

# CSE4334/5334

# DATA MINING

## Lecture 4: Classification (1)

CSE4334/5334 Data Mining, Fall 2014

Department of Computer Science and Engineering, University of Texas at Arlington

Chengkai Li (Slides courtesy of Vipin Kumar)

# Classification: Definition

- Given a collection of records (*training set*)
  - ▣ Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - ▣ A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

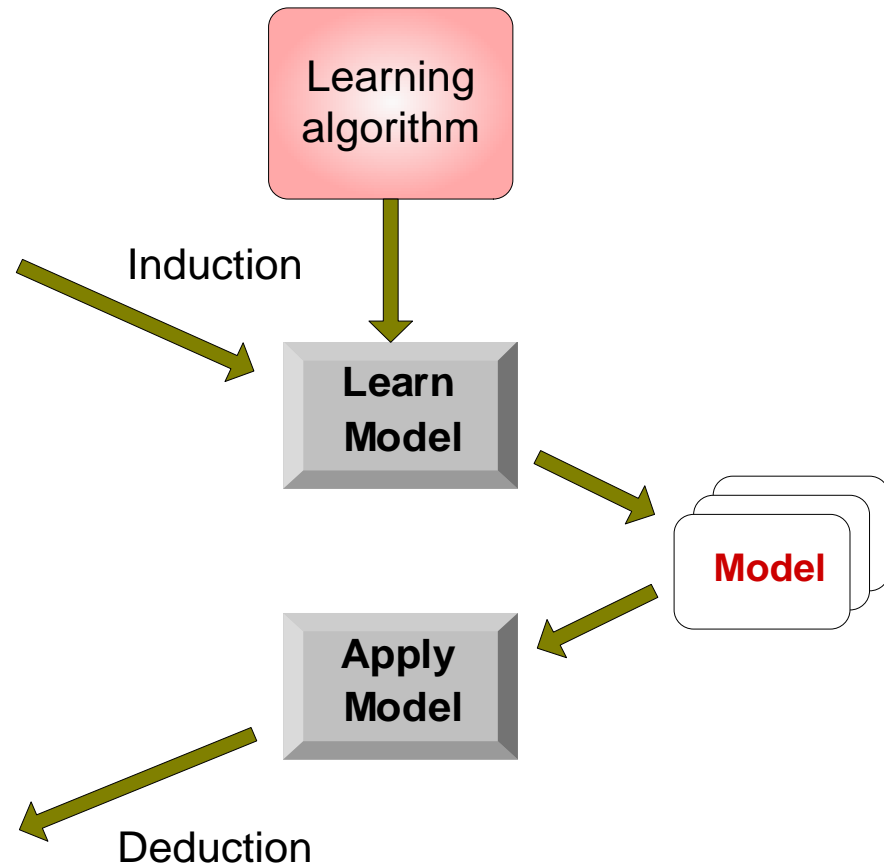
# Illustrating Classification Task

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 Set

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

Test Set



# Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Categorizing news stories as finance, weather, entertainment, sports, etc.
- **Give me more Examples.**



# Classification vs. Prediction

## □ Classification

- predicts categorical class labels
- Most suited for nominal attributes [ex: Gender, Color]
- Less effective for ordinal attributes [ex: temperature, rank]

## □ Prediction

- models continuous-valued functions or ordinal attributes, i.e., predicts unknown or missing values
- e.g., Linear regression

# Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**
  - ▣ Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - ▣ New data is classified based on the training set
- **Unsupervised learning (clustering)**
  - ▣ The class labels of training data is unknown
  - ▣ Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

# Classification Techniques

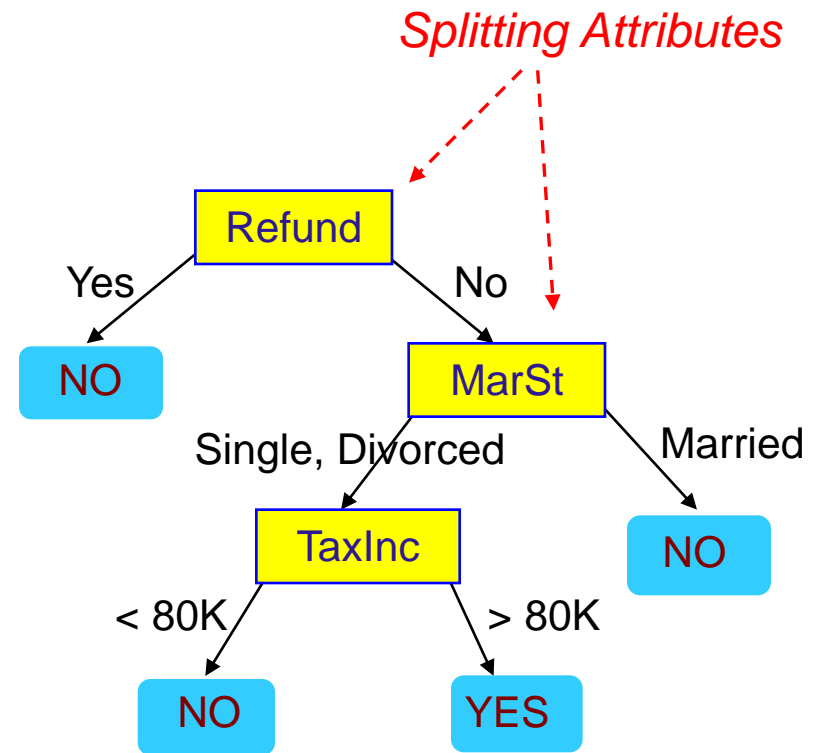
- Decision Tree based Methods
- Rule-based Methods
- Nearest-Neighbor Classifiers
- Naïve Bayes Classifiers and Bayesian Belief Networks
- Neural Networks
- Support Vector Machines

# Example of a 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



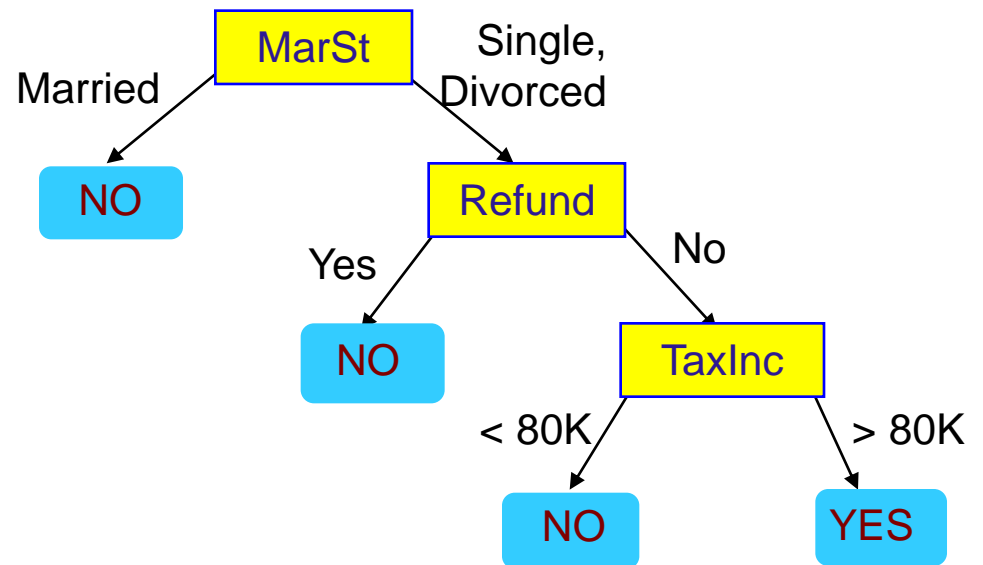
Model: Decision Tree



# Another Example of Decision Tree

categorical      categorical      continuous  
class

<i>Tid</i>	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



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

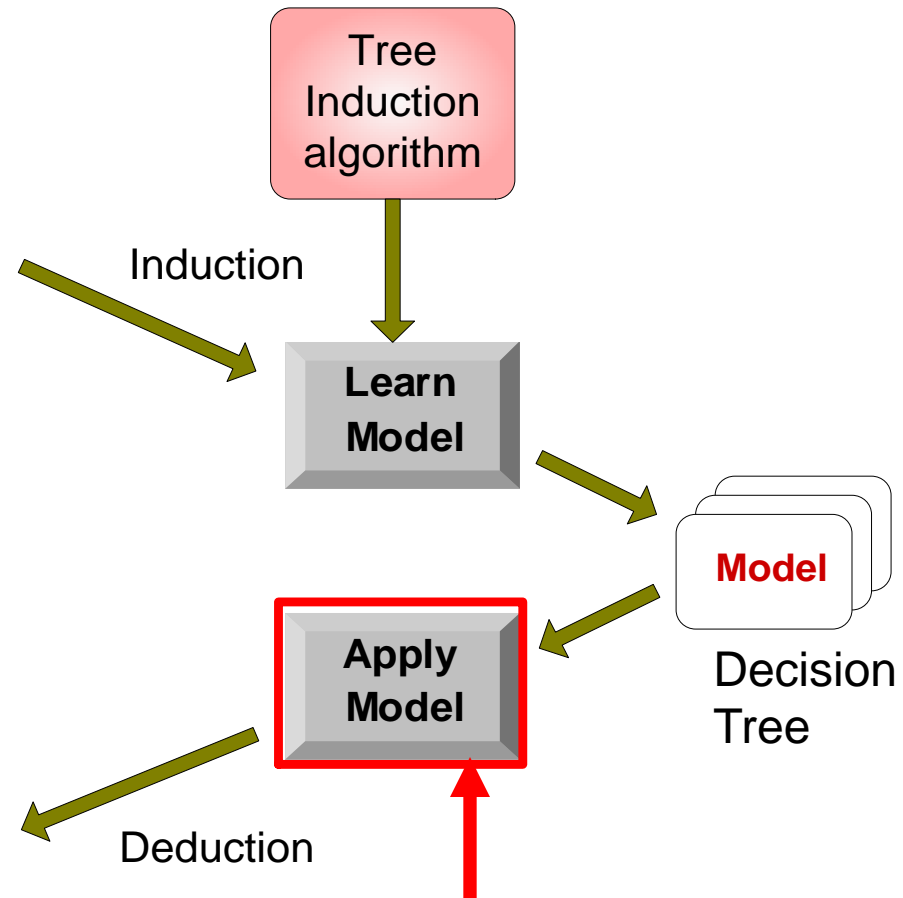
# Decision Tree Classification Task

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 Set

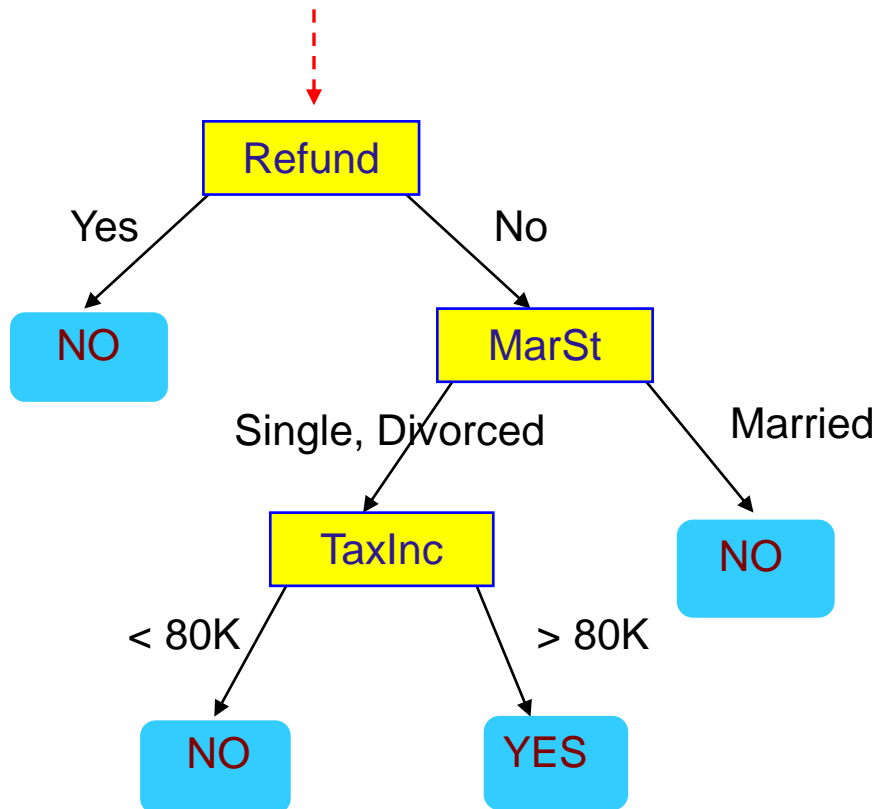
Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

Test Set



# Apply Model to Test Data

Start from the root of tree.



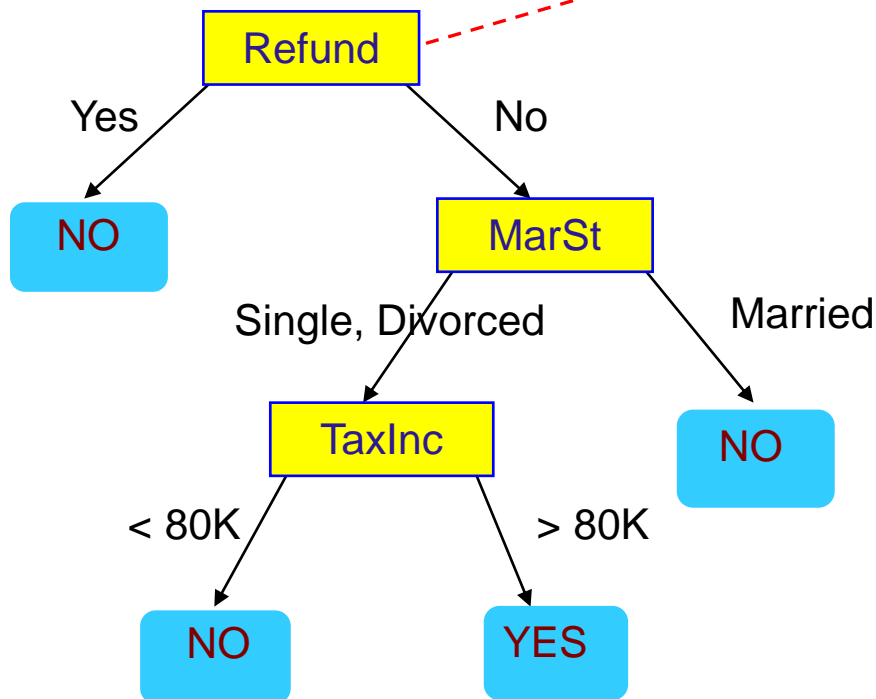
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

# Apply Model to Test Data

Test Data

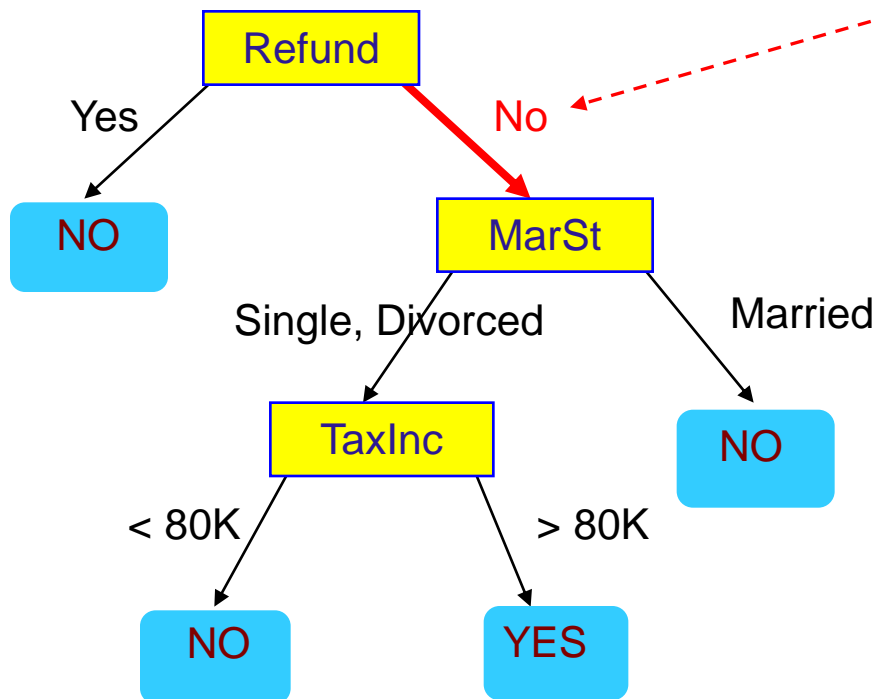
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

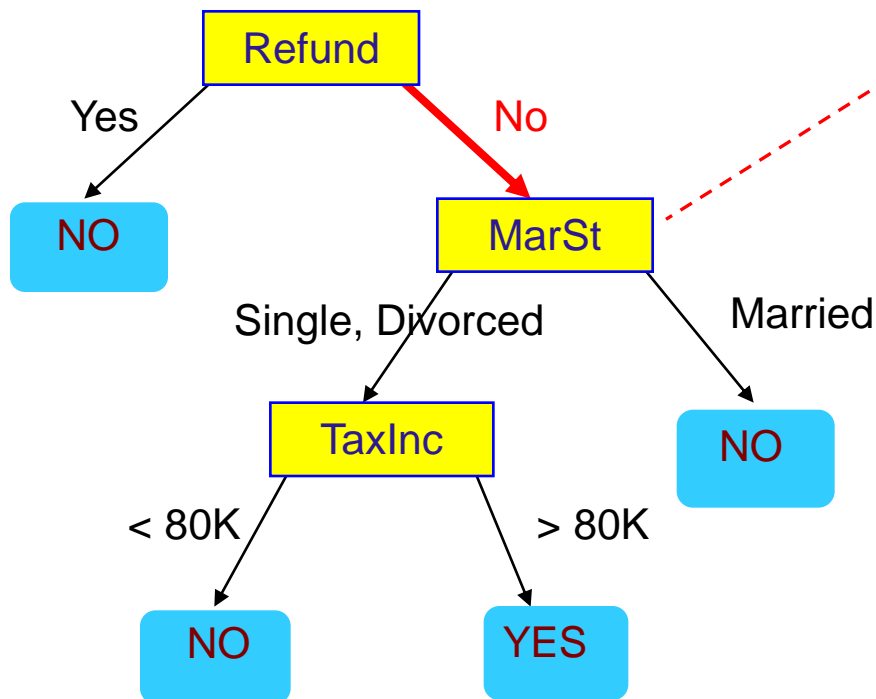
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

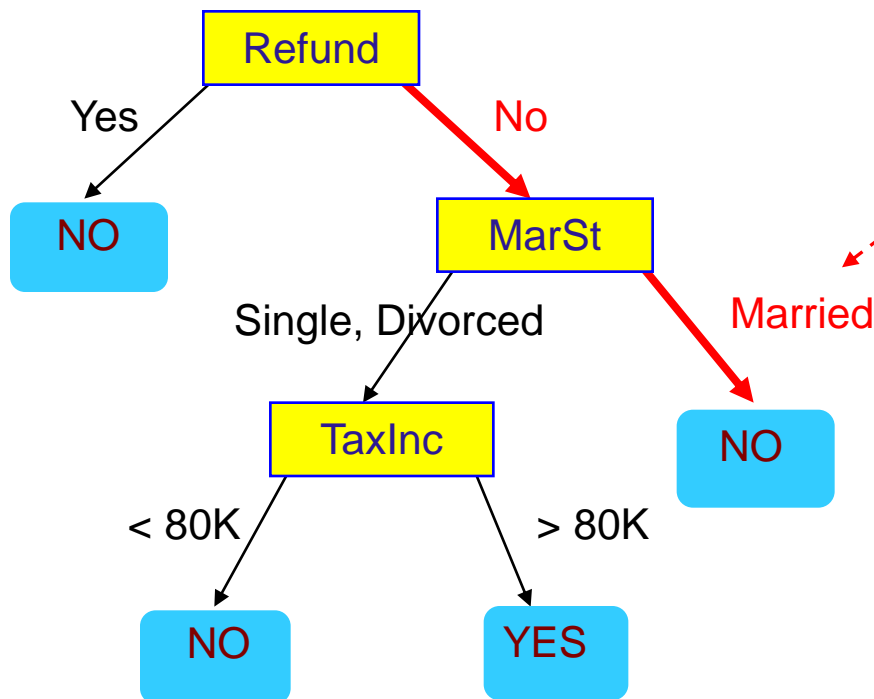
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

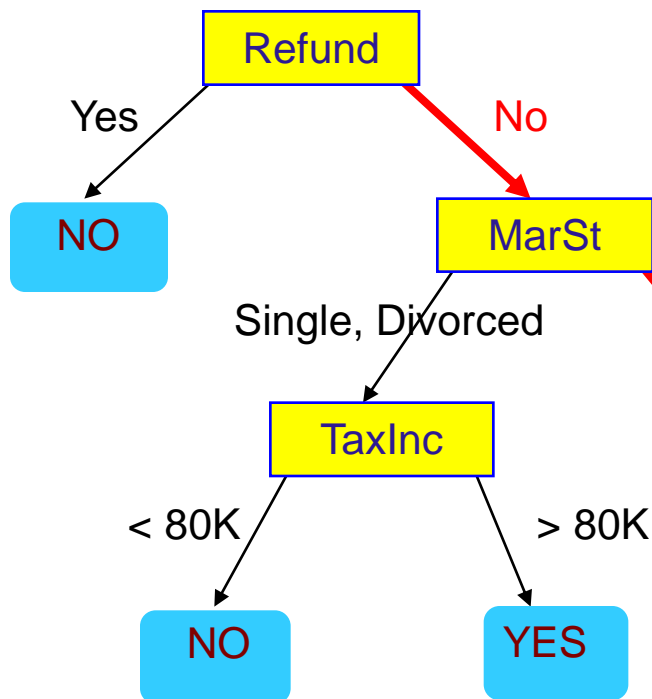
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

## Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"



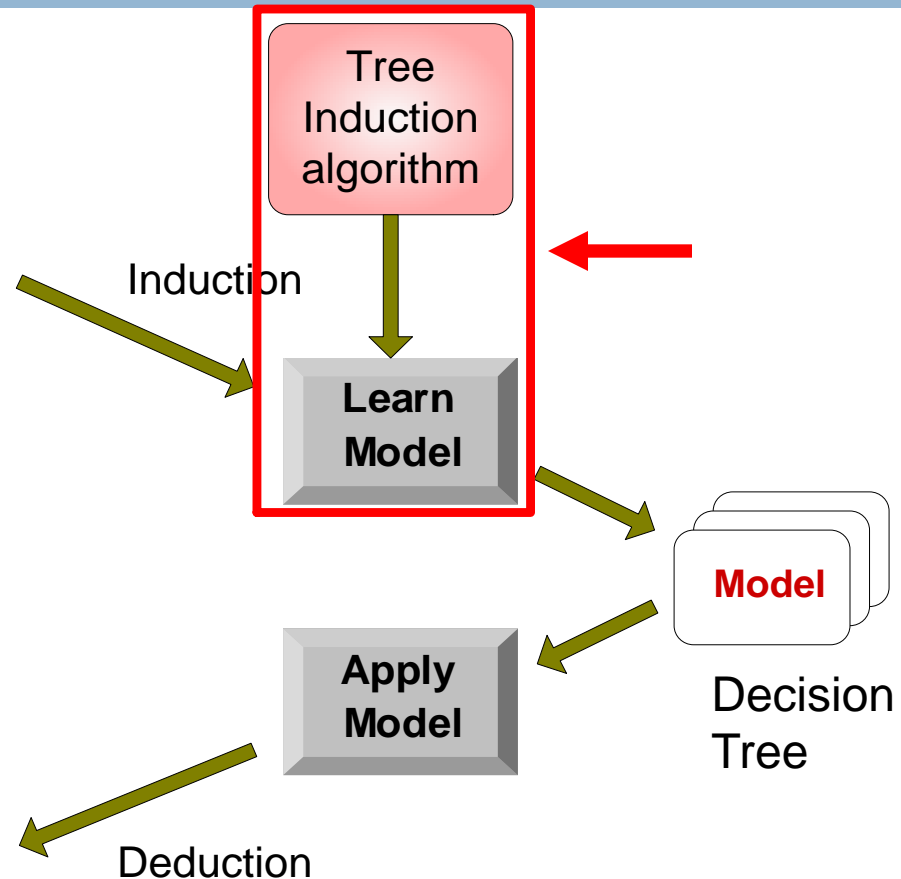
# Decision Tree Classification Task

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 Set

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

Test Set



# Decision Tree Induction

- Large search space
  - ▣ Exponential number of trees, with respect to the set of attributes.
  - ▣ Finding the optimal decision tree is computationally infeasible
- Efficient algorithm for accurate suboptimal decision tree
  - ▣ Greedy strategy
  - ▣ Grow the tree by making locally optimally decisions in selecting the attributes

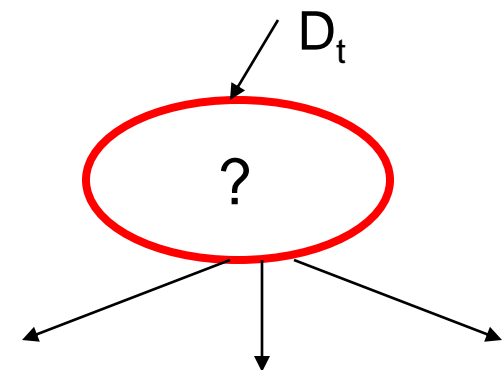
# Decision Tree Induction

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

# General Structure of Hunt's Algorithm

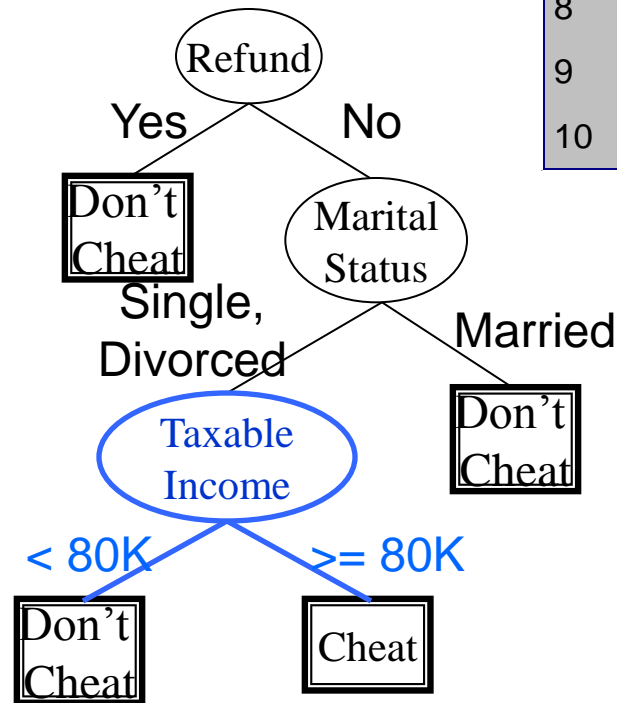
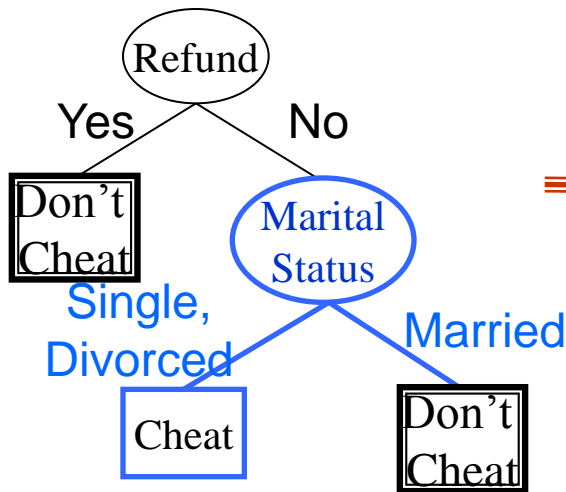
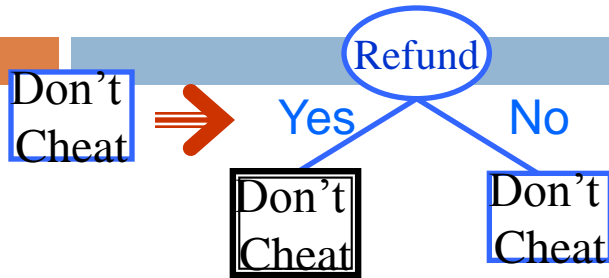
- Let  $D_t$  be the set of training records that reach a node  $t$
- General Procedure:
  - ▣ If  $D_t$  contains records that belong the same class  $y_t$ , then  $t$  is a leaf node labeled as  $y_t$
  - ▣ If  $D_t$  is an empty set, then  $t$  is a leaf node labeled by the default class,  $y_d$
  - ▣ 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 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

<i>Tid</i>	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.
  
- 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.
  
- 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

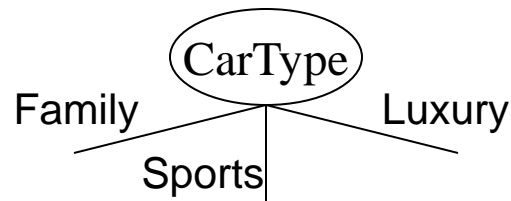
# How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
  
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

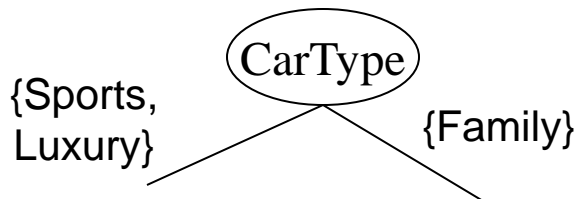


# Splitting Based on Nominal Attributes

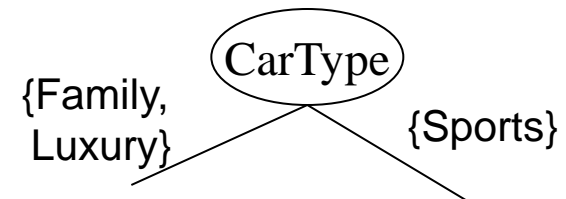
- **Multi-way split:** Use as many partitions as distinct values.



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

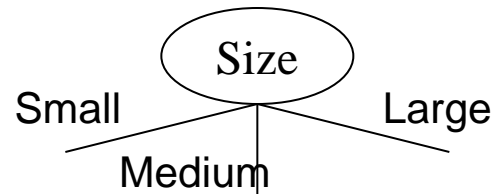


OR

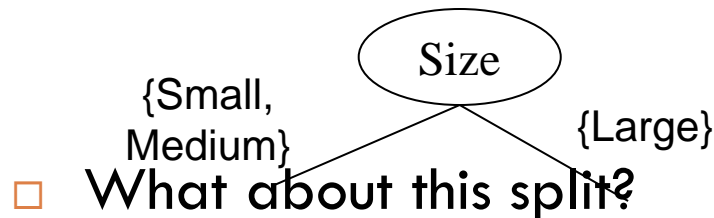


# Splitting Based on Ordinal Attributes

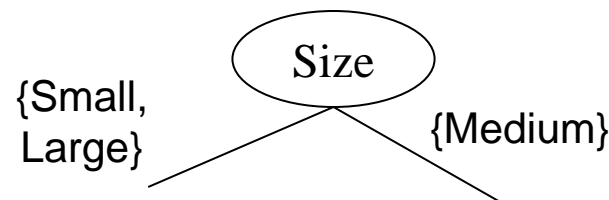
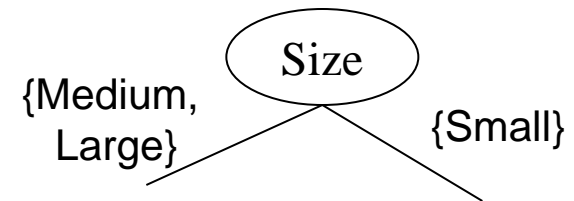
- **Multi-way split:** Use as many partitions as distinct values.



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



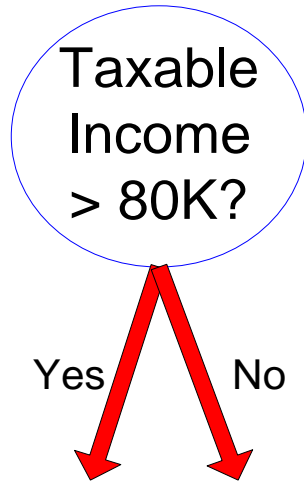
OR



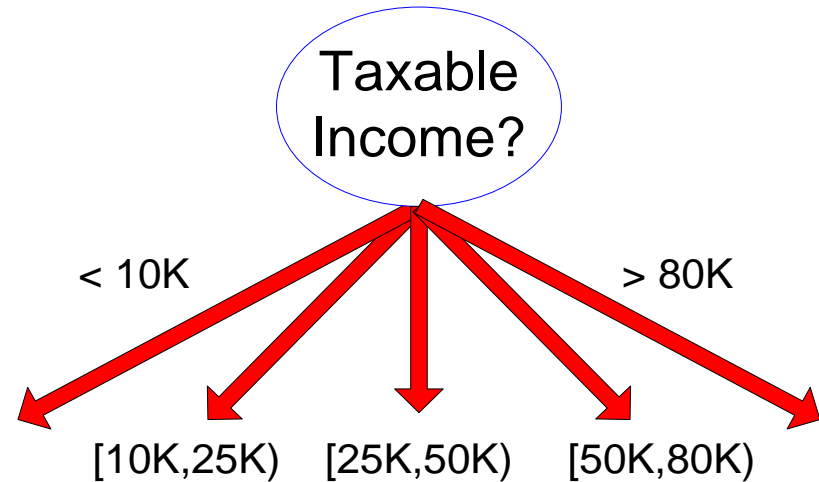
# Splitting Based on Continuous Attributes

- 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 computational intensive

# Splitting Based on Continuous Attributes



(i) Binary split



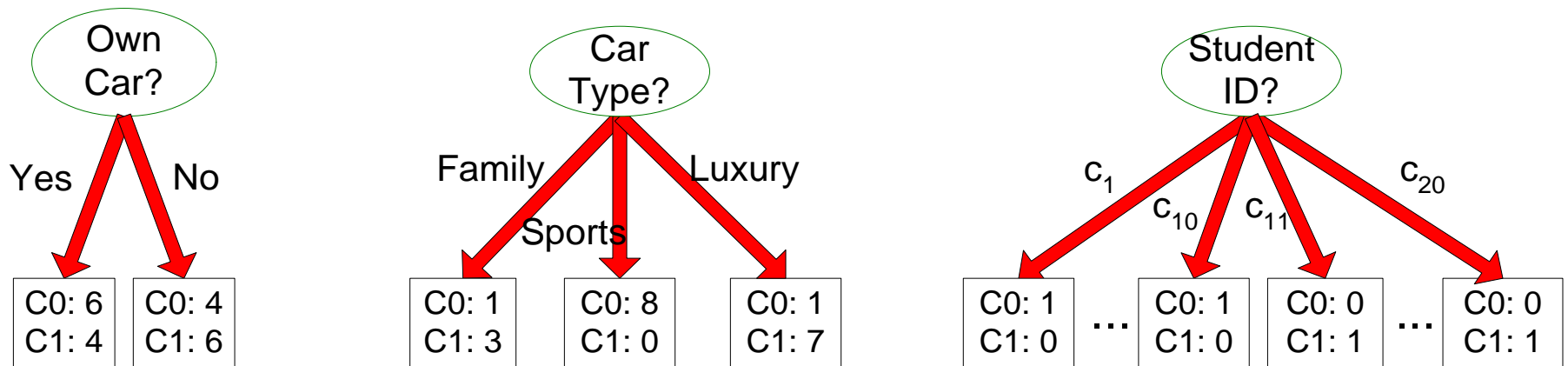
(ii) Multi-way split

# 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

# 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

# Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error



# How to Find the Best Split

Before Splitting:

C0	<b>N00</b>
C1	<b>N01</b>

→ M0

A?

Yes

No

Node N1

Node N2

C0	<b>N10</b>
C1	<b>N11</b>

C0	<b>N20</b>
C1	<b>N21</b>

M1

M2

M12

gain  
(Information gain, if Entropy is used as M)

M0 – M12 vs M0 – M34

B?

Yes

No

Node N3

Node N4

C0	<b>N30</b>
C1	<b>N31</b>

C0	<b>N40</b>
C1	<b>N41</b>

M3

M4

M34

# Measure of Impurity: GINI

- Gini Index for a given node  $t$  :

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

(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

C1	<b>0</b>
C2	<b>6</b>
<b>Gini=0.000</b>	

C1	<b>1</b>
C2	<b>5</b>
<b>Gini=0.278</b>	

C1	<b>2</b>
C2	<b>4</b>
<b>Gini=0.444</b>	

C1	<b>3</b>
C2	<b>3</b>
<b>Gini=0.500</b>	

# Examples for computing GINI

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

C1	<b>0</b>
C2	<b>6</b>

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

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	<b>1</b>
C2	<b>5</b>

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

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

C1	<b>2</b>
C2	<b>4</b>

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

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

# 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  $p$ .

# How to Find the Best Split

Before Splitting:

C0	<b>N00</b>
C1	<b>N01</b>

→ M0

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

A?

B?

Yes

No

Yes

No

Node N1

Node N2

Node N3

Node N4

C0	<b>N10</b>
C1	<b>N11</b>

C0	<b>N20</b>
C1	<b>N21</b>

C0	<b>N30</b>
C1	<b>N31</b>

C0	<b>N40</b>
C1	<b>N41</b>

M1

M2

M3

M4

M12

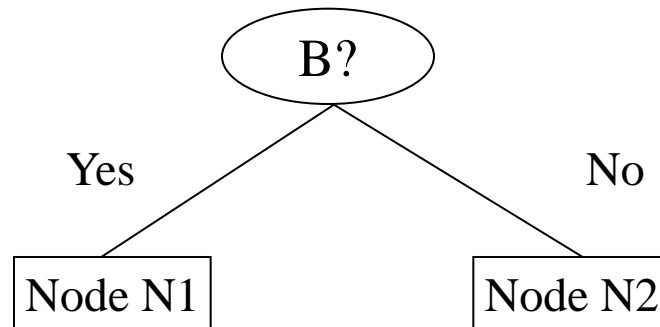
gain  
(Information gain, if Entropy is used as M)

M34

M0 – M12 vs M0 – M34

# Binary Attributes: Computing GINI Index

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



	<b>Parent</b>
C1	<b>6</b>
C2	<b>6</b>
<b>Gini = 0.500</b>	

$$\begin{aligned} \text{Gini}(N1) &= 1 - (5/7)^2 - (2/7)^2 \\ &= 0.408 \end{aligned}$$

$$\begin{aligned} \text{Gini}(N2) &= 1 - (1/5)^2 - (4/5)^2 \\ &= 0.32 \end{aligned}$$

	<b>N1</b>	<b>N2</b>
C1	<b>5</b>	<b>1</b>
C2	<b>2</b>	<b>4</b>
<b>Gini=0.371</b>		

$$\begin{aligned} \text{Gini}(\text{Children}) &= 7/12 * 0.408 + \\ &\quad 5/12 * 0.32 \\ &= 0.371 \end{aligned}$$

# Categorical Attributes: Computing Gini Index

- 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	8	1
C2	3	0	7
Gini	<b>0.163</b>		

Two-way split  
(find best partition of values)

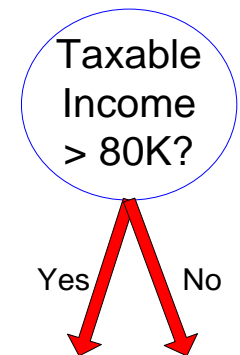
	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	<b>0.468</b>	

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	<b>0.167</b>	

# Continuous Attributes: Computing Gini Index

- Use Binary Decisions based on one value
- Several Choices for the splitting value
  - ▣ Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - ▣ Class counts in each of the partitions,  $A < v$  and  $A \geq v$
- Simple method to choose best  $v$ 
  - ▣ For each  $v$ , scan the database to gather count matrix and compute its Gini index
  - ▣ Computationally Inefficient! Repetition of work.

<i>Tid</i>	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





# Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
  - ▣ Sort the attribute on values
  - ▣ Linearly scan these values, each time updating the count matrix and computing gini index
  - ▣ Choose the split position that has the least gini index

Cheat		No	No	No	Yes	Yes	Yes	No	No	No	No											
		Taxable Income																				
Sorted Values		60	70	75	85	90	95	100	120	125	220											
Split Positions		55	65	72	80	87	92	97	110	122	172	230										
		<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>					
Yes		0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0			
No		0	7	1	6	2	5	3	4	3	4	3	4	4	3	5	2	6	1	7	0	
Gini		0.420		0.400		0.375		0.343		0.417		0.400		<u>0.300</u>		0.343		0.375		0.400		0.420

# Alternative Splitting Criteria based on INFO

- Entropy at a given node  $t$ :

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

(NOTE:  $p(j | t)$  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 least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

# Examples for computing Entropy

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

C1	<b>0</b>
C2	<b>6</b>

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

$$Entropy = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	<b>1</b>
C2	<b>5</b>

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

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

C1	<b>2</b>
C2	<b>4</b>

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

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

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

# Splitting Based on INFO...

## □ Gain Ratio:

$$\mathit{GainRATIO}_{split} = \frac{\mathit{GAIN}_{split}}{\mathit{SplitINFO}} \quad \mathit{SplitINFO} = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

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 Classification Error

- Classification error at a node  $t$  :

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

- 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

# Examples for Computing Error

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

C1	<b>0</b>
C2	<b>6</b>

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

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

C1	<b>1</b>
C2	<b>5</b>

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

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

C1	<b>2</b>
C2	<b>4</b>

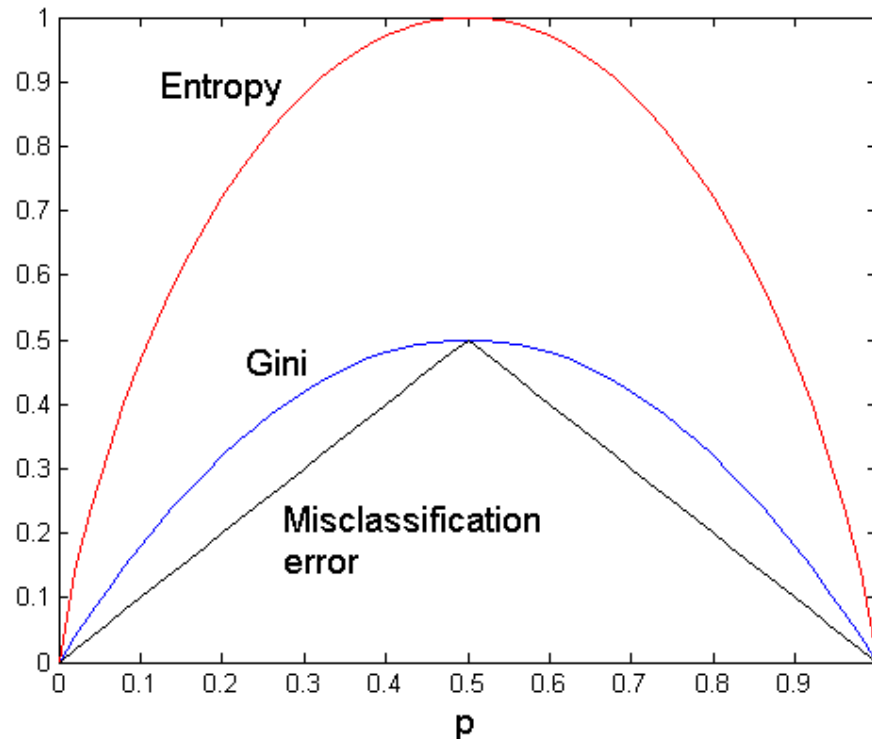
$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$Error = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

# Comparison among Splitting Criteria

For a 2-class problem:

( $p$  is the fraction of records belonging to one of the two classes.)





# 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 same (or similar) attribute values
  - ▣ What to do? majority voting
- Early termination (to be discussed later)

# Decision Tree Based Classification

- Advantages:
  - ▣ Inexpensive to construct
  - ▣ Extremely fast at classifying unknown records
  - ▣ Easy to interpret for small-sized trees
  - ▣ Accuracy is comparable to other classification techniques for many simple data sets

# 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.
  
- You can download the software from:  
<http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz>