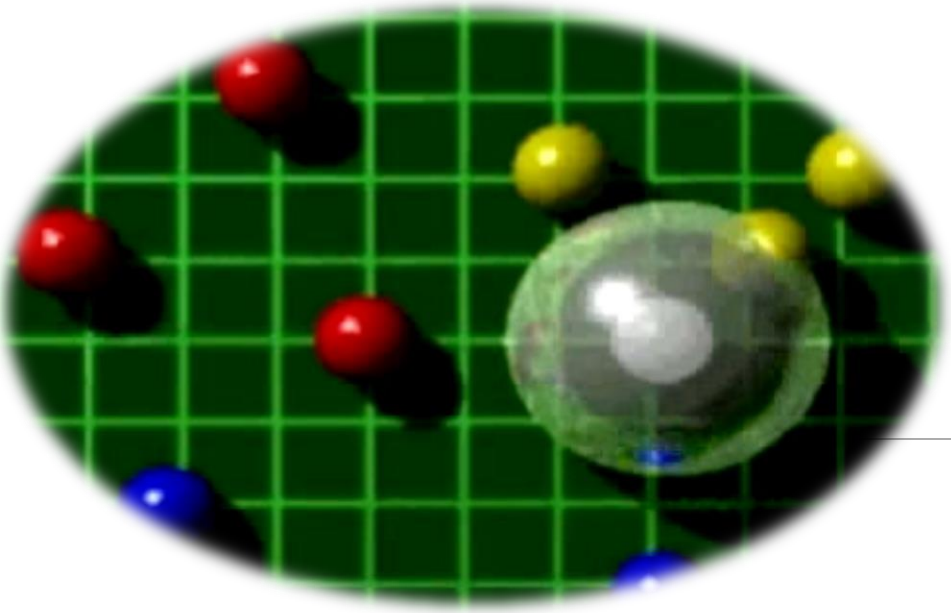
STATISTICAL METHODS IN DATA MINING

Dr. Alper VAHAPLAR



Previously on Course…

- ✓Exploring Data,
- ✓Data Visualization,
- ✓Clustering
- ✓Classification
  - ✓K-nearest neghbor
  - ✓Bayesian Classification

Supervised – Unsupervised Learning

- ✓ **Supervised learning**
  - is a ***machine learning*** technique for creating a function from training data.
  - The ***training data*** consist input objects (typically vectors), and desired outputs.
  - The output can be a continuous value (called ***regression***), or can predict a class label of the input object (called ***classification***).
  - The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples.
  - The learner has to generalize from the presented data to unseen situations in a "reasonable" way
- ✓ **Unsupervised learning**
  - is a method of ***machine learning*** where a model is fit to observations.
  - It is distinguished from ***supervised learning*** by the fact that there is ***no a priori*** output.
  - In unsupervised learning, a data set of input objects is gathered. Unsupervised learning then typically treats input objects as a set of ***random variables***. A joint density model is then built for the data set.

Classification

- ✓ The task of assigning *previously unseen* objects to one of several *predefined categories.*
- ✓ Finding a model for class attribute as a function of other attributes.
- ✓ Predicts categorical labels (unlike estimation or prediction).
- ✓ Is a 2-step process:

1. **Model construction**
   - 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, trees, or mathematical formulae.
2. **Model usage** (Classifying future or unknown objects)
   - Estimate accuracy rate of the model on a test set,
   - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known.

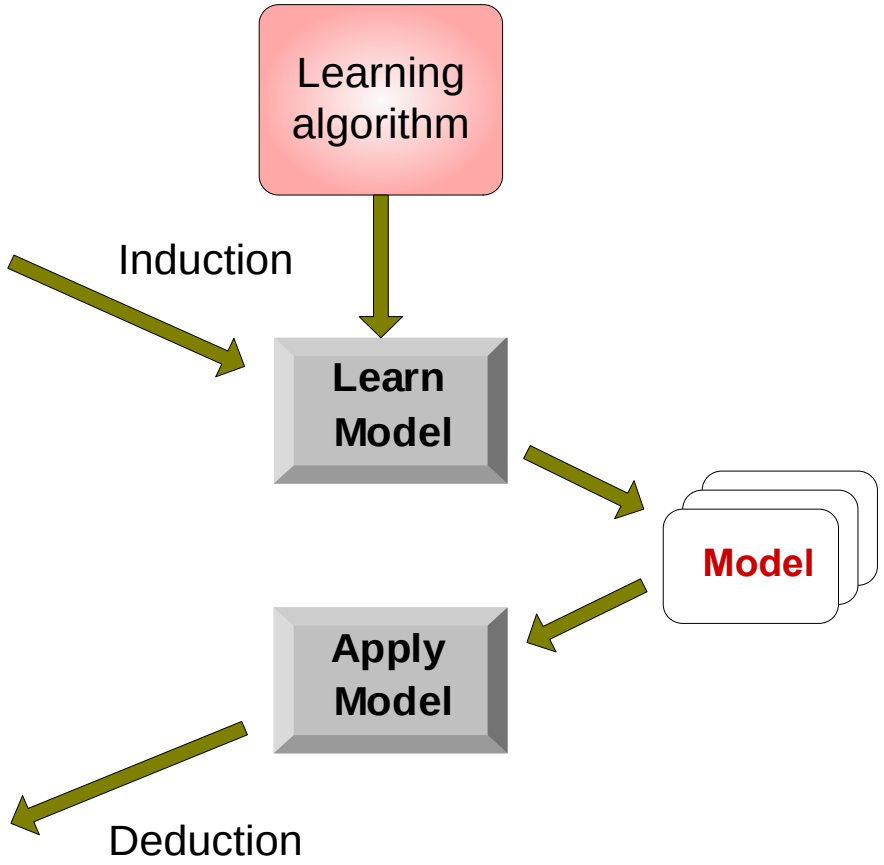
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

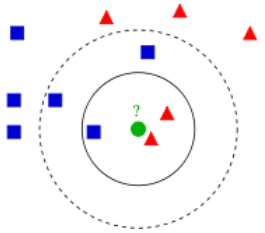
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification Techniques

1. **K-Nearest Neighbor**

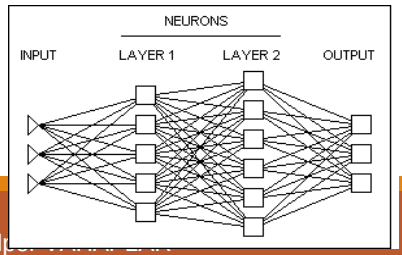


2. **Bayesian Classification**

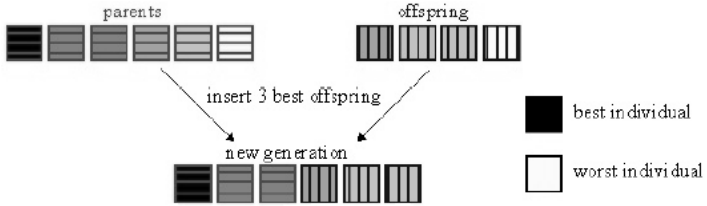
$$c = \max_{c_j} \frac{p(c_j)}{p(d)} \prod_{i=1}^{n} p(a_i \mid c_j)$$

3. **Decision Trees**

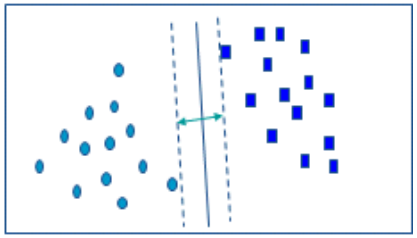
4. **Neural Networks**



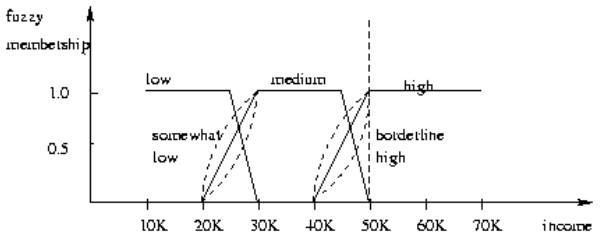
5. **Genetic Algorithms**



6. **Support Vector Machines (SVM)**

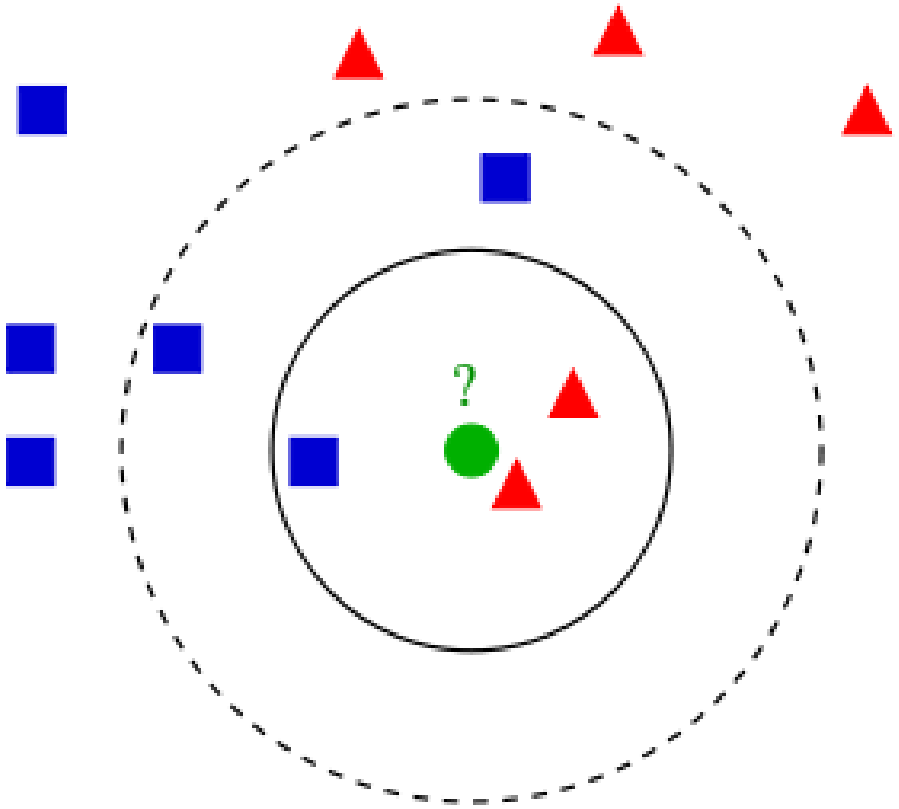


7. **Fuzzy Set Approaches**



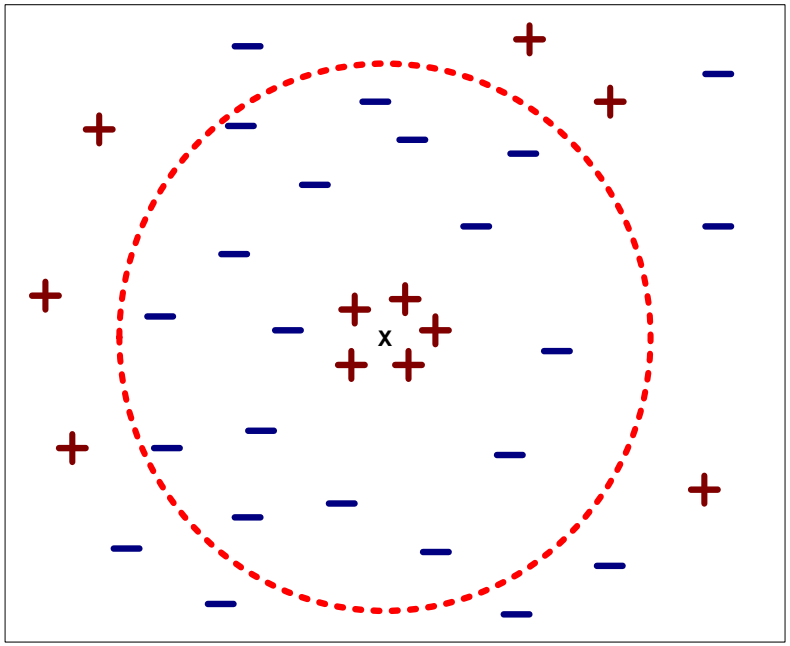
k-nearest Neighborhood Classification

- ✓ Is an example of *instance based learning*,
- ✓ Classification for a new unclassified record is found by comparing it to ***k*** most similar records in the training set.



k-nearest Neighborhood Classification

- ✓ Choosing the value of k:
  - If k is too small, sensitive to outliers or noise.
  - If k is too large, locally interesting behaviour will be overlooked.



k-nearest Neighborhood Classification

- ✓ Advantages:
  - No model is built,
  - Building model is cheap,
  - Simple technique, easily implemented,
  - Well suited for records with multiple class labels,
  - Can sometimes be the best method
- ✓ Disadvantages:
  - Hard to decide *k*,
  - Requires computation of a distance for all new records.

k-NN for Estimation and Prediction

- ✓ k-NN may be used for estimation and prediction as well as for *continuous* valued target variables.
- ✓ Locally Weighted Averaging

$$\hat{y}_{new} = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

$$w_i = 1/\mathrm{d}(\mathrm{new}, x_i)^2$$

Bayesian Classifiers

- ✓ A probabilistic framework for solving classification problems
- ✓ Consider each attribute and class label as random variables
- ✓ Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

- ✓ Equivalent to choosing value of C that maximizes

  $P(A_1, A_2, \ldots, A_n | C)\, P(C)$

Today…

- ✓Decision Trees
- ✓CART,
- ✓C4.5 – C5

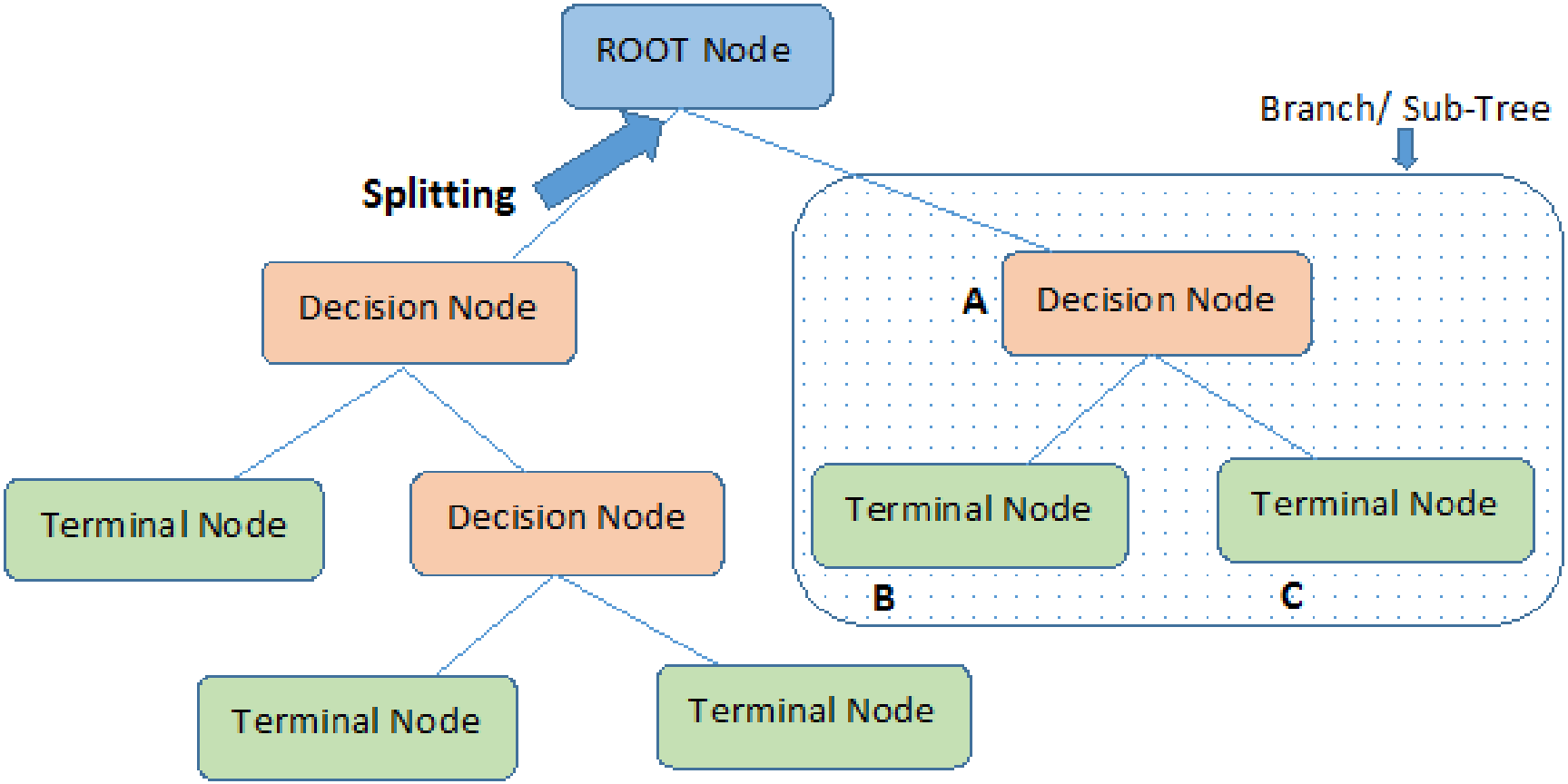
Decision Trees

- A decision tree is a collection of ***decision nodes***, connected by ***branches***, extending downward from the ***root node*** until terminating in ***leaf nodes***.
- Beginning at the ***root node***, which by convention is placed at the top of the decision tree diagram, ***attributes*** are tested at the decision nodes, with each possible outcome resulting in a ***branch***. Each branch then leads either to another ***decision node*** or to a ***terminating*** leaf node.

Decision Trees

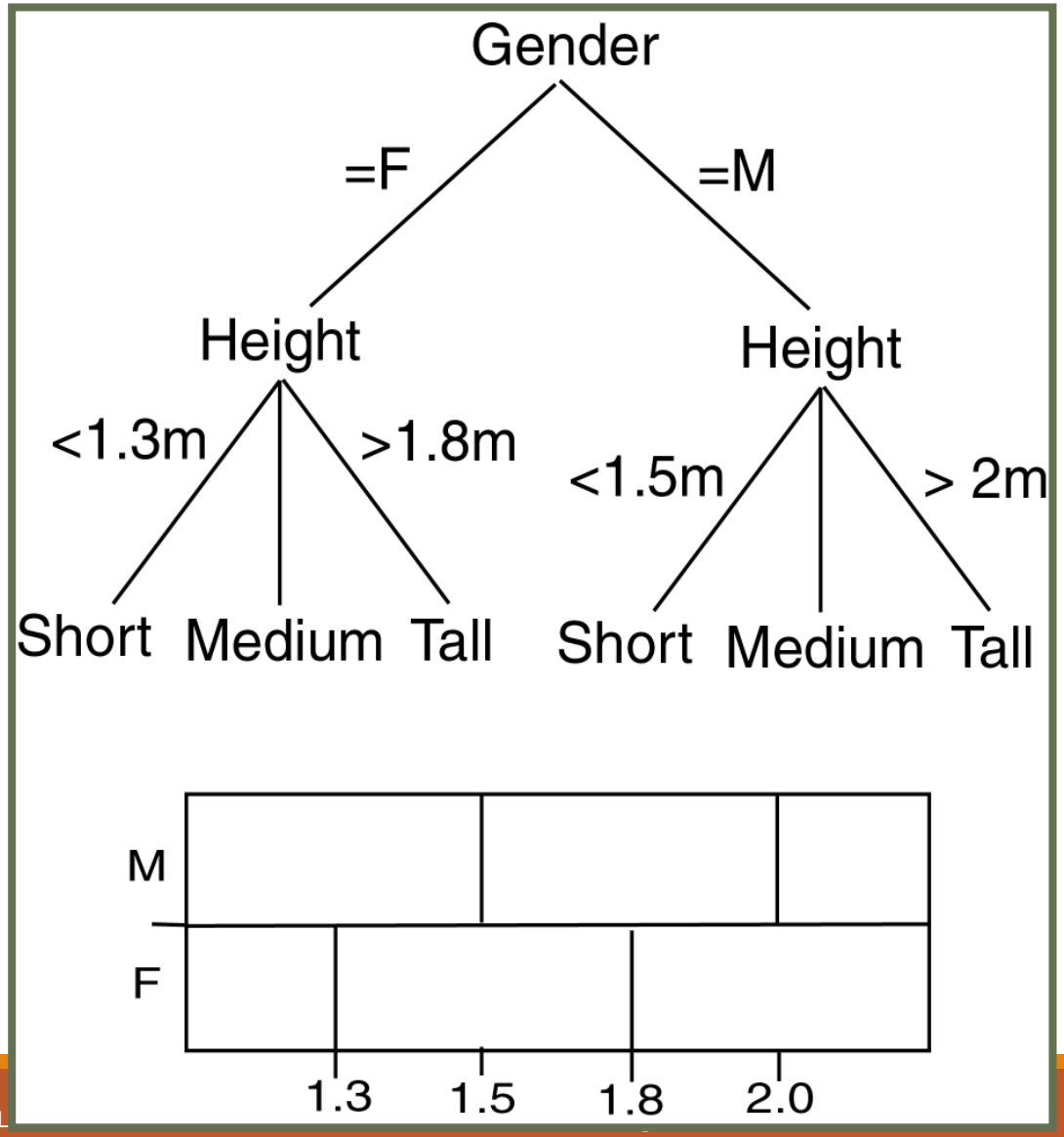
- ***Decision Tree*** is a tree where
  - the model begins with the ***root*** for the training set,
  - ***internal nodes*** are simple decision rules tested on one or more attributes,
  - Each node makes a split into various number of ***branches***, according to the outcome of the test,
  - ***leaf nodes*** represent the prediction for the class labels.
  - If all records in a leaf node are of the same class, it is called a ***pure node***.

Decision Trees



Note:- A is parent node of B and C.

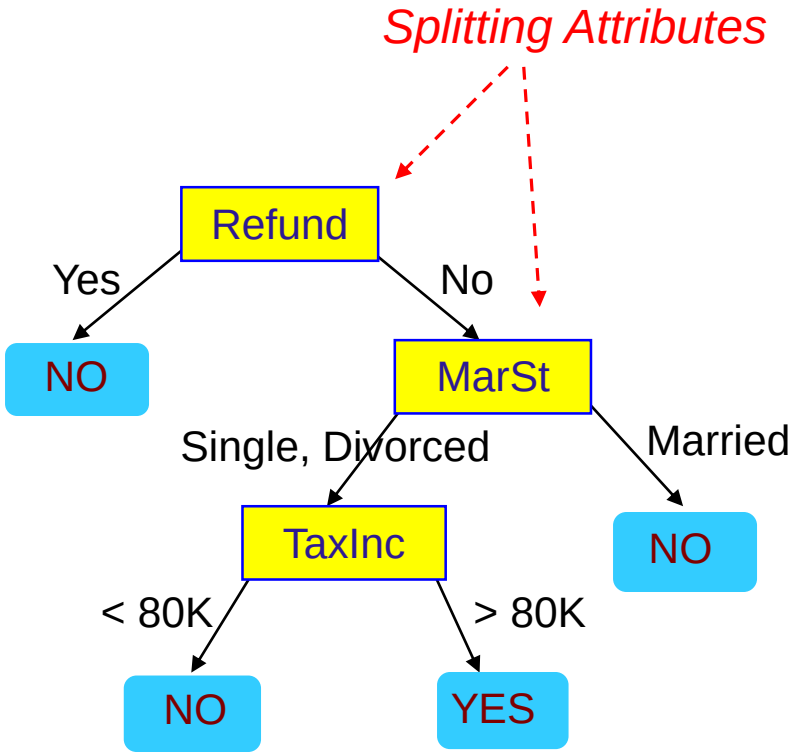
Decision Trees



Decision Trees

Tid	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

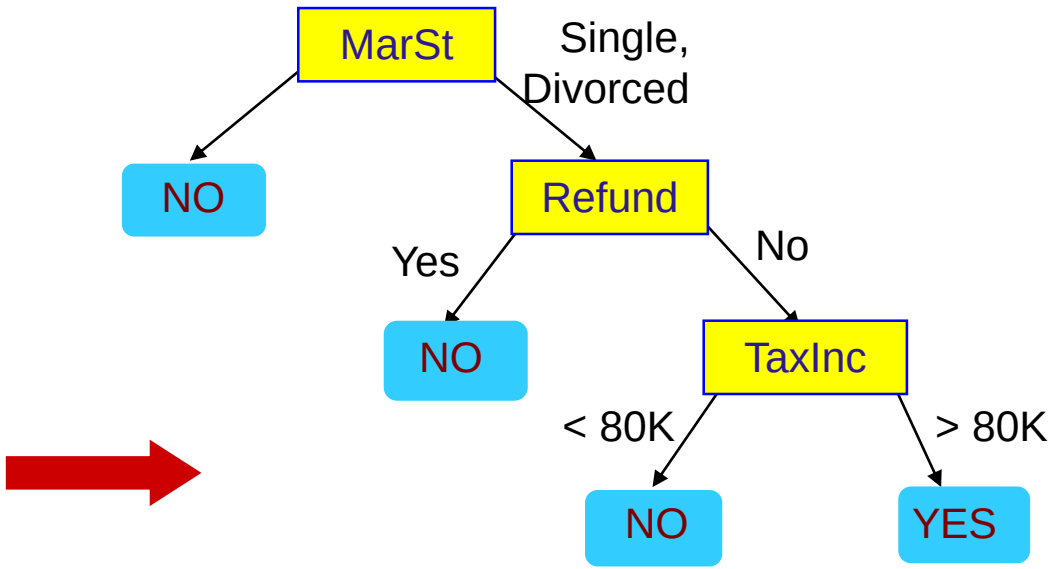


Model: Decision Tree

Decision Trees

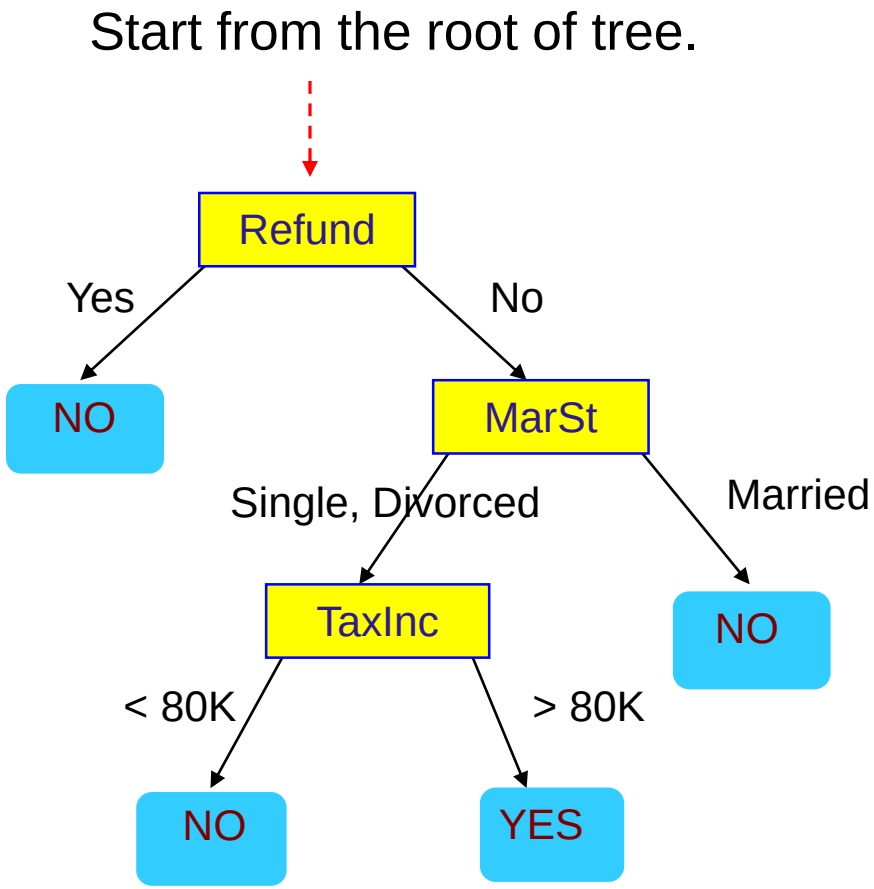
Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



There could be more than one tree that fits the same data!

Apply Model to Test Data



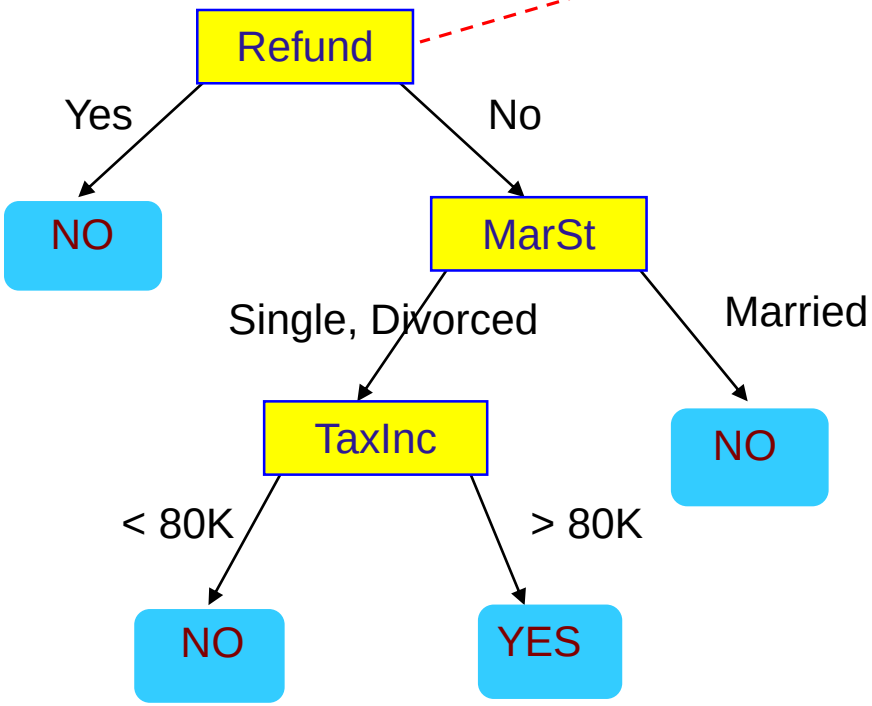
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

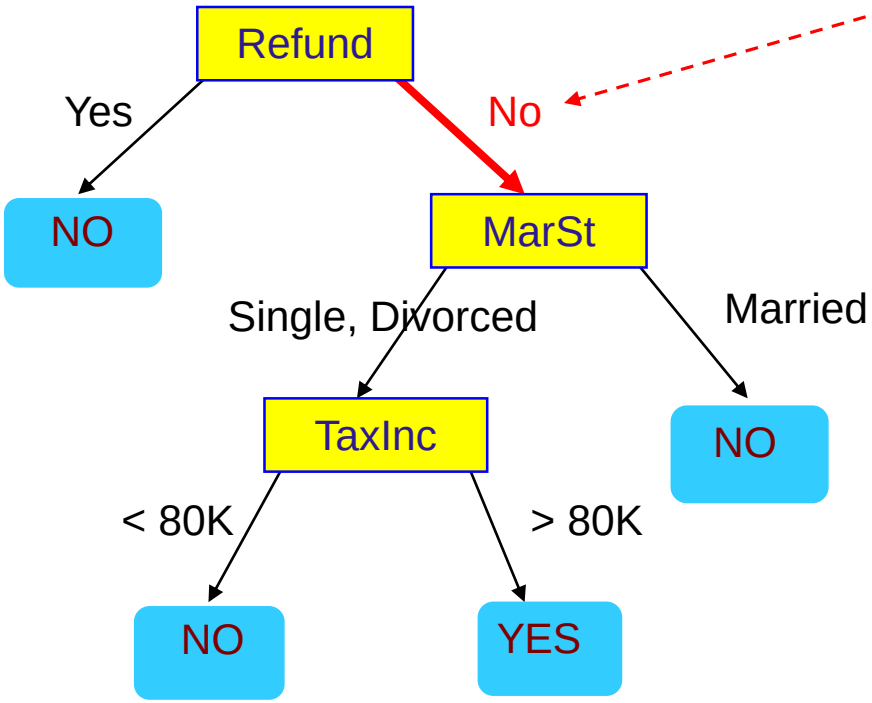
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

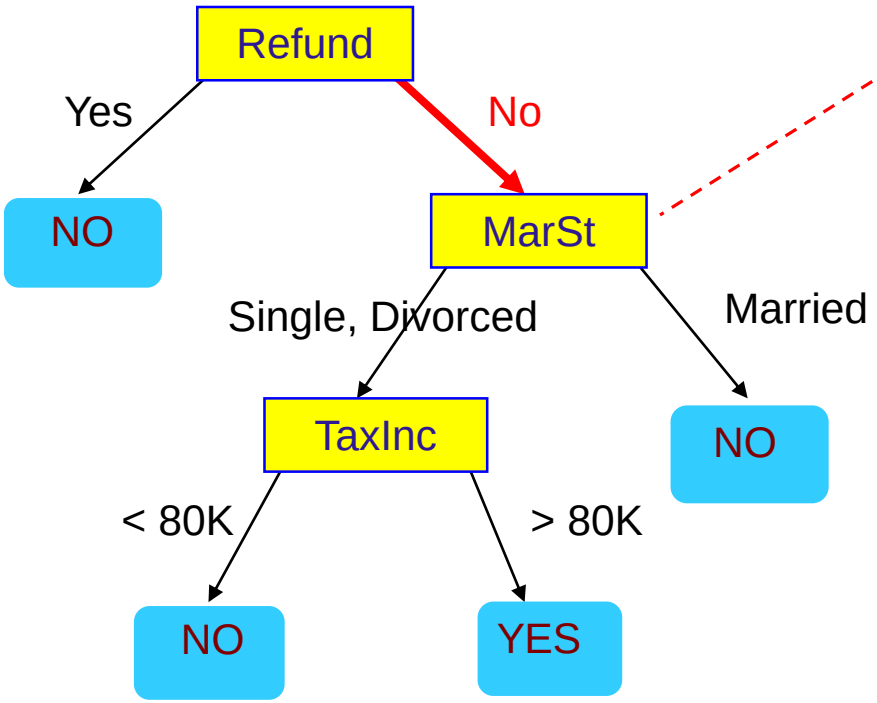
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

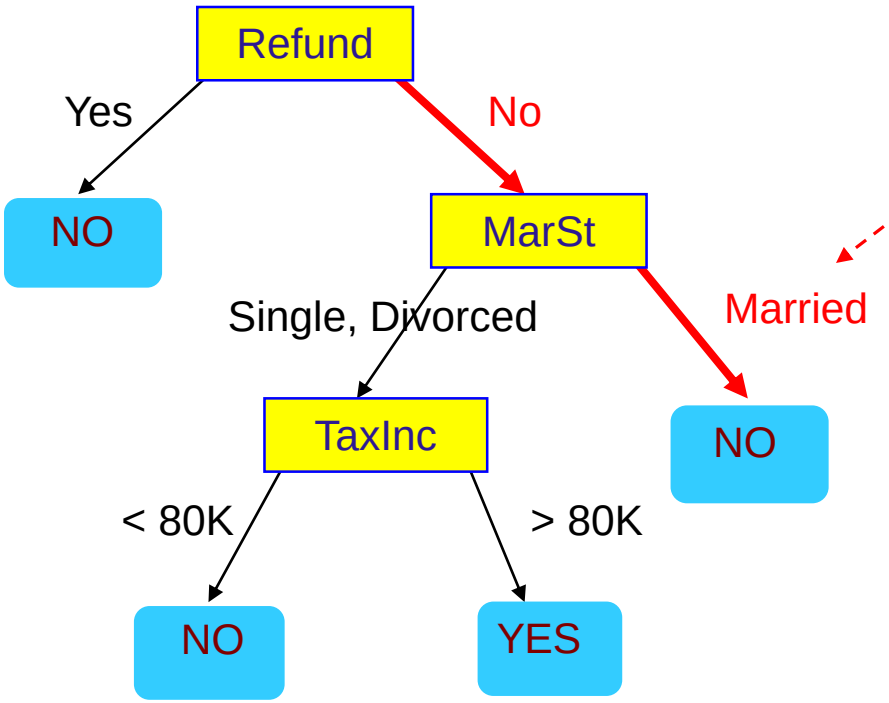
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

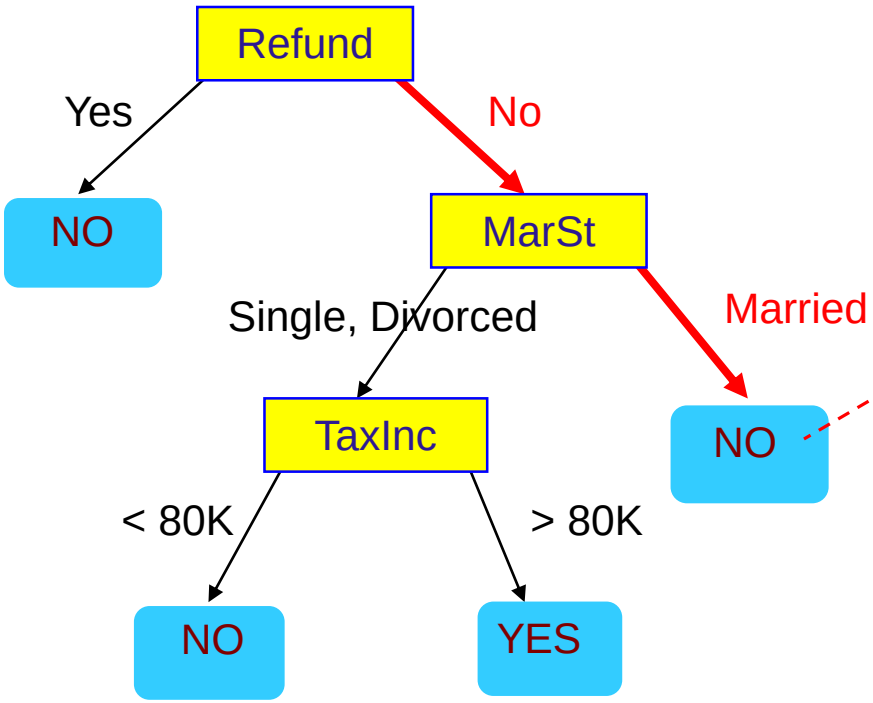
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to *"No"*

Decision Tree Algorithms

Some requirements:

1. Pre-classified target variables.
2. A training data set – rich and varied.
3. Discrete target attribute classes.

✓ Greedy strategy
- Split the records based on an attribute test that optimizes certain criterion.

✓ Issues
- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

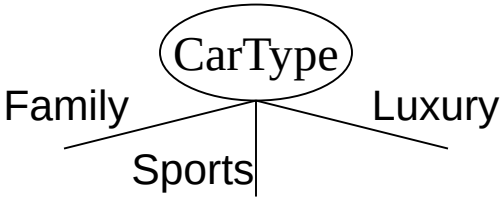
How to Specify Test Condition?

- ✓ Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

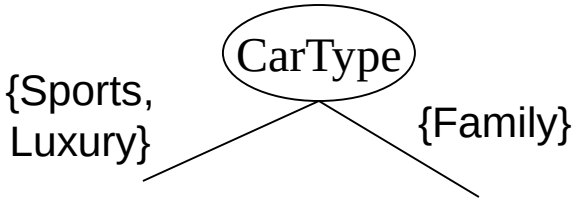
- ✓ Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

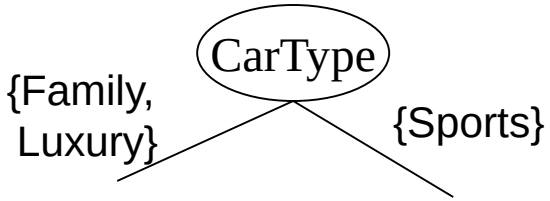
- ✓ Multi-way split: Use as many partitions as distinct values.



- ✓ Binary split: Divides values into two subsets.
  Need to find optimal partitioning.

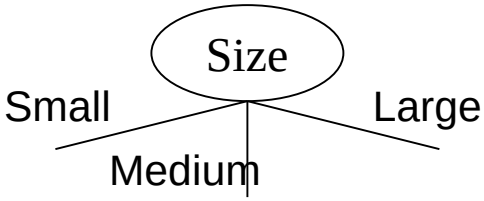


OR

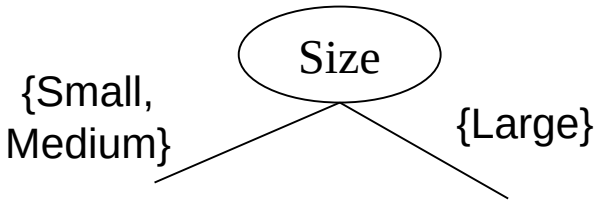


Splitting Based on Ordinal Attributes

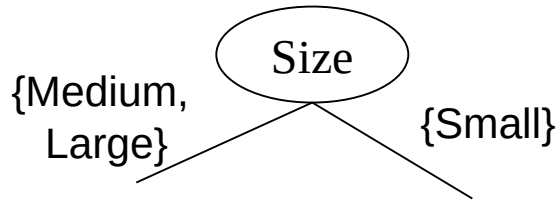
- ✓ Multi-way split: Use as many partitions as distinct values.



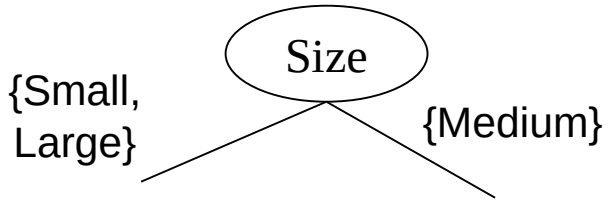
- ✓ Binary split: Divides values into two subsets.
  Need to find optimal partitioning.



OR



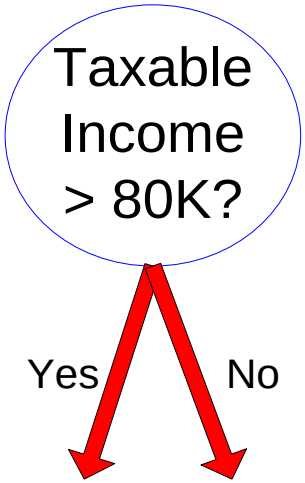
- ✓ What about this split?



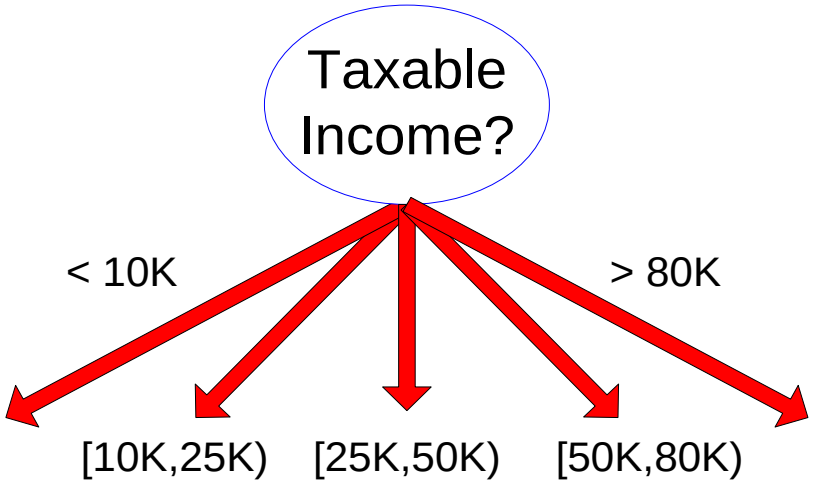
Splitting Based on Continuous Attributes

- ✓ Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval binning, equal frequency binning (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



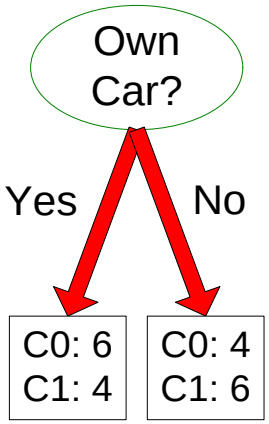
(i) Binary split

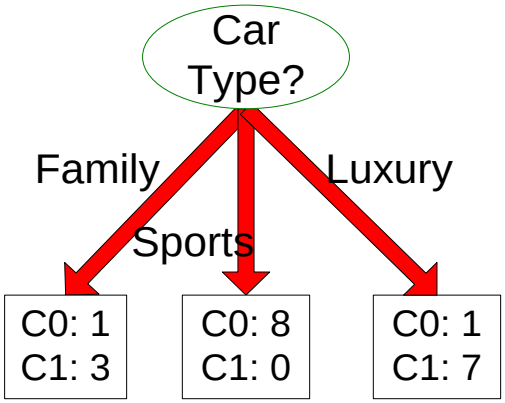


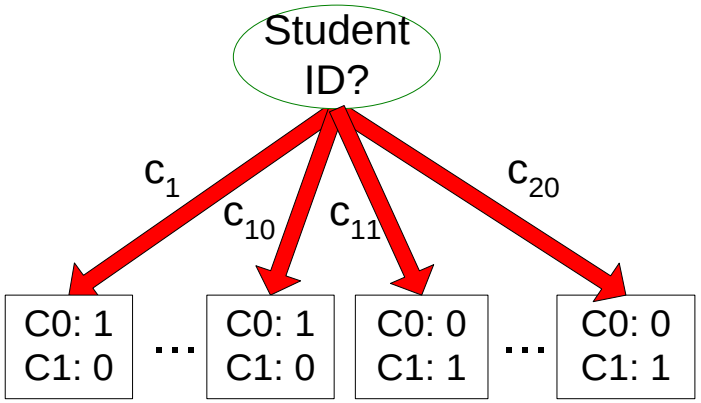
(ii) Multi-way split

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1







Which test condition is the best?

How to determine the Best Split

- ✓ Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- ✓ Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

Decision Tree Algorithms

- ✓ ID3
  - Quinlan (1981)
  - Tries to reduce expected number of comparison
- ✓ C 4.5
  - Quinlan (1993)
  - It is an extension of ID3
  - Used in many data mining applications (C5.0)
  - Also used for rule induction
- ✓ CART
  - Breiman, Friedman, Olshen, and Stone (1984)
  - Classification and Regression Trees
- ✓ CHAID
  - Kass (1980)
  - Oldest decision tree algorithm
  - Well established in database marketing industry
- ✓ QUEST
  - Loh and Shih (1997)

CART

- ✓ Classification And Regression Tree
- ✓ Developed 1974-1984 by 4 statistics professors;
  - Leo Breiman (Berkeley), Charles Stone (Berkeley), Jerry Friedman (Stanford), Richard Olshen (Stanford)
- ✓ CART is a non-parametric tool of discriminant analysis which is designed to represent decision rules in a form of so called binary trees.
- ✓ is ***a binary tree*** – each decision node splits into exactly 2 branches.
- ✓ uses a "***measure of goodness***" for finding optimal split *s* at node *t.*

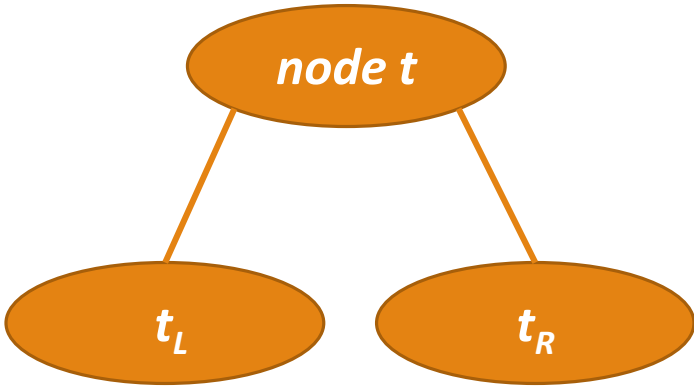
$$\Phi(s|t) = 2P_L P_R \sum_{j=1}^{\#\text{ classes}} |P(j|t_L) - P(j|t_R)|$$

CART

$$\Phi(s|t) = 2 P_L P_R \sum_{j=1}^{\#\ classes} |P(j|t_L) - P(j|t_R)|$$

$t_L = left\ child\ node of\ node\ t$

$t_R = right\ child\ node of\ node\ t$



$$P_L = \frac{number\ of\ records\ at\ t_L}{number\ of\ records\ in\ training\ set}$$

$$P_R = \frac{number\ of\ records\ at\ t_R}{number\ of\ records\ in\ training\ set}$$

$$P(j \mid t_L) = \frac{number\ of\ class\ j\ records\ at\ t_L}{number\ of\ records\ in\ t}$$

$$P(j \mid t_R) = \frac{number\ of\ class\ j\ records\ at\ t_R}{number\ of\ records\ in\ t}$$

CART – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

CART – Example

- ✓ Candidate Binary Splits

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

Candidate Split	Left Child Node, t_L	Right Child Node, t_R
1	$Savings = low$	$Savings \in \{medium, high\}$
2	$Savings = medium$	$Savings \in \{low, high\}$
3	$Savings = high$	$Savings \in \{low, medium\}$
4	$Assets = low$	$Assets \in \{medium, high\}$
5	$Assets = medium$	$Assets \in \{low, high\}$
6	$Assets = high$	$Assets \in \{low, medium\}$
7	$Income \leq \$25{,}000$	$Income > \$25{,}000$
8	$Income \leq \$50{,}000$	$Income > \$50{,}000$
9	$Income \leq \$75{,}000$	$Income > \$75{,}000$

Cand	Left Child Node, t_L	Right Child Node, t_R
1	*Savings* = *low*	*Savings* ∈ {*medium, high*}
2	*Savings* = *medium*	*Savings* ∈ {*low, high*}
3	*Savings* = *high*	*Savings* ∈ {*low, medium*}
4	*Assets* = *low*	*Assets* ∈ {*medium, high*}
5	*Assets* = *medium*	*Assets* ∈ {*low, high*}
6	*Assets* = *high*	*Assets* ∈ {*low, medium*}
7	*Income* ≤ $25,000	*Income* > $25,000
8	*Income* ≤ $50,000	*Income* > $50,000
9	*Income* ≤ $75,000	*Income* > $75,000

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

$$\Phi(s|t) = 2P_L P_R \sum_{i=1}^{\#\text{ classes}} |P(j|t_L) - P(j|t_R)|$$

$$Q(s|t) = \sum_{i=1}^{\#classes} |P(j|t_L) - P(j|t_R)|$$

Split	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_L P_R$	$Q(s\|t)$	$\Phi(s\|t)$
1	0.375	0.625	G: .333 B: .667	G: .8 B: .2	0.46875	0.934	0.4378
2	0.375	0.625	G: 1 B: 0	G: 0.4 B: 0.6	0.46875	1.2	0.5625
3	0.25	0.75	G: 0.5 B: 0.5	G: 0.667 B: 0.333	0.375	0.334	0.1253
4	0.25	0.75	G: 0 B: 1	G: 0.833 B: 0.167	0.375	1.667	0.6248
5	0.5	0.5	G: 0.75 B: 0.25	G: 0.5 B: 0.5	0.5	0.5	0.25

$$\Phi(s|t) = 2P_L P_R \sum_{i=1}^{\#\text{ classes}} |P(j|t_L) - P(j|t_R)|$$

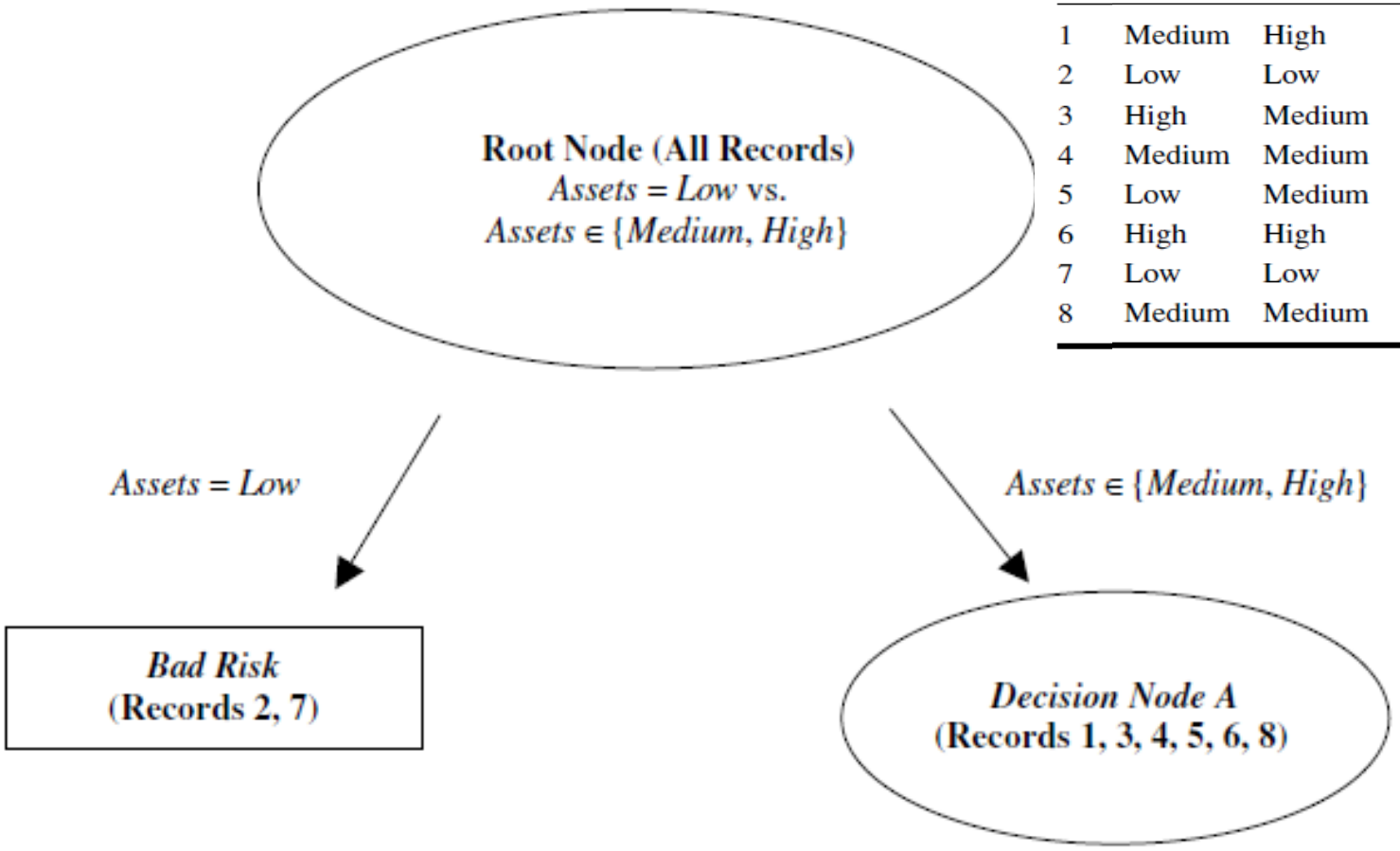
Split	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_L P_R$	$Q(s\|t)$	$\Phi(s\|t)$
1	0.375	0.625	G: .333 B: .667	G: .8 B: .2	0.46875	0.934	0.4378
2	0.375	0.625	G: 1 B: 0	G: 0.4 B: 0.6	0.46875	1.2	0.5625
3	0.25	0.75	G: 0.5 B: 0.5	G: 0.667 B: 0.333	0.375	0.334	0.1253
4	0.25	0.75	G: 0 B: 1	G: 0.833 B: 0.167	0.375	1.667	0.6248
5	0.5	0.5	G: 0.75 B: 0.25	G: 0.5 B: 0.5	0.5	0.5	0.25
6	0.25	0.75	G: 1 B: 0	G: 0.5 B: 0.5	0.375	1	0.375
7	0.375	0.625	G: 0.333 B: 0.667	G: 0.8 B: 0.2	0.46875	0.934	0.4378
8	0.625	0.375	G: 0.4 B: 0.6	G: 1 B: 0	0.46875	1.2	0.5625
9	0.875	0.125	G: 0.571 B: 0.429	G: 1 B: 0	0.21875	0.858	0.1877

Cand	Left Child Node, t_L	Right Child Node, t_R	Split	$\Phi(s\|t)$
1	$Savings = low$	$Savings \in \{medium, high\}$	1	0.4378
2	$Savings = medium$	$Savings \in \{low, high\}$	2	0.5625
3	$Savings = high$	$Savings \in \{low, medium\}$	3	0.1253
4	$Assets = low$	$Assets \in \{medium, high\}$	4	0.6248
5	$Assets = medium$	$Assets \in \{low, high\}$	5	0.25
6	$Assets = high$	$Assets \in \{low, medium\}$	6	0.375
7	$Income \leq \$25{,}000$	$Income > \$25{,}000$	7	0.4378
8	$Income \leq \$50{,}000$	$Income > \$50{,}000$	8	0.5625
9	$Income \leq \$75{,}000$	$Income > \$75{,}000$	9	0.1877

CART – Example

- ✓ The maximum $\Phi(s|t)$ =0.6248 is obtained in candidate split Assets=Low vs. Assets=Medium,High.

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good



CART decision tree after initial split

CART – Example

- ✓ Recompile a table of the candidate splits for Records 1,3,4,5,6,8

Cus	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
8	Medium	Medium	75	Good

Cand	Left Child Node, t_L	Right Child Node, t_R
1	$Savings = low$	$Savings \in \{medium, high\}$
2	$Savings = medium$	$Savings \in \{low, high\}$
3	$Savings = high$	$Savings \in \{low, medium\}$
5	$Assets = medium$	$Assets \in \{low, high\}$
6	$Assets = high$	$Assets \in \{low, medium\}$
7	$Income \leq \$25{,}000$	$Income > \$25{,}000$
8	$Income \leq \$50{,}000$	$Income > \$50{,}000$
9	$Income \leq \$75{,}000$	$Income > \$75{,}000$

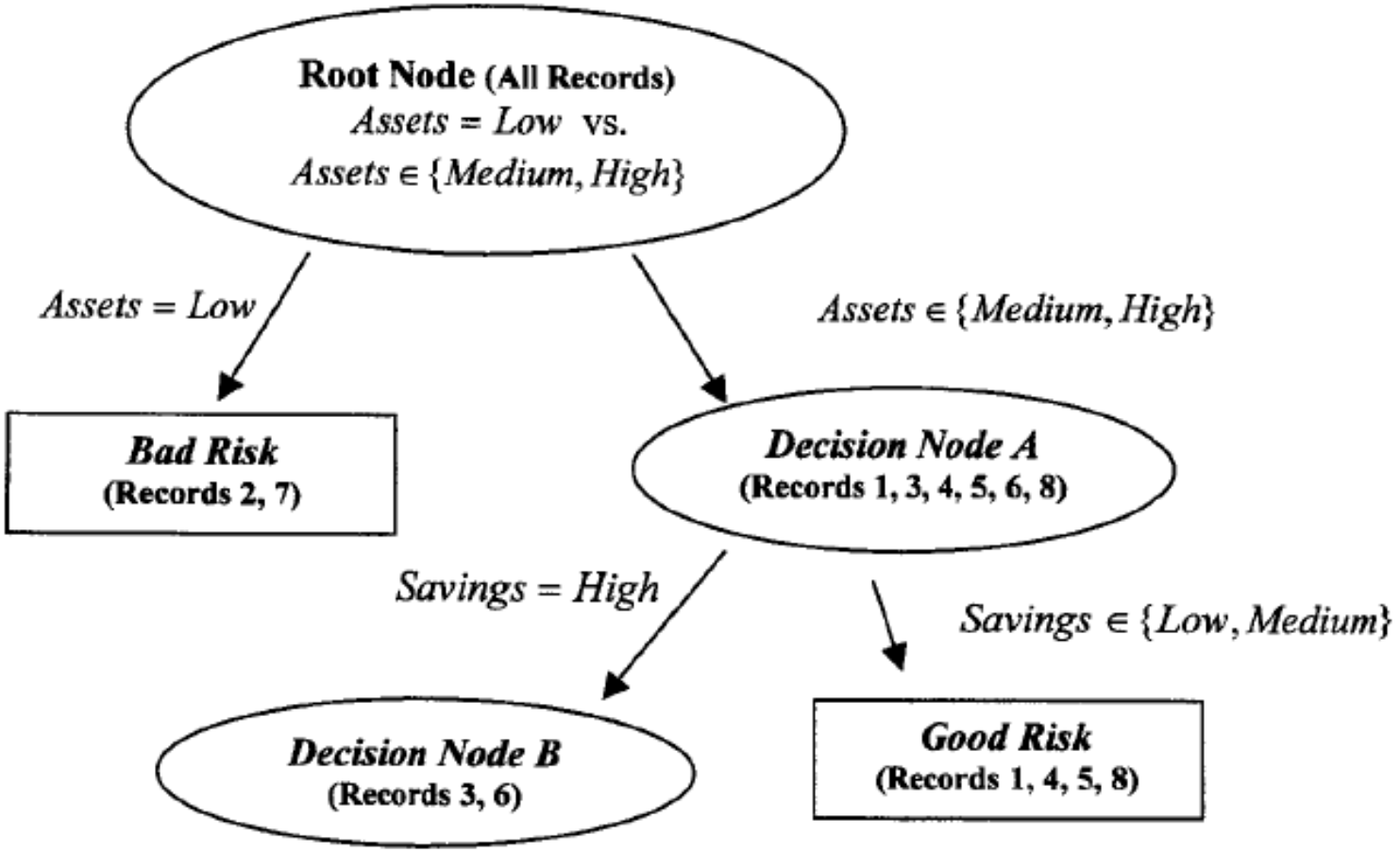
Cand	Left Child Node, t_L	Right Child Node, t_R
1	*Savings* = *low*	*Savings* ∈ {*medium, high*}
2	*Savings* = *medium*	*Savings* ∈ {*low, high*}
3	*Savings* = *high*	*Savings* ∈ {*low, medium*}
5	*Assets* = *medium*	*Assets* = *high*
6	*Assets* = *high*	*Assets* = *medium*
7	*Income* ≤ $25,000	*Income* > $25,000
8	*Income* ≤ $50,000	*Income* > $50,000
9	*Income* ≤ $75,000	*Income* > $75,000

Cus	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
8	Medium	Medium	75	Good

Split	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_L P_R$	$Q(s\|t)$	$\Phi(s\|t)$
1	0.167	0.833	G: 1 B: 0	G: .8 B: .2	0.2782	0.4	0.1112
2	0.5	0.5	G: 1 B: 0	G: 0.667 B: 0.333	0.5	0.6666	0.3333
3	0.333	0.667	G: 0.5 B: 0.5	G: 1 B: 0	0.4444	1	0.4444
5	0.667	0.333	G: 0.75 B: 0.25	G: 1 B: 0	0.4444	0.5	0.2222
6	0.333	0.667	G: 1 B: 0	G: 0.75 B: 0.25	0.4444	0.5	0.2222
7	0.333	0.667	G: 0.5 B: 0.5	G: 1 B: 0	0.4444	1	0.4444
8	0.5	0.5	G: 0.667 B: 0.333	G: 1 B: 0	0.5	0.6666	0.3333
9	0.167	0.833	G: 0.8 B: 0.2	G: 1 B: 0	0.2782	0.4	0.1112

CART – Example

- ✓ Here two candidate splits (3 and 7) share the highest value for $\Phi(s|t)$,0.4444. (Savings=high, Income<25000)



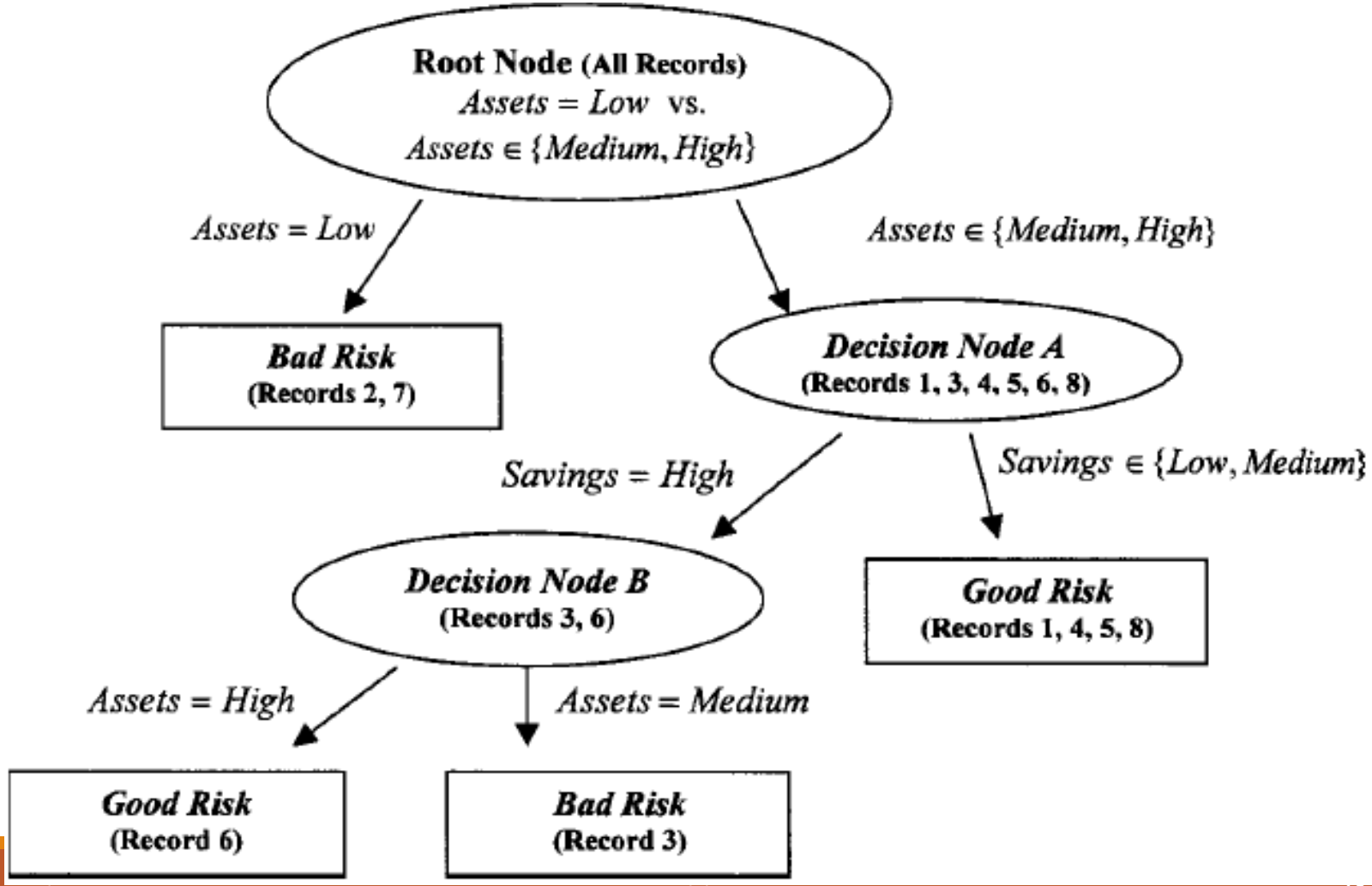
CART decision tree after decision node A split

Cand	Left Child Node, t_L	Right Child Node, t_R
1	*Savings* = *low*	*Savings* ∈ {*medium, high*}
2	*Savings* = *medium*	*Savings* ∈ {*low, high*}
5	*Assets* = *medium*	*Assets* ∈ {*low, high*}
6	*Assets* = *high*	*Assets* ∈ {*low, medium*}
7	*Income* ≤ $25,000	*Income* > $25,000
8	*Income* ≤ $50,000	*Income* > $50,000
9	*Income* ≤ $75,000	*Income* > $75,000

Cus	Savings	Assets	Income ($1000s)	Credit Ri
3	High	Medium	25	Bad
6	High	High	25	Good

CART – Example

- ✓ The complete CART



CART

$$\Phi(s|t) = 2P_L P_R \sum_{j=1}^{\#\text{ classes}} |P(j|t_L) - P(j|t_R)|$$

- ✓ When is $\Phi(s|t)$ large?
- ✓ $\Phi(s|t)$ is large when both of its main components are large: $2P_L P_R$ *and*

$$Q(s|t) = \sum_{j=1}^{\#classes} |P(j|t_L) - P(j|t_R)|$$

- ✓ When is the component $Q(s|t)$ large?
- ✓ When is the component $2P_L P_R$ large?

CART – Example

- ✓ When is the component $2P_L P_R$ large?
- ✓ When P_L and P_R are large,
- ✓ when the proportions of records in the left and right child nodes are equal.
- ✓ The theoretical maximum for $2P_LP_R$ is 2x(0.5)x(0.5) = 0.5.
- ✓ When is the component *Q(s|t)* large?
- ✓ when the distance between $P(j|t_L)$ and $P(j|t_R)$ is maximized across each class.
- ✓ The maximum value would occur when for each class the child nodes are completely uniform (pure).
- ✓ The theoretical maximum value for Q(s|t) is 2 for this component.

$$Q(s|t) = \sum_{j=1}^{\#classes} \left| P(j|t_L) - P(j|t_R) \right|$$

CART

- ✓ Gini Index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444

C4.5 Algorithm

- ✓ Quinlan's extension of his own ID3 algorithm (Quinlan,1992).
- ✓ Multi-way split, (not a binary tree).
- ✓ Uses "***Information gain***" or "***entropy reduction***" to compute impurity to select the optimal split.
- ✓ Let X is an attribute with *k* possible values of probabilities *p1, p2,..., pk* .
- ✓ ***Entropy*** is the smallest number of bits, on average per symbol, needed to transmit a stream of symbols representing the values of X observed.

$$Entropy = H(x) = -\sum_{j} p_j \log_2(p_j)$$

C4.5 Algorithm – Entropy

- ✓ Entropy is a measure of randomness, a measure of the impurity in a collection of training examples.
- ✓ Entropy is a non-negative value.

$$Entropy = H(x) = -\sum_j p_j \log_2(p_j)$$

- ✓ When is entropy minimum?

$$H_{min} = -\sum_j 1 \log_2(1) = 0$$

- ✓ When is entropy maximum?

$$H_{max} = -\sum_{j=1}^{n} \frac{1}{n} \log_2\left(\frac{1}{n}\right) = -\frac{1}{n} n \log_2\left(\frac{1}{n}\right) = -\log_2\left(\frac{1}{n}\right)$$

C4.5 Algorithm – Entropy

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

C4.5 Algorithm – Information gain

- ***Information gain*** is a measure of the effectiveness of an attribute in classifying the training data and measures the expected reduction in entropy by partitioning the examples according to an attribute.

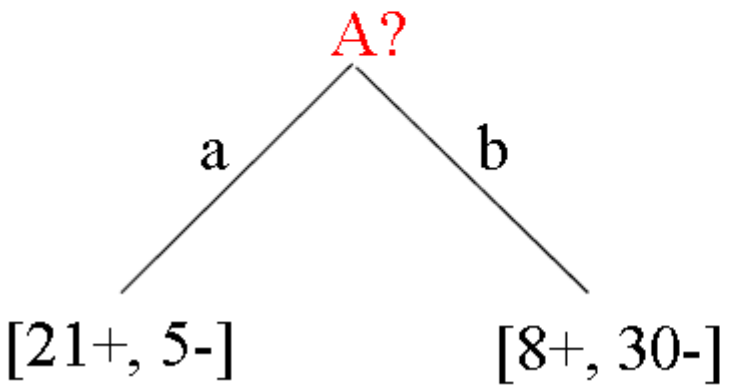
$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} (|S_v| / |S|) \text{Entropy}(S_v)$$

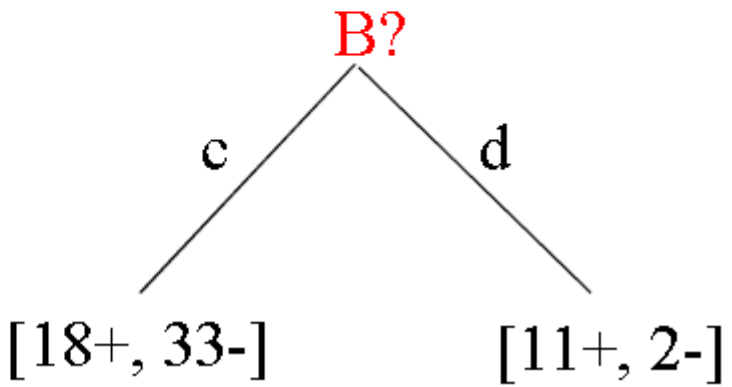
- A – an attribute
- Values(A) – possible values of attribute A
- Sv – the subset of S for which attribute A has value v

C4.5 Algorithm – Information gain

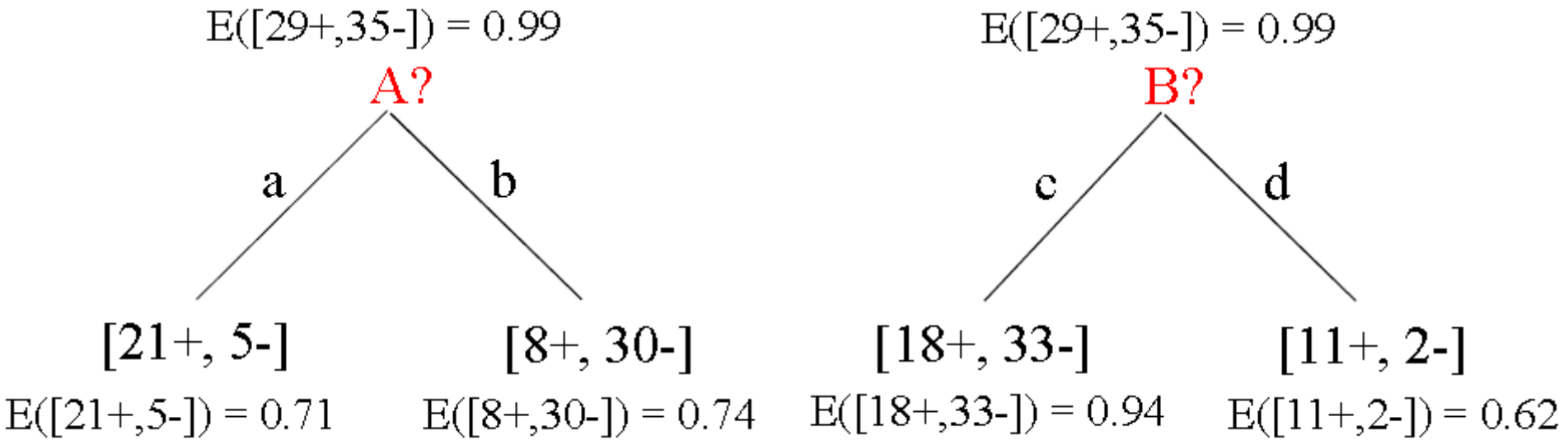
- ✓ Which attribute is the best classifier?

- S: [29+,35-] Attributes: A and B
- possible values for A: a,b possible values for B: c,d





C4.5 Algorithm – Information gain



$$Gain(S,A) = Ent(S) - \frac{26}{64} Ent([21+,5-]) - \frac{38}{64} Ent([8+,30-]) = 0.99 - \frac{26}{64} 0.71 - \frac{38}{64} 0.74 = 0.27$$

$$Gain(S,B) = Ent(S) - \frac{51}{64} Ent([18+,33-]) - \frac{13}{64} Ent([11+,2-]) = 0.99 - \frac{51}{64} 0.94 - \frac{13}{64} 0.62 = 0.12$$

C4.5 Algorithm – Information gain

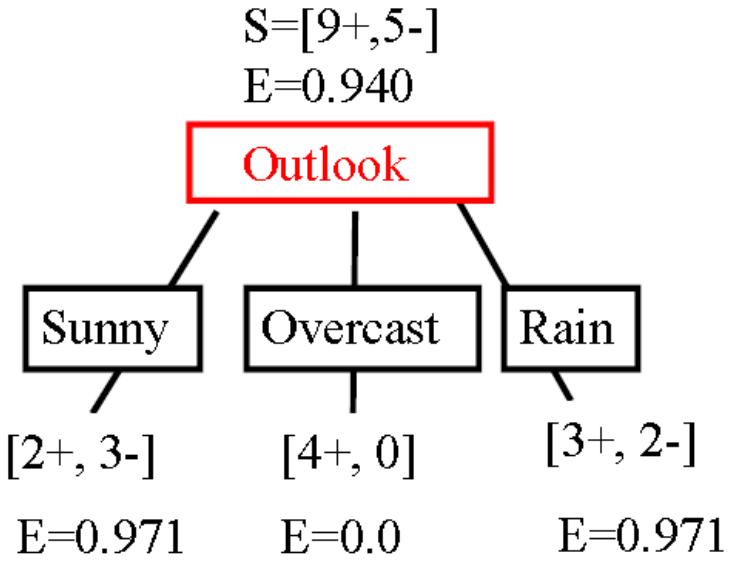
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

C4.5 Algorithm – Informa

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- ✓ $E(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

  $= 0.940$



$$Gain(S, Outlook) = 0.940 - \frac{5}{14}0.971 - \frac{4}{14}0 - \frac{5}{14}0.971$$

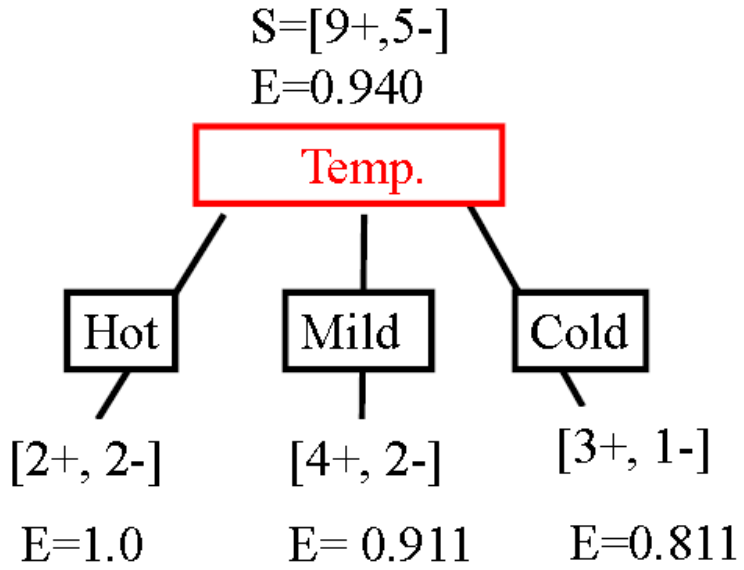
$$Gain(S, Outlook) = 0.247$$

C4.5 Algorithm – Informa

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- ✓ $E(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

  $= 0.940$



$$Gain(S, Temp) = 0.940 - \frac{4}{14}1 - \frac{6}{14}0.911 - \frac{4}{14}0.811$$

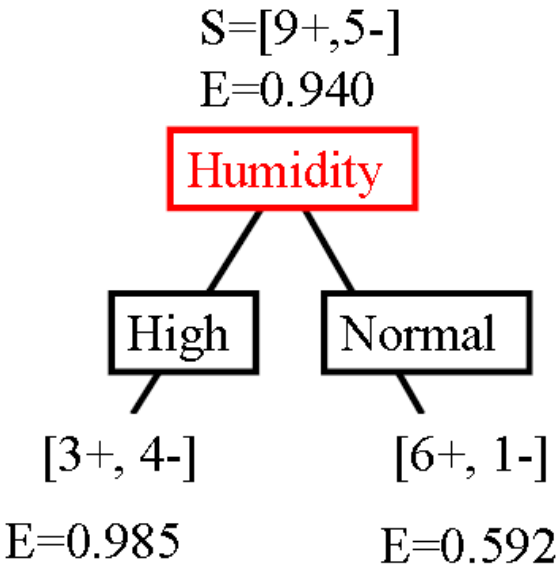
$$Gain(S, Temp) = 0.029$$

C4.5 Algorithm – Informa

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- ✓ $E(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

  $= 0.940$



$$Gain(S, Humidity) = 0.940 - \frac{7}{14}0.985 - \frac{7}{14}0.592$$

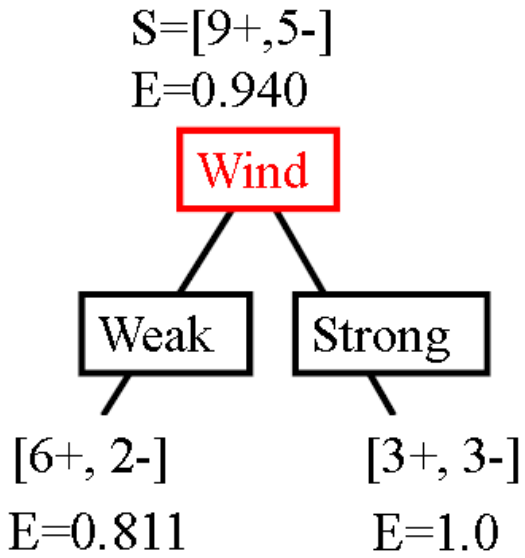
$$Gain(S, Humidity) = 0.151$$

C4.5 Algorithm – Informa

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- ✓ $E(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

  $= 0.940$



$$Gain(S, Wind) = 0.940 - \frac{8}{14}0.811 - \frac{6}{14}1$$

$$Gain(S, Wind) = 0.048$$

C4.5 Algorithm – Informa

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- ✓ $E(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

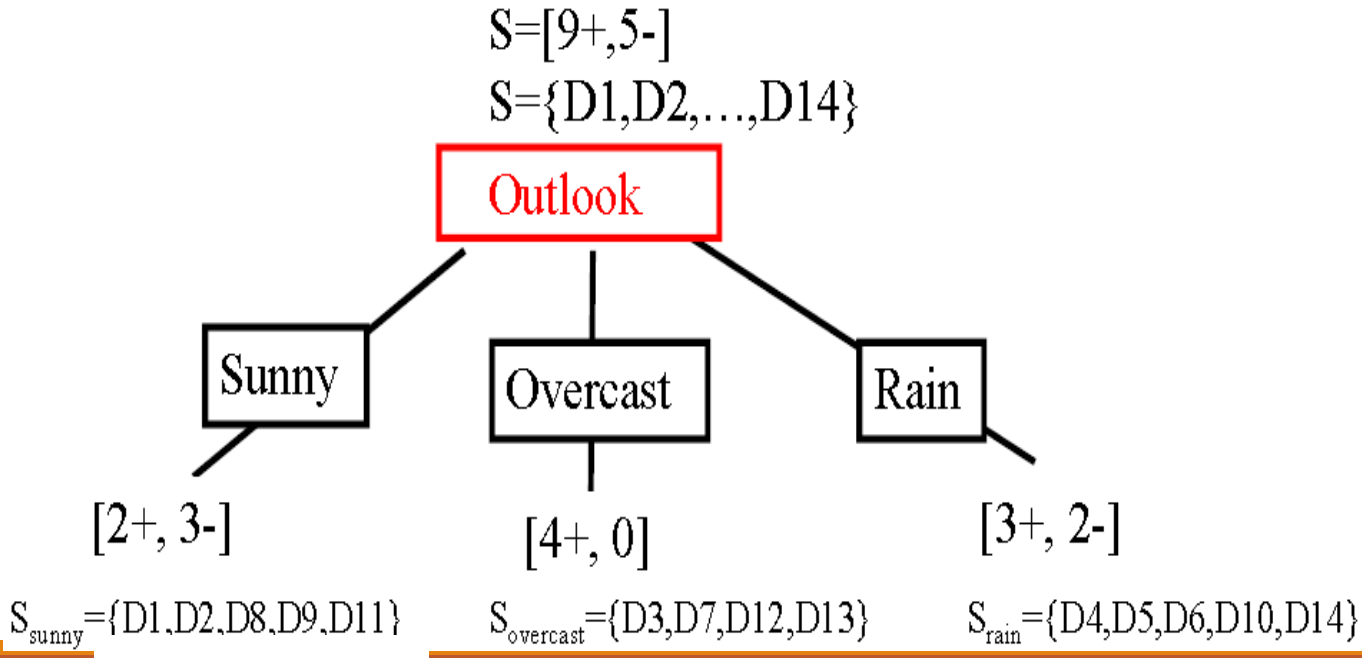
  $= 0.940$

$$Gain(S, Outlook) = 0.247$$

$$Gain(S, Temp) = 0.029$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$



C4.5 – Example

Cus	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

Candidate Splits at Root Node for C4.5 Algorithm

Cand	Child Nodes		
1	*Savings* = *low*	*Savings* = *medium*	*Savings* = *high*
2	*Assets* = *low*	*Assets* = *medium*	*Assets* = *high*
3	*Income* ≤ $25,000	*Income* > $25,000	
4	*Income* ≤ $50,000	*Income* > $50,000	
5	*Income* ≤ $75,000	*Income* > $75,000	

C4.5 – Example

- ✓ 5 Good – 3 Bad Credit Risk

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

$$E(S) = -\sum_j p_j \log_2(p_j) = -\frac{5}{8}\log_2\left(\frac{5}{8}\right) - \frac{3}{8}\log_2\left(\frac{3}{8}\right) = 0.9544$$

$$P_{Savings} \Rightarrow P_{high} = \frac{2}{8} \qquad P_{medium} = \frac{3}{8} \qquad P_{low} = \frac{3}{8}$$

$$H_{savings}(high) = -\sum_j p_j \log_2(p_j) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$H_{savings}(medium) = -\sum_j p_j \log_2(p_j) = -\frac{3}{3}\log_2\left(\frac{3}{3}\right) - \frac{0}{3}\log_2\left(\frac{0}{3}\right) = 0$$

$$H_{savings}(low) = -\sum_j p_j \log_2(p_j) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.9183$$

C4.5 – Example

- ✓ 5 Good – 3 Bad Credit Risk

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

$$E(S) = -\sum_j p_j \log_2(p_j) = -\frac{5}{8}\log_2\left(\frac{5}{8}\right) - \frac{3}{8}\log_2\left(\frac{3}{8}\right) = 0.9544$$

$$P_{Savings} \Rightarrow P_{high} = \frac{2}{8} \qquad P_{medium} = \frac{3}{8} \qquad P_{low} = \frac{3}{8}$$

$$H_{savings}(high) = 1$$

$$H_{savings}(medium) = 0$$

$$H_{savings}(low) = 0.9183$$

$$H_{Savings}(S) = \sum_{i=1}^{k} P_i H_{Savings}(S_i)$$

C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

$$Gain(S, Savings) = E(S) - H_{Savings}(S)$$

$$= 0.9544 - 0.5944 = 0.36$$

C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

- For Assets

$$P_{high}=\frac{2}{8} \qquad P_{medium}=\frac{4}{8} \qquad P_{low}=\frac{2}{8}$$

$$H_{assets}(high)=-\frac{2}{2}\log_2\left(\frac{2}{2}\right)-\frac{0}{2}\log_2\left(\frac{0}{2}\right)=0$$

$$H_{assets}(medium)=-\frac{3}{4}\log_2\left(\frac{3}{4}\right)-\frac{1}{4}\log_2\left(\frac{1}{4}\right)=0.8113$$

$$H_{assets}(low)=-\frac{0}{2}\log_2\left(\frac{0}{2}\right)-\frac{2}{2}\log_2\left(\frac{2}{2}\right)=0$$

$$H_{Assets}(S)=\left(\frac{2}{8}\times 0\right)+\left(\frac{4}{8}\times 0.8113\right)+\left(\frac{2}{8}\times 0\right)=0.4057$$

$$Gain(S,Assets)=H(S)-H_{Assets}(S)=0.9544-0.4057=0.5487\ bits$$

C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

- For Income<=25000

$$P_{income\leq 25K} = \frac{3}{8} \quad P_{income>25K} = \frac{5}{8}$$

$$H_{income\leq 25K}(income \leq 25K) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.9183$$

$$H_{income\leq 25K}(income > 25K) = -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right) = 0.7219$$

$$H_{income\leq 25K}(S) = \left(\frac{3}{8}\times 0.9183\right) + \left(\frac{5}{8}\times 0.7219\right) = 0.7956$$

$$Gain(income \leq 25K) = H(S) - H_{income\leq 25K}(S) = 0.9544 - 0.7956 = 0.1588\ bits$$

C4.5 – Example

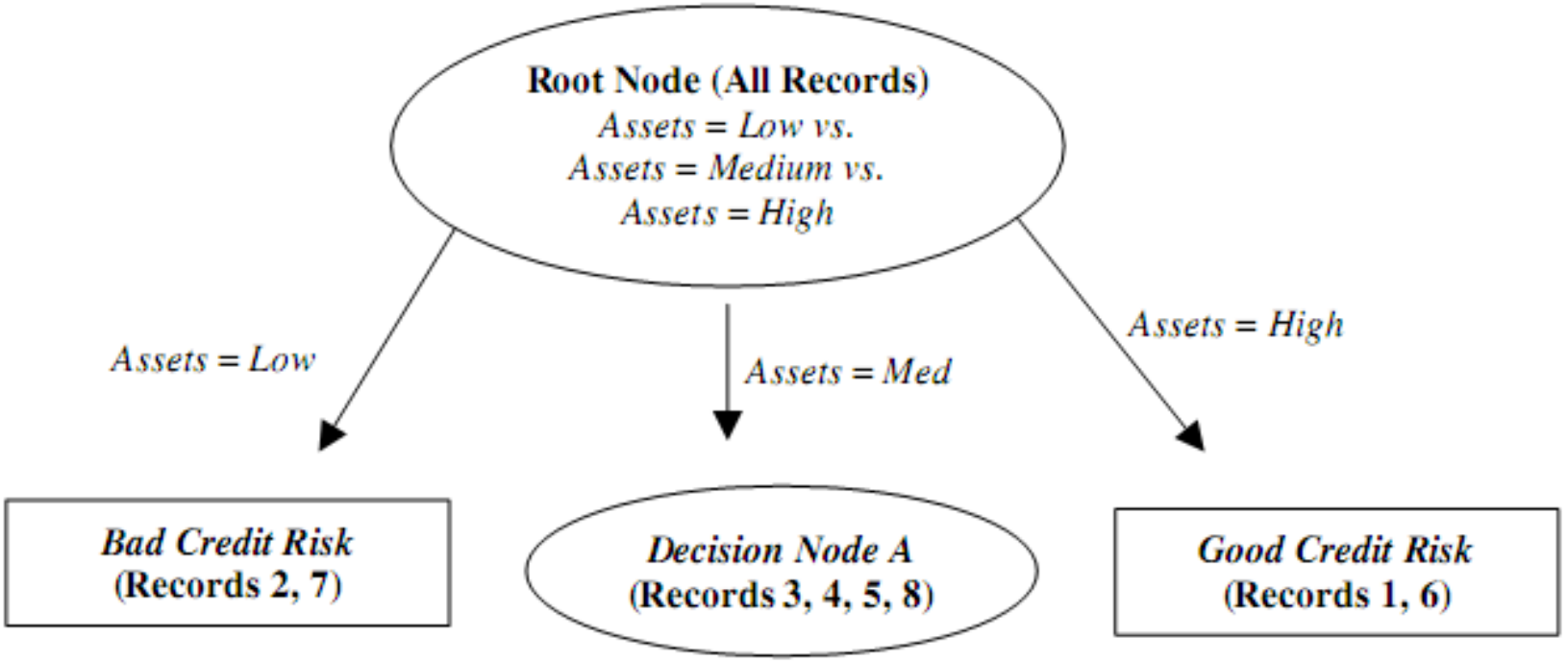
Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

Information Gain for Each Candidate Split at the Root Node

Candidate Split	Child Nodes	Information Gain (Entropy Reduction)
1	*Savings* = *low* *Savings* = *medium* *Savings* = *high*	0.36 bits
2	*Assets* = *low* *Assets* = *medium* *Assets* = *high*	0.5487 bits
3	*Income* ≤ $25,000 *Income* > $25,000	0.1588 bits
4	*Income* ≤ $50,000 *Income* > $50,000	0.3475 bits
5	*Income* ≤ $75,000 *Income* > $75,000	0.0923 bits

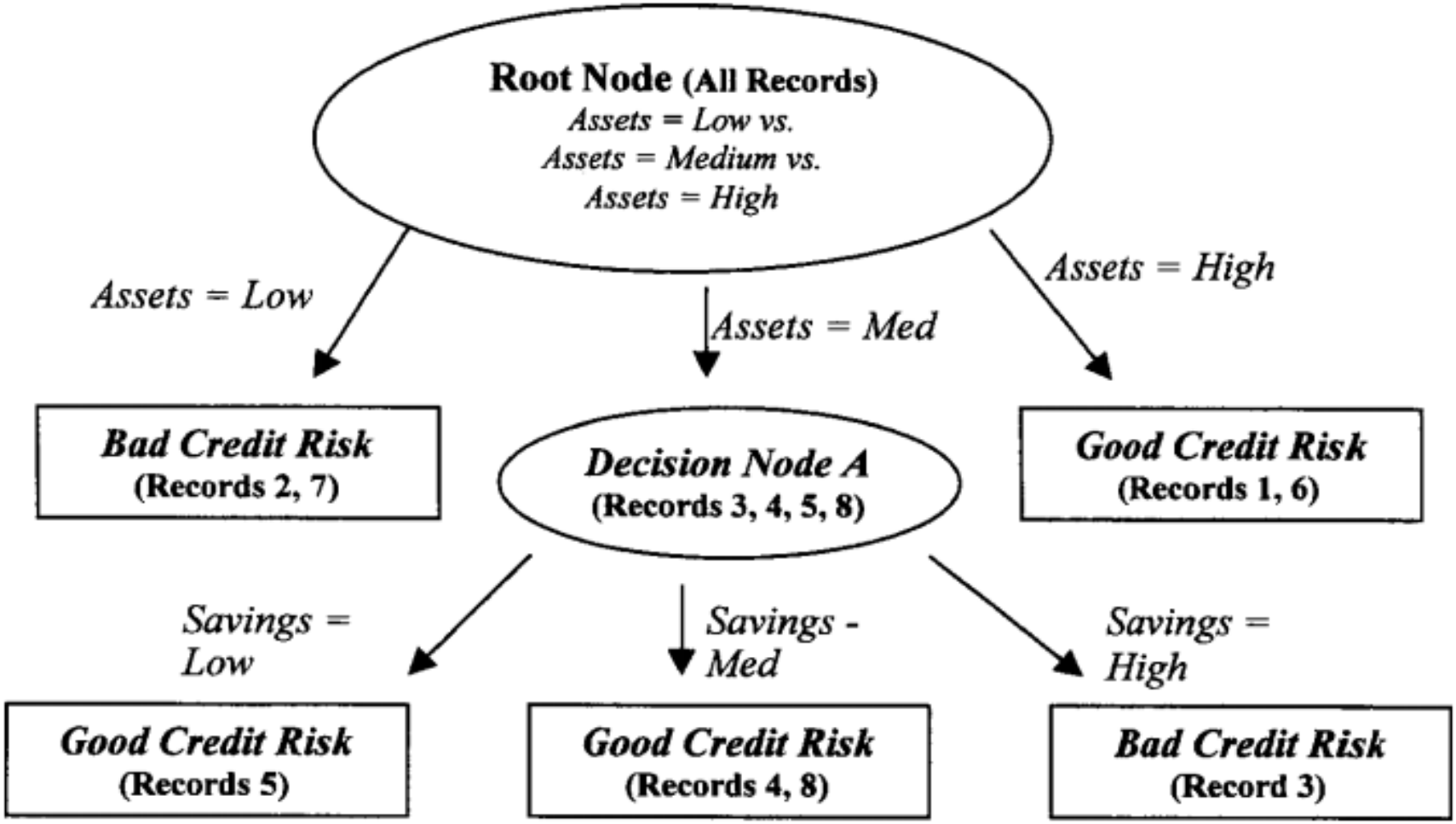
C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good



C4.5 – Example

Cust	Savings	Assets	Income ($1000s)	Credit Ri
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good



Differences between CART and C4.5

- ✓ Unlike CART, the C4.5 algorithm is not restricted to binary splits.
- ✓ For categorical attributes, C4.5 produces a separate branch for each value of the categorical attribute.
  - This may result in more “bushiness” than desired, since some values may have low frequency.
- ✓ C4.5 method for measuring node homogeneity is different from the CART.

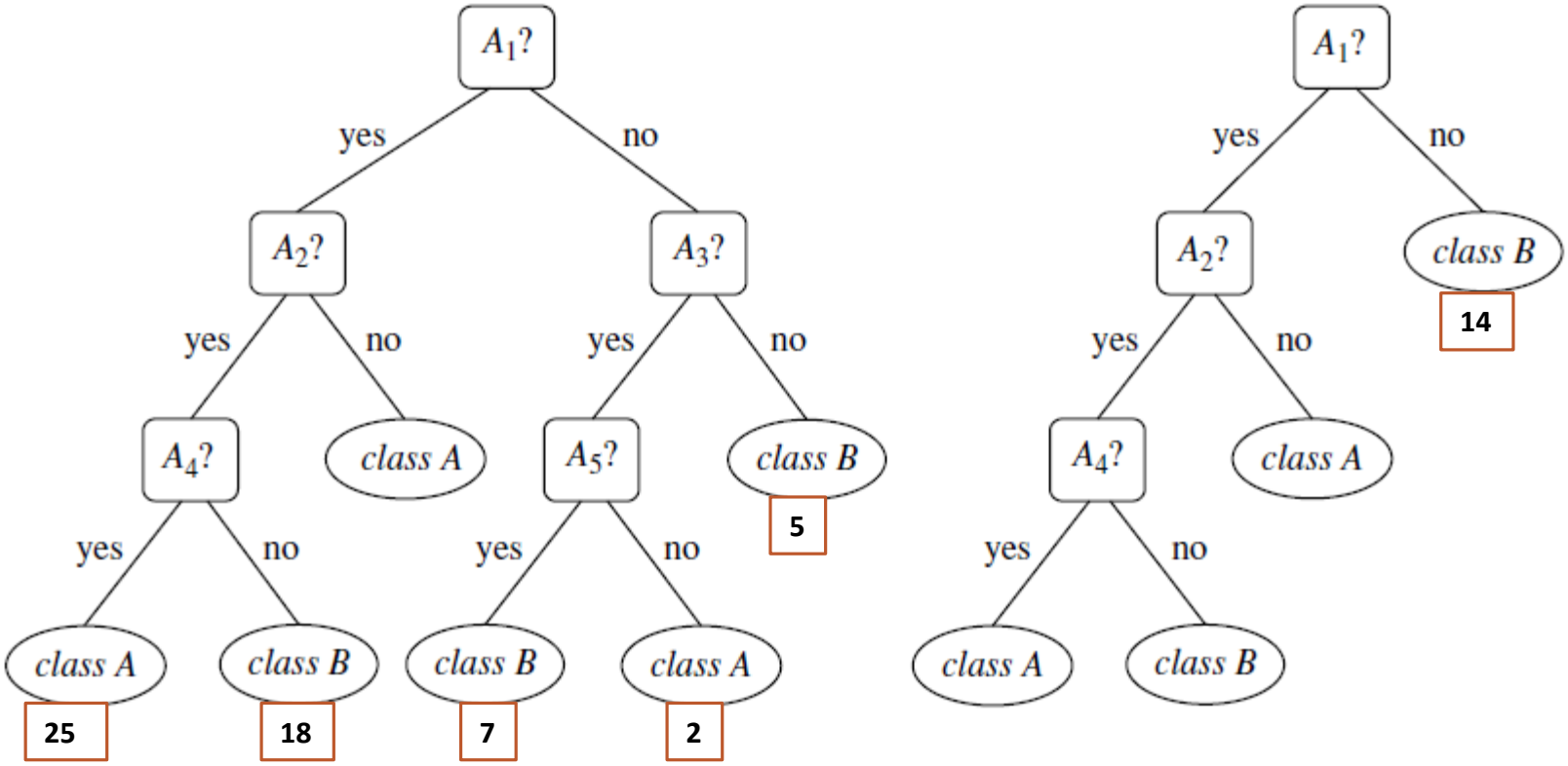
Decision Trees

- ✓ Stopping Rules
  - Pure node
  - Tree depth (levels of tree)
  - Global purity
  - No statistically significant association between any attribute and the class
  - Number of records in a node
  - Percentage of records in a node

Decision Trees

- ✓ Pruning
  - removing bottom-level splits from fully grown tree that do not contribute significantly to the accuracy of the tree.
  - simplify the tree,
  - making it easier to interpret ,
  - improving generalization ,
  - avoid over-fitting.

Decision Trees - Pruning



An unpruned decision tree and a pruned version of it.

Other Tree Examples

- ✓ CHAID
  - Chi-squared Automatic Interaction Detector
- ✓ QUEST
  - Quick, Unbiased, Efficient, Statistical Tree