

# Classification: Basic Concepts, Decision Trees, and Model Evaluation

TDT4300  
TAN Chapter 4



NTNU – Trondheim  
Norwegian University of  
Science and Technology

# 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.

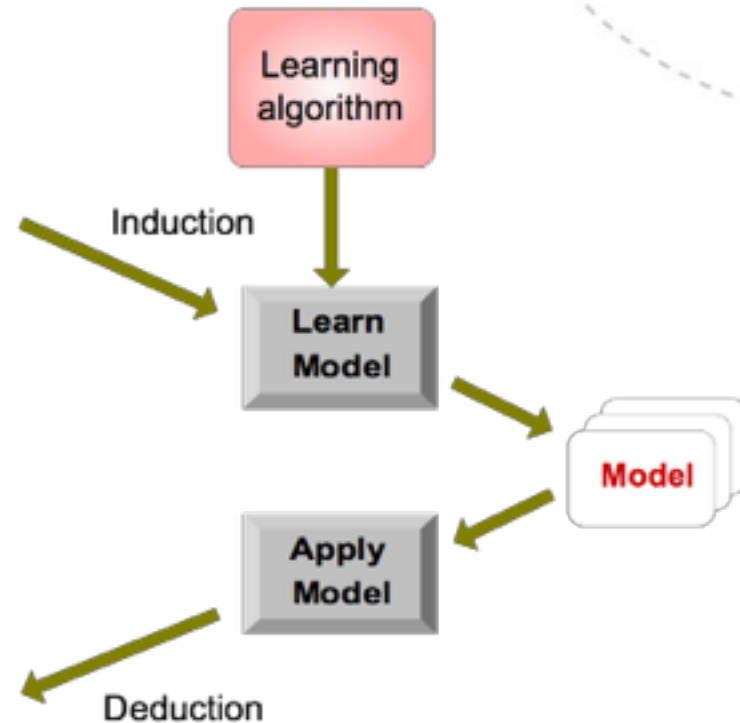
# Illustrating Classification Task

Tid	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

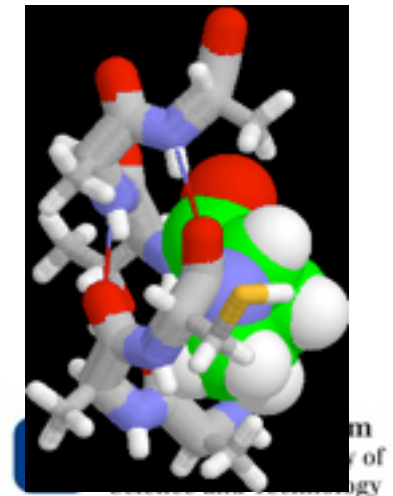
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



# Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc



# Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

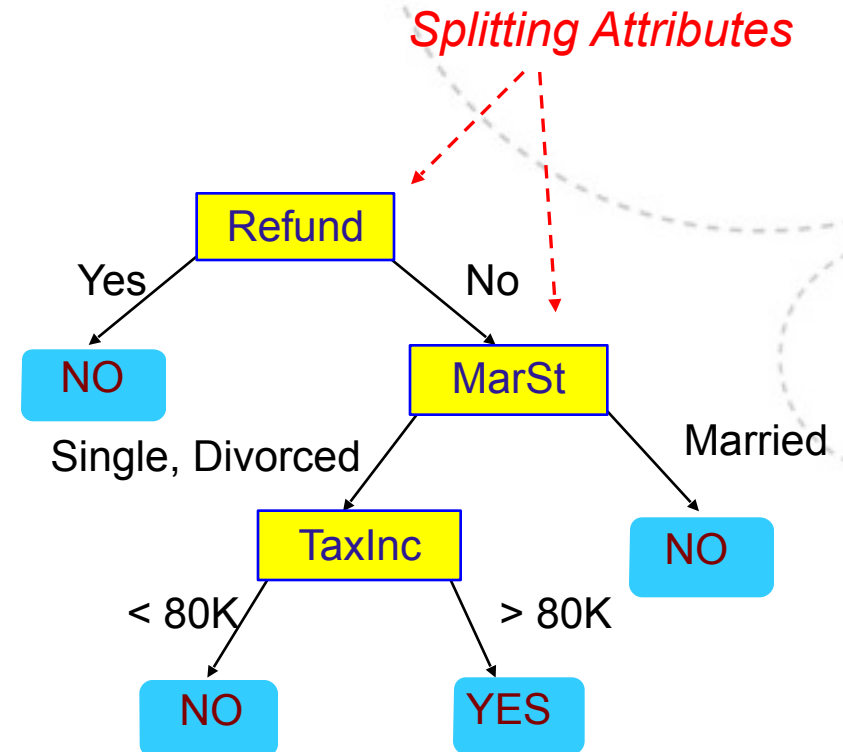


# Example of a Decision Tree

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

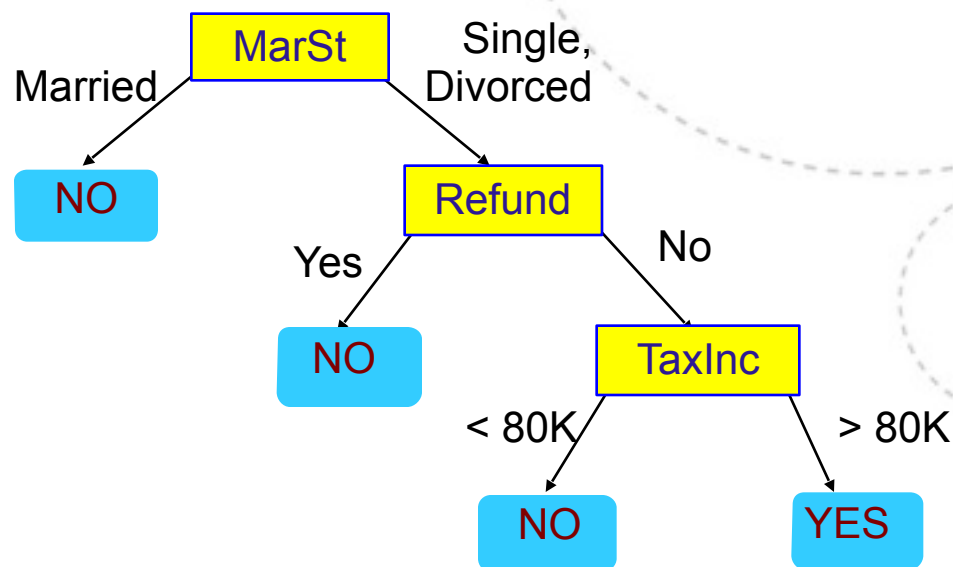


NTNU – Trondheim  
Norwegian University of  
Science and Technology

# Another Example of Decision Tree

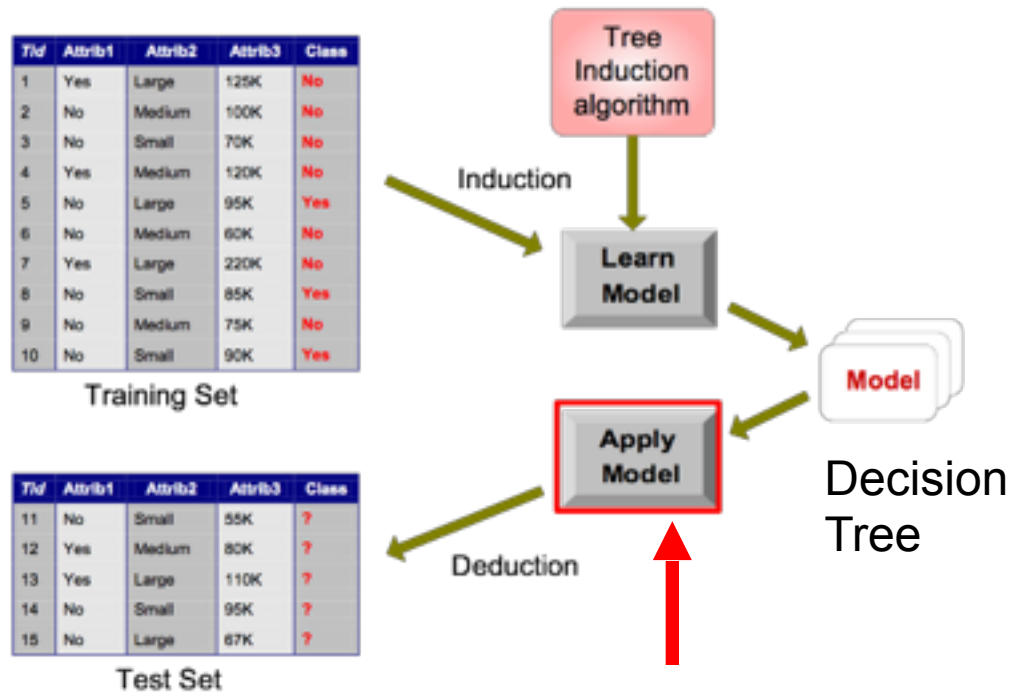
categorical      categorical      continuous      class

<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!

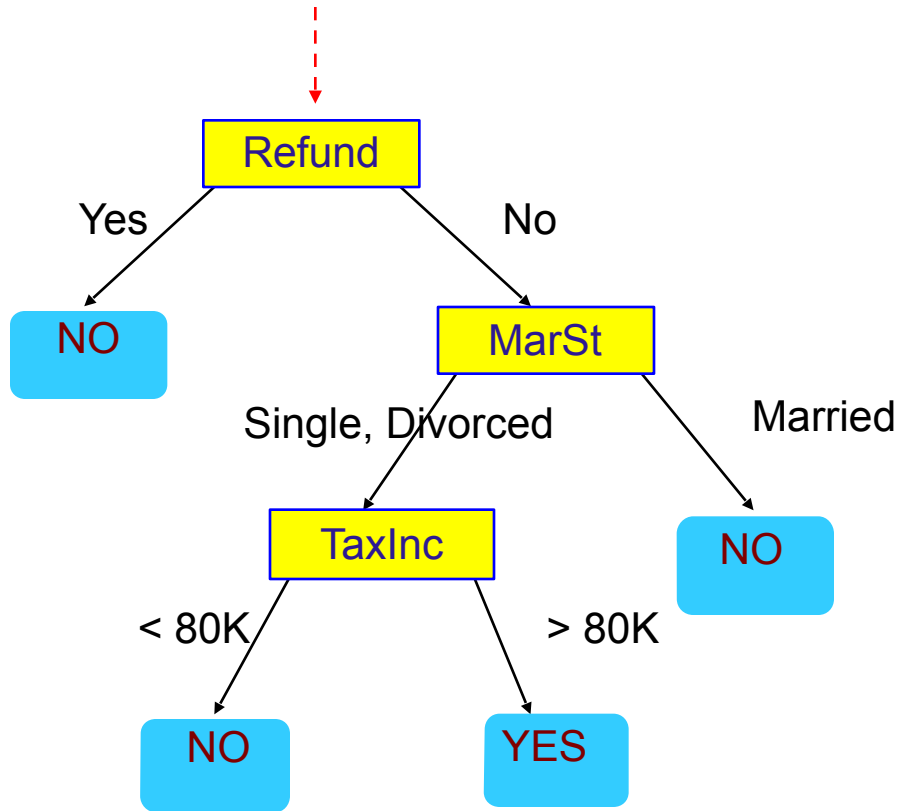
# Decision Tree Classification Task





# Apply Model to Test Data

Start from the root of tree.



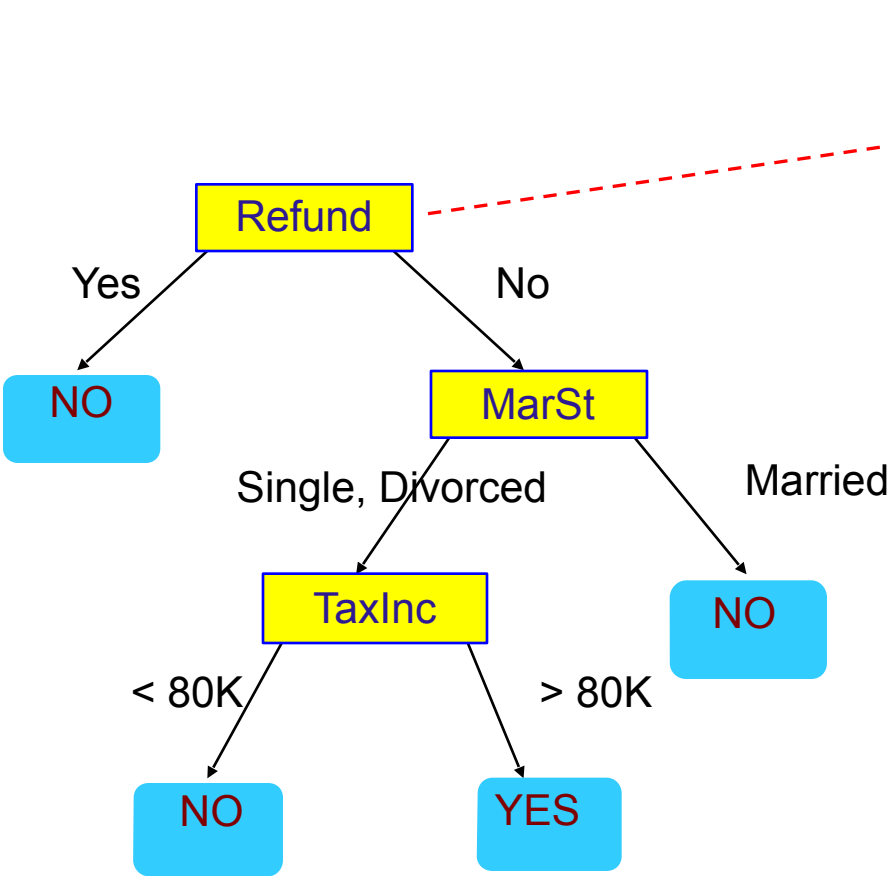
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



NTNU – Trondheim  
Norwegian University of  
Science and Technology

# Apply Model to Test Data



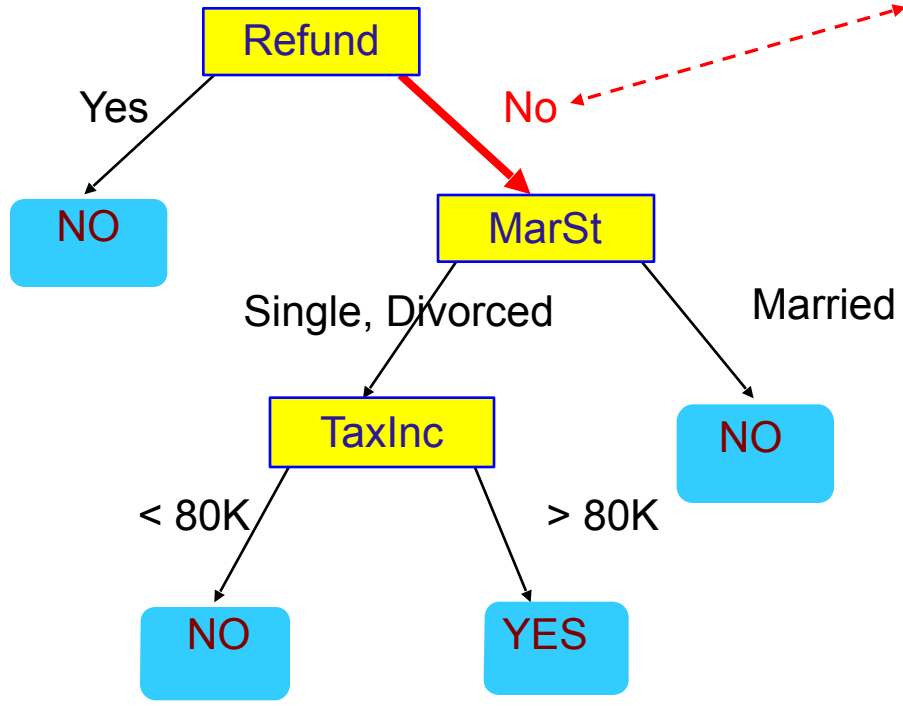
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

# Apply Model to Test Data

Test Data

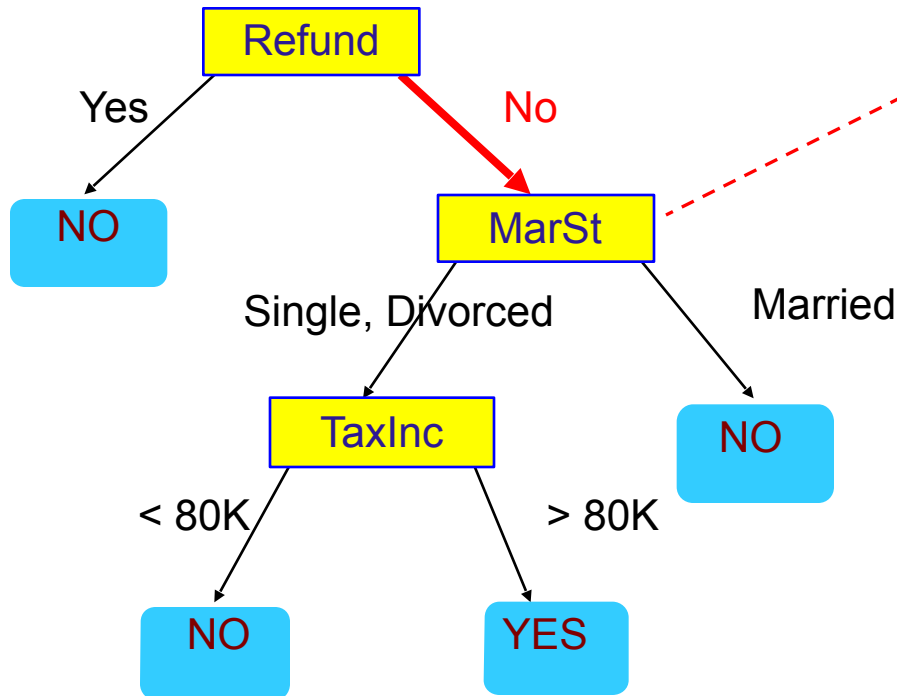
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

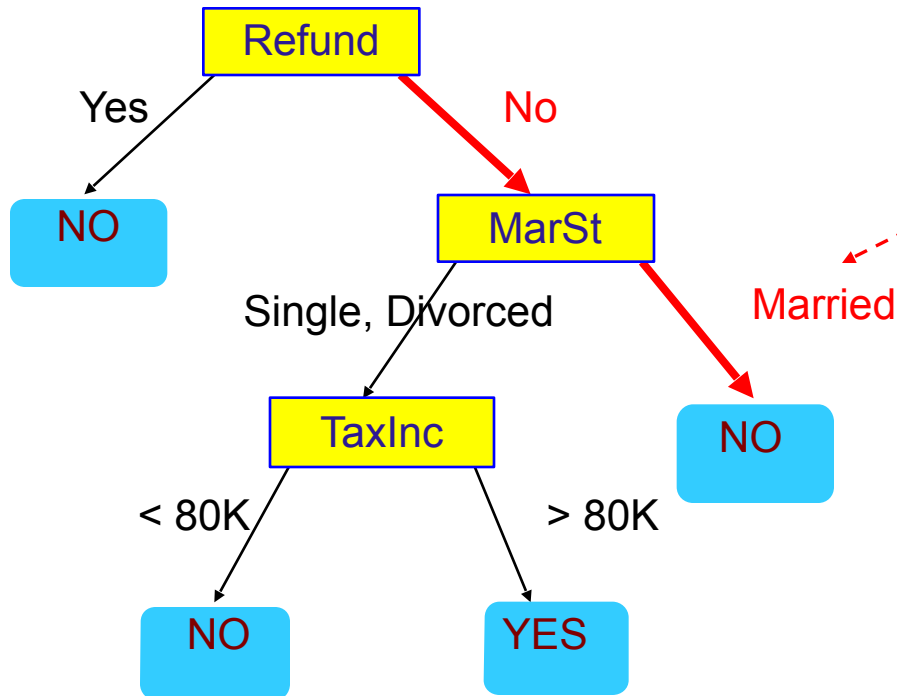
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

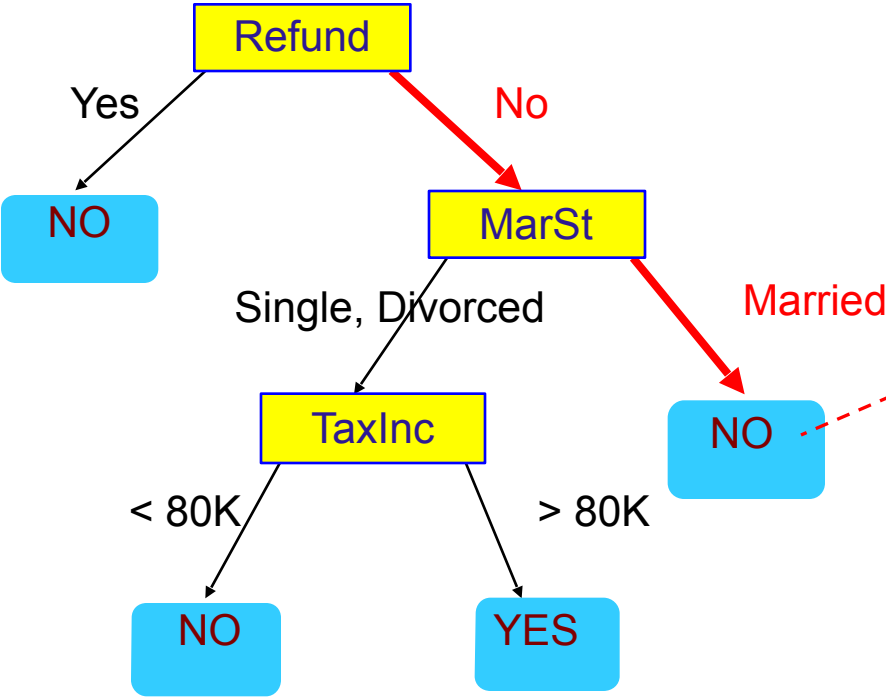
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

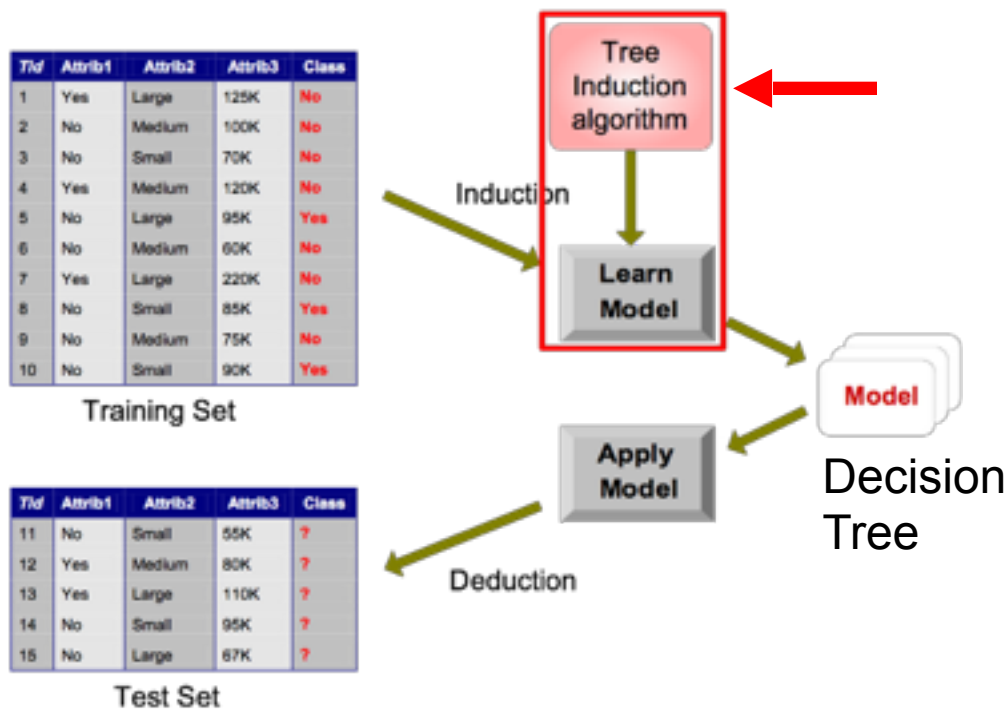
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

# Decision Tree Classification Task



# Decision Tree Induction

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

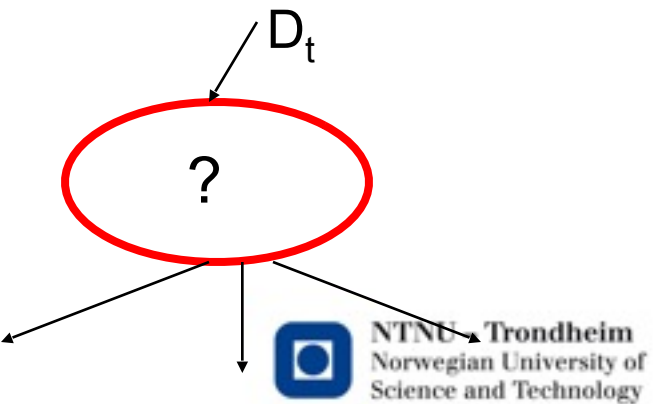




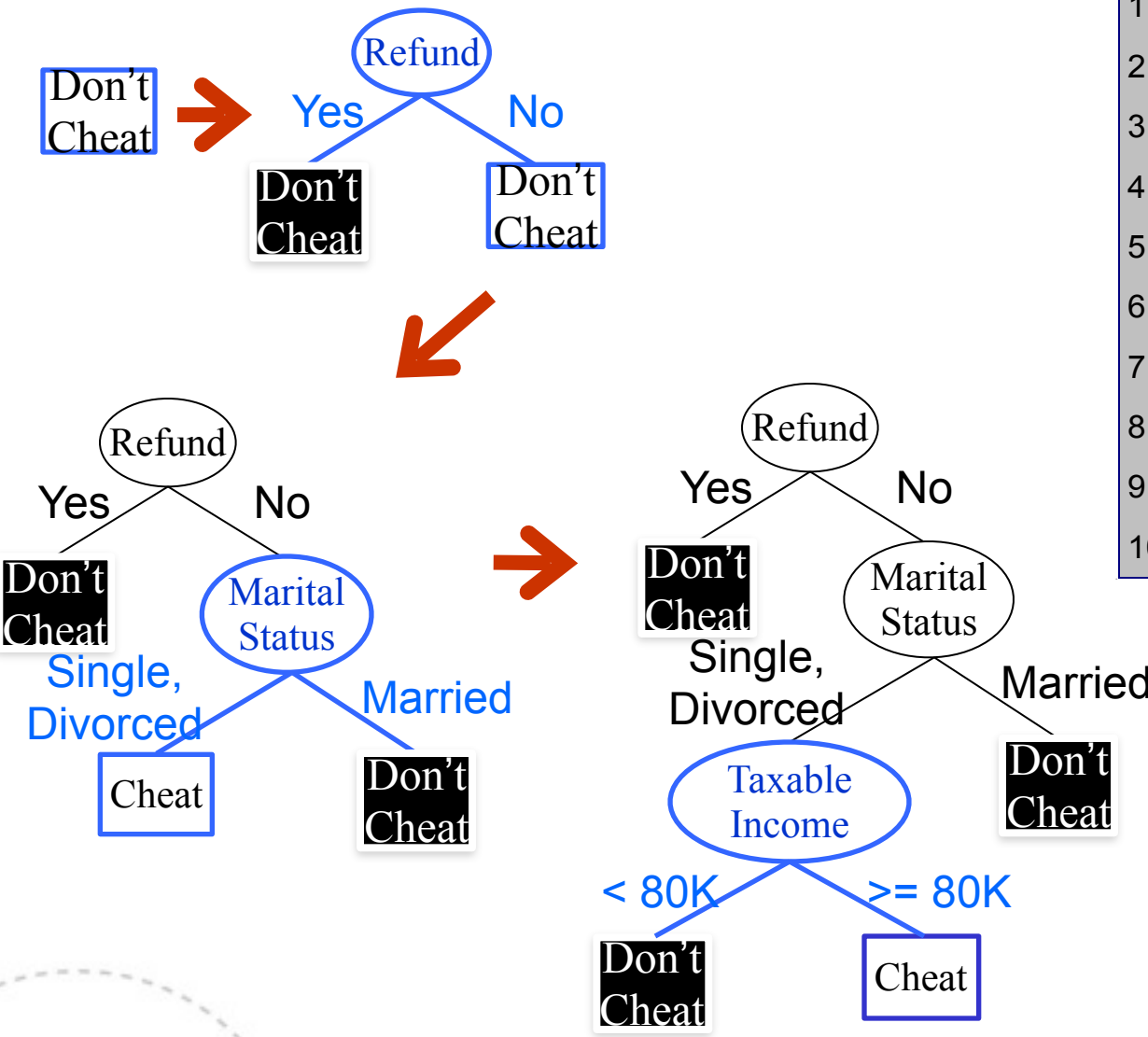
# General Structure of Hunt's Algorithm

- 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.

<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



# 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.
- 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



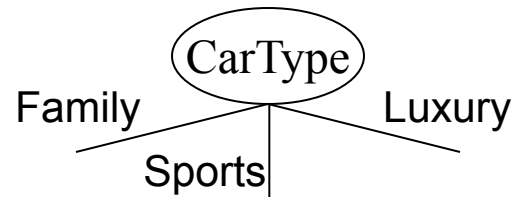
# How to Specify Test Condition?

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

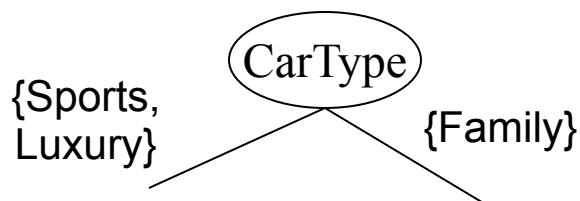


# Splitting Based on Nominal Attributes

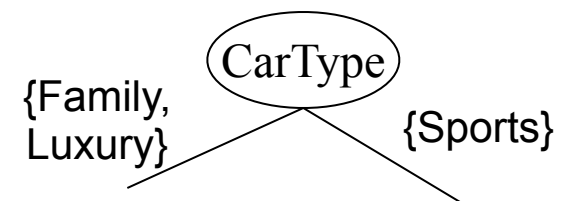
- **Multi-way split:** Use as many partitions as distinct values.



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

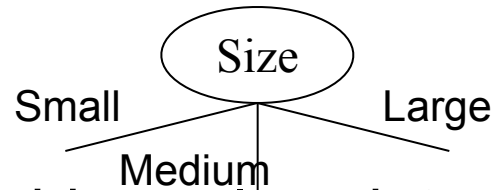


OR

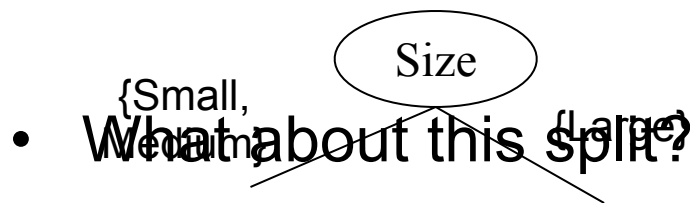


# Splitting Based on Ordinal Attributes

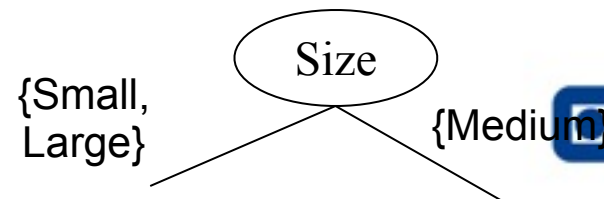
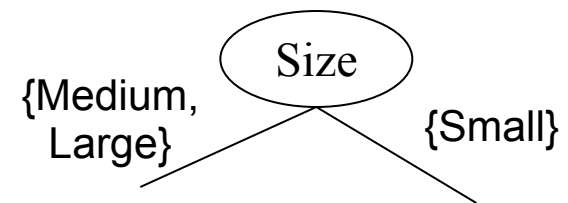
- **Multi-way split:** Use as many partitions as distinct values.



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



OR



NTNU – Trondheim  
Norwegian University of  
Science and Technology

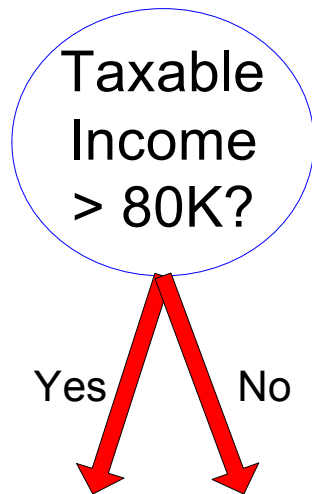
# Splitting Based on Continuous Attributes

- 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 compute intensive

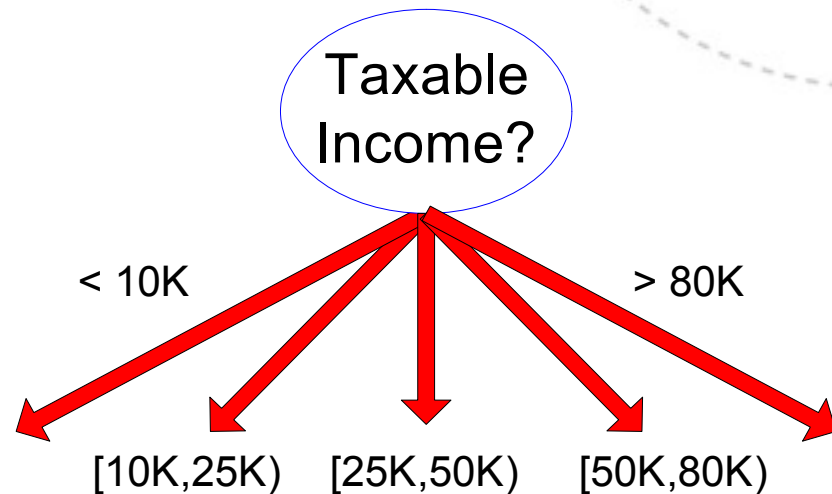




# Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

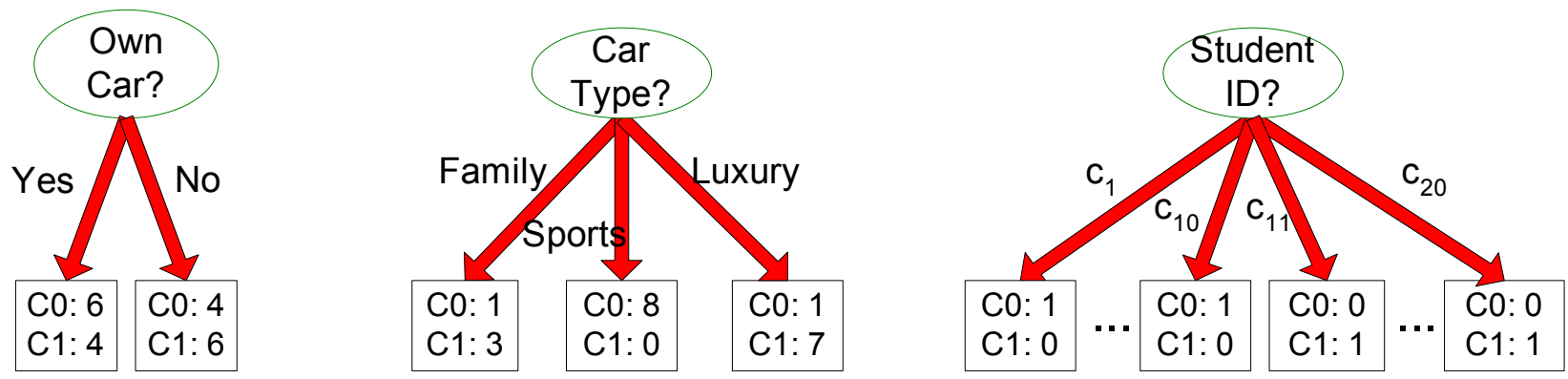
# Tree Induction

- 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



NTNU – Trondheim  
Norwegian University of  
Science and Technology

# Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

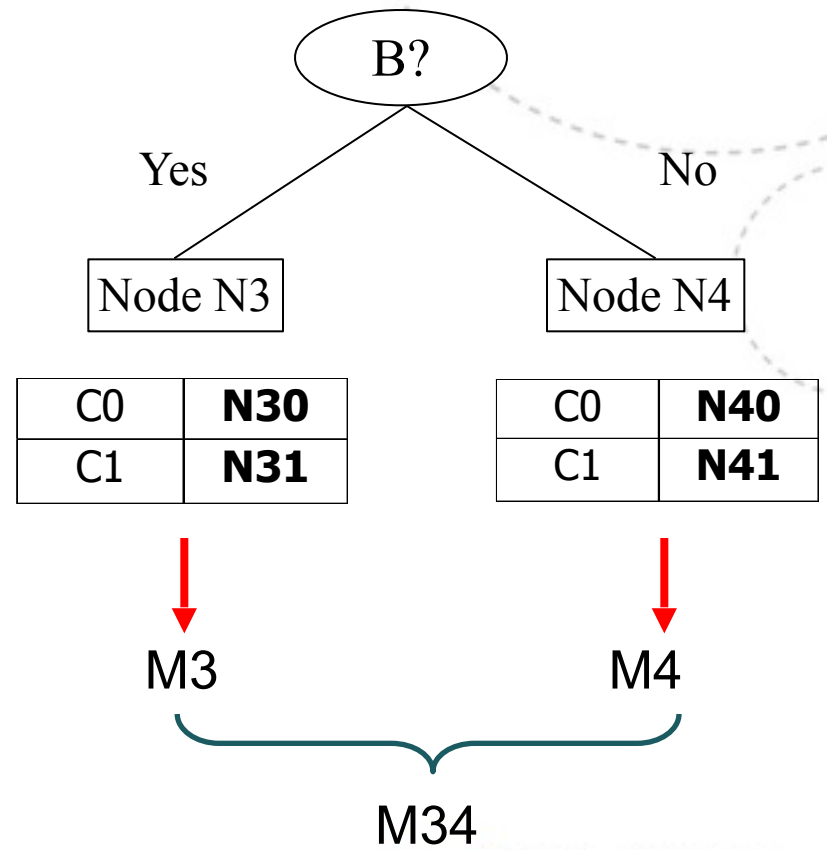
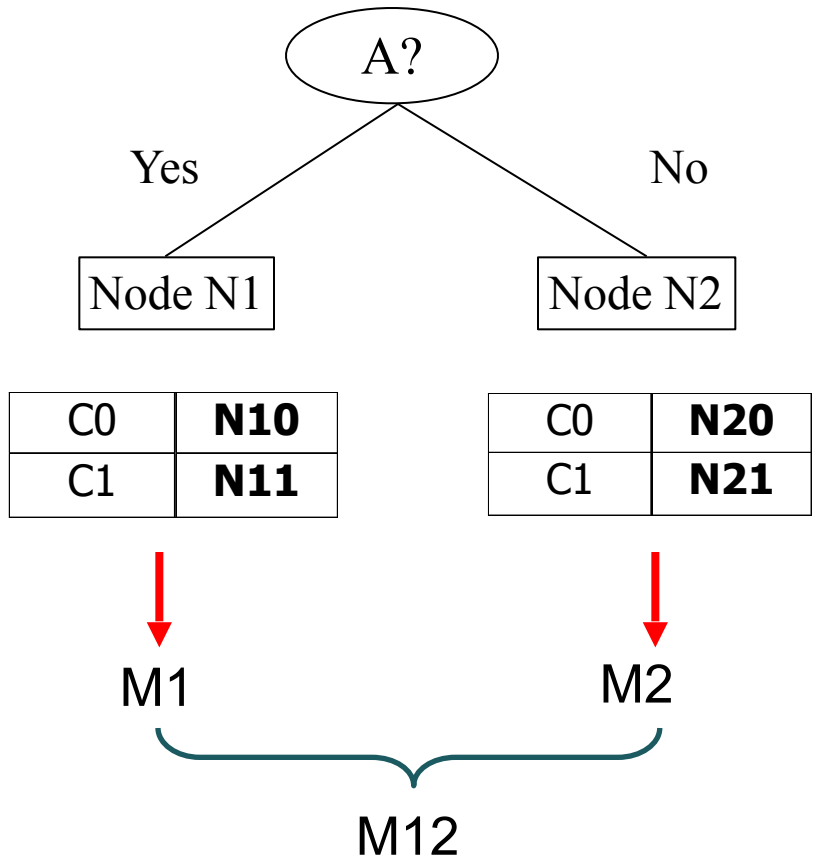


# How to Find the Best Split

Before Splitting:

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

→ M0



Gain = M0 – M12 vs M0 – M34



NTNU – Trondheim  
Norwegian University of  
Science and Technology

# 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 ( $1 - 1/nc$ ) 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,

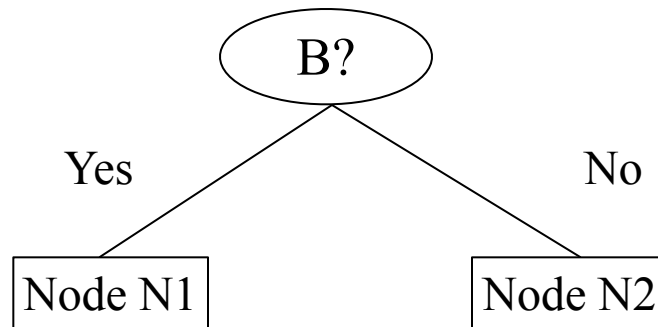
$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

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



# Binary Attributes: Computing GINI Index

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



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

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

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

$$\begin{aligned}
 \text{Gini(Children)} &= 7/12 * 0.408 + \\
 &\quad 5/12 * 0.32 \\
 &= 0.3733
 \end{aligned}$$



# 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	2	1
C2	4	1	1
Gini	0.393		

Two-way split  
(find best partition of values)

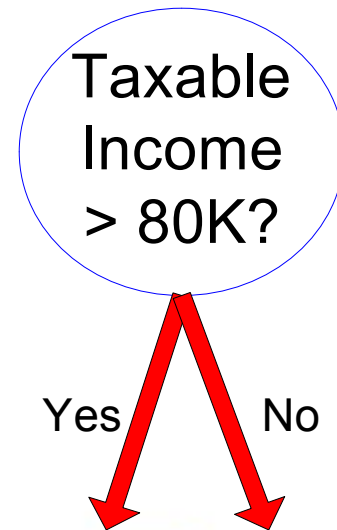
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

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

# 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.

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



NTNU – Trondheim  
Norwegian University of  
Science and Technology

# 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

Sorted Values  
Split Positions

Cheat	No		No		No		Yes		Yes		Yes		No		No		No		No			
→  →	Taxable Income																					
	60		70		75		85		90		95		100		120		125		220			
	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	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		0.300		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 ( $\log nc$ ) 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>

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

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

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:

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

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>

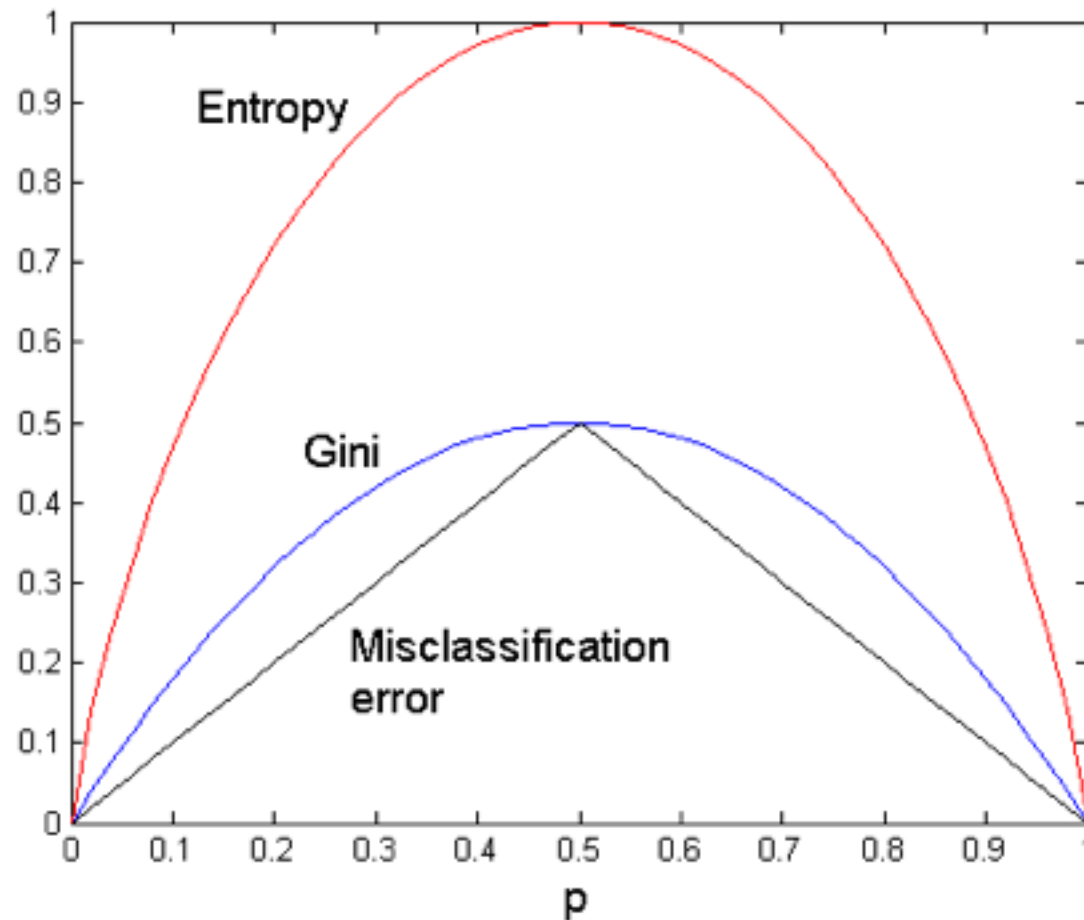
$$P(C1) = 2/6 \quad P(C2) = 4/6$$

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

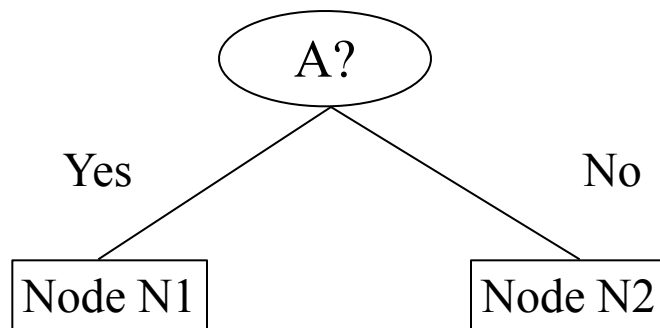


# Comparison among Splitting Criteria

For a 2-class problem:



# Misclassification Error vs Gini



	Parent
C1	<b>7</b>
C2	<b>3</b>
<b>Gini = 0.42</b>	

$$\begin{aligned} \text{Gini}(N1) \\ &= 1 - (3/3)^2 - (0/3)^2 \\ &= 0 \end{aligned}$$

	<b>N1</b>	<b>N2</b>
C1	<b>3</b>	<b>4</b>
C2	<b>0</b>	<b>3</b>
<b>Gini=0.361</b>		

$$\begin{aligned} \text{Gini}(N2) \\ &= 1 - (4/7)^2 - (3/7)^2 \\ &= 0.489 \end{aligned}$$

$$\begin{aligned} \text{Gini(Children)} \\ &= 3/10 * 0 \\ &+ 7/10 * 0.489 \\ &= 0.342 \end{aligned}$$

**Gini improves !!**



# 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)



# 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>

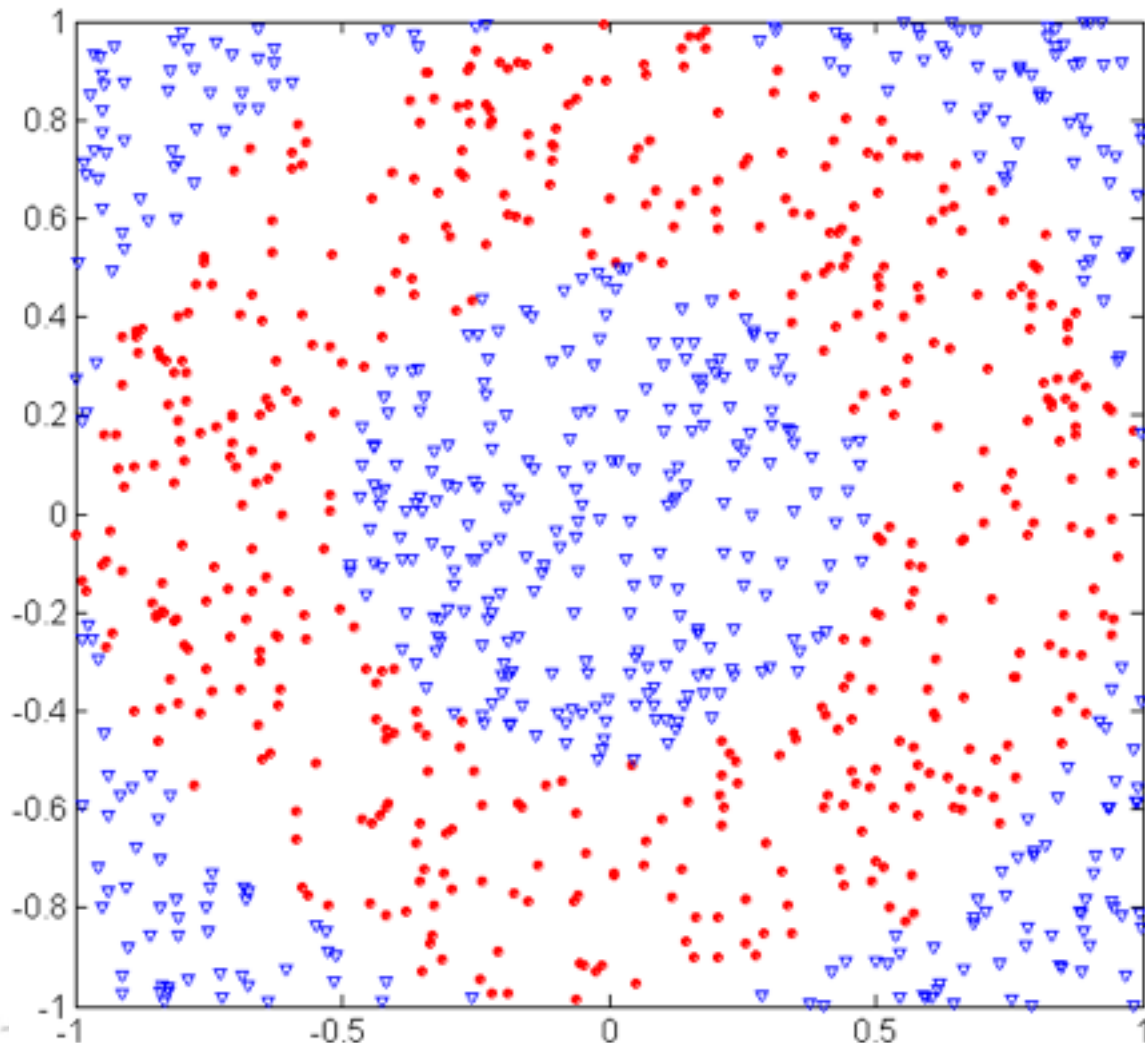


# Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification



# I Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

$$0.5 \leq \sqrt{x_1^2 + x_2^2} \leq 1$$

Triangular points:

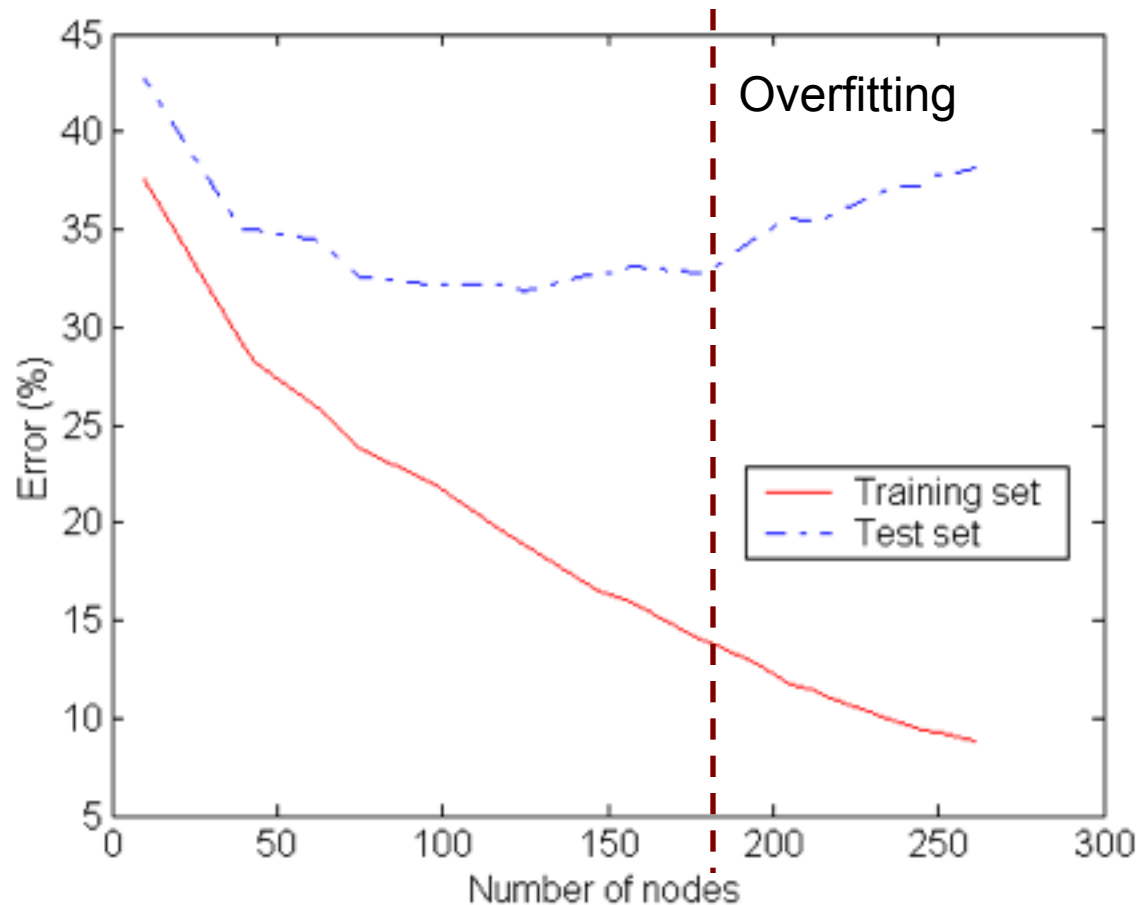
$$\sqrt{x_1^2 + x_2^2} > 0.5 \text{ or}$$

$$\sqrt{x_1^2 + x_2^2} < 1$$



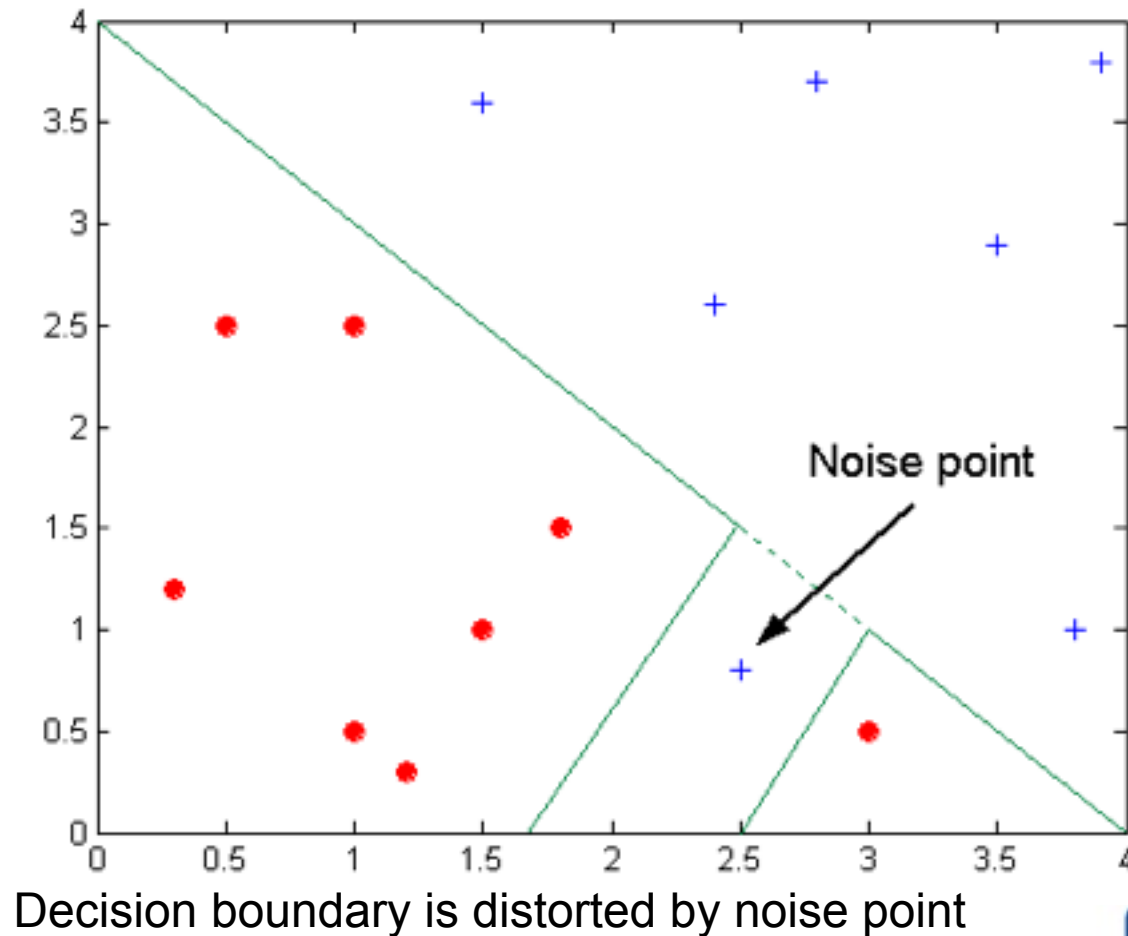
NTNU – Trondheim  
Norwegian University of  
Science and Technology

# Underfitting and Overfitting

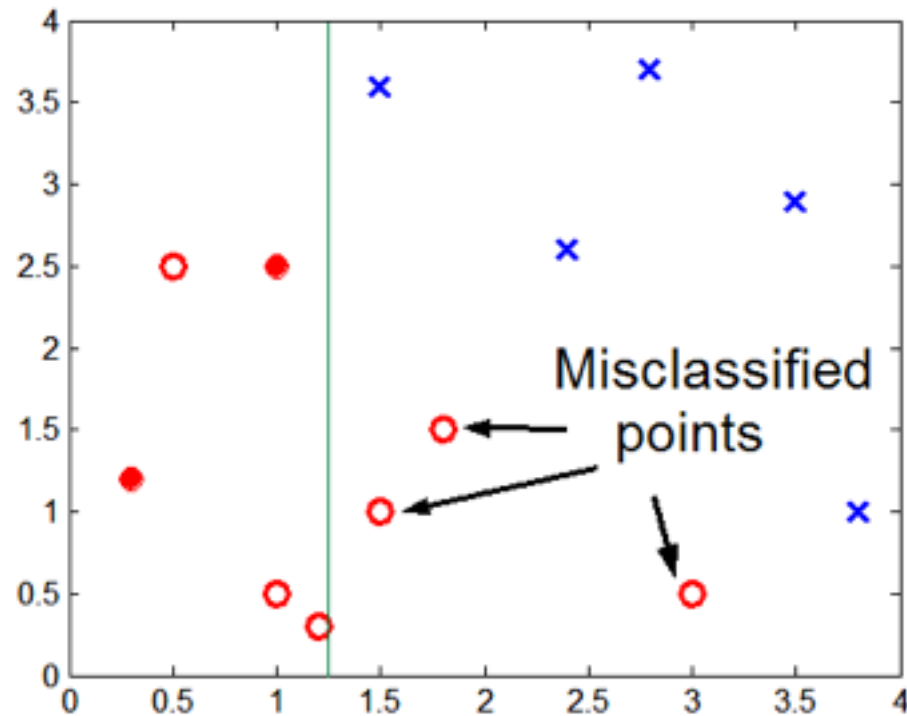


Underfitting: when model is too simple, both training and test errors are large

# Overfitting due to Noise



# Overfitting due to Insufficient Examples



# Notes on Overfitting

- Overfitting results in decision trees that are **more complex than necessary**
- Training error **no longer provides a good estimate** of how well the tree will perform on previously unseen records
- Need **new ways for estimating errors**



# How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
  - 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 class distribution of instances are independent of the available features (e.g., using  $\chi^2$  test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).





# Computing Impurity Measure

<i>Tid</i>	Refund	Marital Status	Taxable Income	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	?	Single	90K	Yes

Missing  
value

Before Splitting:

$$\text{Entropy(Parent)} \\ = -0.3 \log(0.3) - (0.7) \log(0.7) = 0.8813$$

	Class = Yes	Class = No
Refund=Yes	0	3
Refund=No	2	4
Refund=?	1	0

Split on Refund:

$$\text{Entropy(Refund=Yes)} = 0$$

$$\text{Entropy(Refund=No)} \\ = -(2/6) \log(2/6) - (4/6) \log(4/6) = 0.9183$$

$$\text{Entropy(Children)} \\ = 0.3 (0) + 0.6 (0.9183) = 0.551$$

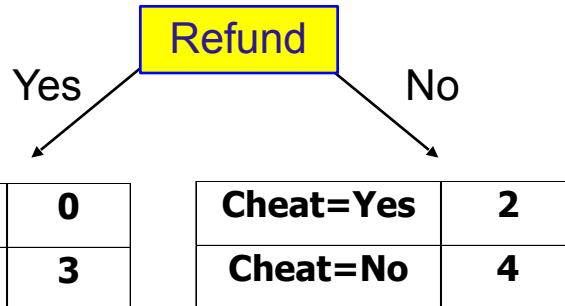
$$\text{Gain} = 0.9 \times (0.8813 - 0.551) = 0.3303$$



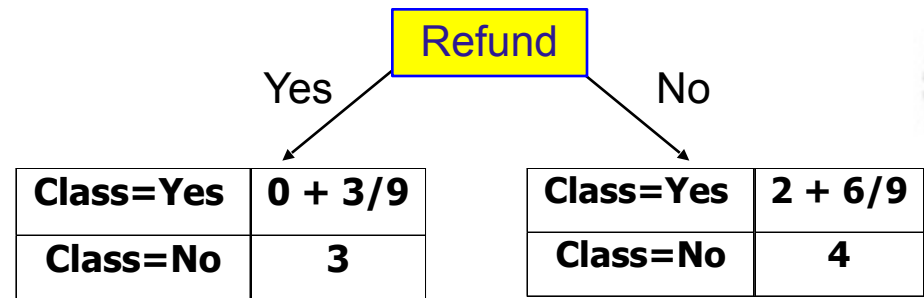
NTNU – Trondheim  
Norwegian University of  
Science and Technology

# Distribute Instances

Tid	Refund	Marital Status	Taxable Income	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



Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes



Probability that Refund=Yes is  $3/9$

Probability that Refund=No is  $6/9$

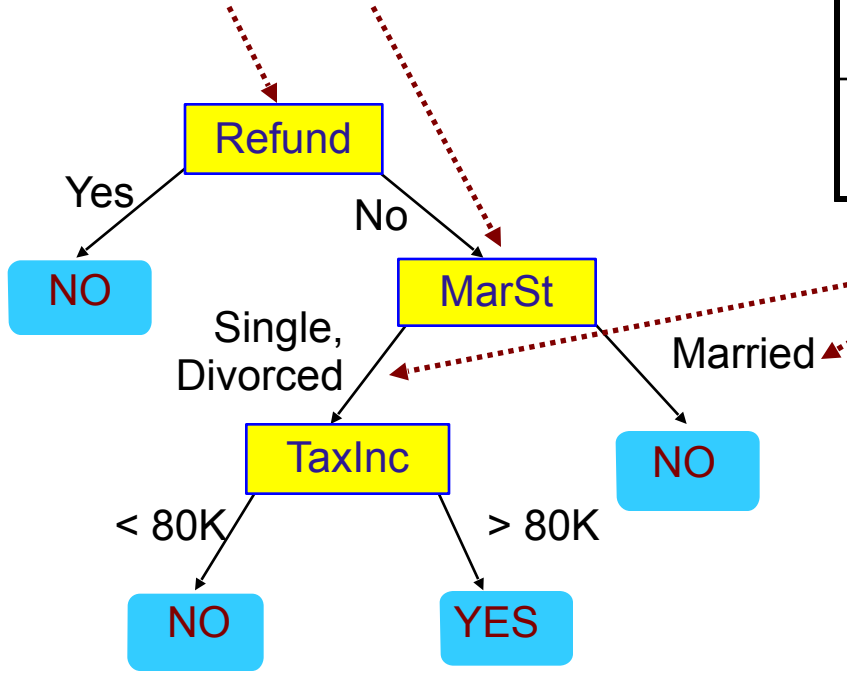
Assign record to the left child with weight =  $3/9$  and to the right child with weight =  $6/9$

# Classify Instances

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?

	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	6/9	1	1	2.67
Total	3.67	2	1	6.67



Probability that Marital Status = Married is  $3.67/6.67$

Probability that Marital Status = {Single, Divorced} is  $3/6.67$

# Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?



# Model Evaluation

- **Metrics for Performance Evaluation**
  - How to evaluate the performance of a model?
- **Methods for Performance Evaluation**
  - How to obtain reliable estimates?
- **Methods for Model Comparison**
  - How to compare the relative performance among competing models?



# Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
		Class=Yes	Class=No
	ACTUAL CLASS		
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

# Metrics for Performance Evaluation...

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is  $9990/10000 = 99.9\%$ 
  - Accuracy is misleading because model does not detect any class 1 example





# Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- **Methods for Performance Evaluation**
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?



# Methods of Estimation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into  $k$  disjoint subsets
  - $k$ -fold: train on  $k-1$  partitions, test on the remaining one
  - Leave-one-out:  $k=n$
- Stratified sampling
  - oversampling vs undersampling
- Bootstrap
  - Sampling with replacement

