Decision Tree

- Decision tree is a classifier in the form of a tree structure
  - Decision node: specifies a test on a single attribute
  - Leaf node: indicates the value of the target attribute
  - Arc/edge: split of one attribute
  - Path: a disjunction of test to make the final decision
- Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

Decision Tree

- Decision trees used in data mining are of two main types:
- **Classification tree** analysis is when the predicted outcome is the class to which the data belongs.
- **Regression tree** analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).

Decision Tree

learning algorithm

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*.

```
create a node N;
if tuples in D are all of the same class, C then
    return N as a leaf node labeled with the class C;
if attribute_list is empty then
    return N as a leaf node labeled with the majority class in D; // majority voting
 apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
 label node N with splitting_criterion;
for each outcome j of splitting_criterion
// partition the tuples and grow subtrees for each partition
  let D_j be the set of data tuples in D satisfying the outcome j; // a partition
  if D_j is empty then
        attach a leaf labeled with the majority class in D to node N;
  else attach the node returned by Generate_decision_tree(D_j, attribute_list) to node N;
endfor
return N;
```

Decision Tree

The decision-tree learning algorithms include:

- ID3
- C4.5
- CART
- CHAID
- MARS

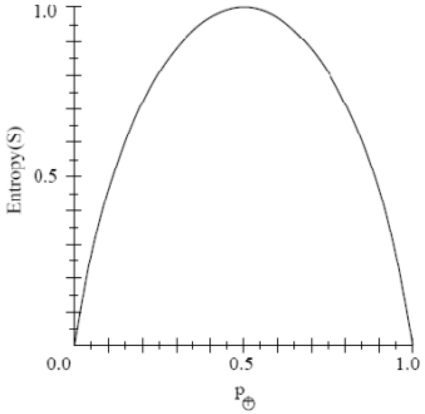
Decision Tree——ID3

- Information entropy

Values range from 0 – 1 to represent the entropy of information

$$Entropy(D) \equiv \sum_{i=1}^{c} - p_i \log_2(p_i)$$

$$Entropy(D) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$



Decision Tree——ID3

- Information entropy

Values range from 0 – 1 to represent the entropy of information

$$Entropy(D) \equiv \sum_{i=1}^{c} - p_i \log_2(p_i)$$

- Information Gain

Information gain is used as an attribute selection measure. Pick the attribute that has the highest Information gain

$$Gain(D, A) = Entropy(D) - \sum_{j=1}^{v} \frac{|D_j|}{|D|} Entropy(D_j)$$

Decision Tree——ID3

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

Decision Tree ——ID3

- Class P: *buys_computer* = "yes"
- Class N: *buys_computer* = "no"

$$Entropy(D) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

- Compute the expected information requirement for each attribute: start with the attribute *age*

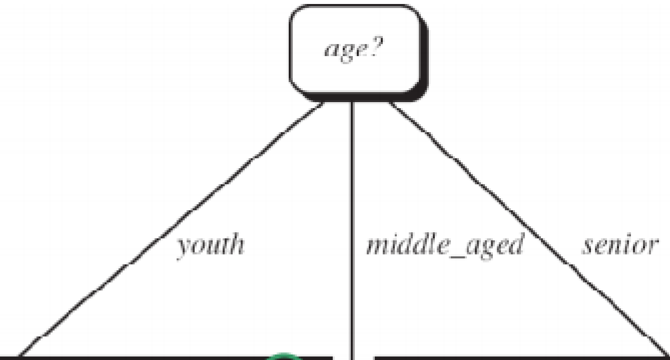
$$\begin{aligned}
&Gain(age, D) \\
&= Entropy(D) - \sum_{v \in \{Youth, Middle-aged, Senior\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(D) - \frac{5}{14} Entropy(S_{youth}) - \frac{4}{14} Entropy(S_{middle_aged}) - \frac{5}{14} Entropy(S_{senior}) \\
&\boxed{= 0.246}
\end{aligned}$$

$$Gain\,(income, D) = 0.029$$

$$Gain\,(student, D) = 0.151$$

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

Decision Tree ——ID3

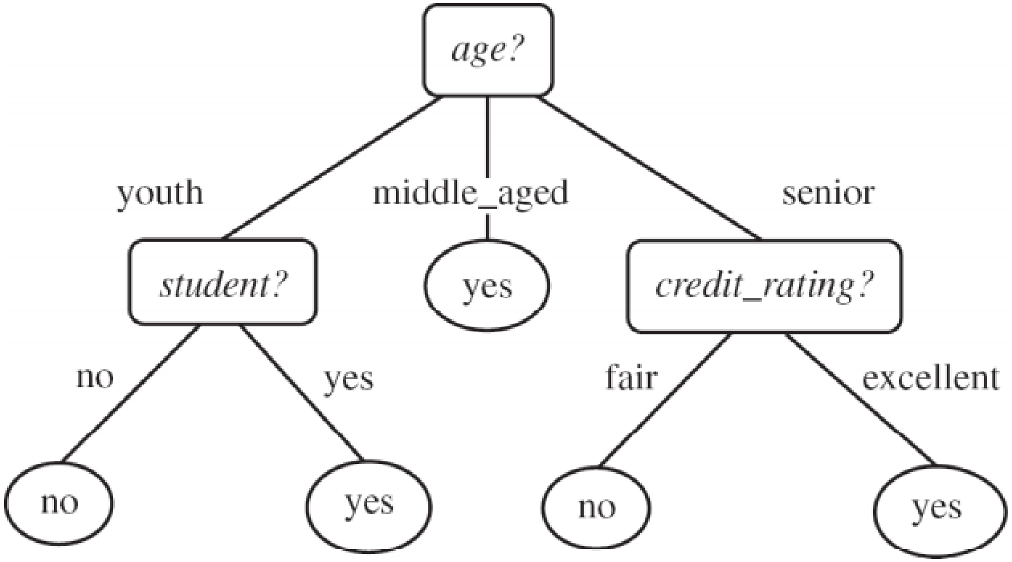


income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

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

Decision Tree ——ID3



Problem:
sensitive to attributes that have many values.

Decision Tree ——C4.5

- Gain ratio

$$GainRatio(D, A) = \frac{Gain(D, A)}{IV(A)}$$

- Intrinsic value

$$IV(A) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}$$

Problem:
sensitive to attributes with less value.

Decision Tree ——CART

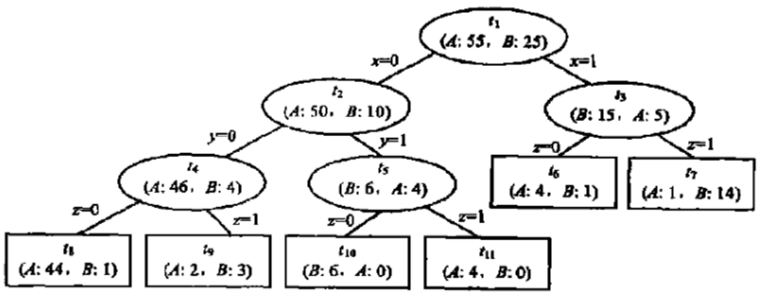
- Gini value

$$Gini(D) = 1 - \sum_{k=1}^{|y|} p_k^2$$

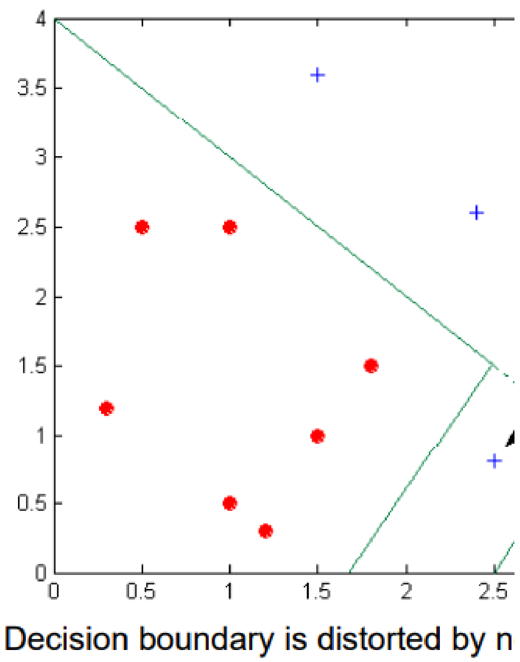
- Gini index

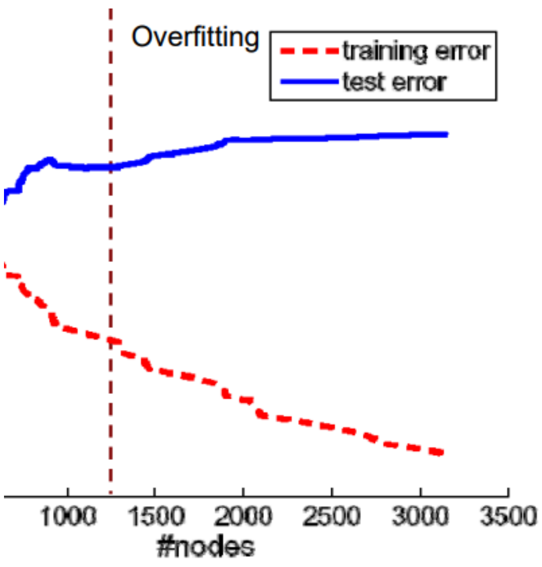
$$Gini\_index(D, A) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v)$$

- Binary tree



Decision Tree ——pruning



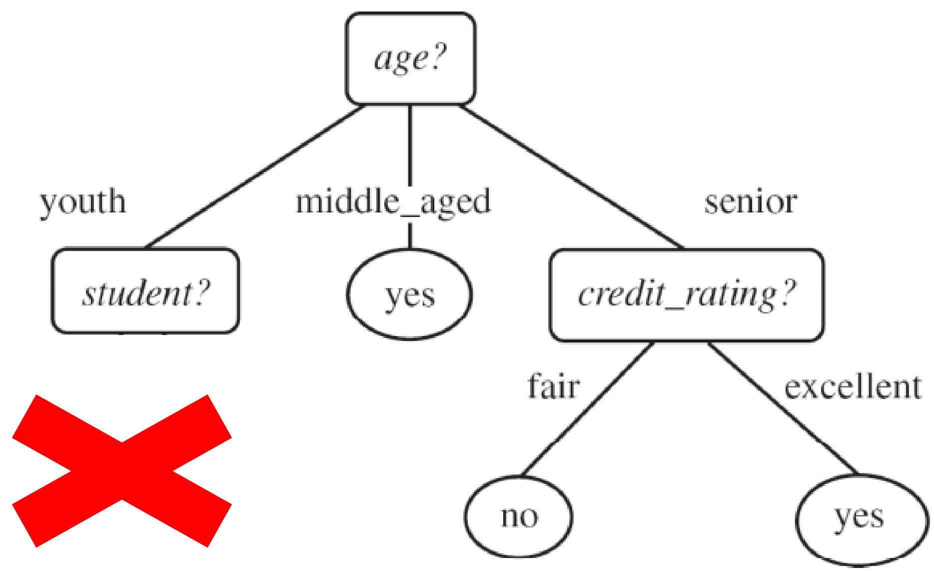


Decision Tree ——pruning

- Pre-pruning

Stop if expanding the current node does not improve the precision of the validation set

Before: 71.4%
After: 57.1%



Before: 42.9%
After: 71.4%

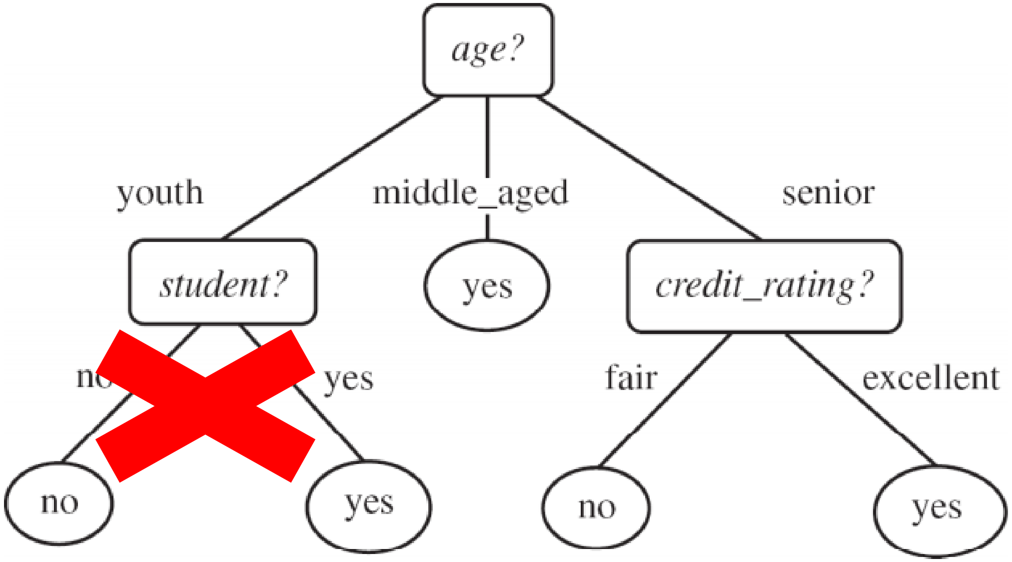
Decision Tree ——pruning

- Post-pruning

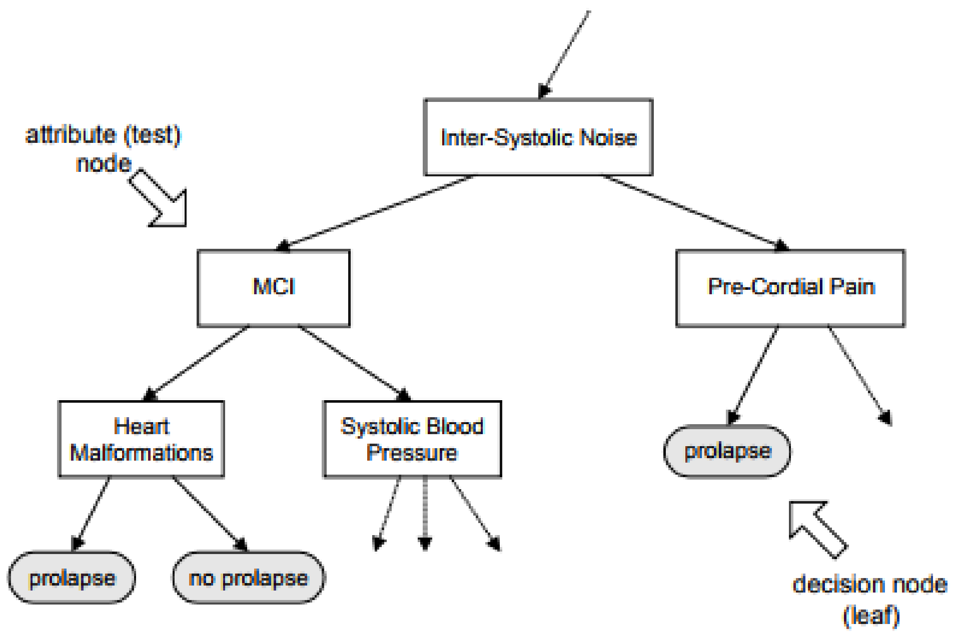
Grow decision tree to its entirety

If generalization error improves after trimming, replace sub-tree by a leaf node. Class label of leaf node is determined from majority class of instances in the sub-tree

Before: 57.1 %
After: 71.4%



Decision Tree



Decision Tree