



# COMP3420: Advanced Databases and Data Mining

Classification and prediction:  
Introduction and Decision  
Tree Induction

# Lecture outline

- Classification versus prediction
- Classification – A two step process
- Supervised versus un-supervised learning
- Evaluating classification methods
- Decision tree induction
- Attribute selection measures
  - Information gain, Gain ratio, and Gini index
- Overfitting and tree pruning
- Enhancements to basic decision tree induction
- Classification in large databases

# Classification versus 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

- Prediction

- Models continuous-valued functions, (predicts unknown or missing values)

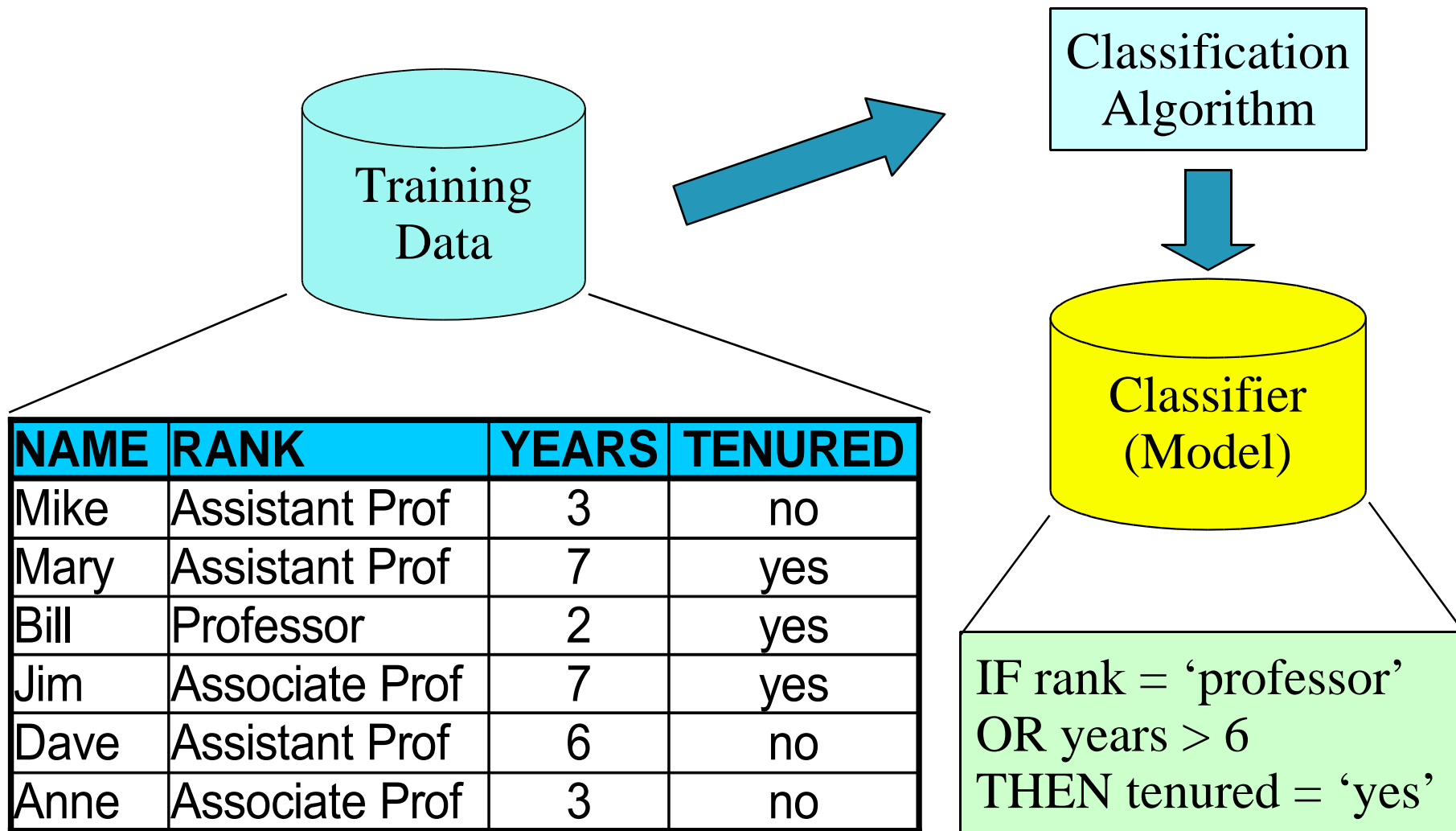
- Typical applications

- Credit approval
- Target marketing
- Medical diagnosis
- Fraud detection

# Classification – A two step process

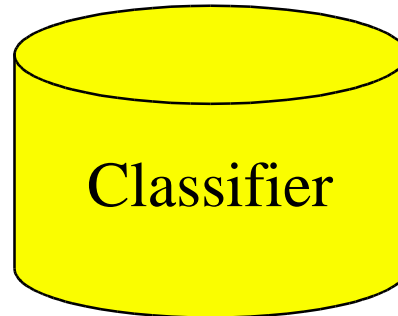
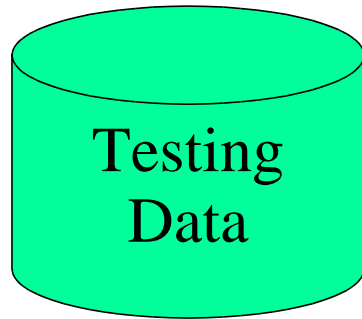
- *Model construction*: Describing a set of predetermined classes
  - Each tuple/sample/record is assumed to belong to a predefined class, as determined by the *class label attribute*
  - The set of tuples used for model construction is the *training set*
  - The model is represented as classification rules, decision trees, or mathematical formulas
- *Model usage*: For classifying future or unknown objects
  - Need to estimate the *accuracy* of the model
  - The known label of a test sample is compared with the classified results from the model
  - *Accuracy rate* is the percentage of test set samples that are correctly classified by the model
  - The test set is independent of the training set, otherwise *over-fitting* will occur
  - If the accuracy is acceptable, use the model to *classify* data objects whose class labels are not known

# Example: Model construction

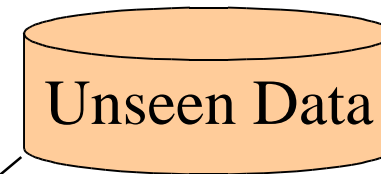


# Example: Using the model in classification

Accuracy: 75%



IF rank = 'professor'  
OR years > 6  
THEN tenured = 'yes'



(Jeff, Professor, 4)

Tenured?



**Yes**

NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

# Supervised versus un-supervised learning

- Supervised learning (classification and prediction)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations (in previous example: attribute 'Tenured' with classes `yes' and `no')
  - New data is classified based on the training set
- Un-supervised learning (clustering and associations)
  - The class label 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

# Evaluating classification methods

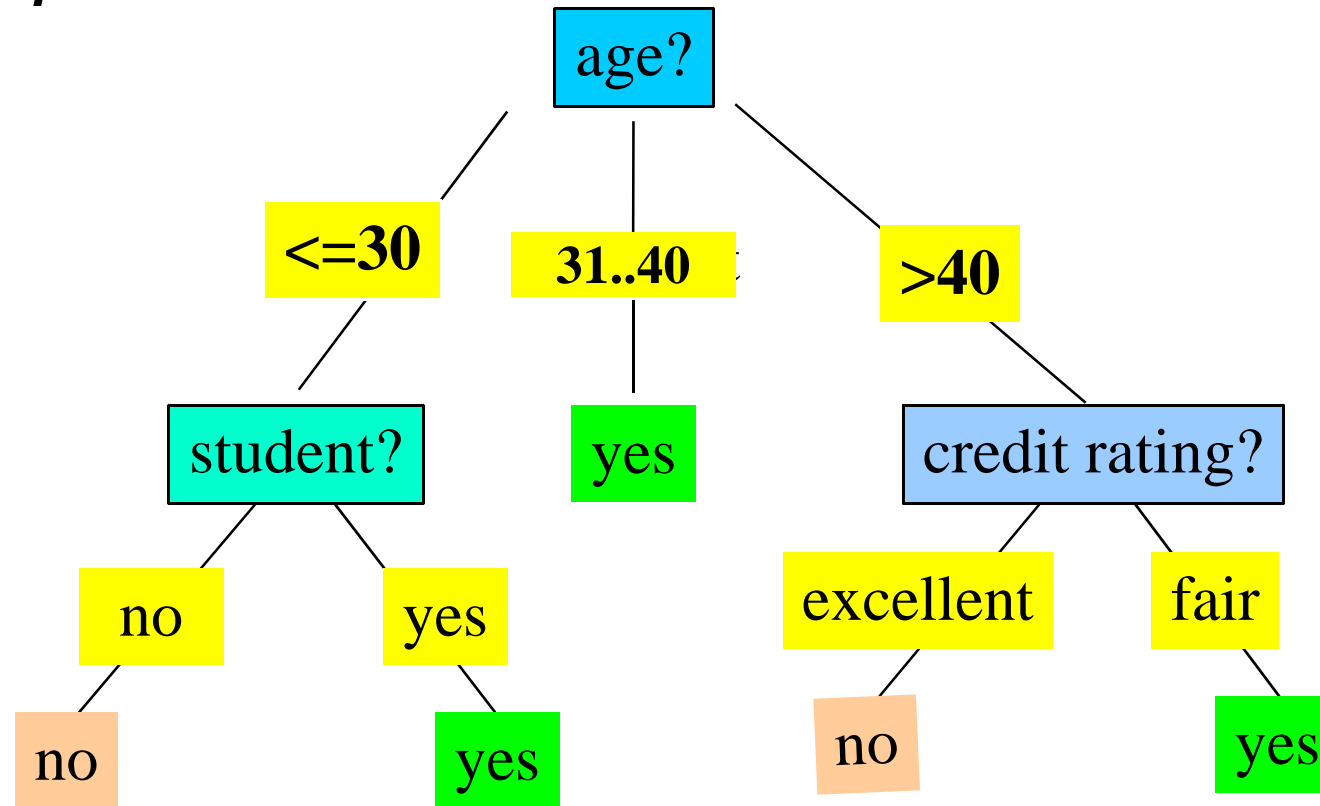
- Accuracy (or other quality measures)
  - Classifier accuracy: Predicting class label
  - Predictor accuracy: Guessing value of predicted attributes
- Speed and complexity
  - Time to construct the model (training time)
  - Time to use the model (classification/prediction time)
  - Scalability: Efficiency in handling disk-based databases
- Robustness: Handling noise and outliers
- Interpretability: Understanding and insight provided by the model
- Other measures (for example goodness of rules)
  - Such as decision tree size or compactness of classification rules

# Decision tree induction: Example training data set

What rule would you learn to identify who buys a computer?

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

# Example output: A decision tree for “*Buys computer*”



# 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 training examples are at the root
  - Take the best immediate (local) decision while building the tree
  - Data is partitioned recursively based on selected attributes
- Attributes are categorical
  - if continuous-valued, they need to be discretised in advance
- Attributes are selected on the basis of a heuristic or statistical measure
  - for example, information gain, gain ratio, or Gini index
- 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

# Attribute selection measure: Information gain

- Used in popular decision tree algorithm *ID3*
- In each step, select the attribute with the highest *information gain*
- Let  $p_i$  be the probability that an arbitrary record in data set  $D$  belongs to class  $C_i$ , estimated as  $|C_i|/|D|$  ( $| \ | = \text{number of...}$ )
- *Expected information* (information entropy) needed to classify a record in  $D$  is ( $m$  being the number of classes):

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

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

- The smaller  $\text{Info}_A(D)$  is, the purer the partitions are

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

- *Information gained* by using attribute  $A$  for branching:

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)_{12}$$

# Information gain example

Class P: "Buys computer" = "yes"

Class N: "Buys computer" = "no"

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

Age	$p_i$	$n_i$	Partition $j$
$\leq 30$	2	3	1
31...40	4	0	2
$> 40$	3	2	3

Age	Income	Student	Credit rating	Buys computer
$\leq 30$	high	no	fair	no
$\leq 30$	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
$\leq 30$	medium	no	fair	no
$\leq 30$	low	yes	fair	yes
$> 40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
$> 40$	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14} Info(D_0) + \frac{4}{14} Info(D_1) + \frac{5}{14} Info(D_2) = 0.694$$

From this follows:

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

And similar:

$$Gain(Income) = 0.029$$

$$Gain(Student) = 0.151$$

$$Gain(Credit\ rating) = 0.048$$

# Calculate information gain for continuous-value attributes

- Let  $A$  be an attribute with continuous values (e.g. income)
- We have to determine the best *split-point* for  $A$ 
  - Sort the values of  $A$  in increasing order
  - Typically, the mid-point 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  $A$ ,  $Info_A(D)$ , is selected as the split-point for  $A$
  - To determine the best split for attribute  $A$ , only one pass through the sorted values is required
- Split:
  - $D_1$  is the set of records in  $D$  satisfying  $A \leq \text{split-point}$ , and  $D_2$  is the set of tuples in  $D$  satisfying  $A > \text{split-point}$

# Attribute selection measure: Gain ratio

- The *information gain* measure is biased towards attributes with a large number of values
- C4.5 (successor of ID3) uses *gain ratio* to overcome the problem (normalisation to information gain):

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

- $GainRatio(A) = Gain(A) / SplitInfo(A)$
- The attribute with the maximum gain ratio is selected as the splitting attribute
- Example: Gain ratio on attribute *income*

$$GainRatio(income) = 0.029 / 0.926 = 0.031$$

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2 \left( \frac{4}{14} \right) - \frac{6}{14} \times \log_2 \left( \frac{6}{14} \right) - \frac{4}{14} \times \log_2 \left( \frac{4}{14} \right) = 0.926$$

# Attribute selection measure: Gini index

- Used in *CART*, *IBM Intelligent Miner*

- If a data set  $D$  contains records from  $m$  classes, the Gini index,  $gini(D)$ , is defined as:

$$gini(D) = 1 - \sum_{j=1}^m p_j^2$$

- $p_j$  is the relative frequency of class  $j$  in  $D$

- If  $D$  is split on attribute  $A$  into two sub-sets  $D_1$  and  $D_2$ , the Gini index,  $gini_A(D)$ , is defined as:

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in impurity:  $gini(A) = gini(D) - gini_A(D)$
- The attribute that provides the smallest  $gini_A(D)$  (or the largest reduction in impurity) is chosen to split a node
  - Need to enumerate all possible splitting subsets/points for each attribute

# Gini index example

- Example data set  $D$  has 9 records in class “*Buys computers*” = “yes” and 5 in “*Buys computers*” = “no”

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

- Suppose the attribute *income* partitions the data set into 10 records in  $D_1$ : {“medium”, “low”} and 4 in  $D_2$ : {“high”}:

$$\begin{aligned} gini_{income \in \{low, medium\}}(D) &= \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2) \\ &= \left(\frac{10}{14}\right) \left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \left(\frac{4}{14}\right) \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443 \end{aligned}$$

- Split on {“low”, “high”} and {“medium”} gives a Gini index of 0.458, and on {“medium”, “high”} and {“low”} gives 0.450
  - Best split for attribute *income* is on {“low”, “medium”} and {“high”} because it minimises the Gini index

# Comparing attribute selection measures

- The three measures, in general, return good results, but...
- Information gain
  - Is biased towards multivalued attributes
- Gain ratio
  - Tends to prefer unbalanced splits in which one partition is much smaller than the others
- Gini index
  - Is biased to multivalued attributes
  - Has difficulty when the number of classes is large
  - Tends to favor tests that result in equal-sized partitions and purity in both partitions
- There are many other selection measures!

# 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
  - Classify training records very well, but poor accuracy for unseen records
- Two approaches to avoid overfitting
  - *Prepruning*: Halt tree construction early: do not split a node if this would result in a 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 (feature) construction**
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication

# 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 fast learning speed (compared to other classification methods)
  - Convertible to simple and easy to understand classification rules
  - Can use SQL queries for accessing databases
  - Comparable classification accuracy with other methods