50.007 Machine Learning

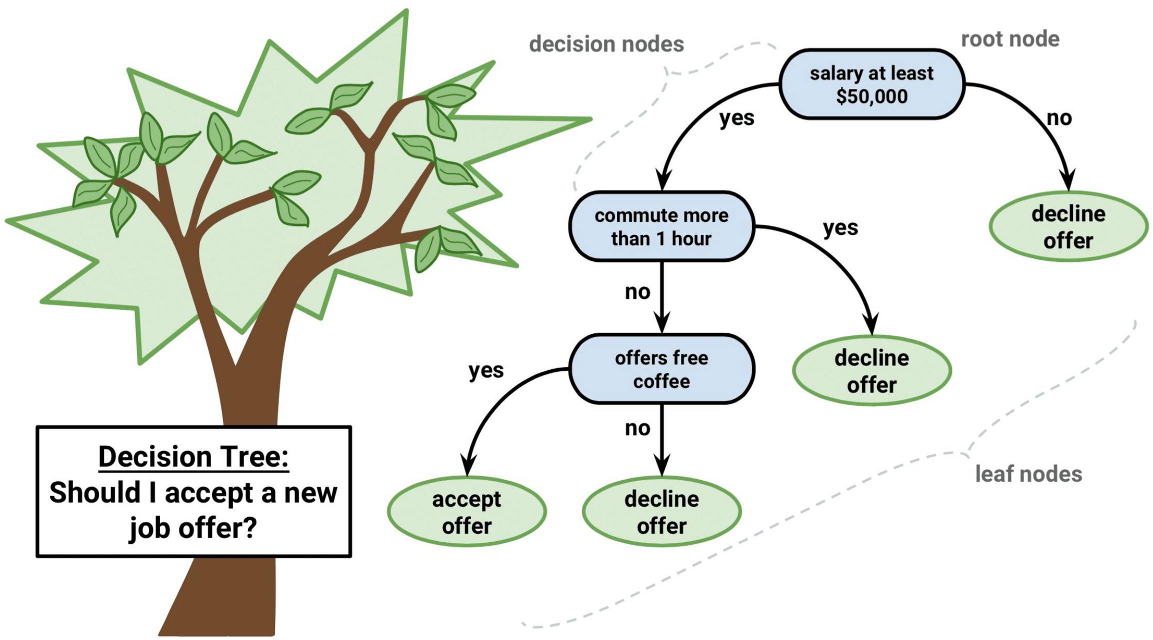
Decision Tree

Roy Ka-Wei Lee
Assistant Professor, DAI/ISTD, SUTD

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Decision Tree

- Decision tree builds classification or regression models in the form of a tree structure.
- Breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- Final result is a tree with **decision nodes** and **leaf nodes**.



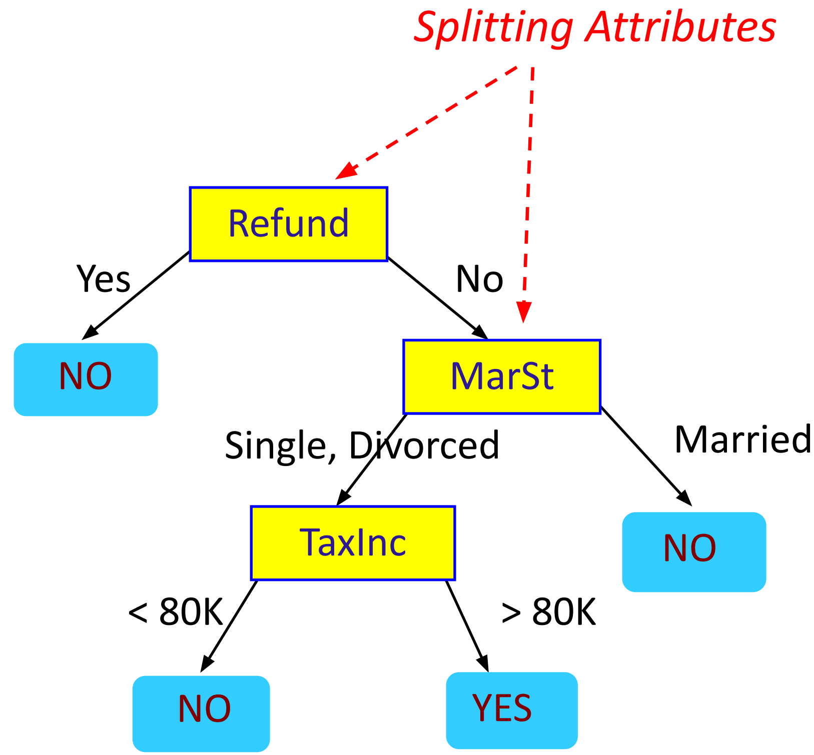
Source: http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

Example of Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
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

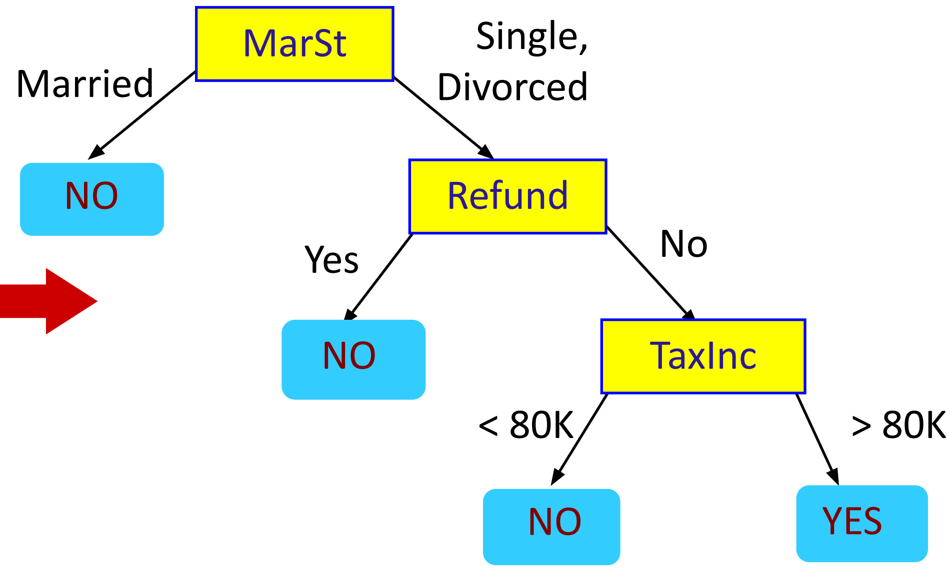


Model: Decision Tree

Example of Decision Tree

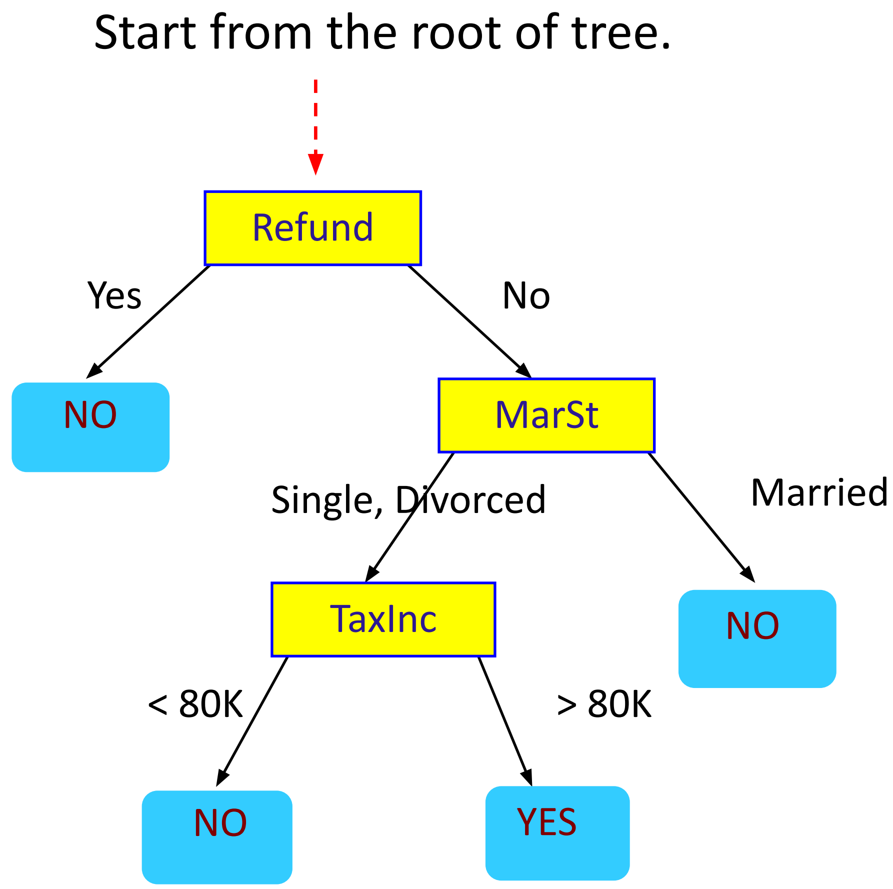
Tid (categorical)	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10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

Training Data



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

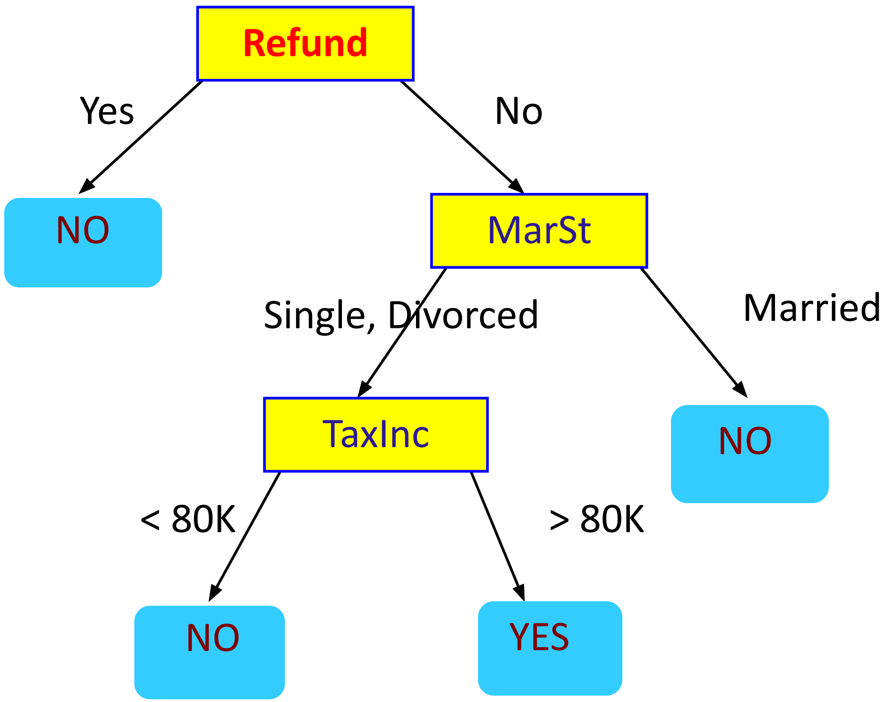
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



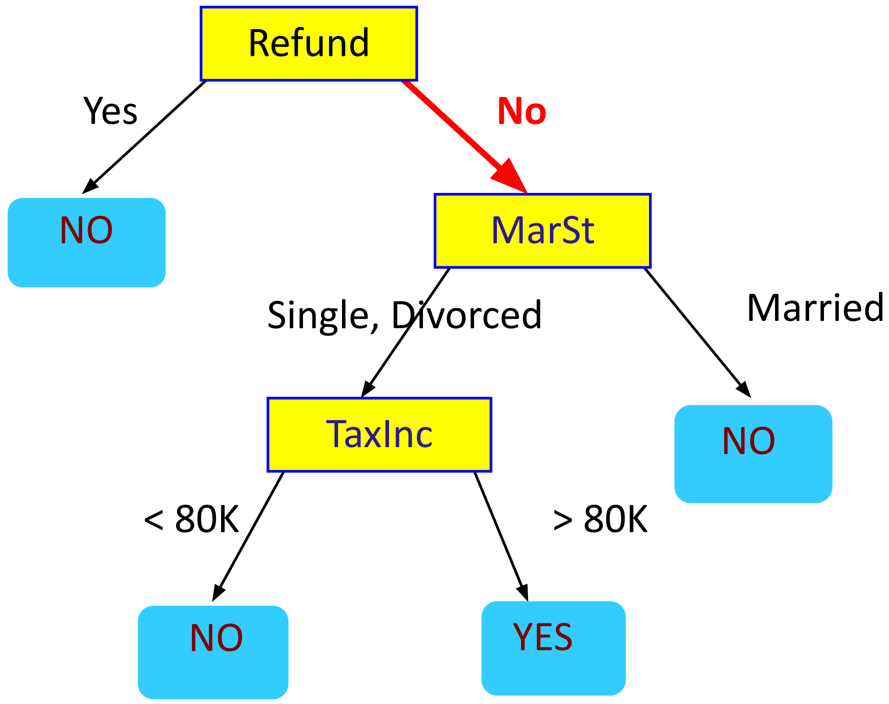
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

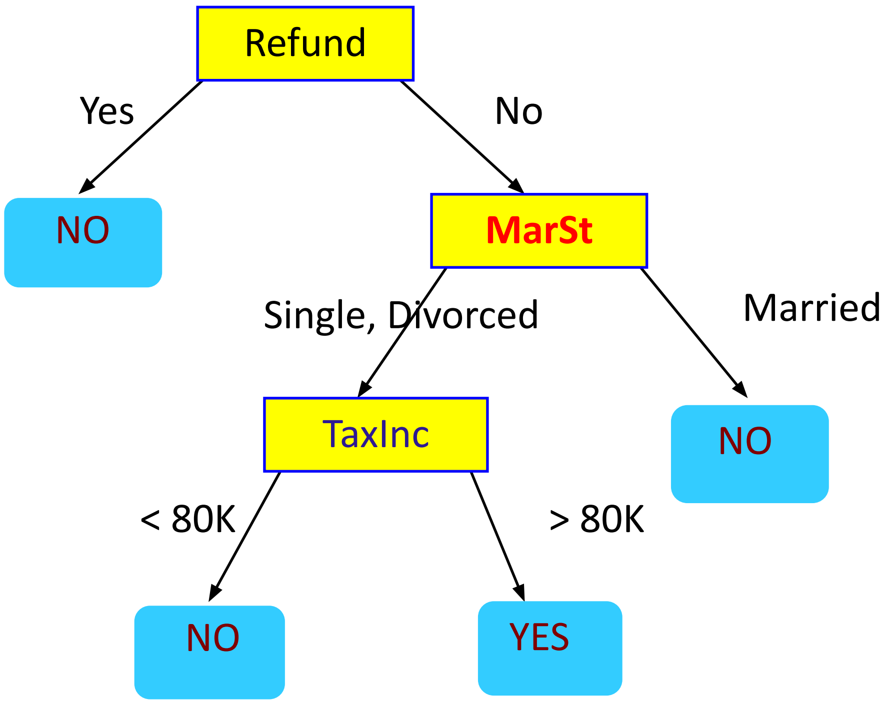
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



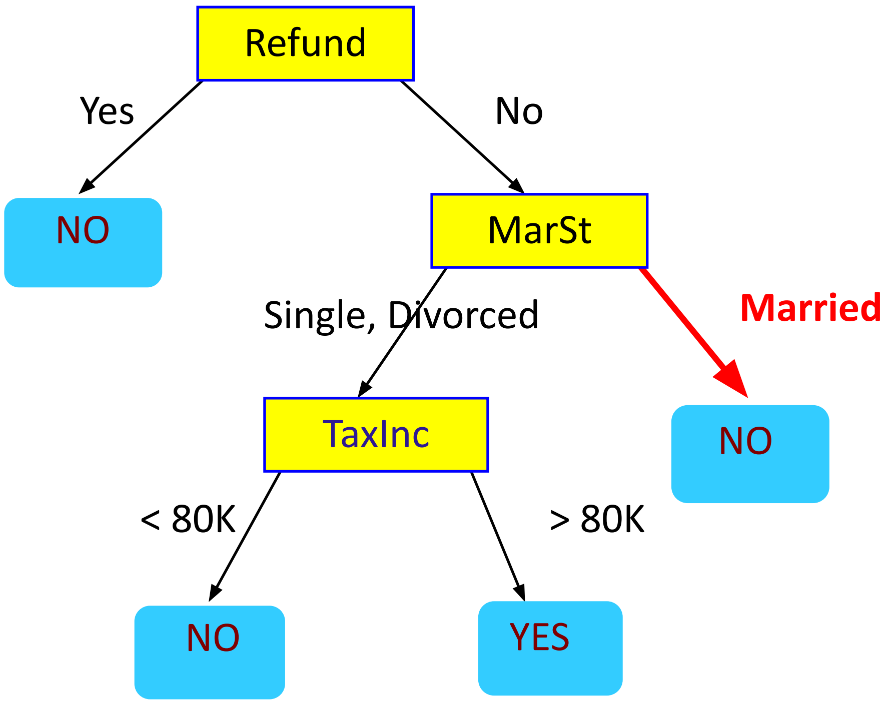
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

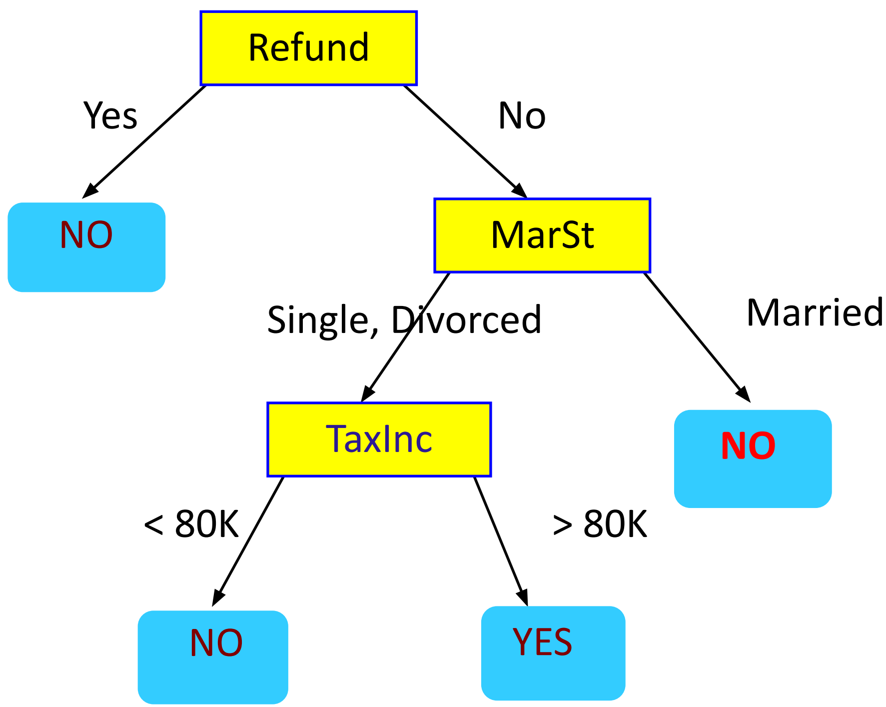


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Apply Model to Test Data



Test Data

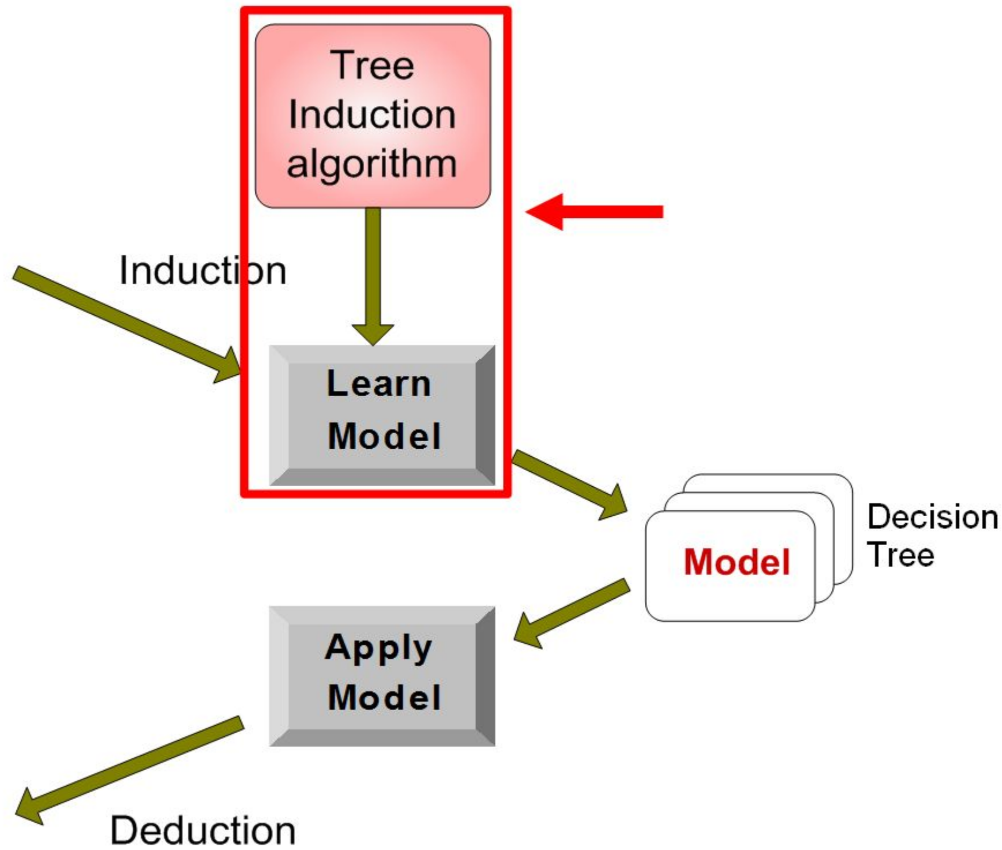
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to "**No**"

Decision Tree 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

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Growing a Tree

1. Features to choose

2. Conditions for splitting

3. Knowing when to stop

4. Pruning

Decision Tree Induction

- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ, SPRINT

Hunt Algorithm

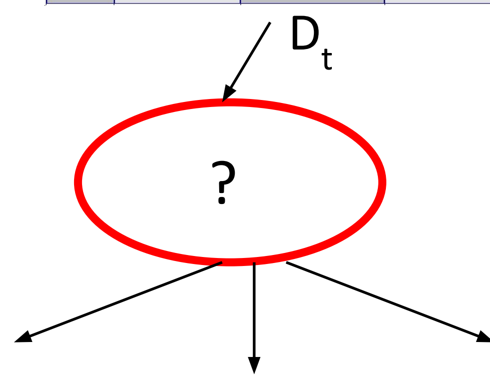
- A decision tree is grown in a recursive fashion by partitioning the training records successively into purer subset
- It is the basis of many existing decision tree induction algorithms

General Structure of Hunt's Algorithm

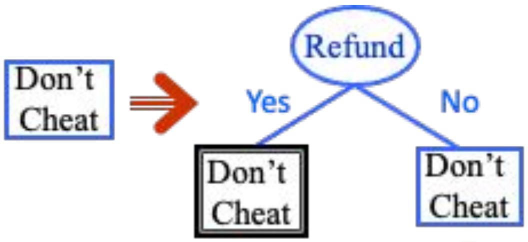
- Let D_t be the set of training records that reach a node **t**
- General Procedure:
  - If D_t contains records that belong the same class y_t, then **t** is a **leaf** node labeled as y_t
  - If D_t is an empty set, then t is a **leaf** node labeled by the default class y_d
  - If D_t contains records that belong to more than one class, use an **attribute test** to **split** the data into smaller subsets.

  Recursively apply the above procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
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



Hunt's Algorithm

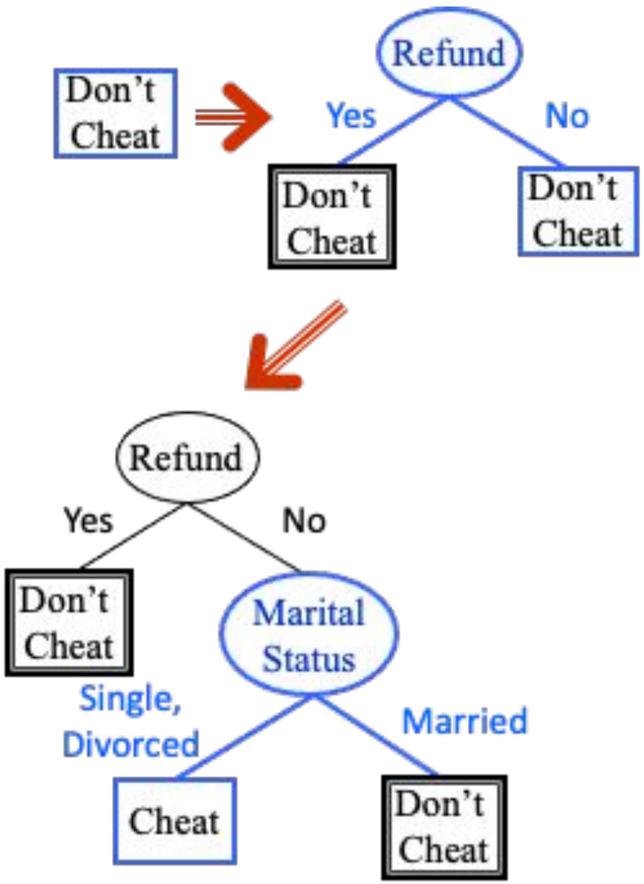


Purely contain label of "No"

Mix of "Yes" and "No" label. So we reclusively split

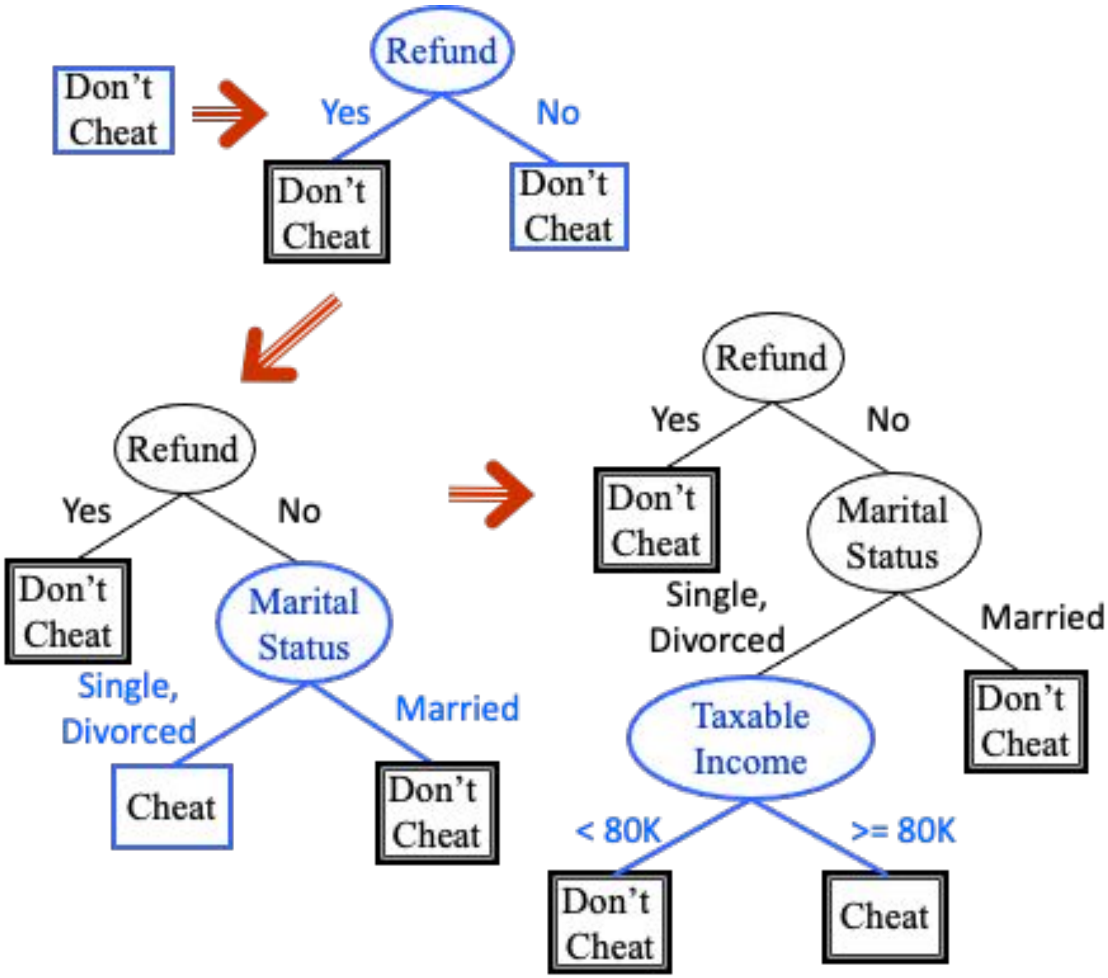
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm



Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
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

Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Cheat
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

Tree Induction

- Greedy strategy.
  - Split the records based on **an attribute test** that optimizes certain criterion (split such that we get most homogenous leaf node)

- 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**

Tree Induction

- Greedy strategy.
  - Split the records based on **an attribute test** that optimizes certain criterion (split such that we get most homogenous leaf node)

- 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**

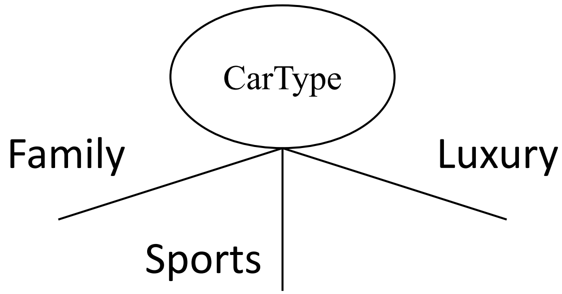
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

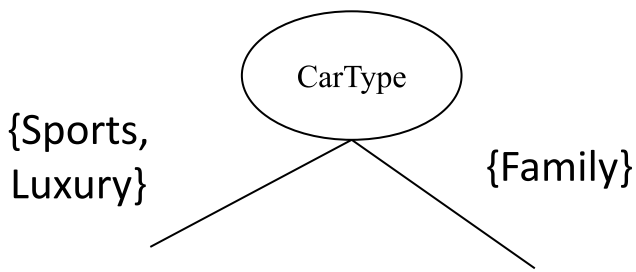
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

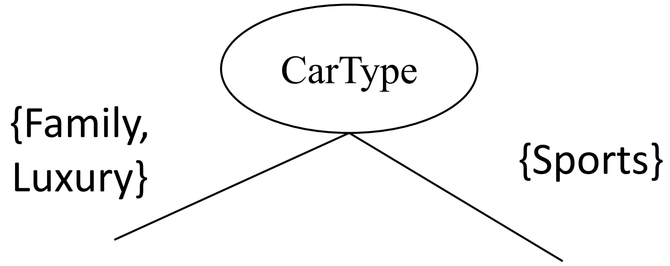
Splitting Based on Nominal Attributes

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



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

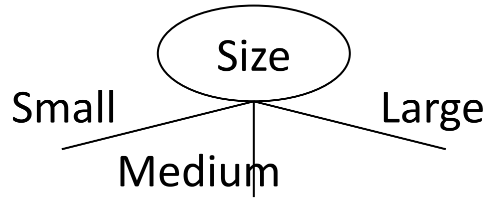




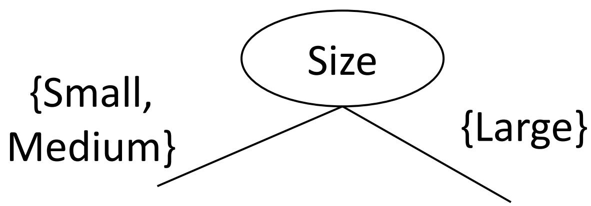
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

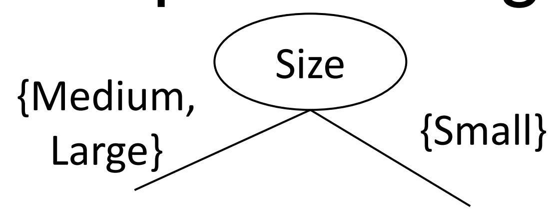
Splitting Based on Ordinal Attributes

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



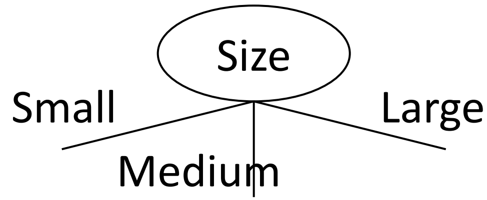
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



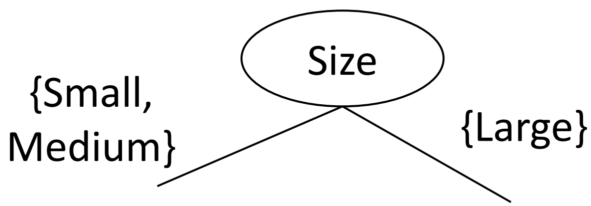


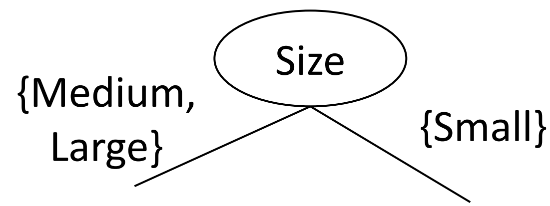
Splitting Based on Ordinal Attributes

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

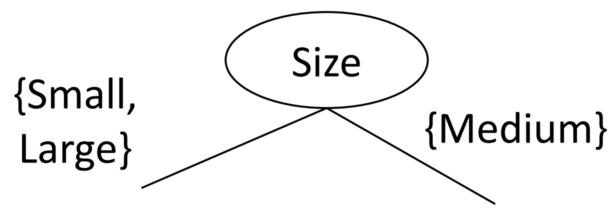


- Binary split: Divides values into two subsets. Need to find optimal partitioning.





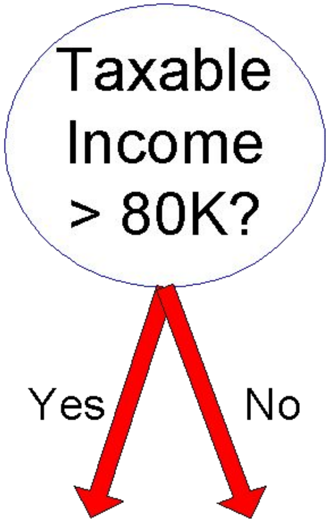
- What about this split?



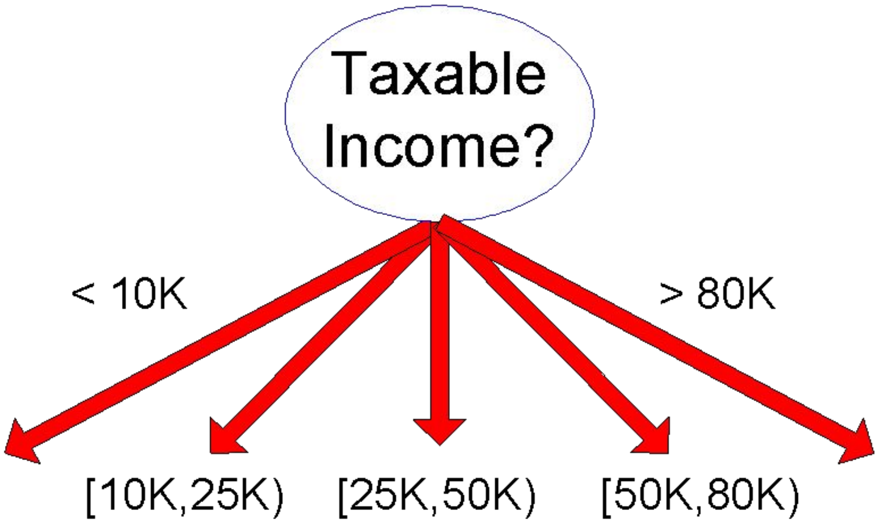
Splitting Based on Continuous Attributes

- 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 bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more computationally intensive

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

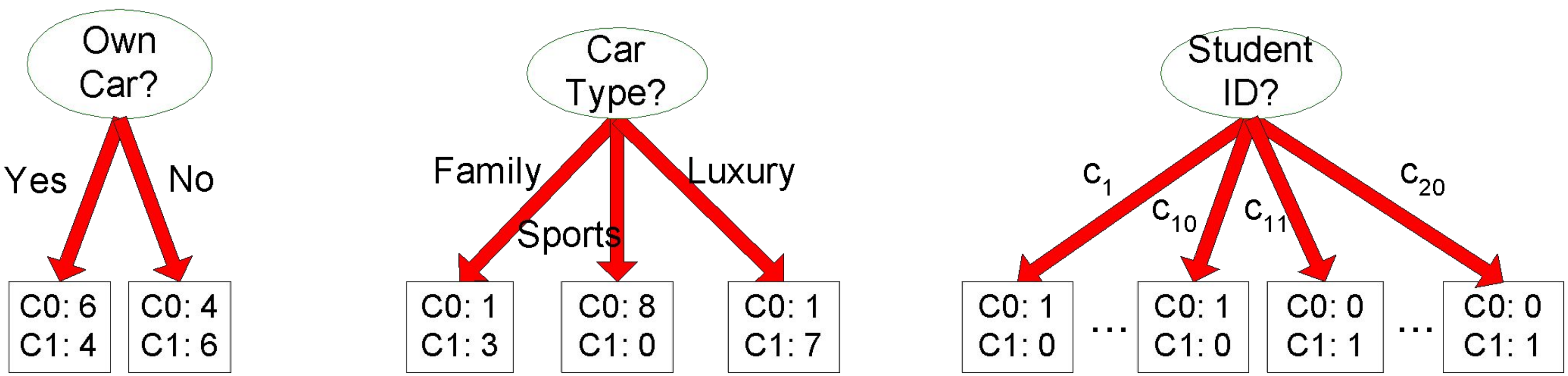
Tree Induction

- Greedy strategy.
  - Split the records based on **an attribute test** that optimizes certain criterion (split such that we get most homogenous leaf node)

- 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 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

Non-homogeneous,

High degree of impurity

C0: 9
C1: 1

Homogeneous,

Low degree of impurity

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

Measures of Impurity: GINI

- Gini Index for a given node t :

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

(NOTE: *p(j | t)* is the relative frequency of class j at node t).

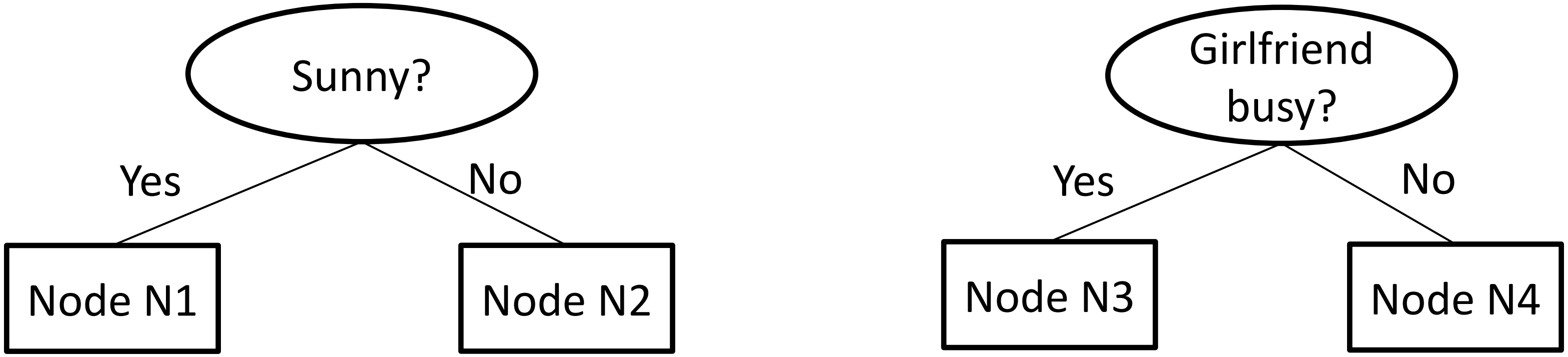
  - Maximum: $(1 - 1/n_c)$ when records are equally distributed among all classes, implying least interesting information
  - Minimum: (0.0) when all records belong to one class, implying most interesting information

Consider this example…

Before splitting

Fish	10
Don't Fish	10

*Note that "fish" or "don't fish" is the class label



Node N1:

Fish	3
Don't Fish	6

Node N2:

Fish	7
Don't Fish	4

Node N3:

Fish	7
Don't Fish	0

Node N4:

Fish	3
Don't Fish	10

Using Gini Index, evaluate which test condition is better

Computing GINI for Fishing Example

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

Before splitting

Fish	10
Don't Fish	10

Gini = 1 – [p(Fish)^2 + p(Don't Fish)^2]

= 1 – [p(10/20)^2 + p(10/20)^2]

= 1 – (0.25 + 0.25) = 0.5

Split by "Sunny?"

Yes

Fish	3
Don't Fish	6

P(Fish) = 3/9 P(Don't Fish) = 6/9

Gini = 1 – [(3/9)^2 +(6/9)^2]= 0.444

No

Fish	7
Don't Fish	4

P(Fish) = 7/11 P(Don't Fish) = 4/11

Gini = 1 – [(7/11)^2 +(4/11)^2]= 0.462

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Computing GINI for Fishing Example

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

Before splitting

Fish	10
Don't Fish	10

Gini = 1 – [p(Fish)^2 + p(Don't Fish)^2]

= 1 – [p(10/20)^2 + p(10/20)^2]

= 1 – (0.25 + 0.25) = 0.5

Split by "Girlfriend busy?"

Yes

Fish	7
Don't Fish	0

No

Fish	3
Don't Fish	10

Computing GINI for Fishing Example

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

Before splitting

Fish	10
Don't Fish	10

Gini = 1 – $[p(\text{Fish})^2 + p(\text{Don't Fish})^2]$

= 1 – $[p(10/20)^2 + p(10/20)^2]$

= 1 – (0.25 + 0.25) = 0.5

Split by "Girlfriend busy?"

Yes

Fish	7
Don't Fish	0

P(Fish) = 7/7 P(Don't Fish) = 0/7

Gini = 1 – $[(7/7)^2 + (0/0)^2]$= 0.0

No

Fish	3
Don't Fish	10

P(Fish) = 3/13 P(Don't Fish) =10/13

Gini = 1 – $[(3/13)^2 + (10/13)^2]$= 0.355

Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where n_i = number of records at child i,
n = number of records at node p.

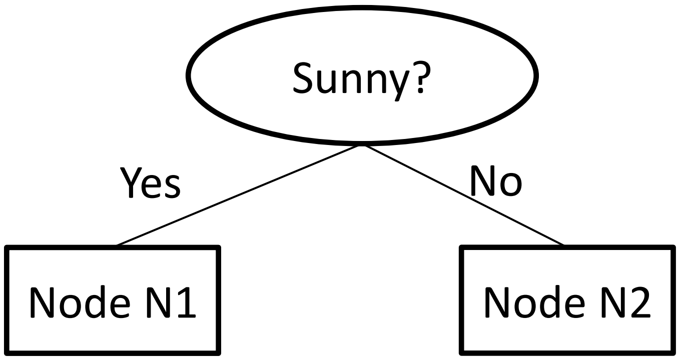
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Fishing Example Continue…

Before splitting

Fish	10
Don't Fish	10

*Note that "fish" or "don't fish" is the class label

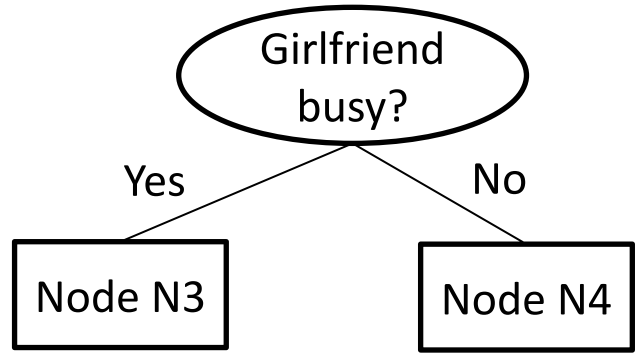


Fish	3
Don't Fish	6

Gini(N1) = 0.444

Fish	7
Don't Fish	4

Gini(N2) = 0.462



Fish	7
Don't Fish	0

Gini(N3) = 0.0

Fish	3
Don't Fish	10

Gini(N4) = 0.355

$$Gini_{sunny} = (9/20 * 0.444) + (11/20 * 0.462) = 0.45$$

$$Gini_{girlfriend} = (7/20 * 0.0) + (13/20 * 0.355) = 0.23$$

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  – Measures homogeneity of a node.
    - Maximum ($\log n_c$) when records are equally distributed among all classes implying the **least** information
    - Minimum (0.0) when all records belong to one class, implying the **most** information

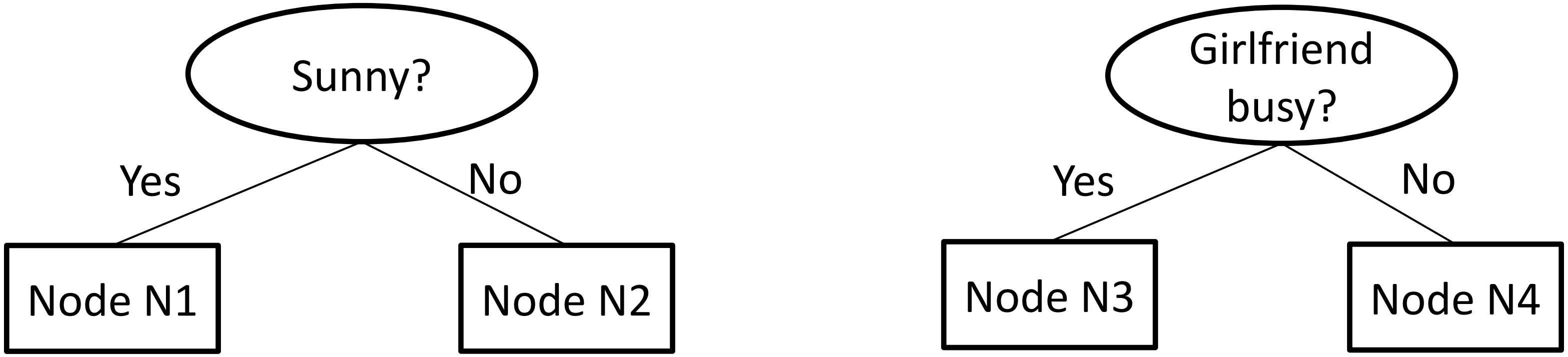
  – Entropy based computations are similar to the GINI index computations

Same Fishing Example…

Before splitting

Fish	10
Don't Fish	10

*Note that "fish" or "don't fish" is the class label



Node N1:

Fish	3
Don't Fish	6

Node N2:

Fish	7
Don't Fish	4

Node N3:

Fish	7
Don't Fish	0

Node N4:

Fish	3
Don't Fish	10

Using Entropy, evaluate which test condition is better

Computing Entropy for Fishing Example

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

Before splitting

Fish	10
Don't Fish	10

Entropy = $-$ [p(Fish)$\log_2$(p(Fish)) + p(Don't Fish)$\log_2$(p(Don't Fish))]

$= -[(10/20)\log_2(10/20) + (10/20)\log_2(10/20)]$

$= -[-0.5 + -0.5) = 1$

Split by "Sunny?"

Yes

Fish	3
Don't Fish	6

P(Fish) = 3/9 P(Don't Fish) = 6/9

Entropy $= -[(3/9)\log_2(3/9) + (6/9)\log_2(6/9)] = 0.918$

No

Fish	7
Don't Fish	4

P(Fish) = 7/11 P(Don't Fish) = 4/11

Entropy $= -[(7/11)\log_2(7/11) + (4/11)\log_2(4/11)] = 0.945$

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Computing Entropy for Fishing Example

$$Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

Before splitting

Fish	10
Don't Fish	10

Entropy = $-$ [p(Fish)$\log_2$(p(Fish)) + p(Don't Fish)$\log_2$(p(Don't Fish))]

= $-$ [(10/20) $\log_2$(10/20) + (10/20) $\log_2$(10/20)]

= $-$ [-0.5 + -0.5) = 1

Split by "Girlfriend busy?"

Yes

Fish	7
Don't Fish	0

No

Fish	3
Don't Fish	10

Computing Entropy for Fishing Example

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

Before splitting

Fish	10
Don't Fish	10

Entropy = – [p(Fish)$\log_2$(p(Fish)) +
p(Don't Fish)$\log_2$(p(Don't Fish))]

= – [(10/20) $\log_2$(10/20) + (10/20) $\log_2$(10/20)]

= – [-0.5 + -0.5) = 1

Split by "Girlfriend busy?"

Yes

Fish	7
Don't Fish	0

P(Fish) = 7/7 P(Don't Fish) = 0/7

Entropy = – [(7/7) $\log_2$(7/7) + (0/7) $\log_2$(0/7)] = 0

No

Fish	3
Don't Fish	10

P(Fish) = 3/13 P(Don't Fish) = 10/13

Entropy = – [(3/13) $\log_2$(3/13) + (10/13) $\log_2$(10/13)] = 0.779

Splitting Based on INFO...

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;

n_i is number of records in partition i

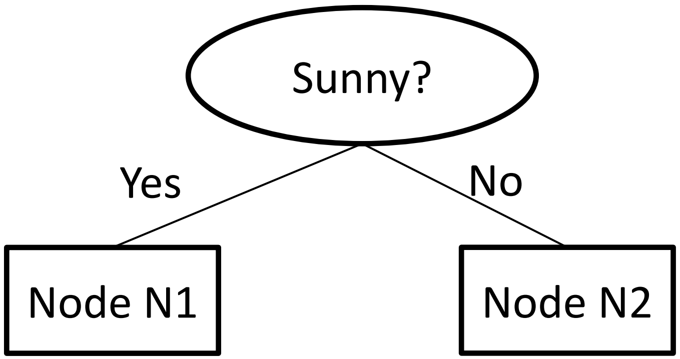
  - Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
  - Used in ID3 and C4.5

Fishing Example Continue…

Before splitting

Fish	10
Don't Fish	10

*Note that "fish" or "don't fish" is the class label

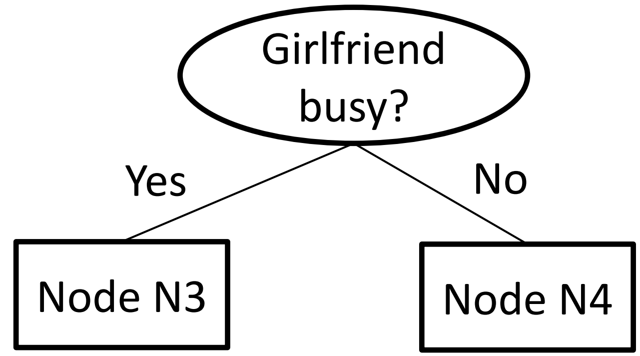


Fish	3
Don't Fish	6

Entropy(N1) = 0.918

Fish	7
Don't Fish	4

Entropy(N2) = 0.945



Fish	7
Don't Fish	0

Entropy(N3) = 0.0

Fish	3
Don't Fish	10

Entropy(N4) = 0.779

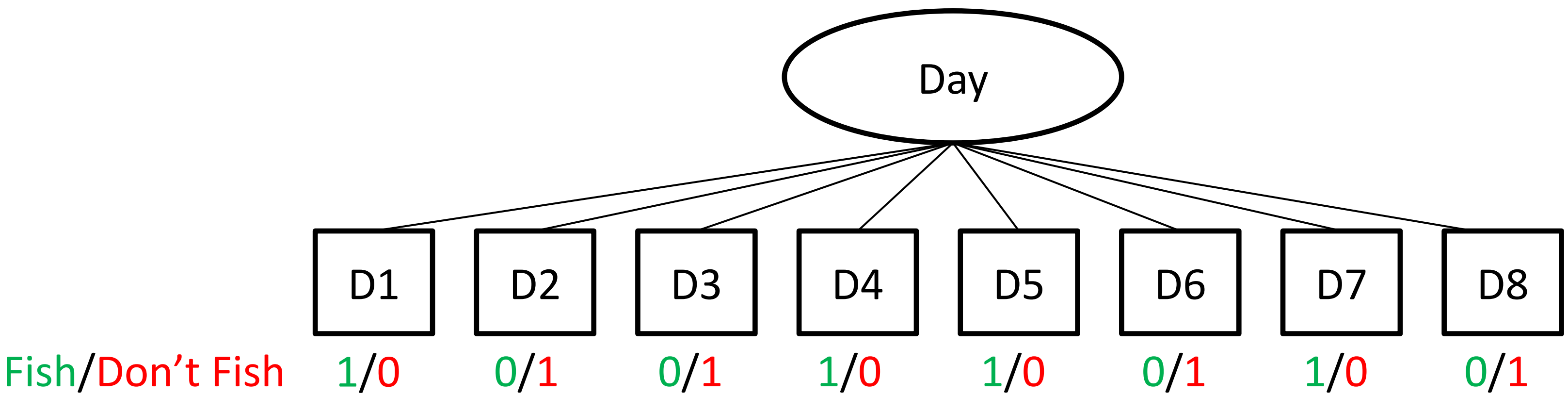
$$Gain_{split} = 1 - (9/20 * 0.918) + (11/20 * 0.945) = 0.06$$

$$Gain_{split} = 1 - (7/20 * 0.0) + (13/20 * 0.779) = 0.493$$

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Problem…

- Disadvantage:
  - Tends to prefer splits that result in large number of partitions, each being small but pure.



All subset are perfectly pure! => optimal split!?

Splitting Based on INFO...

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions
n_i is the number of records in partition i

  – Adjusts Information Gain by the entropy of the partitioning (SplitINFO).
  – Higher entropy partitioning (large number of small partitions) is penalized!
  – Used in C4.5
  – Designed to overcome the disadvantage of Information Gain

Splitting Criteria Based on Misclassification Error

- Misclassification error at a node t :

$$\boxed{Error(t) = 1 - \max_i P(i \mid t)}$$

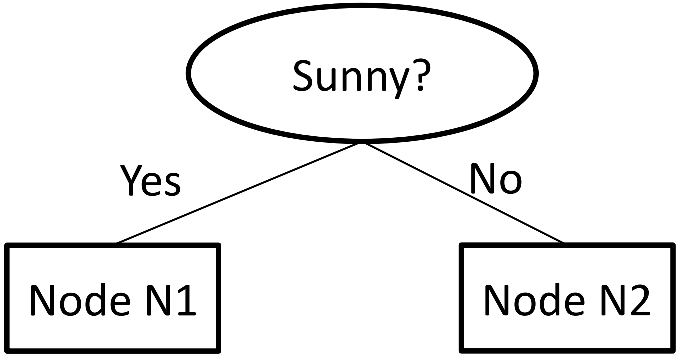
- Measures misclassification error made by a node.
  - Maximum: $(1 - 1/n_c)$ when records are equally distributed among all classes, implying least interesting information
  - Minimum: (0.0) when all records belong to one class, implying most interesting information

Still the Fishing Example…

Before splitting

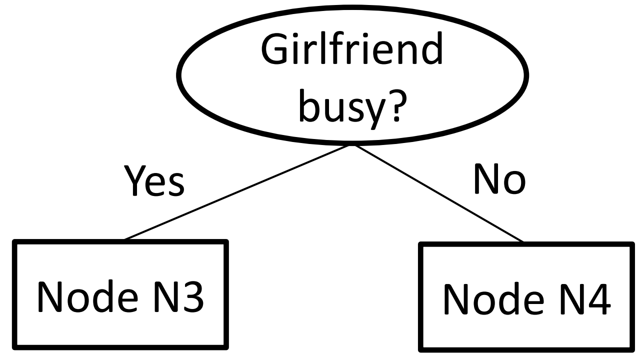
Fish	10
Don’t Fish	10

*Note that “fish” or “don’t fish” is the class label



Fish	3
Don’t Fish	6

Fish	7
Don’t Fish	4



Fish	7
Don’t Fish	0

Fish	3
Don’t Fish	10

Using Classification Error, evaluate which test condition is better

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Computing Error for Fishing Example

$$Error(t) = 1 - \max_i P(i \mid t)$$

Before splitting

Fish	10
Don't Fish	10

Error = 1 – max(p(Fish),p(Don't Fish))

= 1 – max((10/20),(10/20))

= 1 – [0.5) = 0.5

Split by "Sunny?"

Yes

Fish	3
Don't Fish	6

P(Fish) = 3/6 P(Don't Fish) = 6/9

Error = 1 – max((3/9),(6/9)) = 0.333

No

Fish	7
Don't Fish	4

P(Fish) = 7/11 P(Don't Fish) = 4/11

Error = 1 – max((7/11),(4/11)) = 0.363

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Computing Error for Fishing Example

$$Error(t) = 1 - \max_i P(i \mid t)$$

Before splitting

Fish	10
Don't Fish	10

Error = 1 – max(p(Fish),p(Don't Fish))

= 1 – max((10/20),(10/20))

= 1 – [0.5) = 0.5

Split by "Girlfriend busy?"

Yes

Fish	7
Don't Fish	0

No

Fish	3
Don't Fish	10

Computing Error for Fishing Example

$$Error(t) = 1 - \max_i P(i \mid t)$$

Before splitting

Fish	10
Don’t Fish	10

Error = 1 – max(p(Fish),p(Don’t Fish))

= 1 – max((10/20),(10/20))

= 1 – [0.5) = 0.5

Split by “Girlfriend busy?”

Yes

Fish	7
Don’t Fish	0

P(Fish) = 7/7 P(Don’t Fish) = 0/7

Error = 1 – max((7/7),(0/7)) = 0

No

Fish	3
Don’t Fish	10

P(Fish) = 3/13 P(Don’t Fish) = 10/13

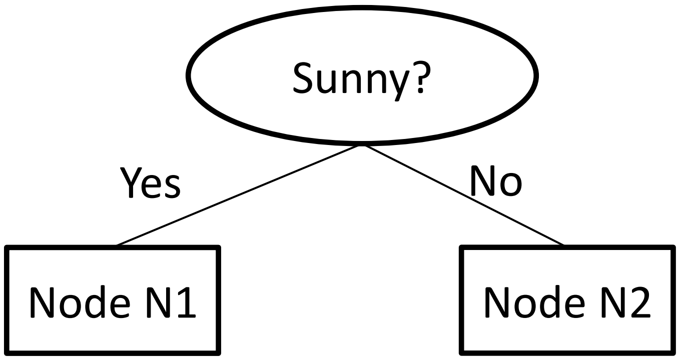
Error = 1 – max((3/13),(10/13)) = 0.231

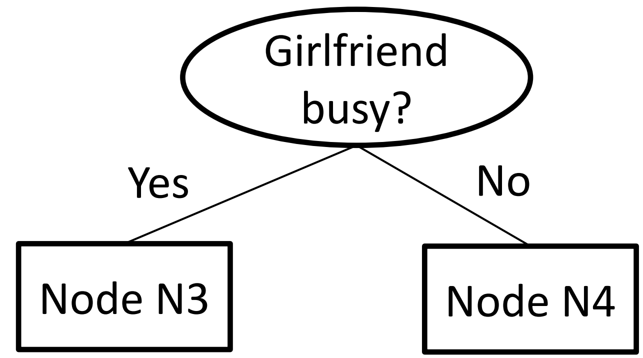
Fishing Example Continue…

Before splitting

Fish	10
Don't Fish	10

*Note that "fish" or "don't fish" is the class label





Node N1:

Fish	3
Don't Fish	6

Gini(N1) = 0.333

Node N2:

Fish	7
Don't Fish	4

Gini(N2) = 0.363

Node N3:

Fish	7
Don't Fish	0

Gini(N3) = 0.0

Node N4:

Fish	3
Don't Fish	10

Gini(N4) = 0.231

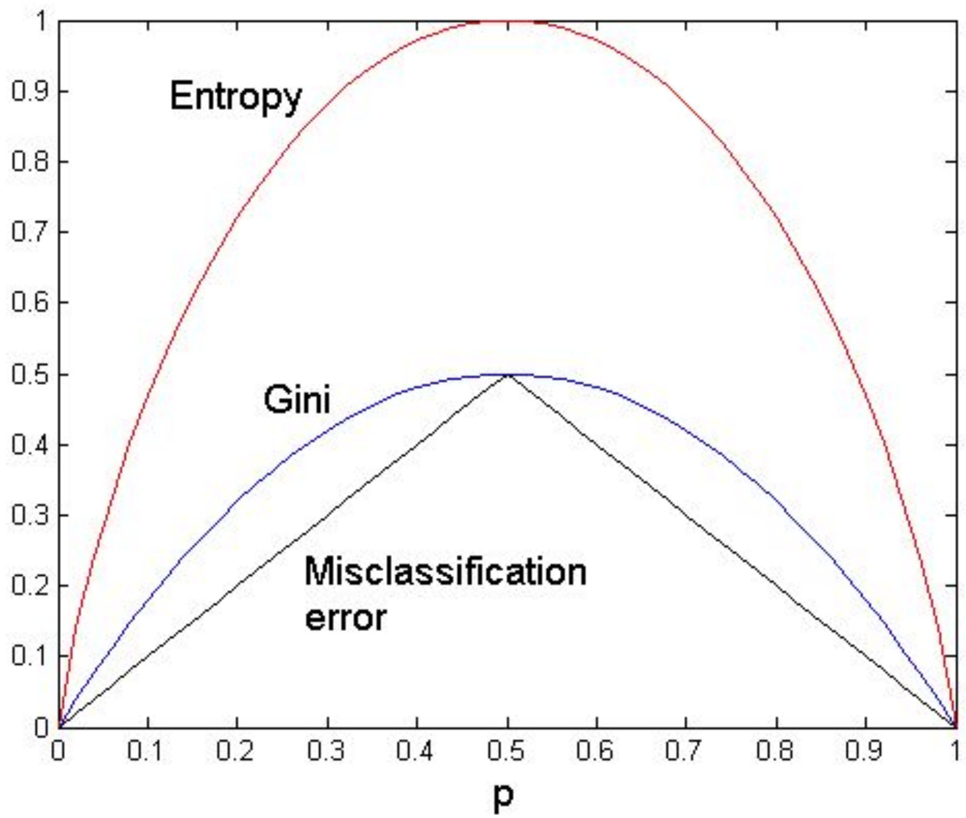
$$Gini_{sunny} = (9/20 * 0.444) + (11/20 * 0.462) = 0.45$$

$$Gini_{girlfriend} = (7/20 * 0.0) + (13/20 * 0.231) = 0.15$$

Comparison Among Splitting Criteria

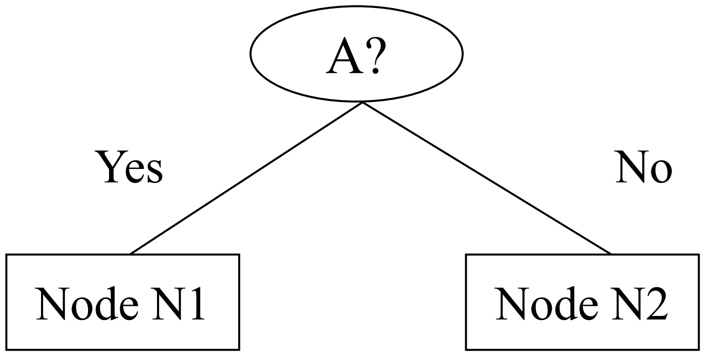
- For a 2-class problem, which curve is “entropy”, “Gini”, “error”?

Impurity measure



Misclassification Error vs GINI

- Which measure gives bigger impurity gain?



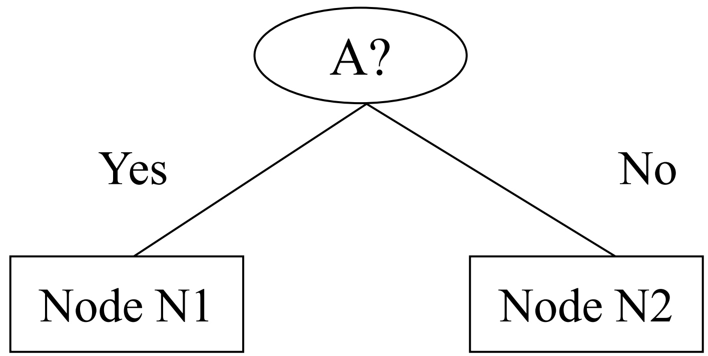
	Parent
C1	7
C2	3

	N1	N2
C1	3	4
C2	0	3

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Misclassification Error vs GINI

- Which measure gives bigger impurity gain?



	Parent
C1	7
C2	3
Error = 0.3	

Error(N1)
= 1 – max((3/3),(0/3))
= 0

Error(N2)
= 1 – max((4/7),(3/7))
= 0.428

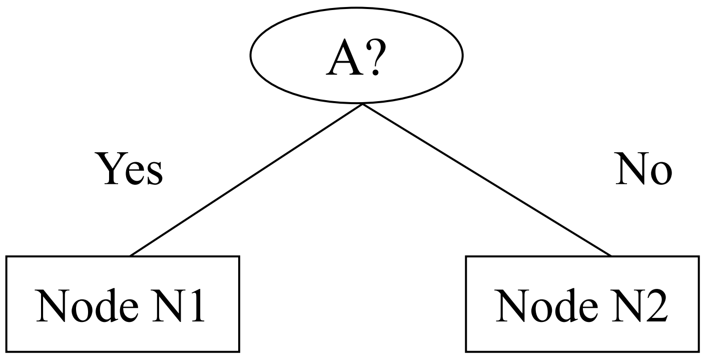
Error(Children)
= 3/10 * 0 + 7/10 * 0.428
= 0.3

	N1	N2
C1	3	4
C2	0	3
Error = 0.3		

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Misclassification Error vs GINI

- Which measure gives bigger impurity gain?



	Parent
C1	7
C2	3
Gini = 0.42	

Gini(N1)
$= 1 - (3/3)^2 - (0/3)^2$
$= 0$

Gini(N2)
$= 1 - (4/7)^2 - (3/7)^2$
$= 0.489$

Gini(Children)
$= 3/10 * 0 + 7/10 * 0.489$
$= 0.342$

Gini improves !!

	N1	N2
C1	3	4
C2	0	3
Gini = 0.342		

Tree Induction

- 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

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

Example: C4.5

- Simple depth-first construction.
- Uses Information Gain
- Sorts Continuous Attributes at each node.
- Needs entire data to fit in memory.
- Unsuitable for Large Datasets.
  - Needs out-of-core sorting
- Software download
  - You can download the C4.5 software from:
    http://www.rulequest.com/Personal/c4.5r8.tar.gz
  - And the advanced C5.0 software from http://www.rulequest.com/r207.html

Pros and Cons

- Pros
  - Simple to understand, interpret and visualize
  - Can handle both categorical and numerical data
  - Extremely fast at classifying unknown records
  - Accuracy is comparable to other classification techniques for many simple data sets
  - Non-linear relationship between variables does not affect the performance

- Cons
  - Prone to overfitting
  - Unstable because small variation in the data result in completely different trees generated
  - Greedy algorithm cannot guarantee the return of globally optimal decision tree