no
low	low	no
low	high	no
low	high	no
low	high	no

Attribute Selection by Information Gain Computation

attribute1	attribute2	class label
high	high	yes
high	high	yes
high	high	yes
high	low	yes
high	low	yes
high	low	yes
high	low	no
low	low	no
low	low	no
low	high	no
low	high	no
low	high	no

Consider the attribute1:

	Class label	
attribute1	yes	no
high	6	1
low	0	5

Consider the attribute2:

	Class label	
attribute2	yes	no
high	3	3
low	3	3

Attribute Selection by Information Gain Computation

attribute1	attribute2	class label
high	high	yes
high	high	yes
high	high	yes
high	low	yes
high	low	yes
high	low	yes
high	low	no
low	low	no
low	low	no
low	high	no
low	high	no
low	high	no

Consider the attribute1:

	Class label	
attribute1	yes	no
high	6	1
low	0	5

Consider the attribute2:

	Class label	
attribute2	yes	no
high	3	3
low	3	3

attribute1 is better than attribute2 for classification purpose !

Information Gain (ID3/C4.5)

- To measure the degree of diversity, we can use the entropy function
- Assume there are two groups, *P* and *N*
  - Suppose there are *p* elements of group *P* and *n* elements of group *N*
  - The entropy function *I(p,n)* is defined as:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Some examples:

$$I(2,3) = -\frac{2}{5}\,log_2\left(\frac{2}{5}\right) - \frac{3}{5}\,log_2\left(\frac{3}{5}\right) = 0.971$$

$$I(5,0) = -\frac{5}{5}log_2\left(\frac{5}{5}\right) - \frac{0}{5}log_2\left(\frac{0}{5}\right) = 0$$

Note: We define $log_2(0)$ as 0

Information Gain in Decision Tree Induction

- Assume that using attribute *A,* a set *S* will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$ where $\{1,\ldots, v\}$ are the possible values of *A*
  - If S_i contains p_i examples of *P* and n_i examples of *N*, the *entropy*, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on A.

$$Gain(A) = I(p, n) - E(A)$$

- The attribute with the highest information gain is chosen as the test attribute for the given set *S*.
- Generalize the above to *m* classes.

Attribute Selection by Information Gain Computation

Consider the attribute age:

age	yes	no	$I(p_i, n_i)$
<=30	2	3	0.971
31..40	4	0	0
>40	3	2	0.971

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$$I(p, n) = I(9, 5) = 0.940$$

$$Gain(age) = I(p,n) - E(age)$$

$$Gain(age) = 0.94 - 0.694 = 0.246$$

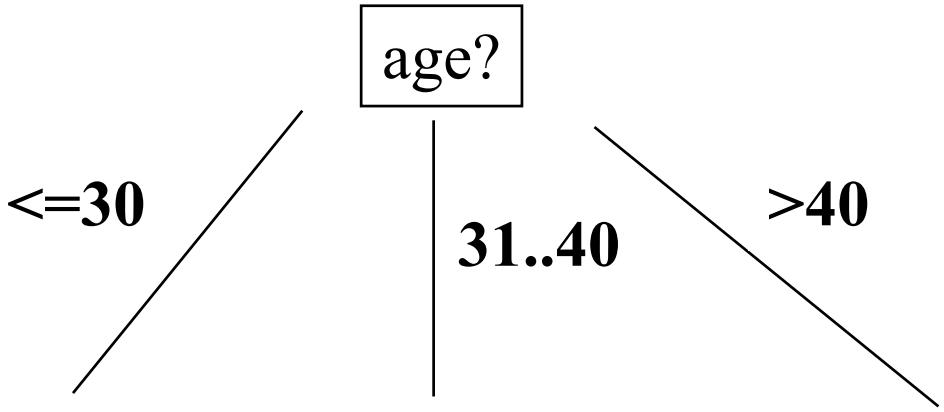
Consider other attributes in a similar way:

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

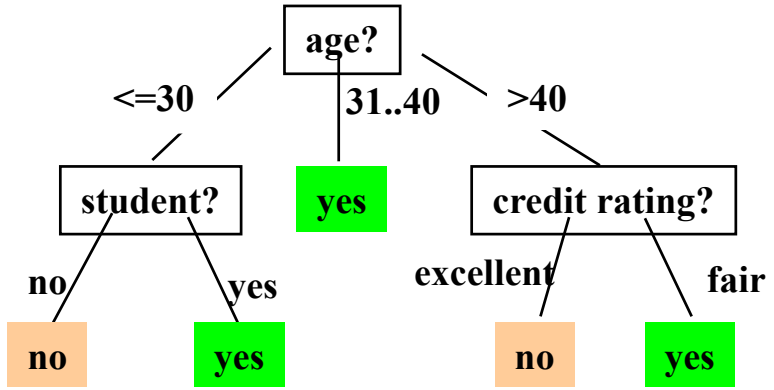
$$Gain(credit\_rating) = 0.048$$

Learning (Constructing) a Decision Tree



Extracting Classification Rules from Trees

- Represent the knowledge in the form of *IF-THEN* rules
- One rule is created for each *path* from the root to a leaf
- Each *attribute-value pair* along a path forms a *conjunction*
- The leaf node holds the class prediction
- Rules are easier for humans to understand
- Example



- *IF age = "<=30" AND student = "no" THEN buys_computer = "no"*
- *IF age = "<=30" AND student = "yes" THEN buys_computer = "yes"*
- *IF age = "31…40" THEN buys_computer = "yes"*
- *IF age = ">40" AND credit_rating ="excellent" THEN buys_computer= "no"*
- *IF age = "<=30" AND credit_rating = "fair" THEN buys_computer = "yes"*

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - comparable classification accuracy with other methods

Presentation of Classification Results