Data Mining
Classification: Basic Concepts and Techniques

Lecture Notes for Chapter 3

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

Classification: Definition

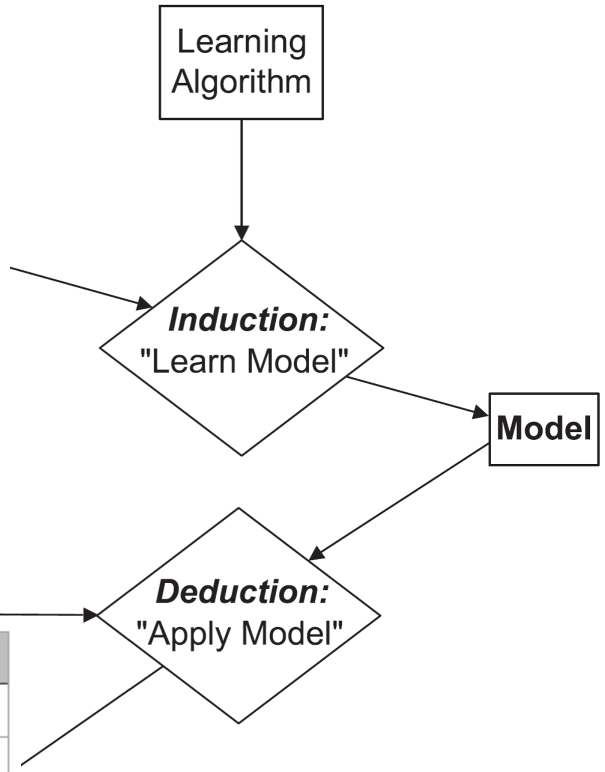
- Given a collection of records (training set)
  - Each record is by characterized by a tuple (x,y), where x is the attribute set and y is the class label
    - x: attribute, predictor, independent variable, input
    - y: class, response, dependent variable, output
- Task:
  - Learn a model that maps each attribute set x into one of the predefined class labels y

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

General Approach for Building Classification Model

Attrib1	Attrib2	Attrib3	Class
Yes	Single/Divor	No	No
No	Married	No	No
No	Single/Divor	Yes	No
Yes	Married	No	No
No	Single/Divor	No	Yes
No	Married	Yes	No
Yes	Single/Divor	No	No
No	Single/Divor	No	Yes
No	Married	Yes	No
No	Single/Divor	No	Yes



Attrib1	Attrib2	Attrib3	Class
No	Married	No	?
Yes	Single/Divor	Yes	?
Yes	Single/Divor	Yes	?
No	Married	Yes	?
No	Married	No	?

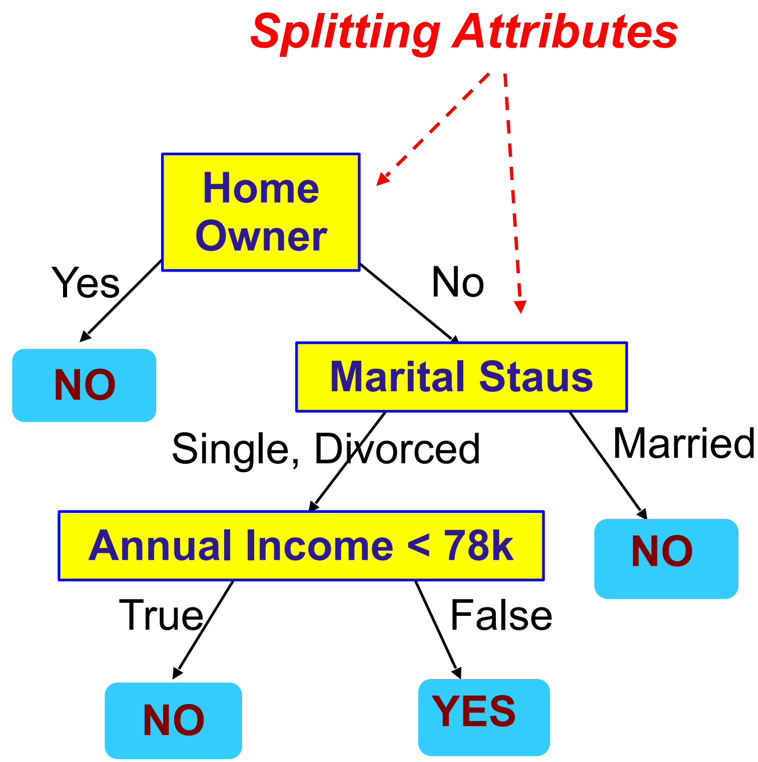
Figure 3.3. General framework for building a classification model.

Classification Technique: Decision Tree

Attribute Attribute Attribute class

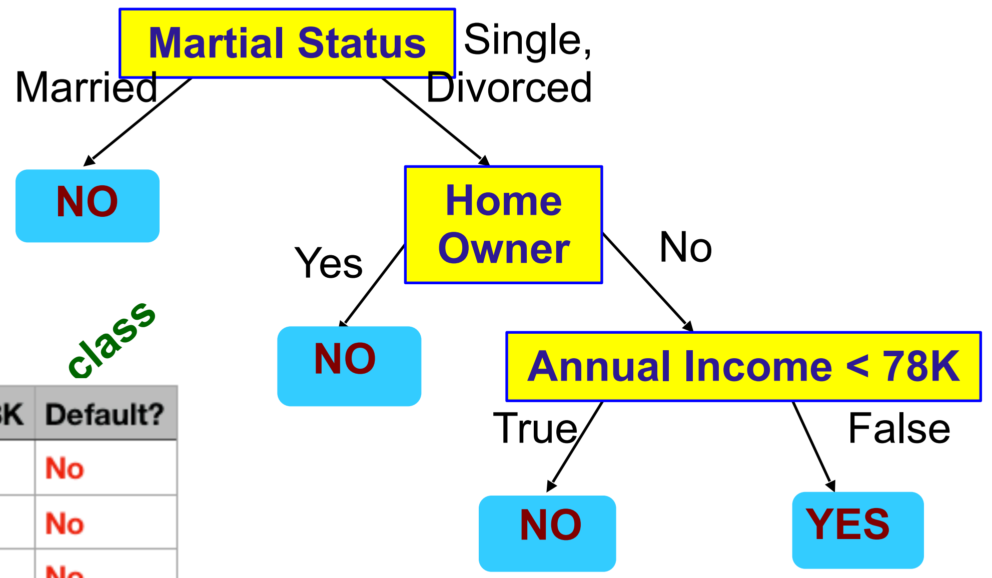
Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes

Training Data



Model: Decision Tree

Another Example of Decision Tree



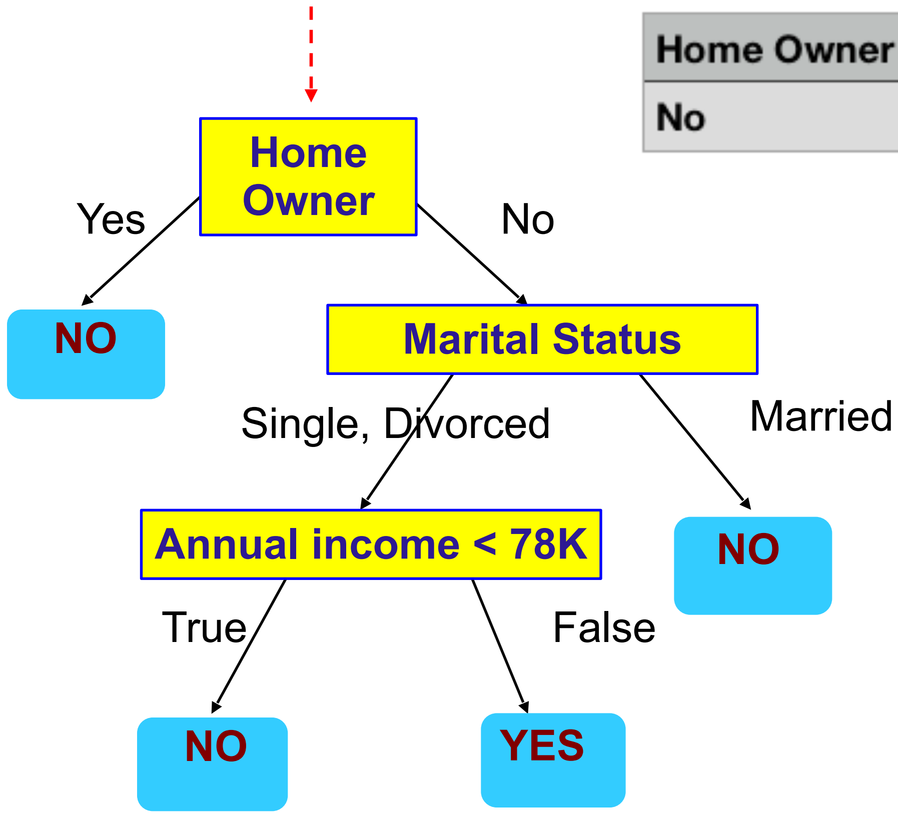
Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes

Attribute, Attribute, Attribute, class

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

Apply Model to Test Data

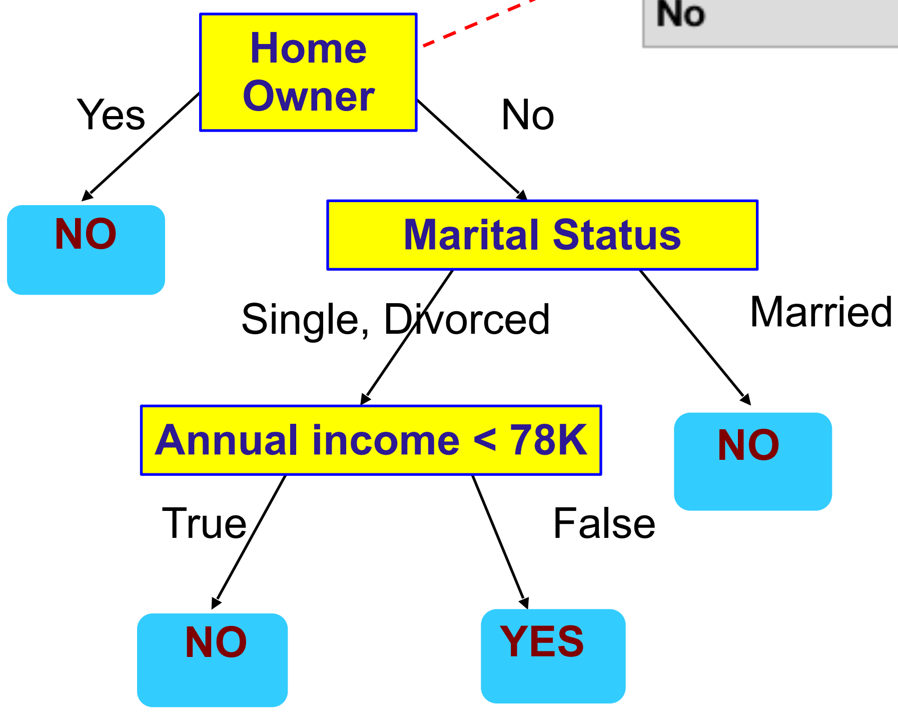
Start from the root of tree.



Test Data

Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?

Apply Model to Test Data



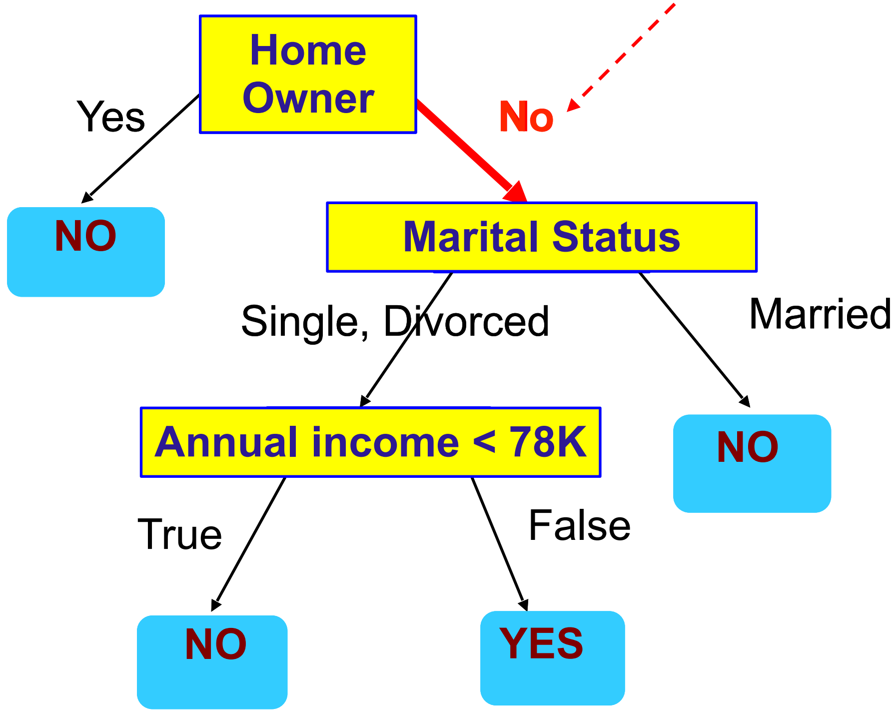
Test Data

Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?

Apply Model to Test Data

Test Data

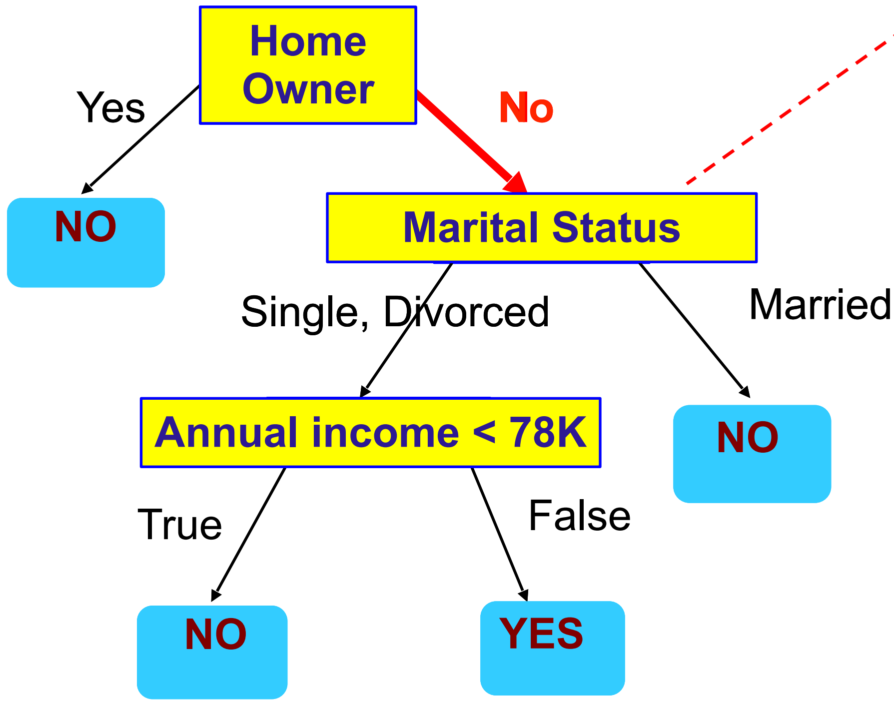
Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?



Apply Model to Test Data

Test Data

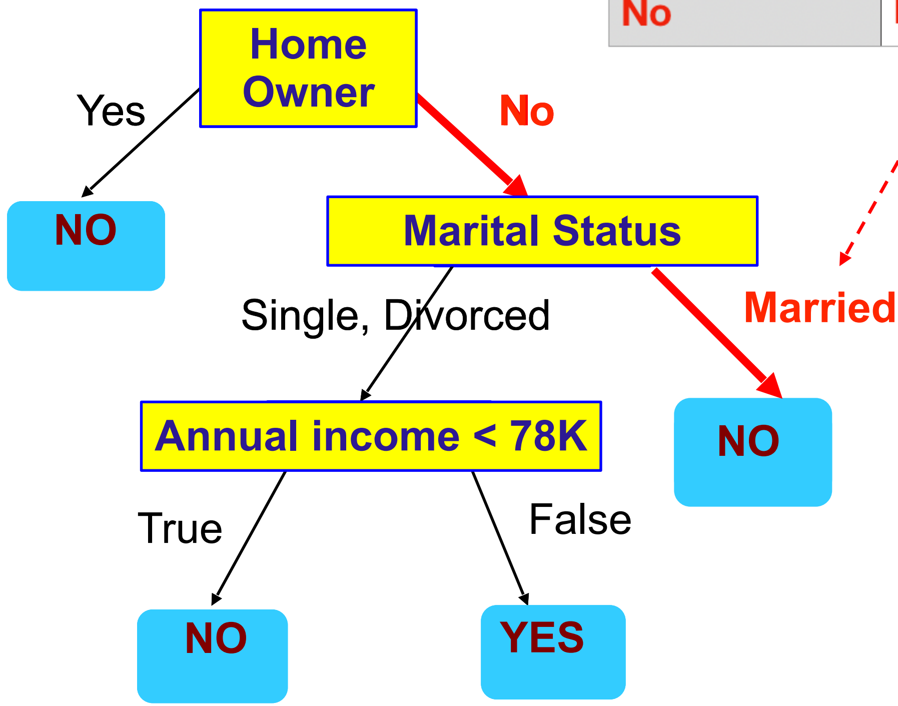
Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?



Apply Model to Test Data

Test Data

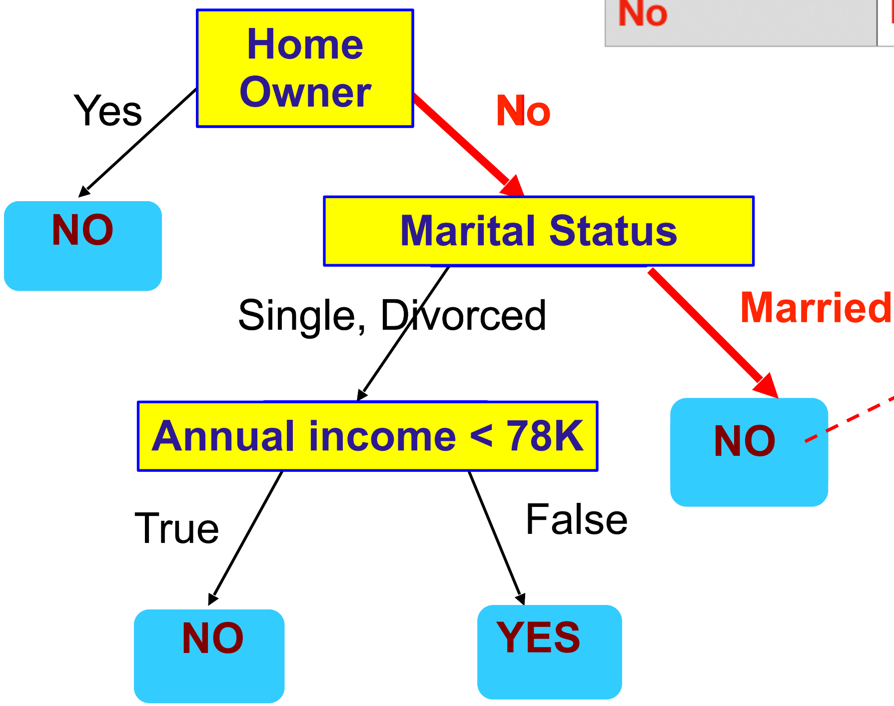
Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?



Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual income < 78K	Default?
No	Married	No	?



Assign Defaulted to "No"

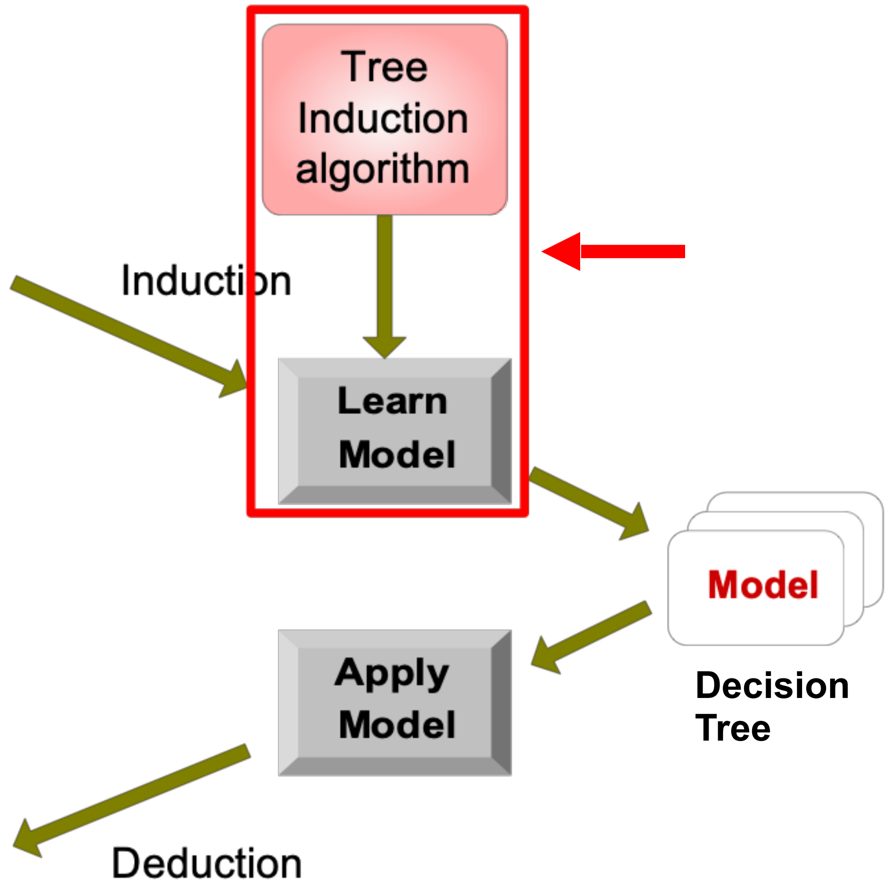
Decision Tree Induction

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

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	?

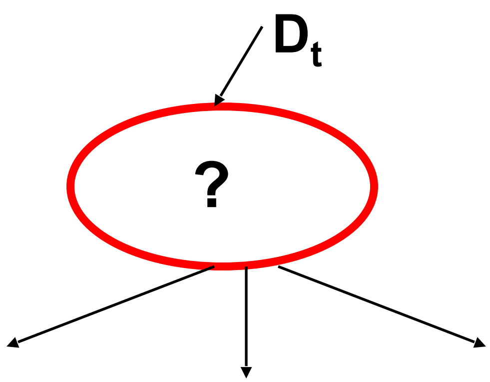
Test Set

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 contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes



Hunt's Algorithm

{1,2,3,4,5,6,7,8,9,10}

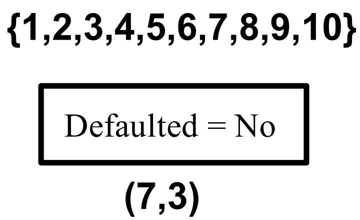
$\boxed{\text{Defaulted = No}}$

(7,3)

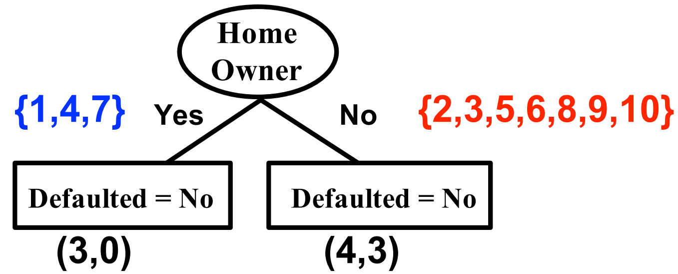
(a)

Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	**No**
No	Married	No	**No**
No	Single/Divorced	Yes	**No**
Yes	Married	No	**No**
No	Single/Divorced	No	**Yes**
No	Married	Yes	**No**
Yes	Single/Divorced	No	**No**
No	Single/Divorced	No	**Yes**
No	Married	Yes	**No**
No	Single/Divorced	No	**Yes**

Hunt's Algorithm



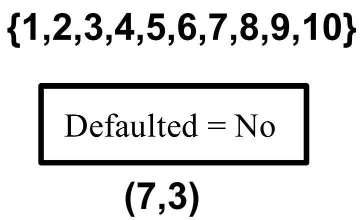
(a)



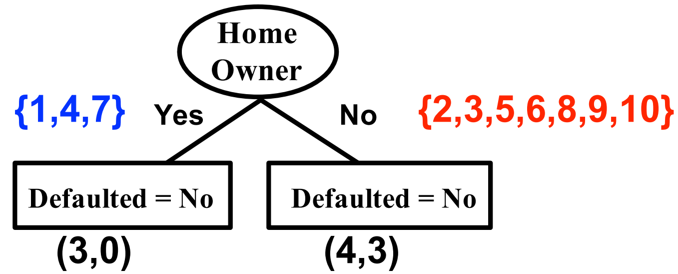
(b)

Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes

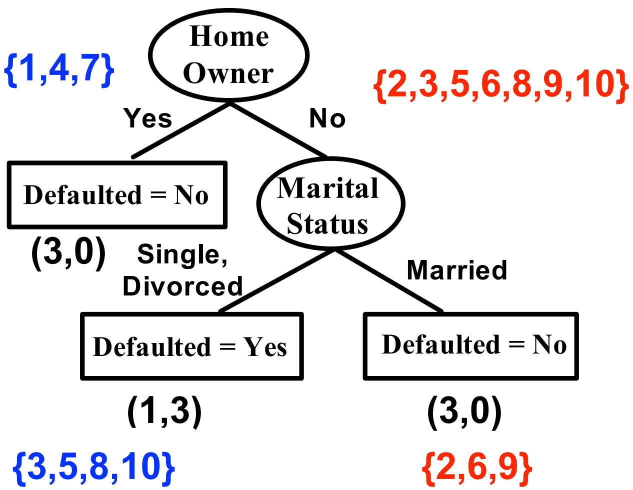
Hunt's Algorithm



(a)



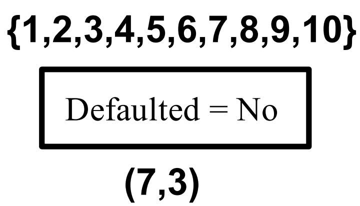
(b)



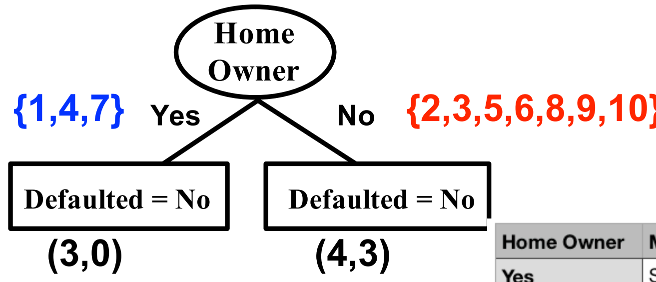
(c)

Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes

Hunt's Algorithm

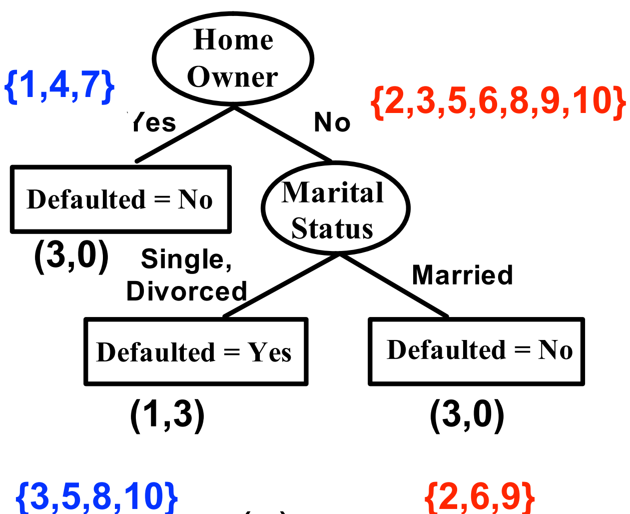


(a)

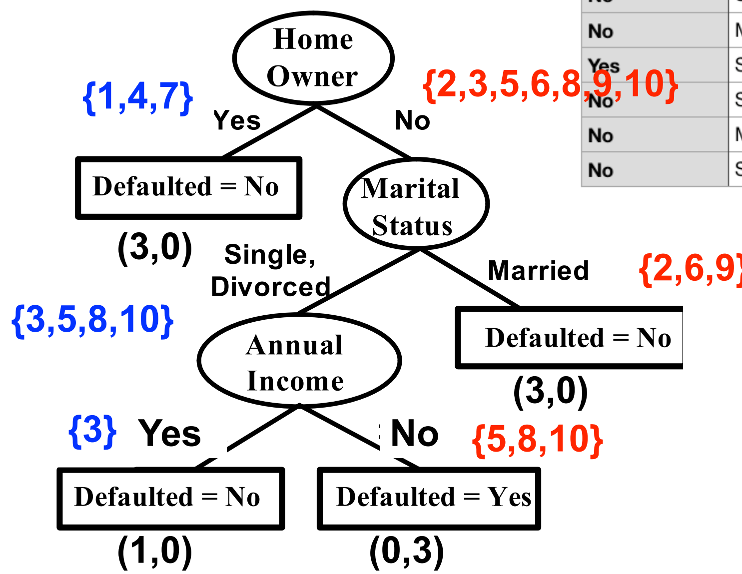


(b)

Home Owner	Marital Status	Annual income < 78K	Default?
Yes	Single/Divorced	No	No
No	Married	No	No
No	Single/Divorced	Yes	No
Yes	Married	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
Yes	Single/Divorced	No	No
No	Single/Divorced	No	Yes
No	Married	Yes	No
No	Single/Divorced	No	Yes



(c)



(d)

Design Issues of Decision Tree Induction

- How should training records be split?
  - Method for expressing test condition
    - depending on attribute types
  - Measure for evaluating the goodness of a test condition

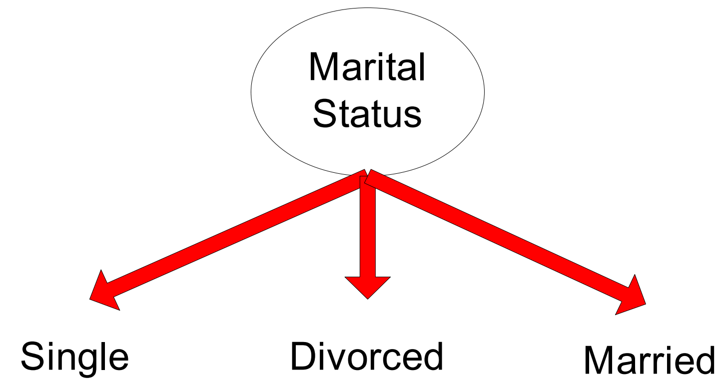
- How should the splitting procedure stop?
  - Stop splitting if all the records belong to the same class or have identical attribute values
  - Early termination

Methods for Expressing Test Conditions

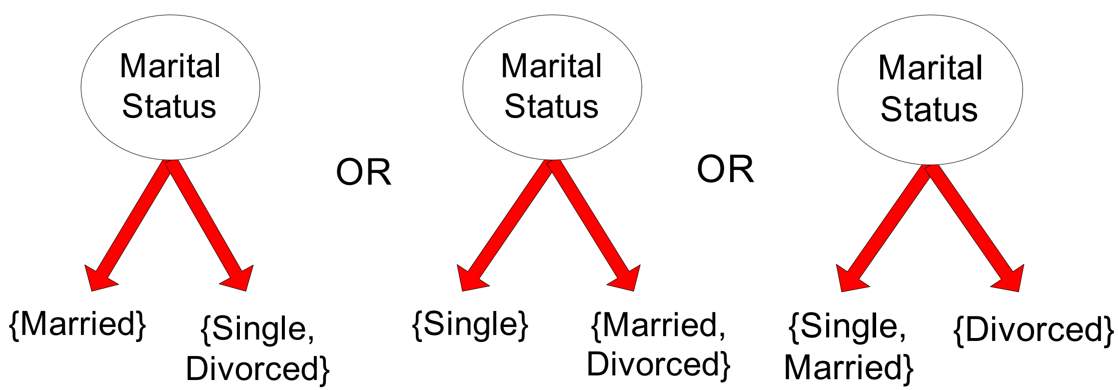
- Depends on attribute types
  – Binary
  – Nominal
  – Ordinal
  – Continuous

Test Condition for Nominal Attributes

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

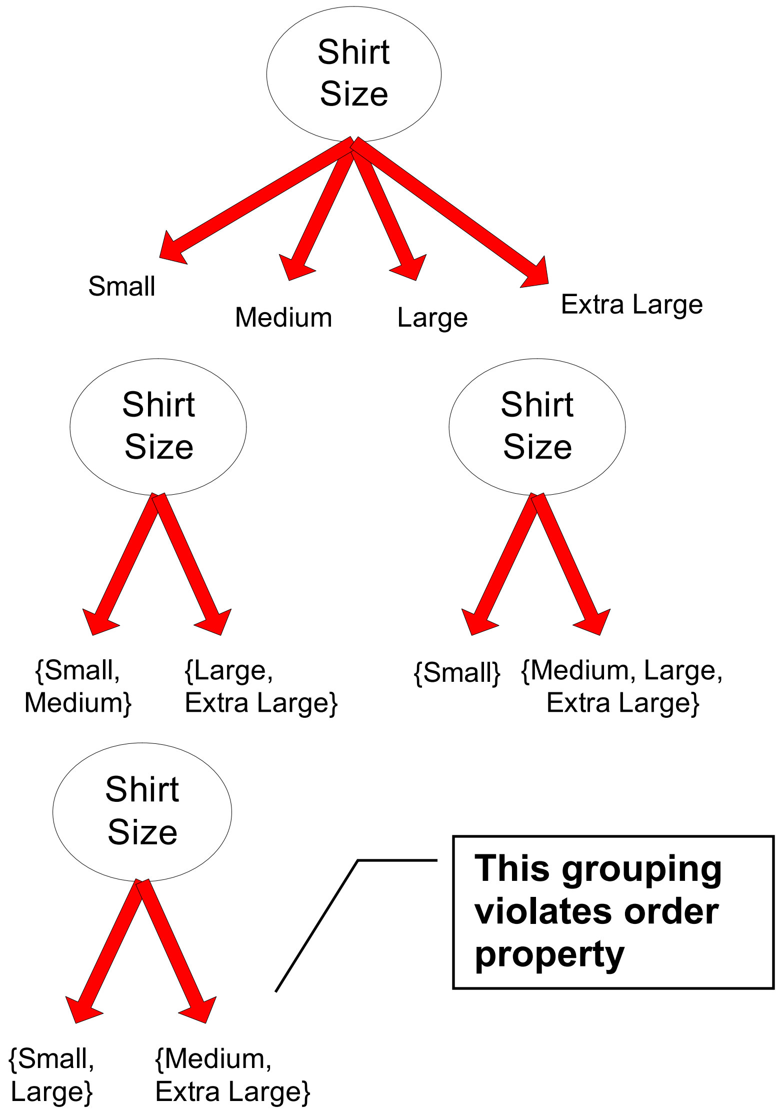


- Binary split:
  - Divides values into two subsets

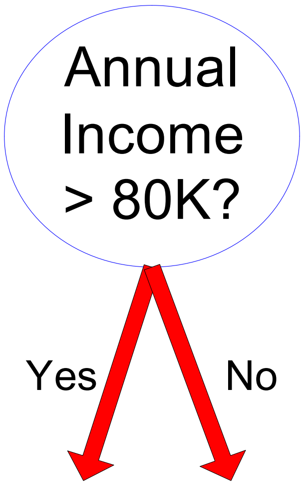


Test Condition for Ordinal Attributes

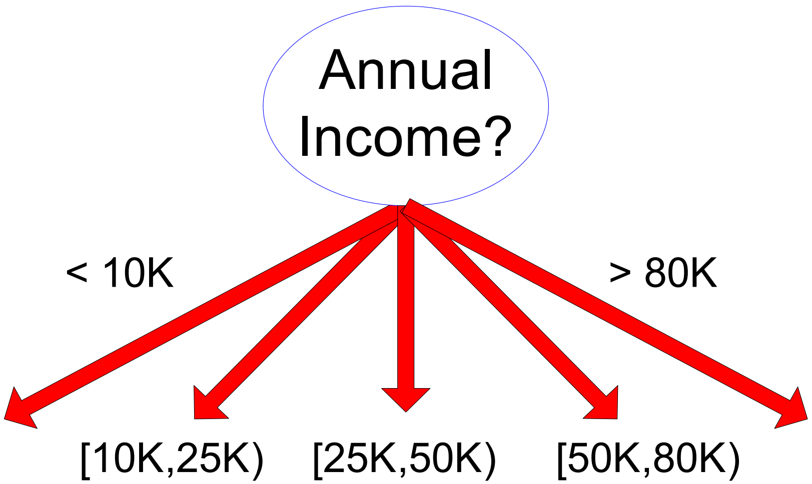
- Multi-way split:
  - Use as many partitions as distinct values
- Binary split:
  - Divides values into two subsets
  - Preserve order property among attribute values



Test Condition for Continuous Attributes



(i) Binary split

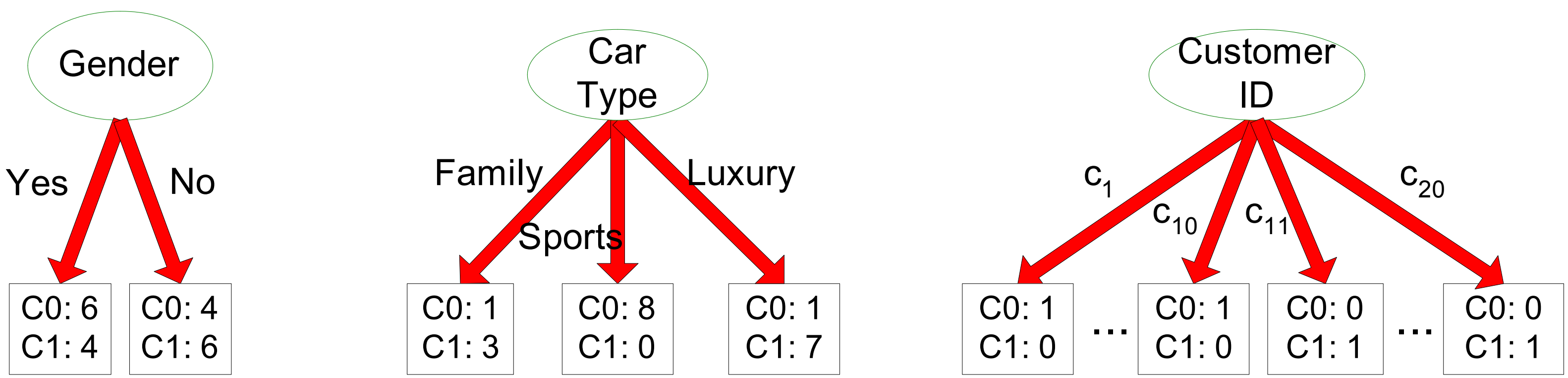


(ii) Multi-way split

How to determine the Best Split

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

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 purer class distribution are preferred

- Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1
High degree of impurity	**Low degree of impurity**

Measures of Node Impurity

- Gini Index

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node **t**, and c is the total number of classes

- Entropy

$$Entropy = -\sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

- Misclassification error

$$Classification\ error = 1 - \max[p_i(t)]$$

Finding the Best Split

1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting
   - Compute impurity measure of each child node
   - M is the weighted impurity of children
3. Choose the attribute test condition that produces the highest gain

   Gain = P – M

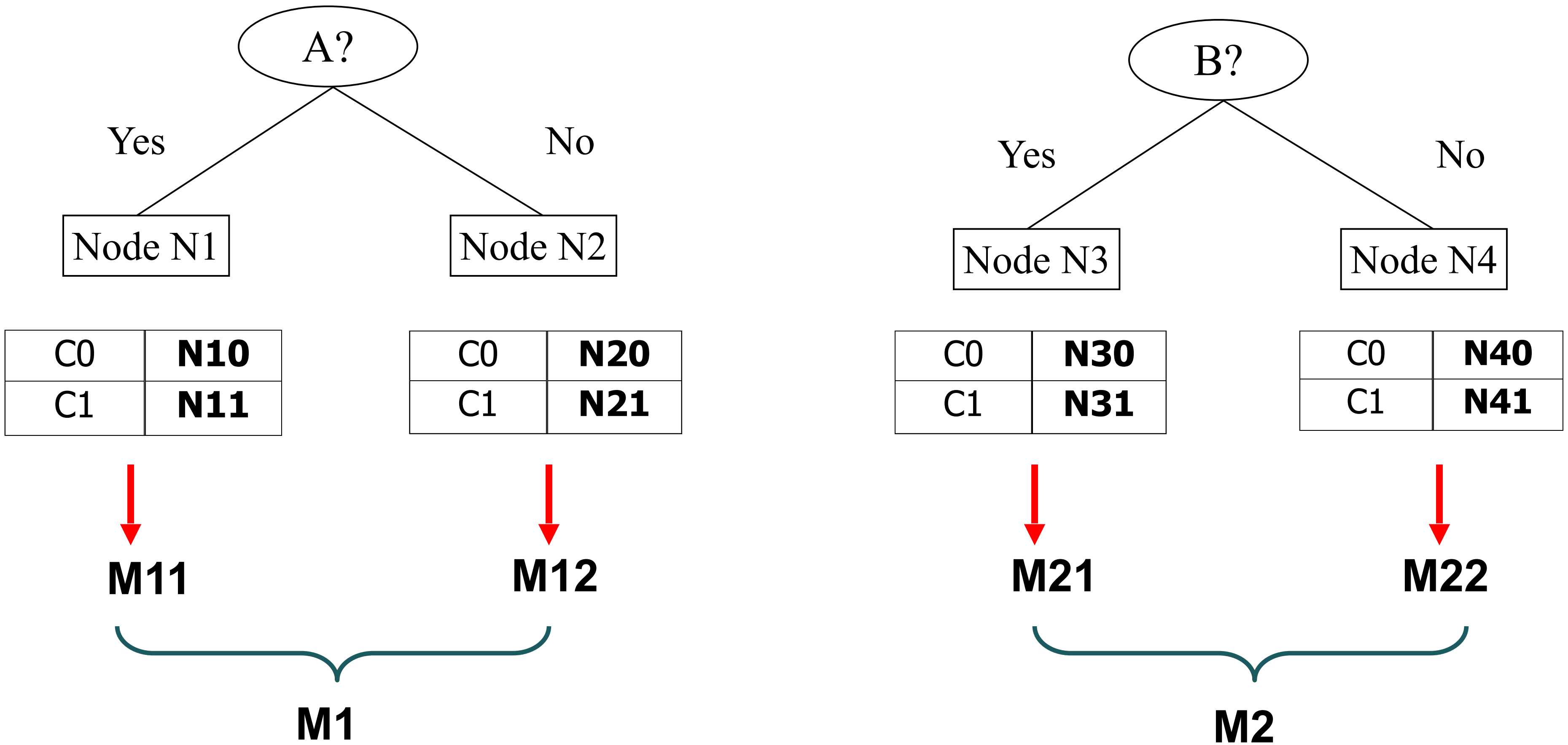
   or equivalently, lowest impurity measure after splitting (M)

Finding the Best Split

Before Splitting:

C0	**N00**
C1	**N01**

→ **P**



Gain = P – M1 vs P – M2

Measure of Impurity: GINI

- Gini Index for a given node t

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t, and c is the total number of classes

  – Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
  – Minimum of 0 when all records belong to one class, implying the most beneficial situation for classification
  – Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

Measure of Impurity: GINI

- Gini Index for a given node t :

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1 C2

- For 2-class problem (p, 1 – p):
  - GINI = $1 – p^2 – (1 – p)^2 = 2p (1-p)$

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Computing Gini Index of a Single Node

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

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

Computing Gini Index for a Collection of Nodes

- When a node p is split into k partitions (children)

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

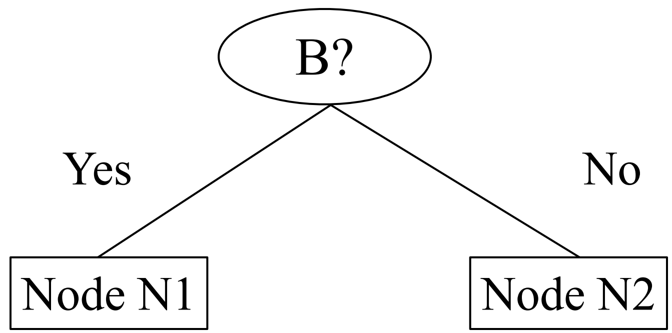
where, n_i = number of records at child i,

n = number of records at parent node p.

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  – Larger and Purer Partitions are sought for.

	Parent
C1	7
C2	5
Gini = 0.486	



Gini(N1)
$= 1 - (5/6)^2 - (1/6)^2$
$= 0.278$

Gini(N2)
$= 1 - (2/6)^2 - (4/6)^2$
$= 0.444$

	N1	N2
C1	5	2
C2	1	4
Gini=0.361		

Weighted Gini of N1 N2
$= 6/12 * 0.278 +$
$6/12 * 0.444$
$= 0.361$

Gain = 0.486 – 0.361 = 0.125

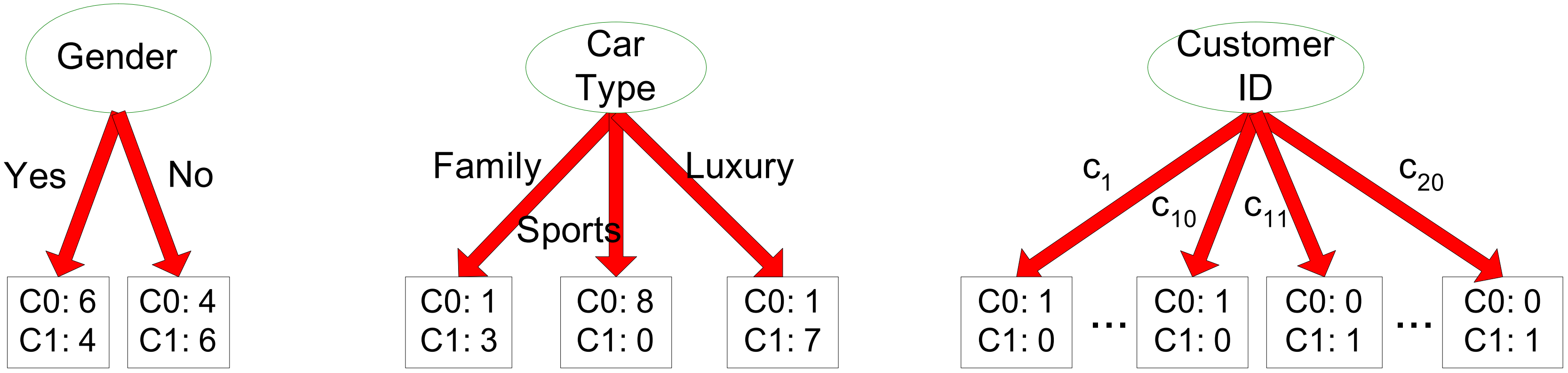
Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Problem with large number of partitions

- Node impurity measures tend to prefer splits that result in large number of partitions, each being small but pure



  – Customer ID has highest information gain because node impurity measure for all the children is zero

Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info} \qquad Split\ Info = -\sum_{i=1}^{k} \frac{n_i}{n} log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

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

Gain Ratio

- 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

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52