Unit-III
Classification and Prediction

Dr. K. Raghava Rao
Professor of CSE
Dept. of MCA
KL University

Classification and Prediction

What is classification & Prediction

Classification and prediction are two forms of data analysis

that can be used to extract models describing important data

classes or to predict future data trends.

Classification and Prediction

What is classification & 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
- Regression or Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values
  -
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

Classification and Prediction

What is classification & Prediction

- Credit approval
  - A bank wants to classify its customers based on whether they are expected to pay back their approved loans
  - The history of past customers is used to train the classifier
  - The classifier provides rules, which identify potentially reliable future customers
  - Classification rule:
    - If age = “31...40” and income = high then credit_rating = excellent
  - Future customers
    - Paul: age = 35, income = high $\Rightarrow$ excellent credit rating
    - John: age = 20, income = medium $\Rightarrow$ fair credit rating

Classification and Prediction

What is classification & Prediction

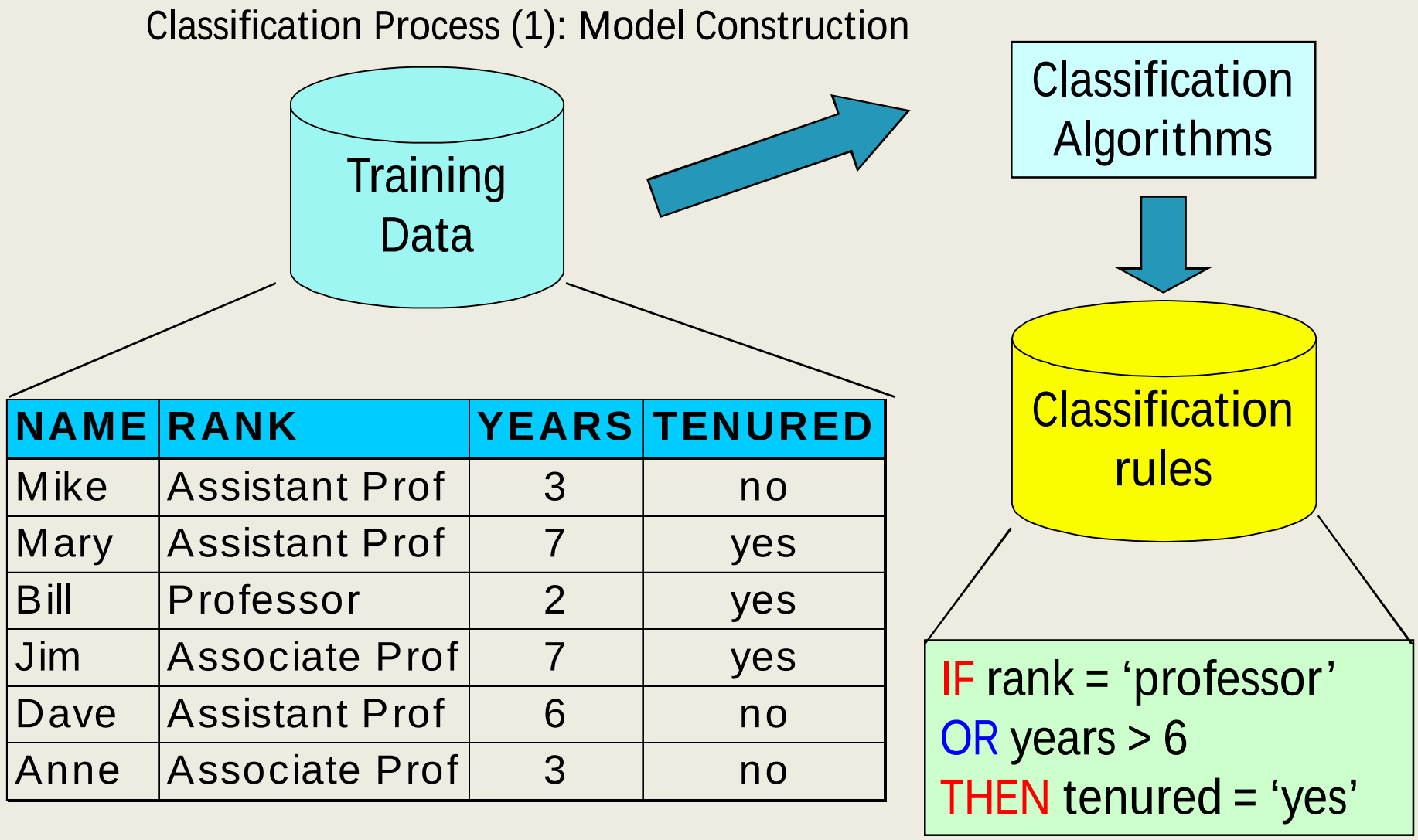
Classification—A Two-Step Process

- Model construction: describing a set of predetermined 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 model is represented as classification rules, decision trees, or mathematical formulae

- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test samples 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

Classification and Prediction

What is classification & Prediction

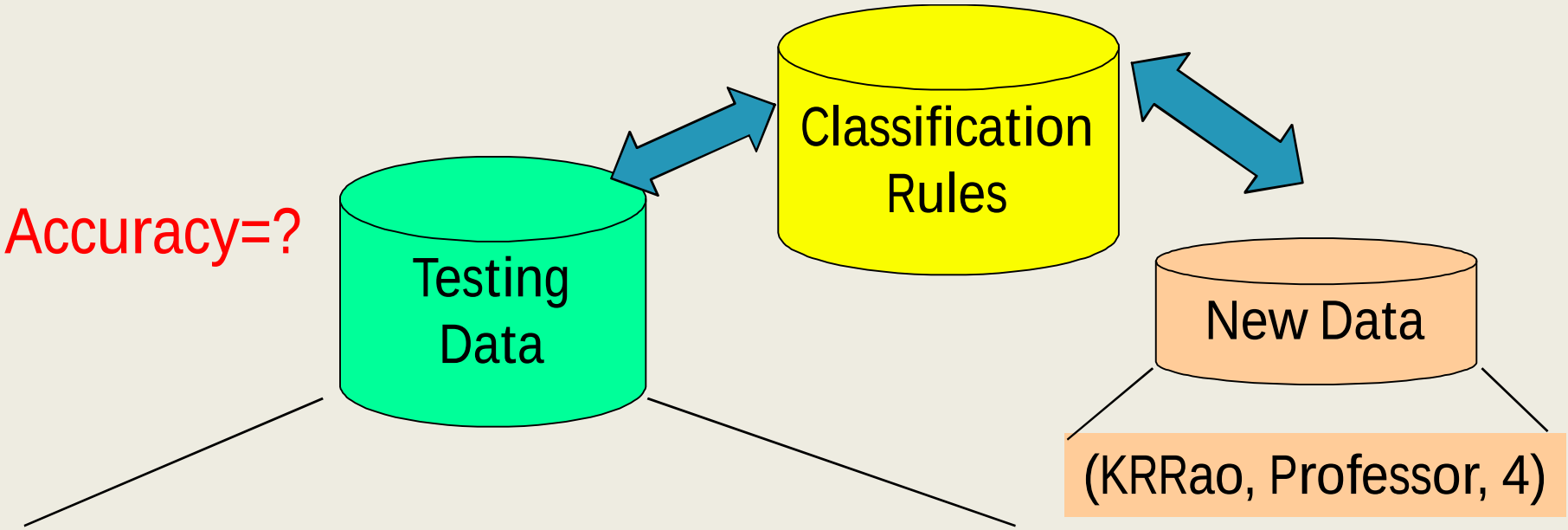


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification and Prediction

What is classification & Prediction

Classification Process (2): Use the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Mellisa	Associate Prof	7	no
KRRAO	Professor	5	yes
Joseph	Assistant Prof	7	yes

Classification and Prediction

What is classification & Prediction

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

Classification and Prediction
Issues regarding classification and prediction

Preparing the Data for Classification and Prediction

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
- Data transformation
  - Generalize and/or normalize data
    - numerical attribute income $\Rightarrow$ categorical {low,medium,high}
    - normalize all numerical attributes to [0,1)

Classification and Prediction
Issues regarding classification and prediction

Comparing Classification and Prediction Methods

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

Classification and Prediction
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
- Decision tree generation consists of two phases
  - 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
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Classification and Prediction
Classification by Decision Tree Induction

An example from Ross Quinlan's ID3

Training Dataset

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

Classification and Prediction
Classification by Decision Tree Induction

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)
  - Samples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Classification and Prediction
Classification by Decision Tree Induction

Decision Tree Induction

Algorithm GenDecTree(Sample S, Attlist A)

Input: Data partition S- which is set of training tuples with class labels
attribute_list- set of candidate attributes
attribute_selection_method-a procedure to determine splitting criterion that "best" partition the data tuples into individual classes..

1. create a node N
2. If all samples are of the same class C then label N with C; terminate;
3. If A is empty then label N with the most common class C in S (majority voting); terminate;
4. Select $a \in A$, with the highest information gain; Label N with a;
5. For each value v of a:
   a. Grow a branch from N with condition a=v;
   b. Let S_v be the subset of samples in S with a=v;
   c. If S_v is empty then attach a leaf labeled with the most common class in S;
   d. Else attach the node generated by GenDecTree(S_v, A-a)

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

- Information gain (ID3/C4.5)
  - All attributes are assumed to be categorical
  - Can be modified for continuous-valued attributes
- Gini index (IBM IntelligentMiner)
  - All attributes are assumed continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
  - Can be modified for categorical attributes

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there are two classes, *P* and *N*
  - Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

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

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Information Gain in Decision Tree Induction

- Assume that using attribute A a set *S* will be partitioned into sets {S_1, S_2 , …, S_v}
  - 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*

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

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

an example from Quinlan's ID3
Training Dataset

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

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Attribute Selection by Information Gain Computation

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for *age*:

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

Hence

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

Similarly

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

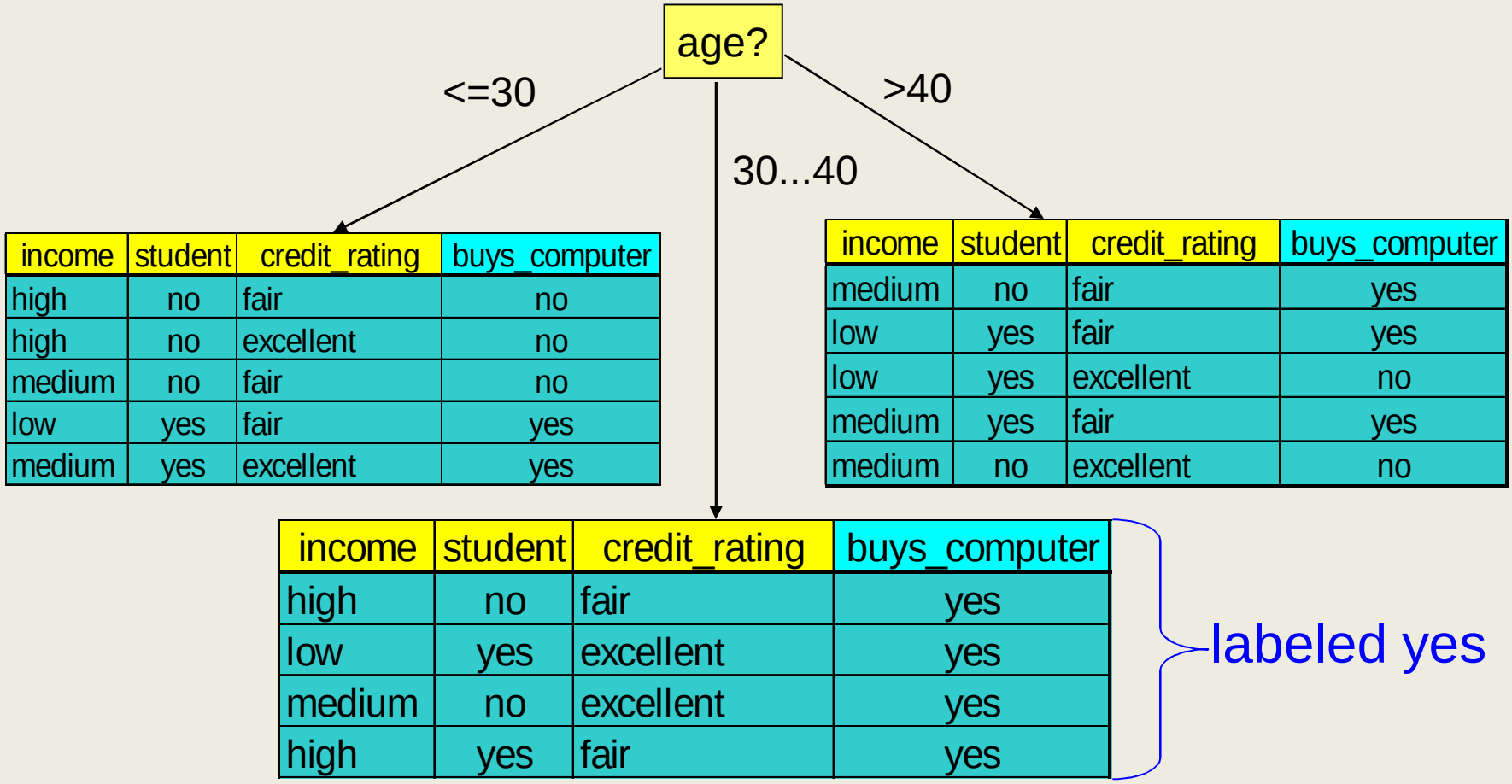
$$Gain(credit\_rating) = 0.048$$

- $\frac{5}{14} I(2,3)$ means “age <=30” has 5 out of 14 samples, with 2 yes’s and 3 no’s.
- I(2,3) = -2/5 * log(2/5) – 3/5 * log(3/5)

age	p_i	n_i	I(p_i, n_i)
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures



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

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

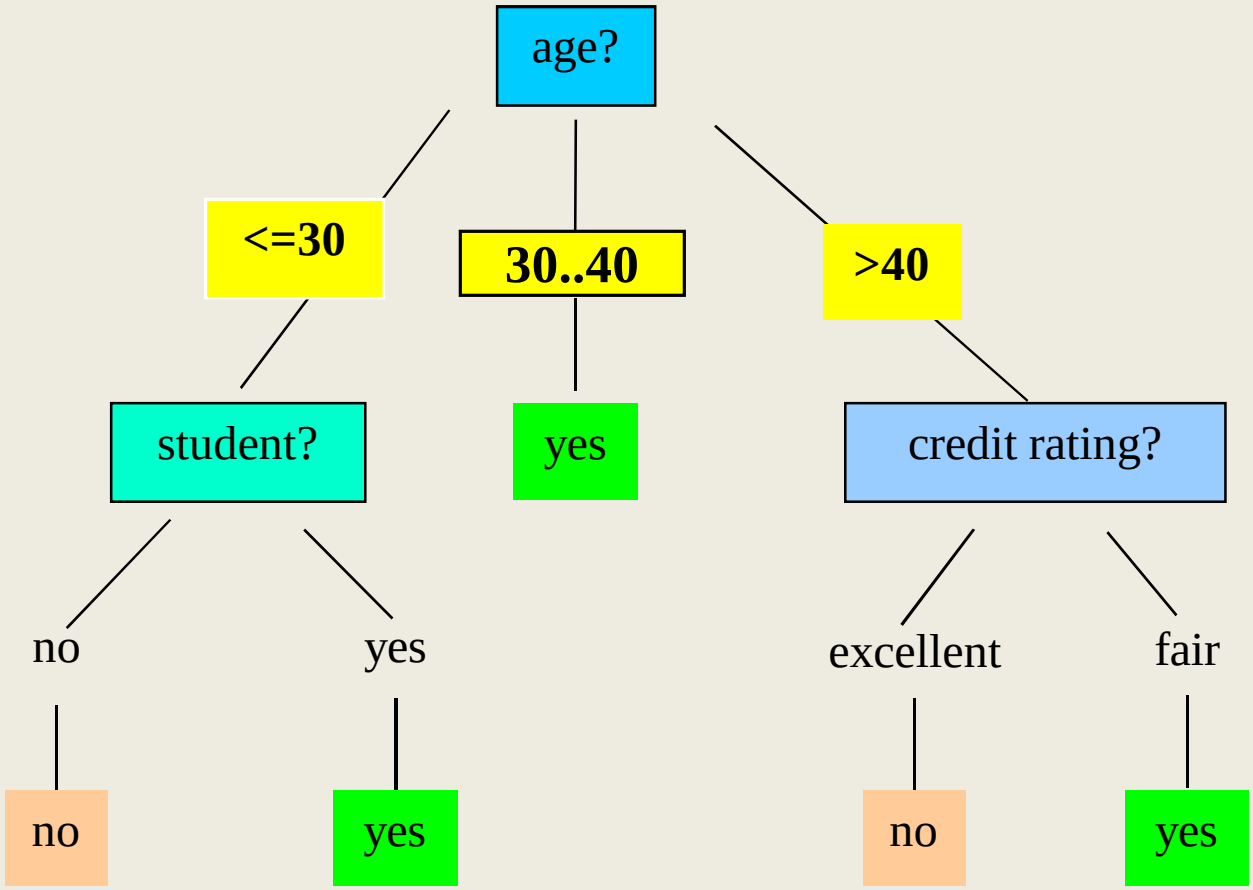
income	student	credit_rating	buys_computer
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

labeled yes

Splitting the samples using *age*

Classification and Prediction
Classification by Decision Tree Induction

Output: A Decision Tree for *"buys_computer"*



Classification and Prediction
Classification by Decision Tree Induction

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* = "*yes*"
IF *age* = ">40" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Gain Ratio

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem of bias od ID3 (it applies normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

-This value represents the potential information generated by splitting the training data set, D, into V partitions, corresponding to v outcomes of test on attribute A.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

-GainRatio(A) = Gain(A)/SplitInfo(A)

- Ex.-gain_ratio(income) = 0.029/0.926= 0.0231
- The attribute with the maximum gain ratio is selected as the splitting attribute

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Gini Index (IBM IntelligentMiner)

• Gini index considers a binary split for each attribute .

• To determine best binary split on A, we examine all of possible subsets that can be found using known values of A.

•If A has v possible values, the there are 2^v possible subsets.

Example: if income has three possible values namely {low,medium,high} possible subsets : {low,medium,high},{low,medium},{low,high},{medium,high},{low},{medium}, {high}, and {}.

We exclude power set , {low,medium,high}, empty set because they do not represent a split. Therefore 2^v -2 possible ways to form two partition of data, T,based on binbary split on A.

-When considering a binary split we comput weighted sum of impurity of each resulting partition . If binary split on A partitions T into T1 and T2 the giniindex of T is given in next slide.

Classification and Prediction
Classification by Decision Tree Induction

Attribute Selection Measures

Gini Index (IBM IntelligentMiner)

- Giniindex measures impurity of D, a data partition or set of training samples as

$$gini\ (T) = 1 - \sum_{j=1}^{n} p_j^2$$

- If a data set *T* contains examples from *n* classes, gini index, *gini*(*T*) is defined as where p_j is the relative frequency of class *j* in *T.*

- If a data set *T* is split into two subsets T_1 and T_2 with sizes N_1 and N_2 respectively, the *gini* index of the split data contains examples from *n* classes, the *gini* index *gini*(*T*) is defined as

$$gini_{split}\ (T) = \frac{N_1}{N} gini\ (T_1) + \frac{N_2}{N} gini\ (T_2)$$

- The attribute which provides the smallest $gini_{split}(T)$ is chosen to split the node (*we need to enumerate all possible splitting points for each attribute*).

Classification and Prediction
Classification by Decision Tree Induction

Gini Index Example problem

- Ex. D has 9 tuples in buys_computer = “yes” and 5 in “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_1)$$

$$= \frac{10}{14}(1 - (\frac{6}{10})^2 - (\frac{4}{10})^2) + \frac{4}{14}(1 - (\frac{1}{4})^2 - (\frac{3}{4})^2)$$
$$= 0.450$$
$$= Gini_{income \in \{high\}}(D)$$

but $gini_{\{medium,high\}}$ is 0.30 and thus the best since it is the lowest

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Classification and Prediction
Classification by Decision Tree Induction
Attribute Selection Measures
Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - Information gain:
    - biased towards multivalued attributes
  - Gain ratio:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Classification and Prediction
Classification by Decision Tree Induction
Attribute Selection Measures

Other Attribute Selection Measures

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistics: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Classification and Prediction
Classification by Decision Tree Induction

Tree Pruning

Overfitting: An induced tree may overfit the training data

Too many branches, some may reflect anomalies due to noise or outliers

Poor accuracy for unseen samples

Two approaches to avoid Overfitting

- Prepruning:
  - Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
  - Difficult to choose an appropriate threshold

Postpruning:

- Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees
- Use a set of data different from the training data to decide which is the “best pruned tree”

Classification and Prediction
Classification by Decision Tree Induction

Scalability and Decision Tree Induction

- ID3, C4.5, and CART are not efficient when the training set doesn't fit the available memory. Instead the following algorithms are used
  - SLIQ
    - Builds an index for each attribute and only class list and the current attribute list reside in memory
  - SPRINT
    - Constructs an attribute list data structure
  - RainForest
    - Builds an AVC-list (attribute, value, class label)
  - BOAT
    - Uses bootstrapping to create several small samples