TILBURG UNIVERSITY

Business Intelligence and Data Management

academic year 2019-2020

Dr. Ekaterini Ioannou, Department of Management, University of Tilburg

Lecture 9

topic: Decision Trees

material: Chapters 9 (book “Data Mining for Business Intelligence”)

https://www.youtube.com/watch?v=AmCV4g7_-QM

Steps in the DM Process:

Business understanding → Data preparation → Model building → Testing & Evaluation → Deployment

Agenda

Date		Lecture contents	Lecturer	Lab topics	Test
Jan-07	1	Intro. to BI+ Data Management	Caron		
Jan-30				SQL-1	1
Jan-28	2	Data warehousing	Caron		
Feb-06				SQL-2	2
Feb-03	3	OLAP business databases & dashboard	Caron		
Feb-13				SQL-3 & OLAP	3a & 3b
Feb-10	4	Data mining introduction	Ioannou		
	5	Regression models	Ioannou		
Feb-17	6	Naïve Bayes	Ioannou		
	7	k nearest neighbors	Ioannou		
Feb-20				Bayes & neighbors	4
Feb-27	8	Performance measures	Ioannou		
Mar-02	9	Decision trees	Ioannou		
Mar-05				Dec. trees	5
Mar-09	10	Association rules	Ioannou		
Mar-11,12&13				Ass. Rules	6
Mar-16	11	Clustering (+20 mins exam preparation)	Ioannou		
Mar-19				Clustering	7

Classification: Learning & Applying

X y

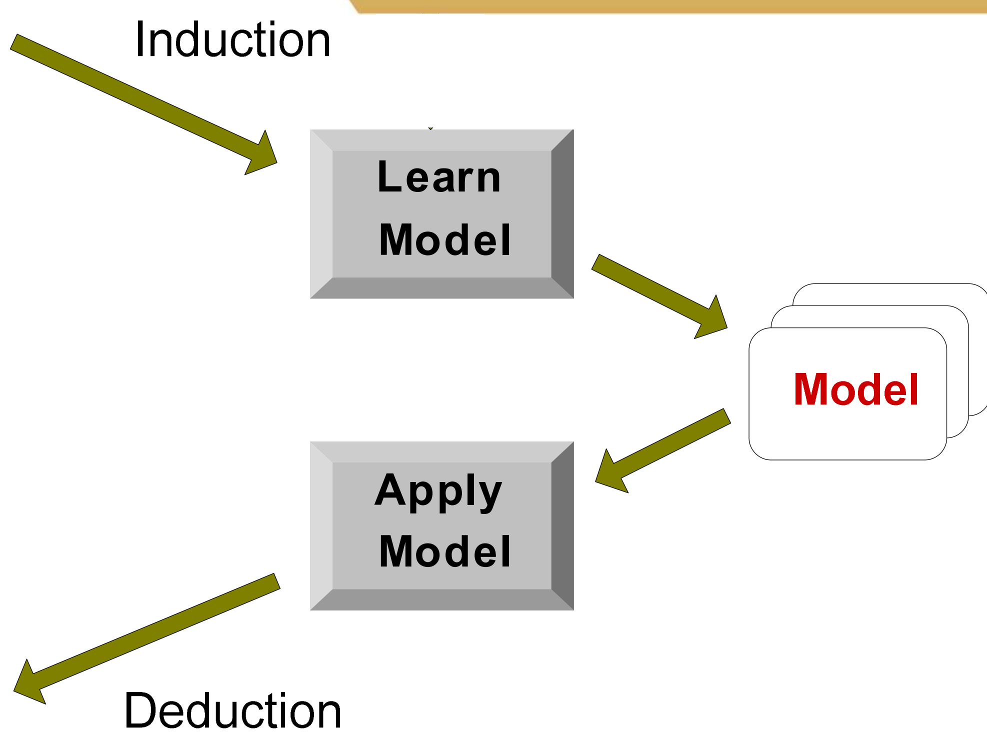
Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

X’ y'

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



Decision Tree

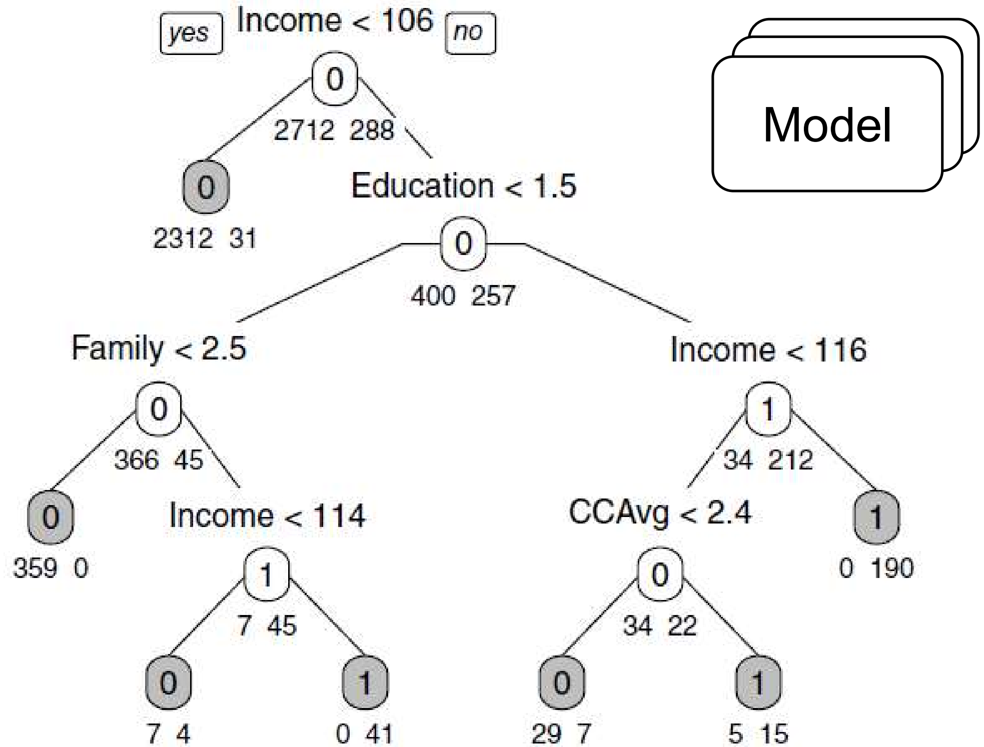
- Data-driven method
- Popular classification technique

Reasons

- Performs well across a wide range of situations
- Does not require much effort from the analyst
- Easy understandable by the consumers
  - At least when the trees are not too large
- Can be used for both
  - Classification, called classification trees
  - Prediction, called regression trees

Example

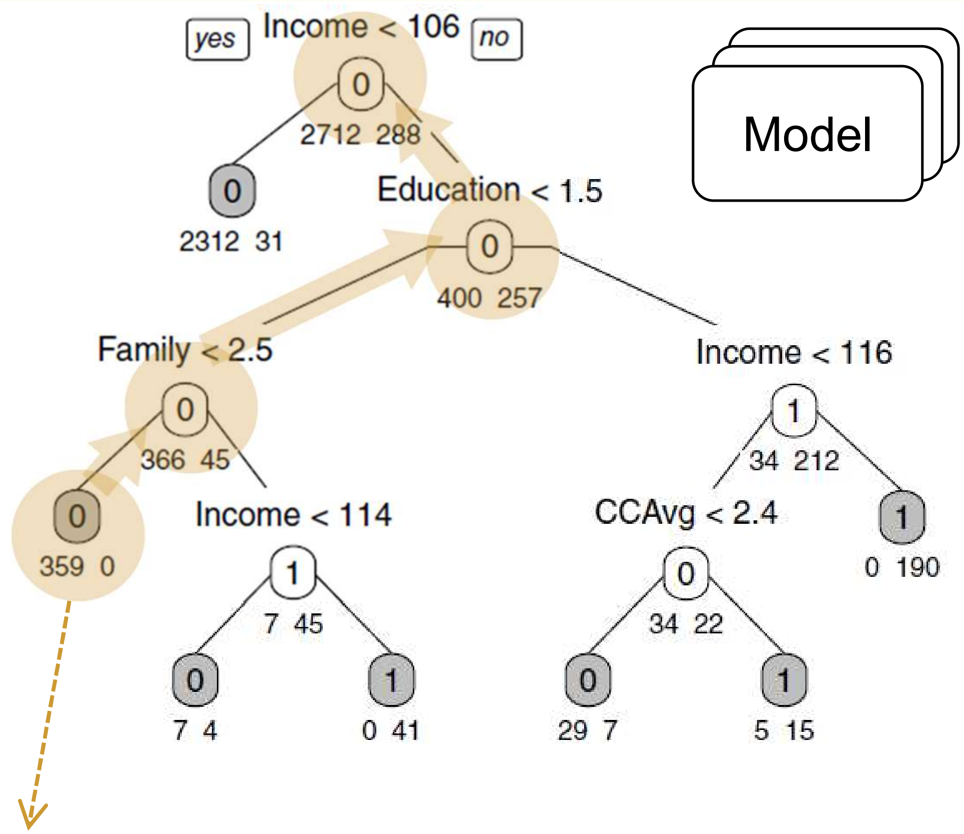
- Classifying bank customers as acceptors on not of a loan offer
- Decision tree:
  - Based on various attributes, e.g., income, education level
  - Created using the data from the available records



- Values above the white nodes give the splitting value on a predictor
- Values below the nodes gives the number of records in the split
- The gray nodes, named terminals, marked with 0 or 1 corresponding to a non acceptor (marked with 0) or acceptor (marked with 1)

Example

- Trees are easy translated into a set of rules
- In this example, rules are for classifying one bank customer



IF Family<2.5 AND Education<1.5 AND Income≥106

THEN Class=0

Main processing

- Separate records into subgroups by creating splits on predictors
- Splits create logical rules that are transparent and easily understandable
    - E.g., IF … THEN Class = ·
- Resulting subgroups should be more homogeneous in terms of the outcome variable
    - Creating useful prediction or classification rules

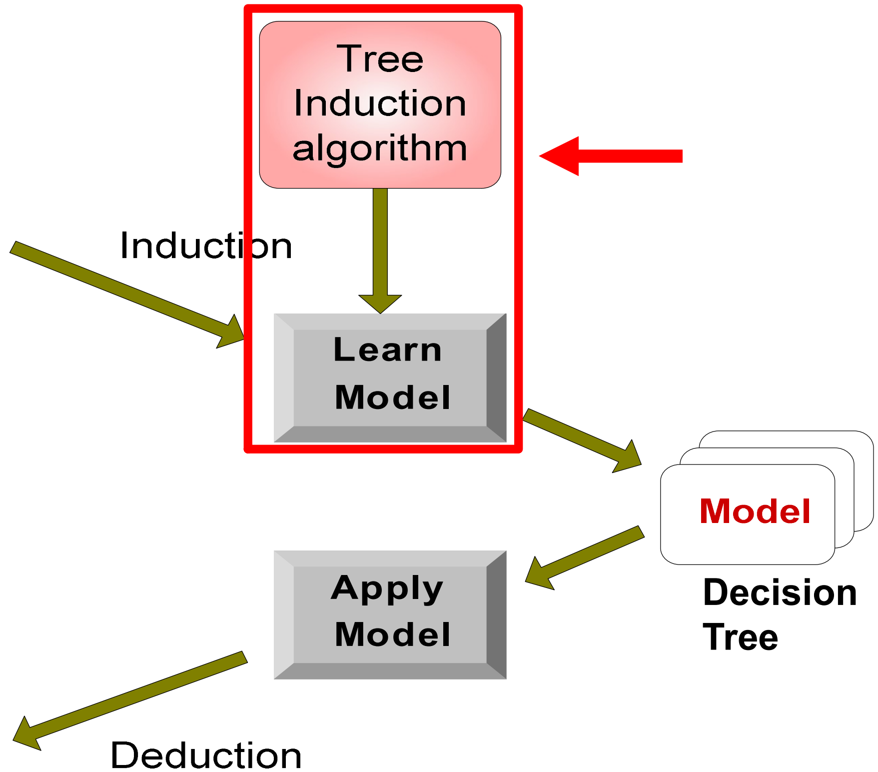
Induction

	X_1	X_2	X_3	y
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

	X_1'	X_2'	X_3'	y
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



Induction (with a Greedy Strategy)

- Tree is constructed in a top-down recursive divide-and-conquer manner
- At start, all the training instances are at the root
- Instances, i.e., from the training set, are then partitioned recursively based on selected attributes

Induction (with a Greedy Strategy)

Issues, discussed in the next slides:

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

Specifying Test Condition

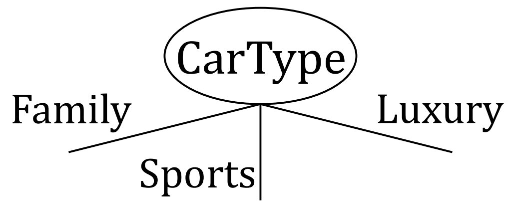
- Depends on attribute type
  - Nominal
  - Ordinal
  - Continuous

- Depends on number of ways to split
  - Binary split, i.e., 2-way
  - Multi-way split

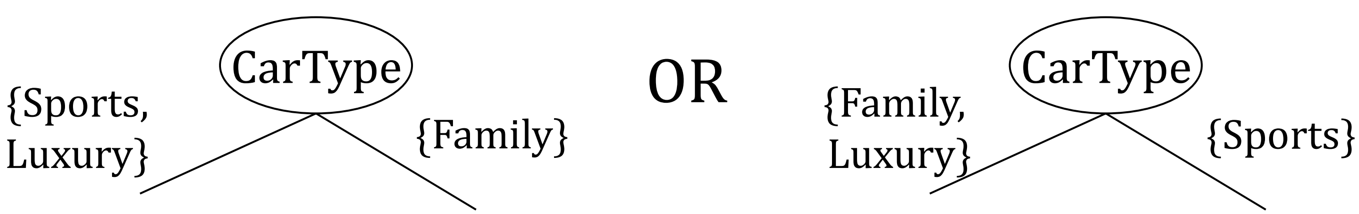
Splitting Based on Nominal Attributes

X_1	X_2	...	y
CarType			Class
Family			
Family			
Luxury			
Sports			
Sports			

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



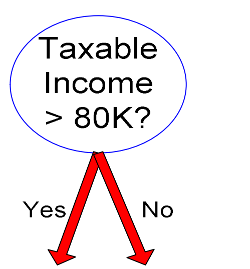
- Binary split:
  - Divides values into two subsets



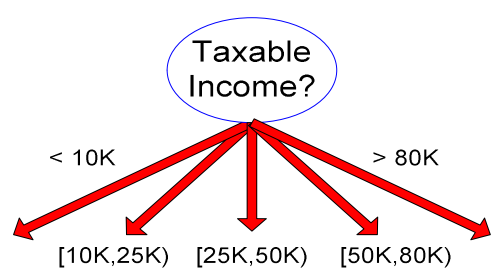
Splitting Based on Continuous Attributes

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 (i.e., $X_i < v$ or $X_i \geq v$)
  - Finds the best cut among all possible splits
  - Can be more compute intensive



(i) Binary split



(ii) Multi-way split

X_1	...	X_n	y
		Income	Class
		10K	
		20K	
		23K	
		100K	
		25K	

Recursive Partitioning

X_1	…	X_i	…	y
				Class

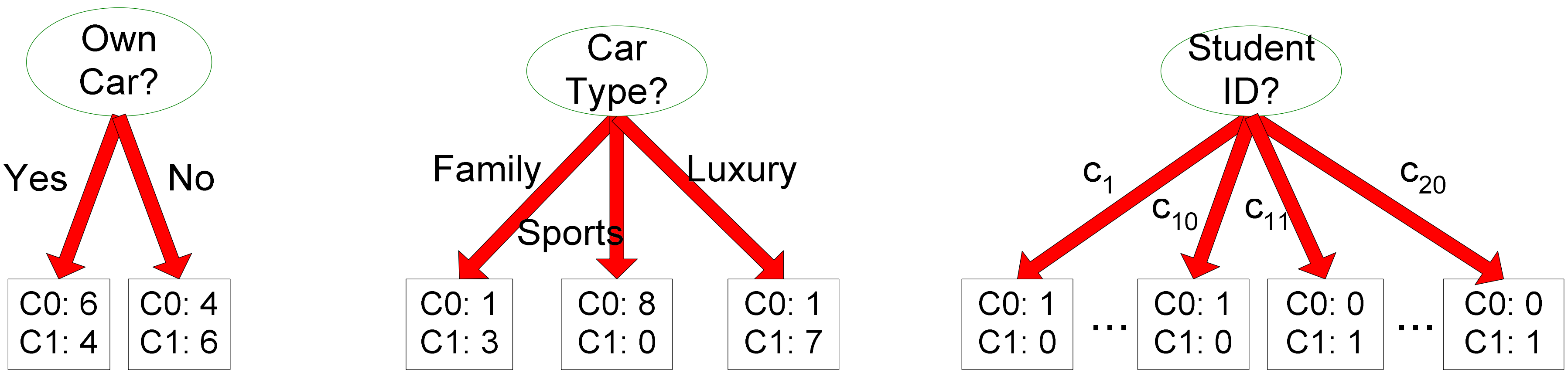
Algorithm examines:

1. Each predicator variable
   - I.e., X_1, X_2, …
2. All possible split values
   - I.e., each value in Xi

Determining the Best Split

X_1	X_2	X_1	y
Own Car	Car type	Student ID	Class

- Before Splitting: 10 records of class $C0$ and 10 records of class $C1$



→ Which test condition is the best?

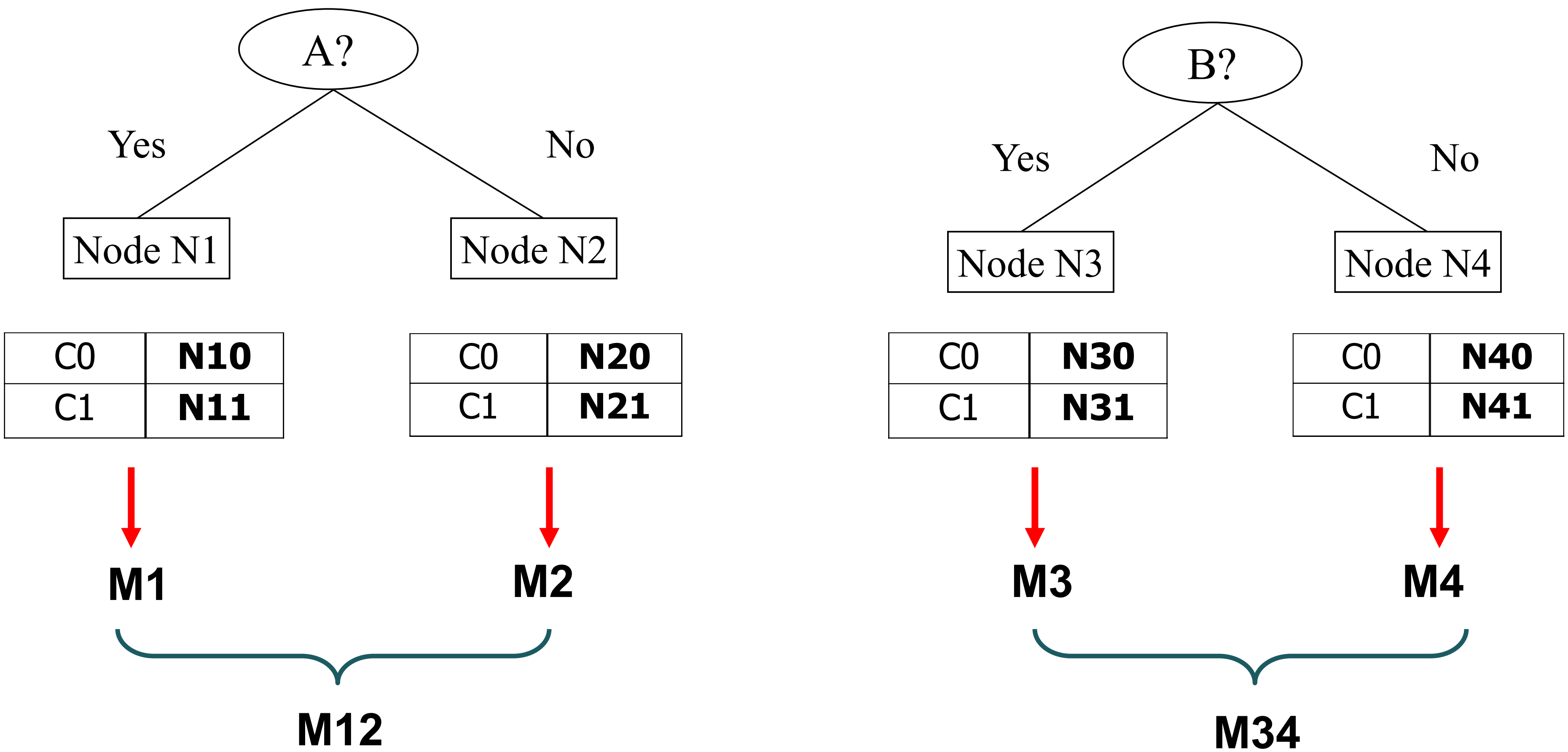
Strategy:

- Split the records based on an attribute test that optimizes certain criterion → Information Gain
- Need a measure of node impurity → Gini Index, Entropy, etc.

Information Gain

Before Splitting: → **M0**

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



Gain = M0-M12 vs. M0-M34

Gini Index

$$\text{GINI(node)} = 1 - \sum_{k=1}^{m} (pk)^2$$

We denote:

- The m classes of the response variable by $\mathrm{k}=1, 2, \ldots, \mathrm{m}$
- The proportion of instances that belong to class k as $\mathrm{p_k}$

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

$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$

$\text{GINI(a)} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$

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

$P(C1) = 1/6 \qquad P(C2) = 5/6$

$\text{GINI(b)} = 1 - (1/6)^2 - (5/6)^2 = 0.278$

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

$P(C1) = 2/6 \qquad P(C2) = 4/6$

$\text{GINI(c)} = 1 - (2/6)^2 - (4/6)^2 = 0.444$

Entropy measure

$$\text{entropy(node)} = -\sum_{k=1}^{m} p_k \log_2(p_k)$$

We denote:

- The m classes of the response variable by $k=1, 2, ..., m$
- The proportion of instances that belong to class k as p_k

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

$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$

$\text{Entropy(a)} = -\, 0 \log_2(0) - 1 \log_2(1) = -\, 0 - 0 = 0$

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

$P(C1) = 1/6 \qquad P(C2) = 5/6$

$\text{Entropy(b)} = -\,(1/6) \log_2(1/6) - (5/6) \log_2(5/6) = 0.65$

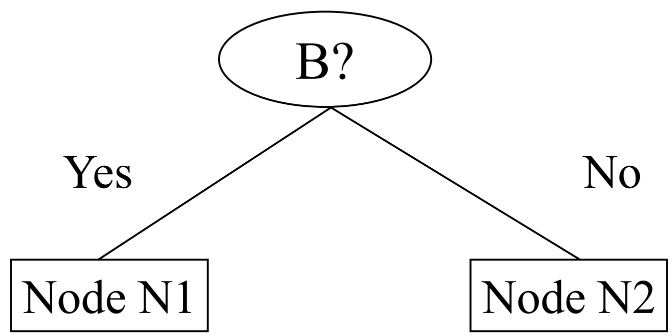
C1	**2**
C2	**4**

$P(C1) = 2/6 \qquad P(C2) = 4/6$

$\text{Entropy(b)} = -\,(2/6) \log_2(2/6) - (4/6) \log_2(4/6) = 0.92$

Combined impurity

- The combined impurity created by a split is a weighted average of the impurity measures
- Weighted by the number of records in each split



	Parent
C1	**6**
C2	**6**
Gini = 0.500	

Gini(N1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.41$

Gini(N2)
$= 1 - (1/5)^2 - (4/5)^2$
$= 0.32$

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.37		

Gini(Children)
$= 7/12 * 0.41 +$
$5/12 * 0.32$
$= 0.37$

➔ GINI impurity index decreases from 0.5 before the split to 0.37 after the split

Categorical Attributes

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

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

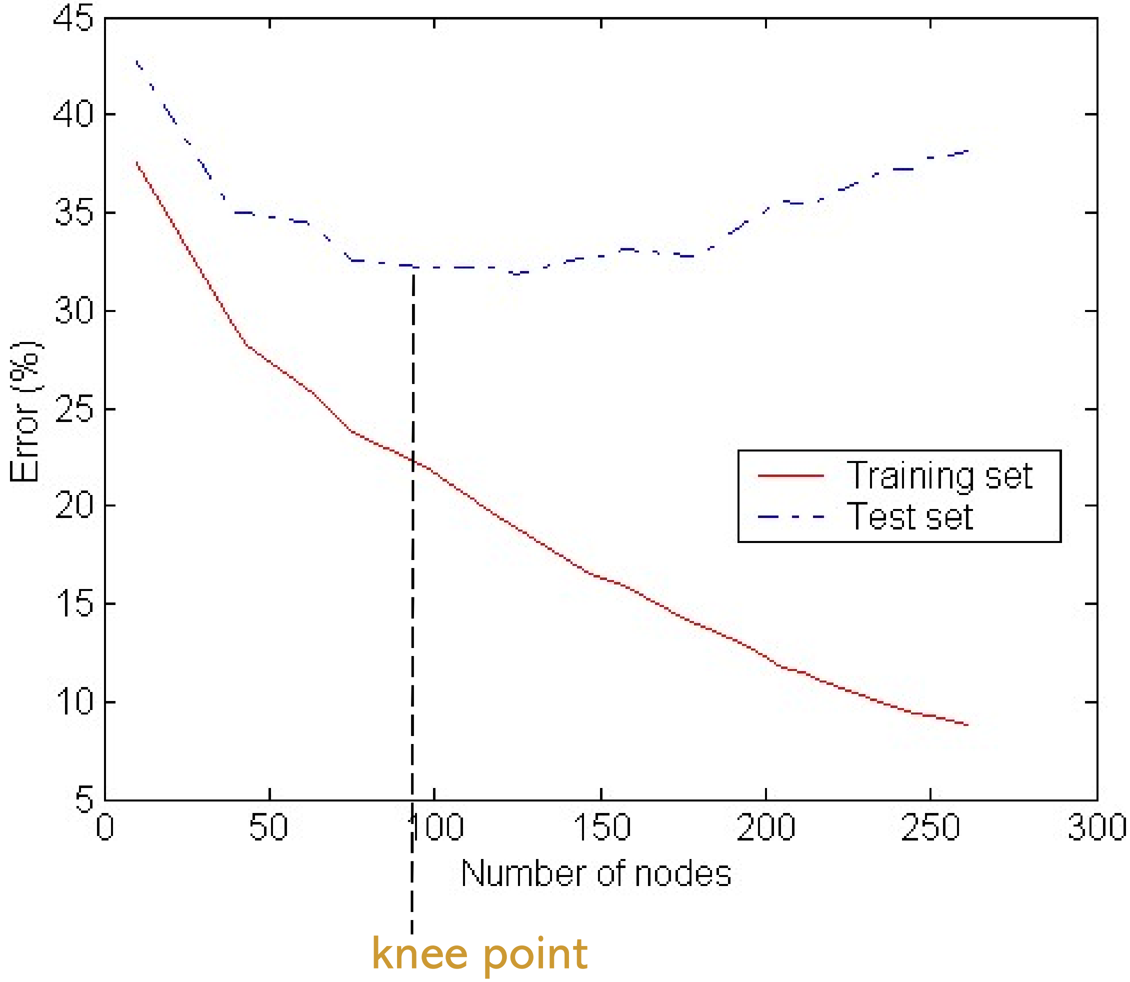
Stopping Criteria for Tree Induction

- 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)

Overfitting



Underfitting: when model is too simple, both training and test errors are large

How to Address Overfitting

Pre-Pruning

- Stop the algorithm before it becomes a fully-grown tree
- Typical stopping conditions for a node:
  - Stop if all instances belong to the same class
  - Stop if all the attribute values are the same
- More restrictive conditions:
  - Stop if number of instances is less than some user-specified threshold
  - Stop if expanding the current node does not improve impurity measures, e.g., Gini or information gain

How to Address Overfitting…

Post-pruning

- Grow decision tree to its entirety
- Trim the nodes of the decision tree in a bottom-up fashion
- 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

Proc and Cons

- Advantages:
  - Easy to understand (domain experts love them!)
  - Easy to generate rules
- Disadvantages:
  - May suffer from overfitting
  - Classifies by rectangular partitioning (so does not handle correlated features very well)
  - Can be quite large – pruning is necessary
  - Does not handle streaming data easily
    - … but a few successful ideas/techniques exist