

# **Data Mining:**

---

## **Concepts and Techniques**


**(3<sup>rd</sup> ed.)**

Jiawei Han, Micheline Kamber, and Jian Pei  
University of Illinois at Urbana-Champaign &  
Simon Fraser University

©2011 Han, Kamber & Pei. All rights reserved.

# Chapter 8. Classification: Basic Concepts

---

- Classification: Basic Concepts 
- Decision Tree Induction
- Model Evaluation and Selection

# Supervised vs. Unsupervised Learning

---

- 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
- 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

# Prediction Problems: Classification vs. Numeric Prediction

---

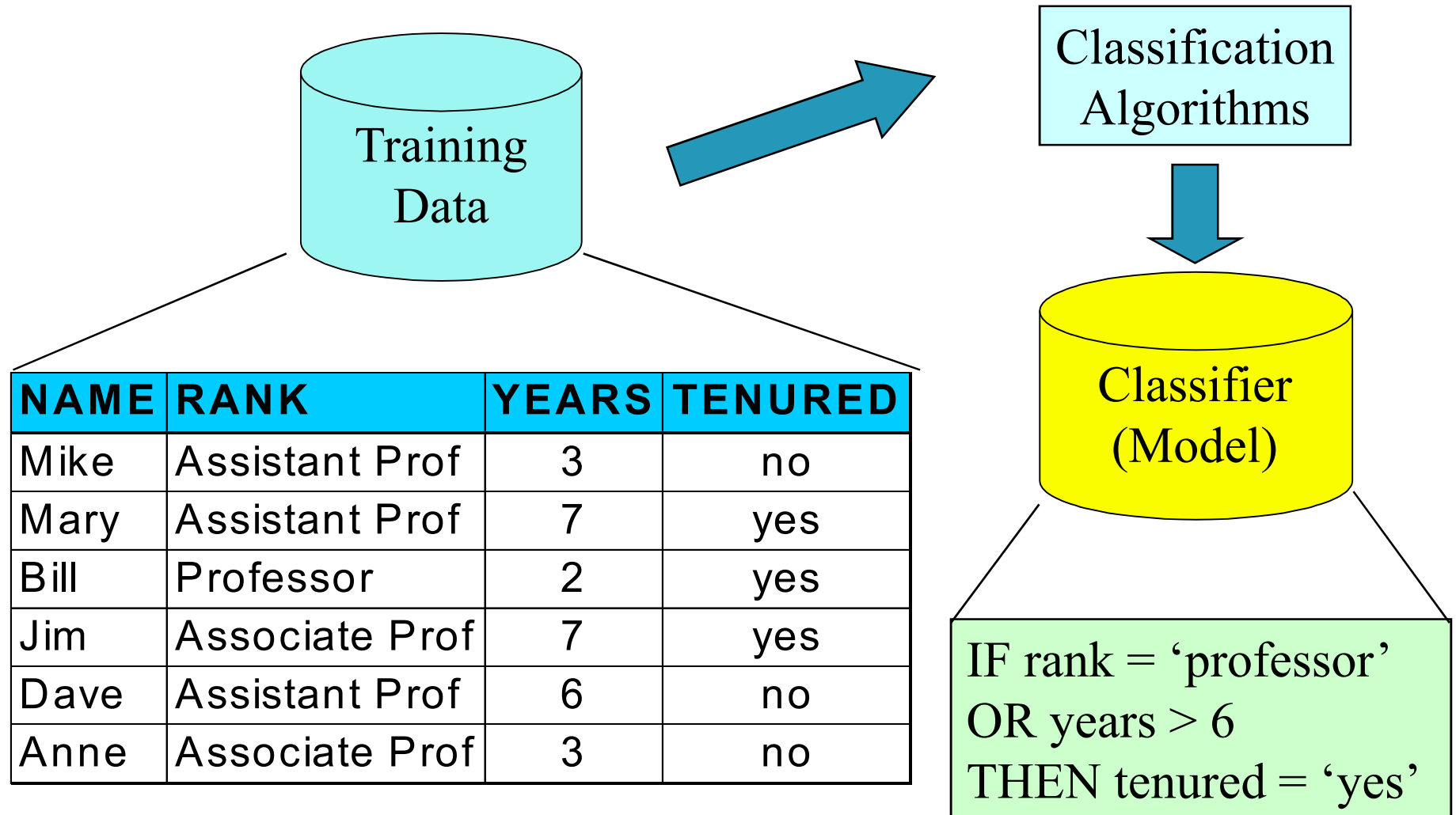
- **Classification**
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data
- **Numeric Prediction**
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit/loan approval:
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is

# Classification—A Two-Step Process

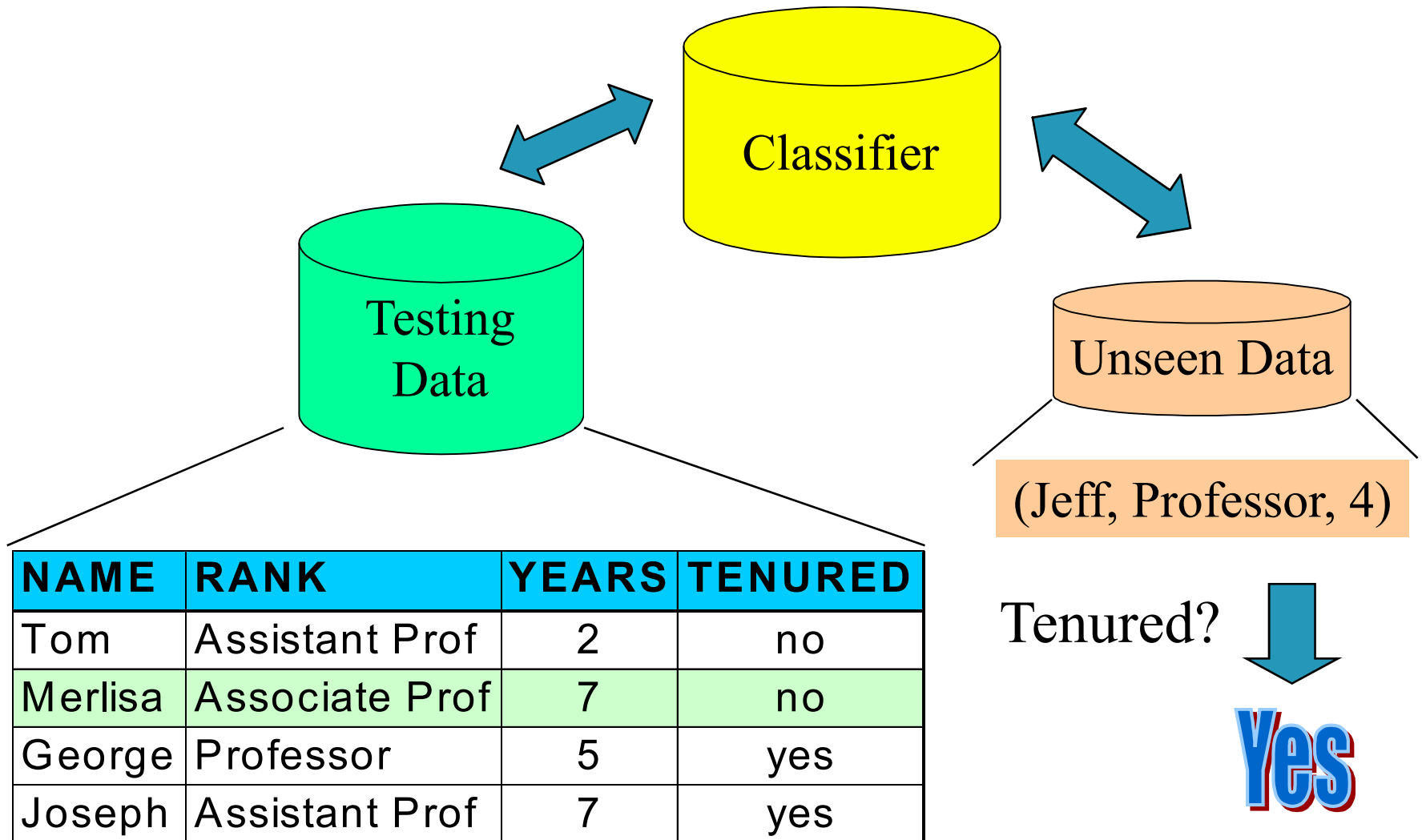
---

- **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 is **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 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 overfitting)
  - If the accuracy is acceptable, use the model to **classify new data**
- Note: If *the test set* is used to select models, it is called **validation (test) set**

# Process (1): Model Construction




## Process (2): Using the Model in Prediction



# Chapter 8. Classification: Basic Concepts

---

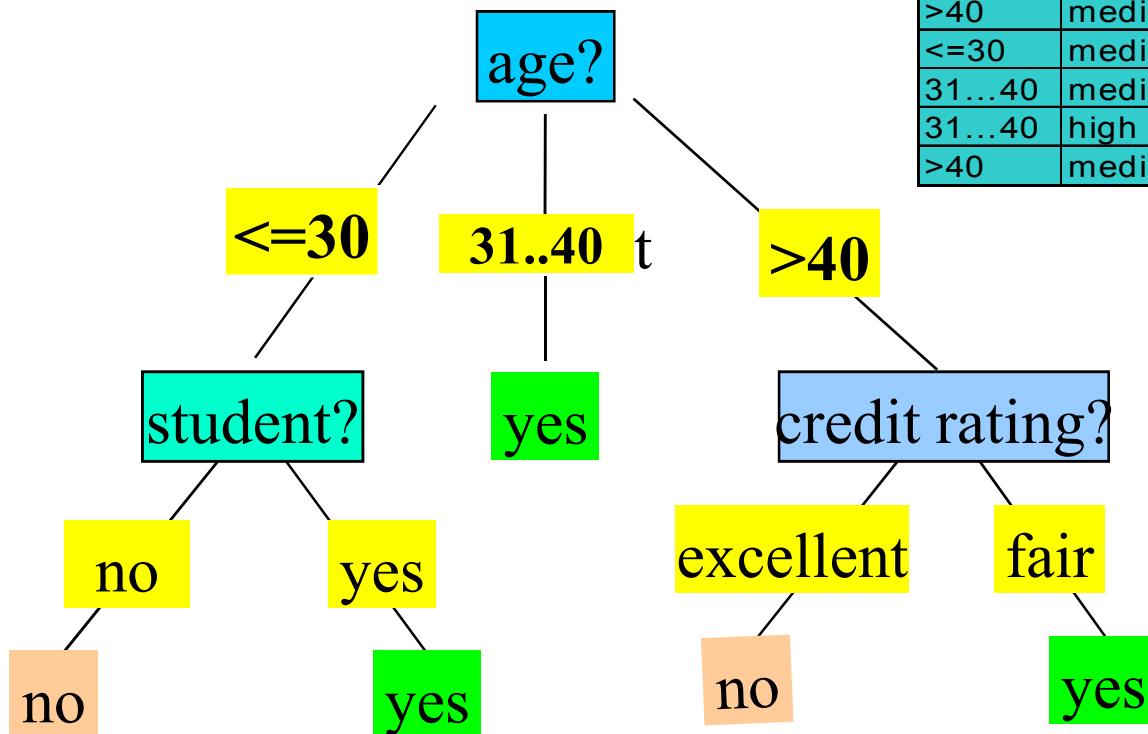
- Classification: Basic Concepts
- Decision Tree Induction 
- Model Evaluation and Selection



# Decision Tree Induction: An Example

- Training data set:  
Buys\_computer
- The data set follows an example of Quinlan's ID3
- Resulting tree:

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
  - 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

# Brief Review of Entropy

- Entropy (Information Theory)

- A measure of uncertainty associated with a random variable

- Calculation: For a discrete random variable  $Y$  taking  $m$  distinct values  $\{y_1, \dots, y_m\}$ ,

- $H(Y) = -\sum_{i=1}^m p_i \log(p_i)$ , where  $p_i = P(Y = y_i)$

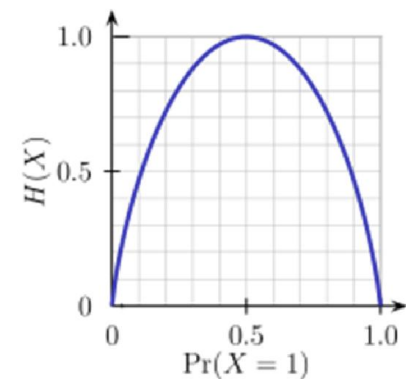
- Interpretation:

- Higher entropy => higher uncertainty

- Lower entropy => lower uncertainty

- Conditional Entropy

- $H(Y|X) = \sum_x p(x)H(Y|X = x)$



**m = 2**

# The Buys\_Computer Example

## ■ Entropy of D

$$\begin{aligned} \text{Entropy}(D) &= \text{Entropy}([9+, 5-]) \\ &= -\frac{9}{14} \log\left(\frac{9}{14}\right) - \frac{5}{14} \log\left(\frac{5}{14}\right) \\ &= 0.940 \end{aligned}$$

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

# Attribute Selection Measure: Information Gain (ID3/C4.5)

---

- Select the attribute with the highest information gain
- Let  $p_i$  be the probability that an arbitrary tuple in  $D$  belongs to class  $C_i$ , estimated by  $|C_{i,D}|/|D|$

- **Expected information** (entropy) needed to classify a tuple in  $D$ :

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

- **Information** needed (after using  $A$  to split  $D$  into  $v$  partitions) to classify  $D$ :

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

- **Information gained** by branching on attribute  $A$

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

# Buys\_computer data

---

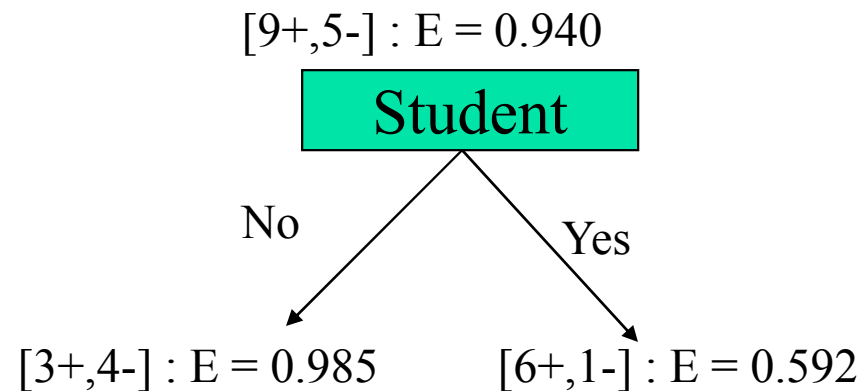
	age	income	student	credit_rating	buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31...40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31...40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	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 Buys\_Computer Example (cont'd)

## ■ Attribute: Student

- $D_{No} = [3+, 4-]$

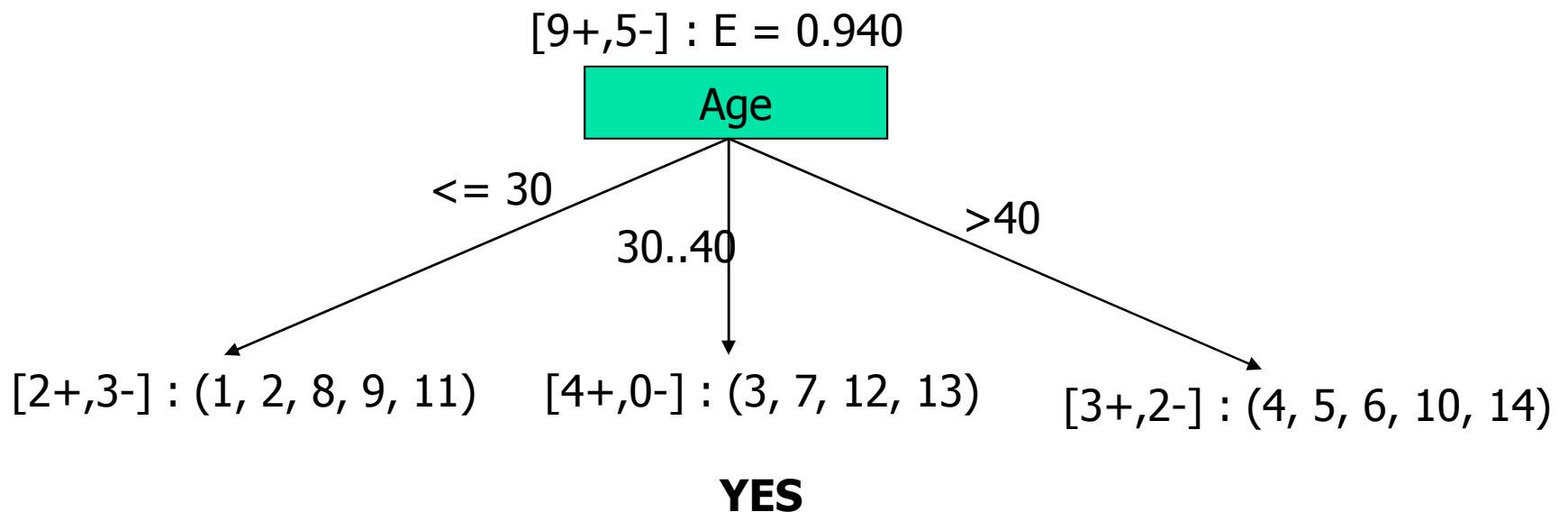
- $D_{Yes} = [6+, 1-]$



$$\begin{aligned} \text{Gain}(D, \text{Student}) &= \text{Entropy}(D) - \sum_{v \in \{\text{yes}, \text{no}\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v) \\ &= \text{Entropy}(D) - \frac{7}{14} \text{Entropy}(D_{std=\text{no}}) - \frac{7}{14} \text{Entropy}(D_{std=\text{yes}}) \\ &= 0.940 - \frac{7}{14} 0.985 - \frac{7}{14} 0.592 = 0.151 \end{aligned}$$

# The Buys\_Computer Example (cont'd)

- Similarly, compute the Gain for the other attributes, i.e. Age, Income and Credit\_rating.
- Best Attribute for root?
  - $\text{Gain}(D, \text{Age}) = 0.246$
  - $\text{Gain}(D, \text{Student}) = 0.151$
  - $\text{Gain}(D, \text{Credit\_rating}) = 0.048$
  - $\text{Gain}(D, \text{Income}) = 0.029$

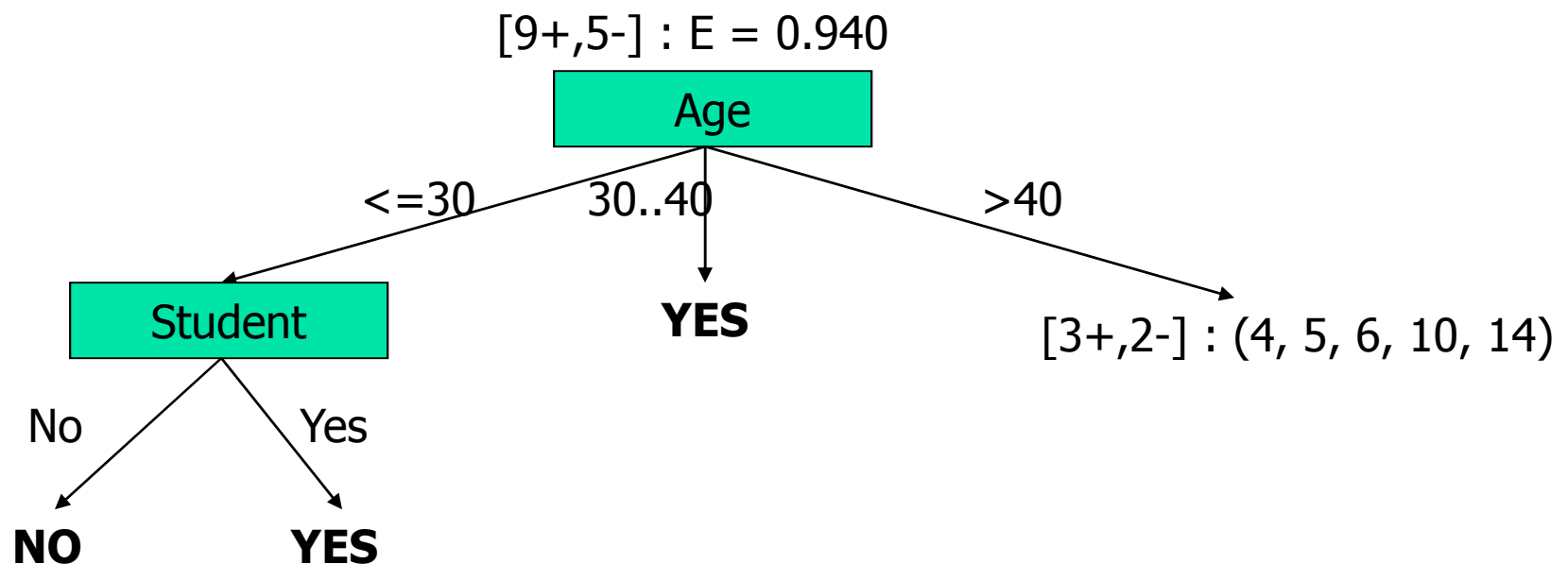




# The Buys\_Computer Example (cont'd)

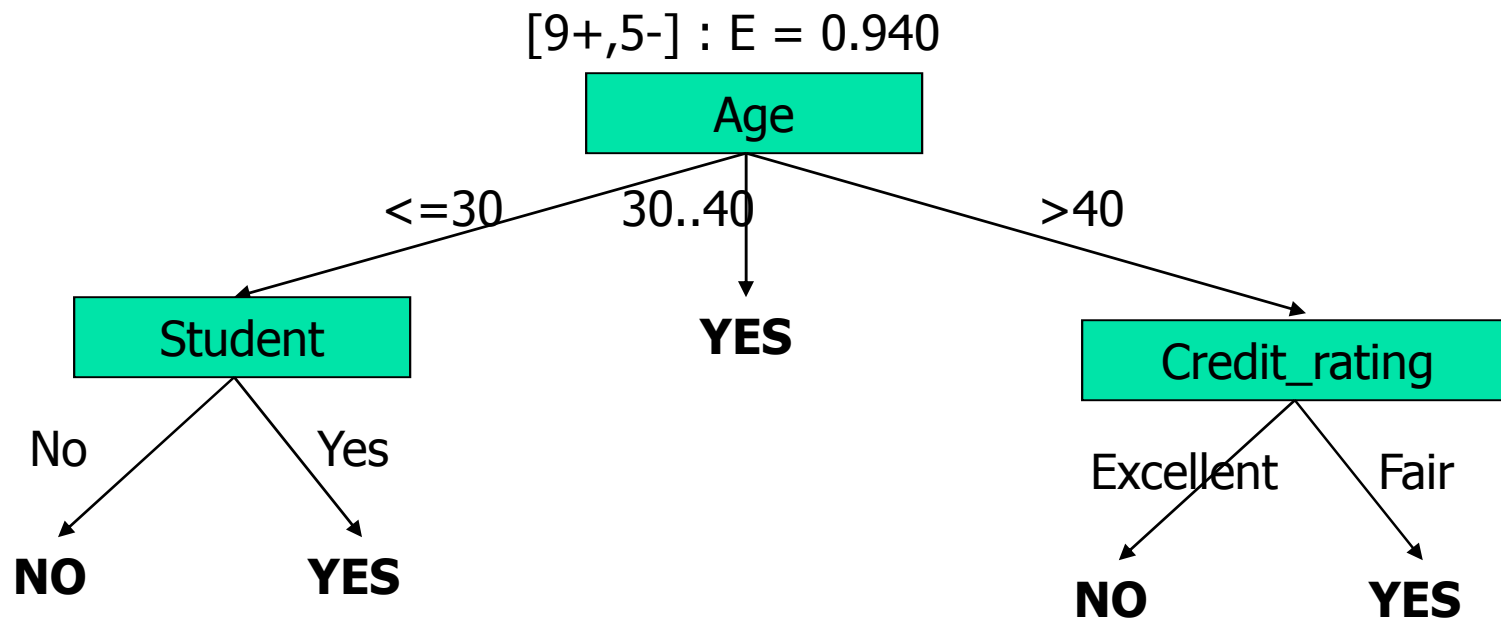
- Best Attribute for sub-tree  $\text{Age} \leq 30$ ?

- $\text{Gain}(D, \text{Student}) = 0.971$
- $\text{Gain}(D, \text{Credit\_rating}) = 0.020$
- $\text{Gain}(D, \text{Income}) = 0.571$



# The Buys\_Computer Example (cont'd)

- Best Attribute for sub-tree Age>40?
  - $\text{Gain}(D, \text{Student}) = 0.020$
  - $\text{Gain}(D, \text{Credit\_rating}) = 0.971$
  - $\text{Gain}(D, \text{Income}) = 0.020$



# Computing Information-Gain for Continuous-Valued Attributes

---

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
  - Sort the value A in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $(a_i + a_{i+1})/2$  is the midpoint between the values of  $a_i$  and  $a_{i+1}$
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 is the set of tuples in D satisfying  $A \leq \text{split-point}$ , and D2 is the set of tuples in D satisfying  $A > \text{split-point}$

# Gain Ratio for Attribute Selection (C4.5)

- 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 (normalization to information gain)

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

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

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

- $gain\_ratio(income) = 0.029/1.557 = 0.019$

- The attribute with the maximum gain ratio is selected as the splitting attribute

# 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 (not discussed here):**
    - 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

# Other Attribute Selection Measures

---

- CHAID: a popular decision tree algorithm, measure based on  $\chi^2$  test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistic: has a close approximation to  $\chi^2$  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

# Overfitting and 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”

# Enhancements to Basic Decision Tree Induction

---

- Allow for **continuous-valued attributes**
  - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle **missing attribute values**
  - Assign the most common value of the attribute
  - Assign probability to each of the possible values
- **Attribute construction**
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication




# Let's Play

---



# Chapter 8. Classification: Basic Concepts

---

- Classification: Basic Concepts
- Decision Tree Induction
- Model Evaluation and Selection 

# Model Evaluation and Selection

---

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy:
  - Holdout method, random subsampling
  - Cross-validation
- Comparing classifiers:
  - Cost-benefit analysis and ROC Curves

# Classifier Evaluation Metrics: Confusion Matrix

## Confusion Matrix:

Actual class\Predicted class	$C_1$	$\neg C_1$
$C_1$	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

## Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

- Given  $m$  classes, an entry,  $\mathbf{CM}_{i,j}$  in a **confusion matrix** indicates # of tuples in class  $i$  that were labeled by the classifier as class  $j$
- May have extra rows/columns to provide totals

# Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

- **Classifier Accuracy**, or recognition rate: percentage of test set tuples that are correctly classified

$$\text{Accuracy} = (TP + TN) / \text{All}$$

- **Error rate**:  $1 - \text{accuracy}$ , or  
 $\text{Error rate} = (FP + FN) / \text{All}$

- **Class Imbalance Problem:**

- One class may be *rare*, e.g. fraud, or HIV-positive
- Significant *majority of the negative class* and minority of the positive class
- **Sensitivity**: True Positive recognition rate
  - **Sensitivity** =  $TP / P$
- **Specificity**: True Negative recognition rate
  - **Specificity** =  $TN / N$

# Classifier Evaluation Metrics:

## Precision and Recall, and F-measures

- **Precision:** exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- Inverse relationship between precision & recall
- **F measure ( $F_1$  or F-score):** harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- $F_\beta$ : weighted measure of precision and recall
  - assigns  $\beta$  times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

# Classifier Evaluation Metrics: Example

---

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	<b>90</b>	<b>210</b>	300	30.00 ( <i>sensitivity</i> )
cancer = no	<b>140</b>	<b>9560</b>	9700	98.56 ( <i>specificity</i> )
Total	230	9770	10000	96.40 ( <i>accuracy</i> )

■ *Precision* =  $90/230 = 39.13\%$

*Recall* =  $90/300 = 30.00\%$

# Evaluating Classifier Accuracy: Holdout & Cross-Validation Methods

---

## ■ Holdout method

- Given data is randomly partitioned into two independent sets
  - Training set (e.g., 2/3) for model construction
  - Test set (e.g., 1/3) for accuracy estimation
- Random sampling: a variation of holdout
  - Repeat holdout  $k$  times, accuracy = avg. of the accuracies obtained

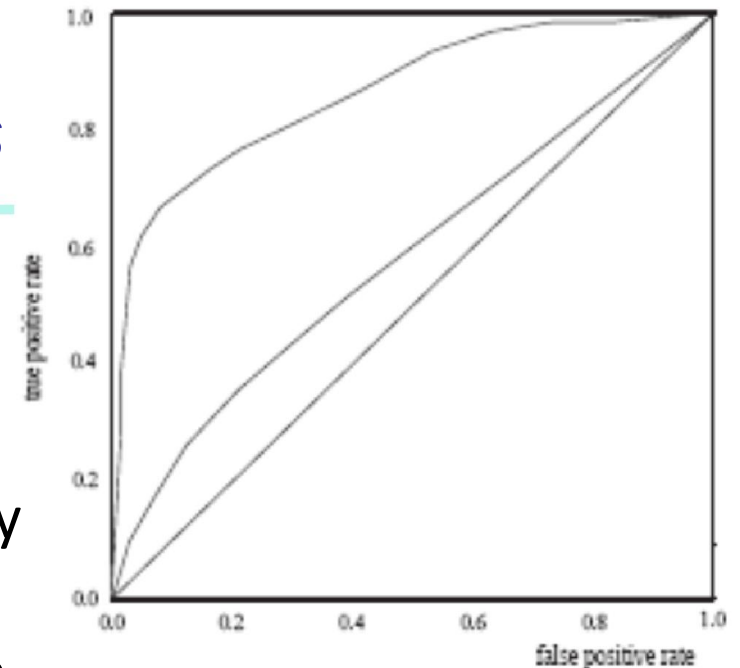
## ■ Cross-validation ( $k$ -fold, where $k = 10$ is most popular)

- Randomly partition the data into  $k$  *mutually exclusive* subsets, each approximately equal size
- At  $i$ -th iteration, use  $D_i$  as test set and others as training set
- Leave-one-out:  $k$  folds where  $k = \#$  of tuples, for small sized data



# Model Selection: ROC Curves

- **ROC** (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

# Issues Affecting Model Selection

---

- **Accuracy**
  - classifier accuracy: predicting class label
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability**
  - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules