Week 3

Classification (Part I)

Seokho Chi
Professor | Ph.D.
SNU Construction Innovation Lab

Source: *Tan, Kumar, Steinback (2006)*

서울대학교 건설혁신연구실
Construction Innovation Lab SNU

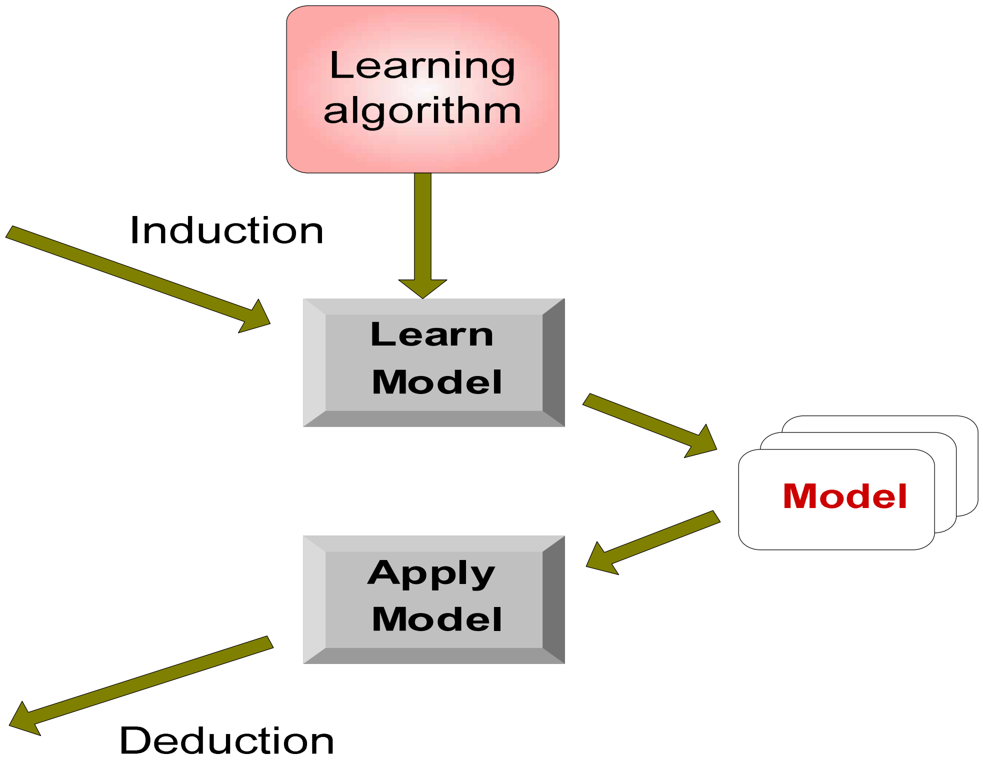
Classification: Definition

- 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 the 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

양성종양 악성종양

- 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

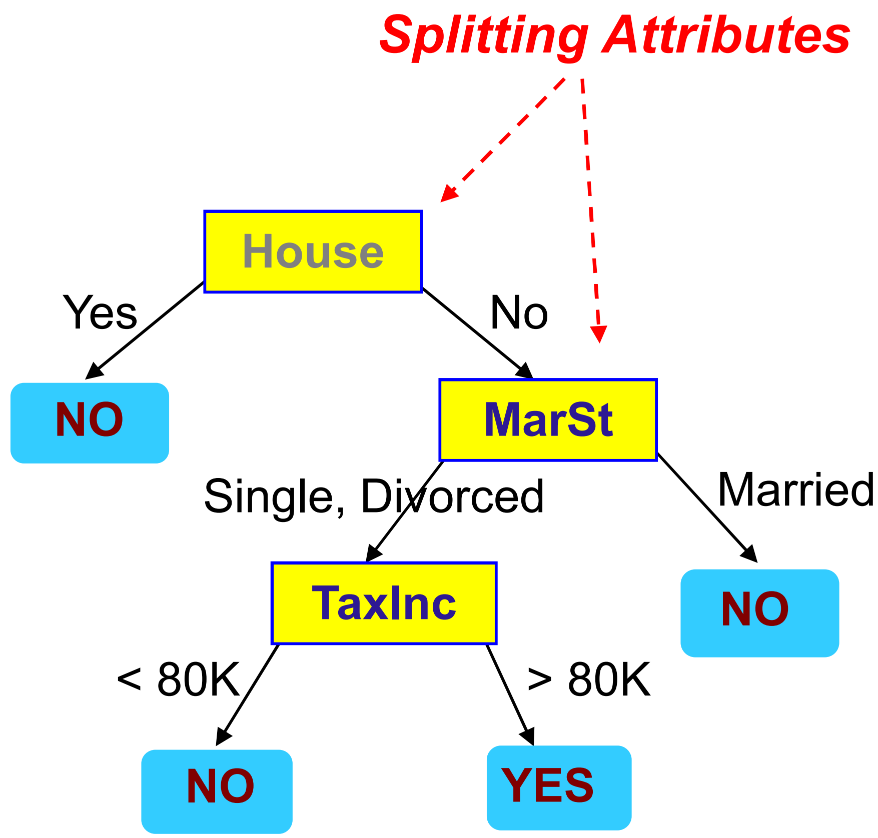
- Decision Tree based Methods
- Rule-based Methods
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Example of a Decision Tree

categorical categorical continuous class

Tid	House Owner	Marital Status	Taxable Income	Default?
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

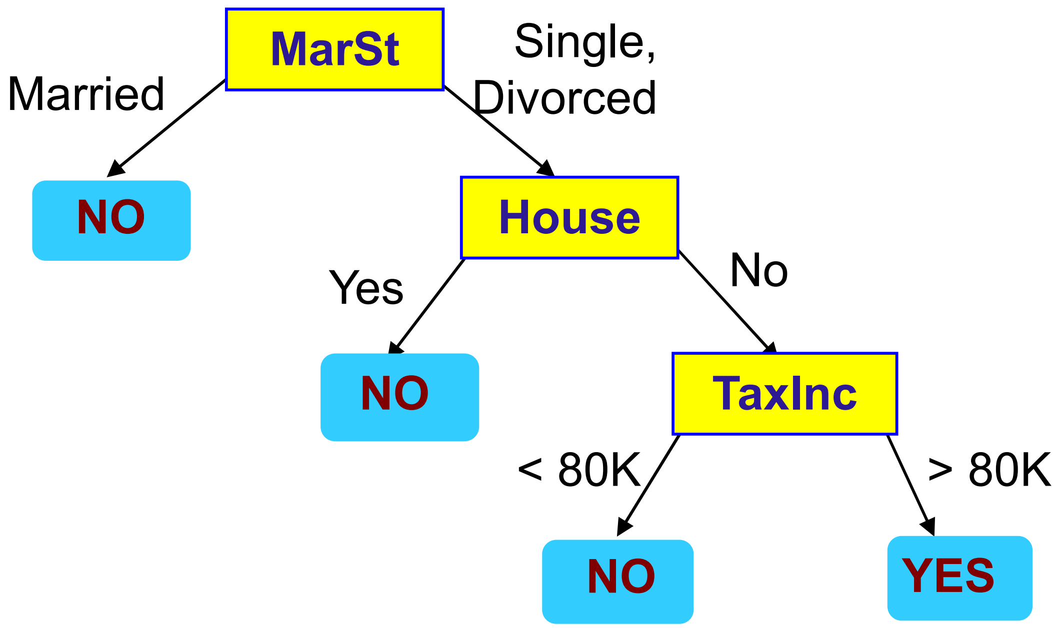


Model: Decision Tree

Another Example of Decision Tree

categorical categorical continuous class

Tid	House Owner	Marital Status	Taxable Income	Default?
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!

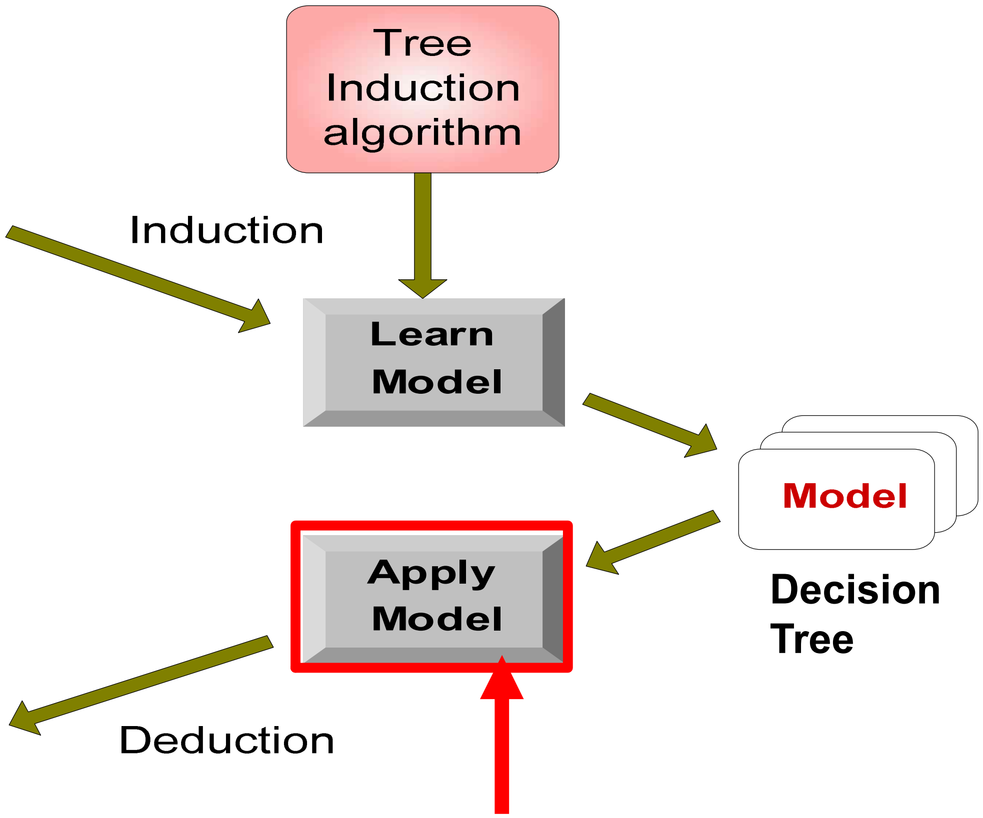
Decision Tree Classification Task

Tid	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

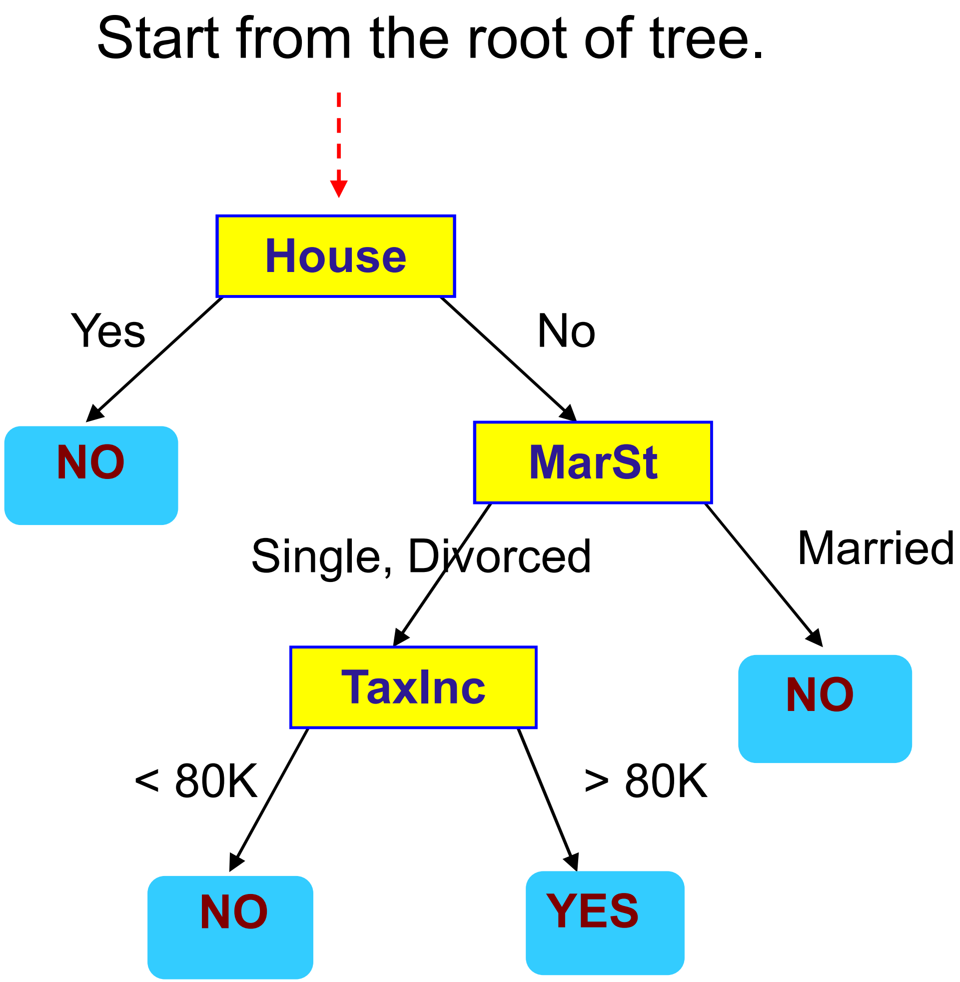
Training Set

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



Apply Model to Test Data



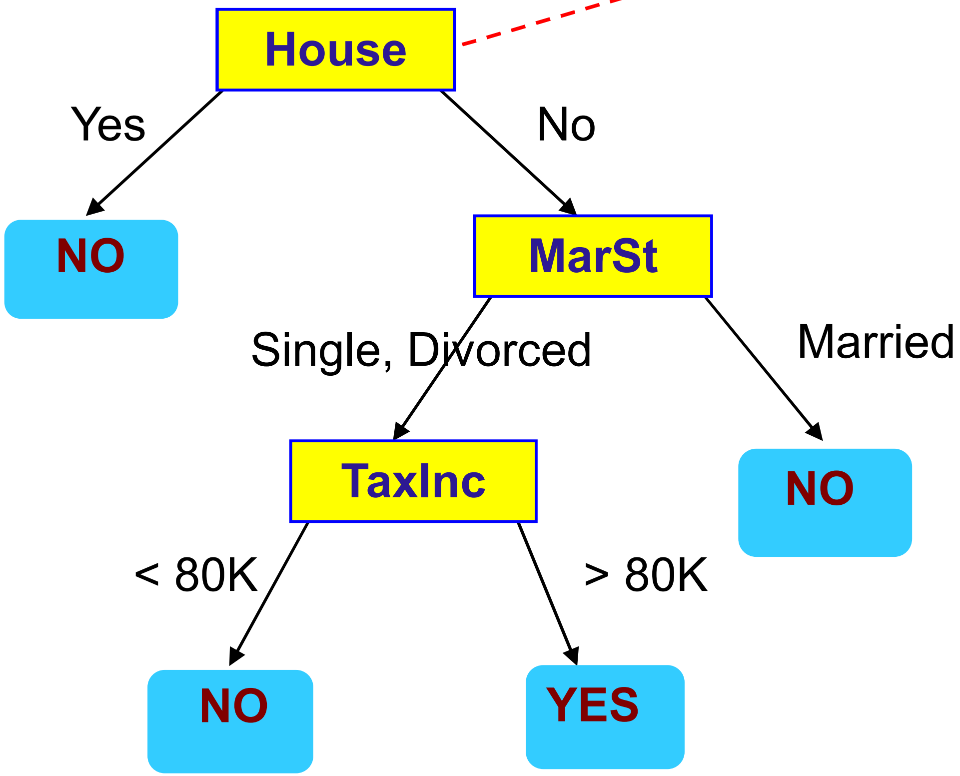
Test Data

House Owner	Marital Status	Taxable Income	Defau lt?
No	Married	80K	?

Apply Model to Test Data

Test Data

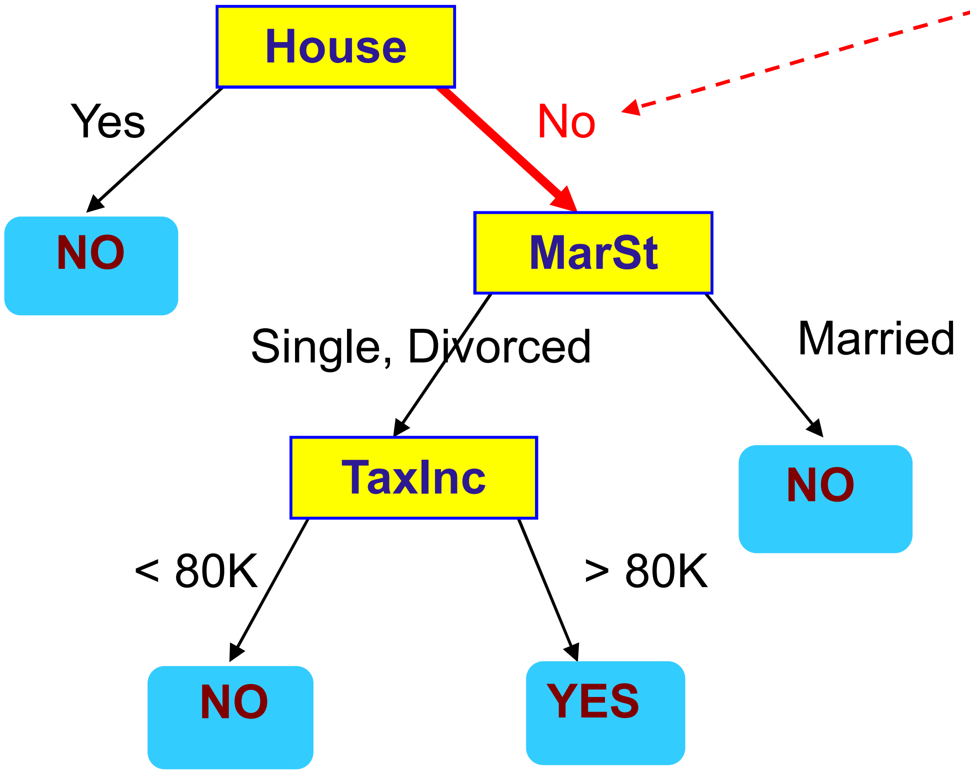
House Owner	Marital Status	Taxable Income	Default?
No	Married	80K	?



Apply Model to Test Data

Test Data

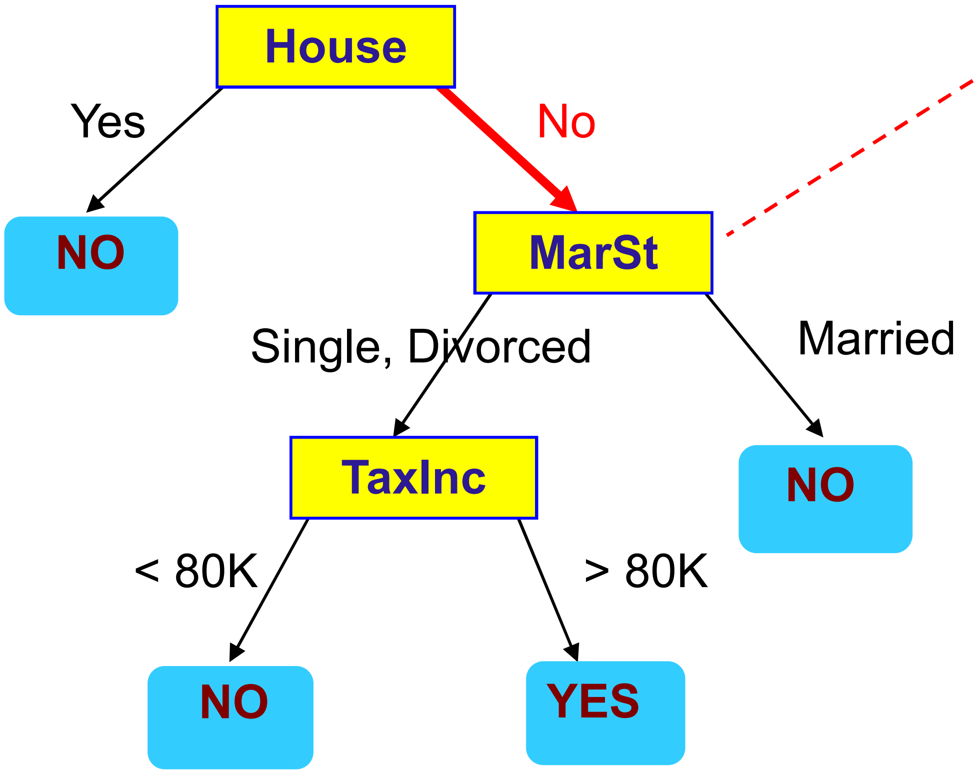
House Owner	Marital Status	Taxable Income	Default?
No	Married	80K	?



Apply Model to Test Data

Test Data

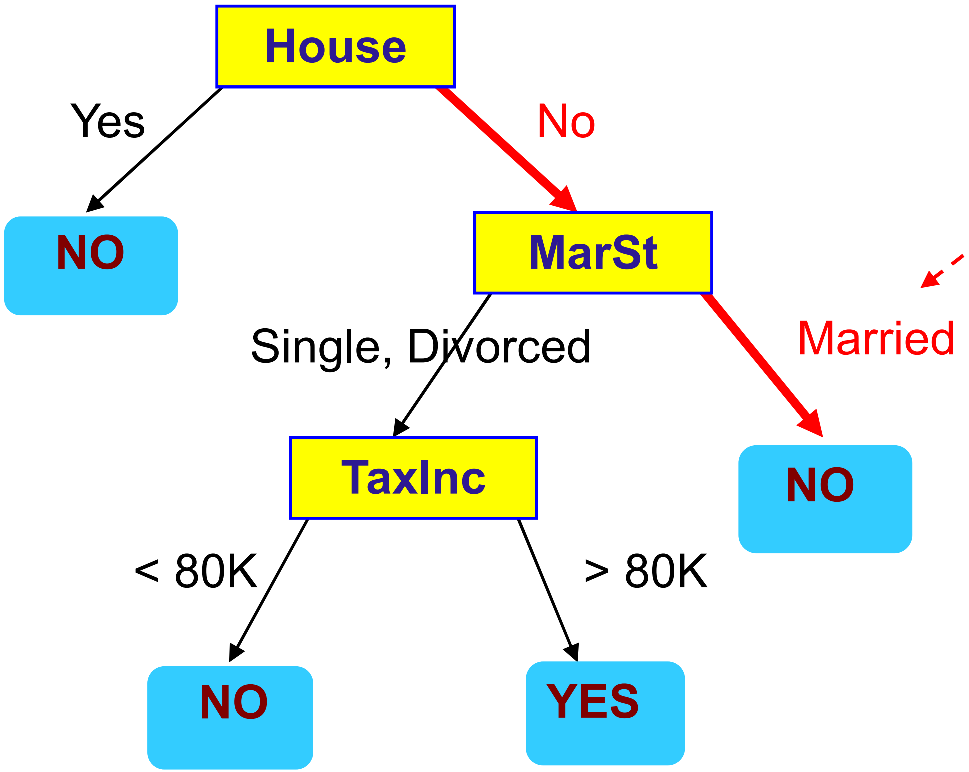
House Owner	Marital Status	Taxable Income	Default It?
No	Married	80K	?



Apply Model to Test Data

Test Data

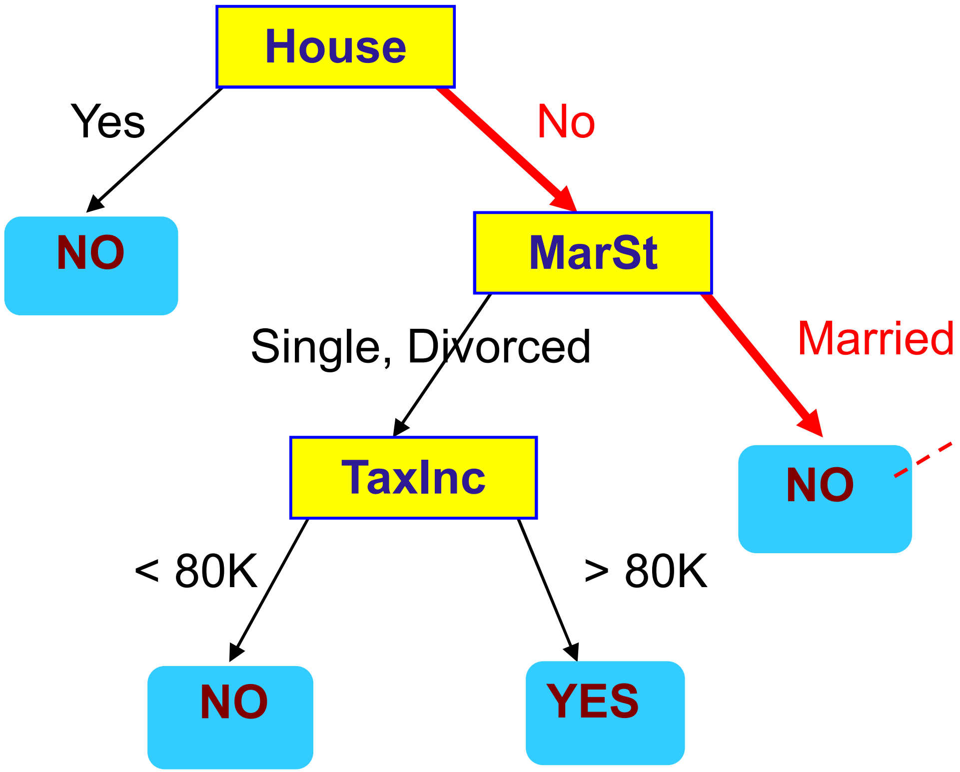
House Owner	Marital Status	Taxable Income	Default?
No	Married	80K	?



Apply Model to Test Data

Test Data

House Owner	Marital Status	Taxable Income	Default It?
No	Married	80K	?



Assign Default to “No”

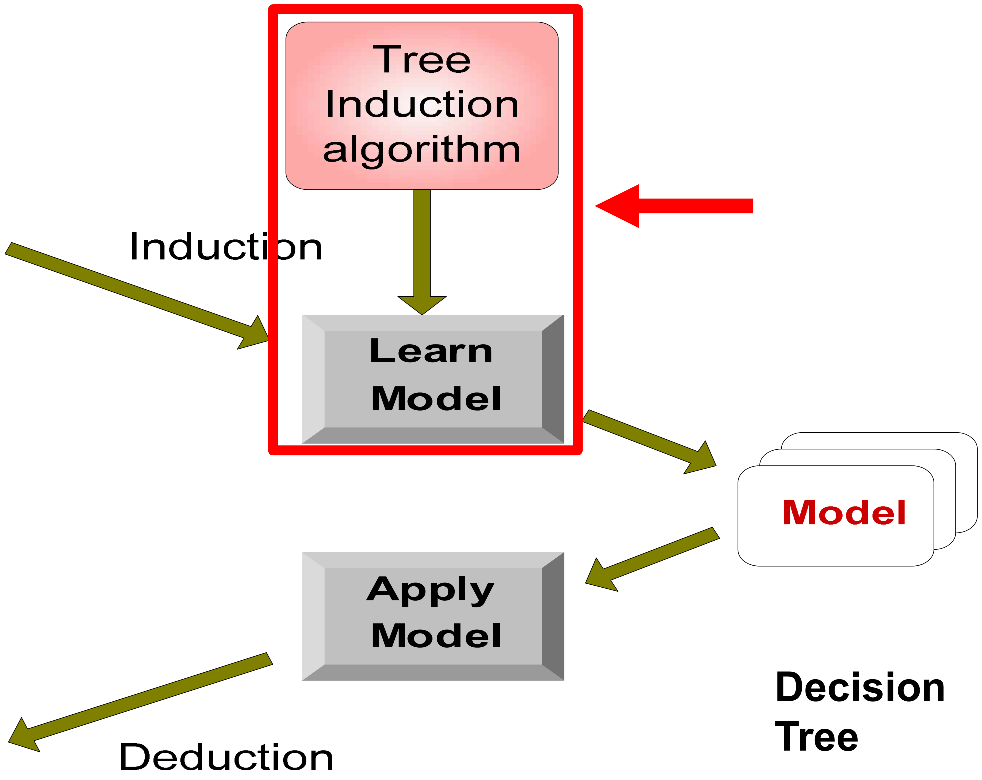
Decision Tree Classification Task

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



Decision Tree Induction

- Greedy strategy
  - Tree is constructed in a top-down recursive divide-and-conquer(분할하여 각각의 문제들을 해결) manner
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - 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

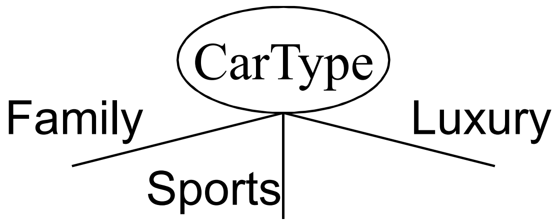
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

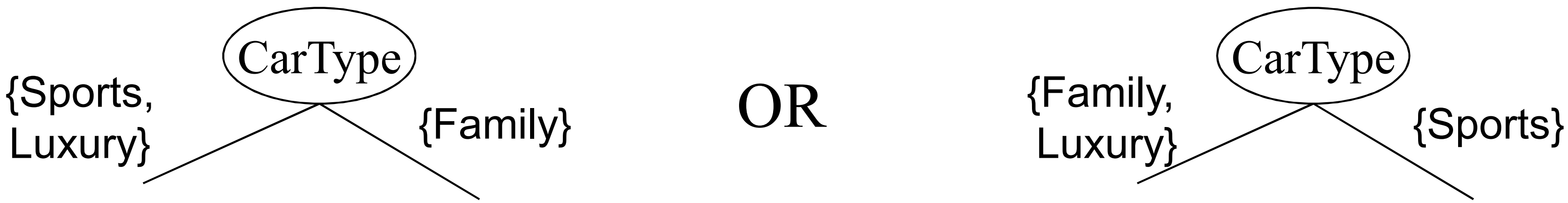
- Depends on number of ways to split
  - 2-way (binary) split
  - Multi-way split

Splitting Based on Nominal Attributes

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

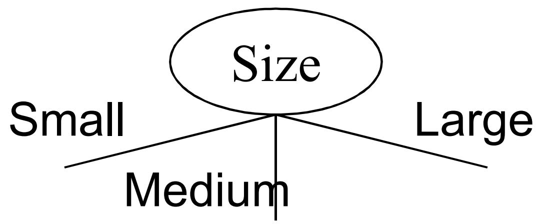


- Binary split: Divides values into two subsets. Need to find optimal partitioning.

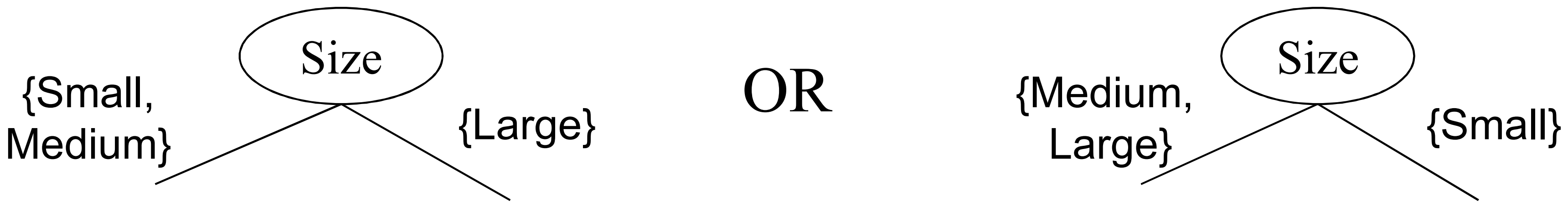


Splitting Based on Ordinal Attributes

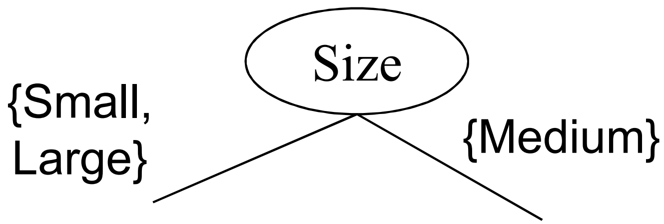
- Multi-way split: Use as many partitions as distinct values.



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

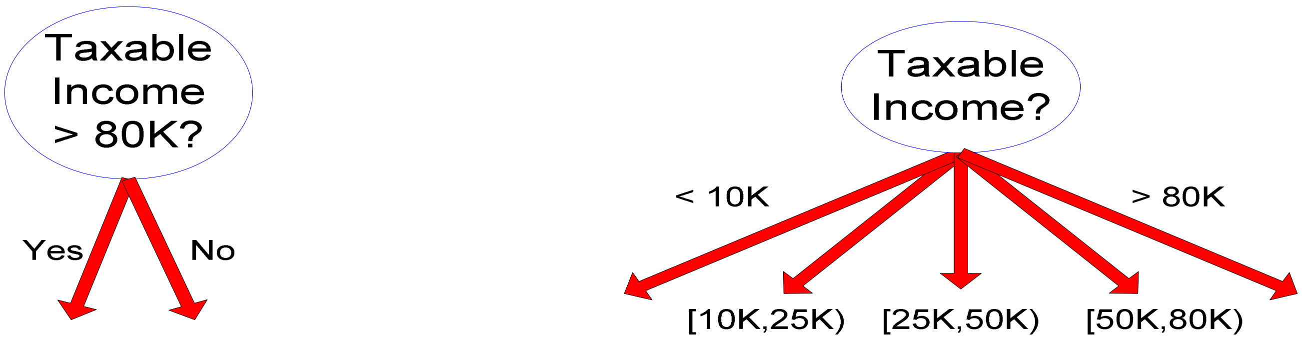


- What about this split?



Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Discretize once at the beginning
    - Equal interval bucketing, equal frequency bucketing, clustering
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - Consider all possible splits and finds the best cut
    - Can be more compute intensive



(i) Binary split

(ii) Multi-way split

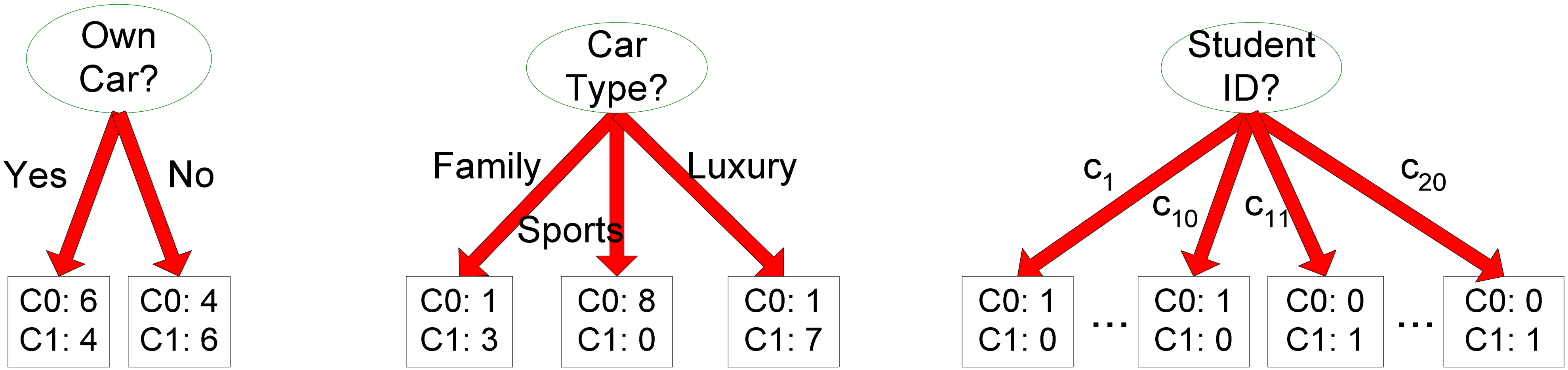
Tree Induction

- 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? (cont.)

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1
Non-homogeneous,	**Homogeneous (대표성),**
High degree of impurity	**Low degree of impurity (명확성)**

Best split : determined by measuring impurity of nodes!

Measures of Node Impurity

- Information

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

- Gini Index

$$Gini(s_1, s_2, \ldots, s_m) = 1 - \sum_{i=1}^{m} \left(\frac{s_i}{s}\right)^2$$

- Classification error

$$Error(s_1, s_2, \ldots, s_m) = 1 - \max\left(\frac{s_i}{s}\right)$$

Node N_1	Count
Class=0	0
Class=1	6

$$\begin{aligned} \text{Gini} &= 1 - (0/6)^2 - (6/6)^2 = 0 \\ \text{Info.} &= -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0 \\ \text{Error} &= 1 - \max[0/6, 6/6] = 0 \end{aligned}$$

Node N_2	Count
Class=0	1
Class=1	5

$$\begin{aligned} \text{Gini} &= 1 - (1/6)^2 - (5/6)^2 = 0.278 \\ \text{Info.} &= -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650 \\ \text{Error} &= 1 - \max[1/6, 5/6] = 0.167 \end{aligned}$$

Node N_3	Count
Class=0	3
Class=1	3

$$\begin{aligned} \text{Gini} &= 1 - (3/6)^2 - (3/6)^2 = 0.5 \\ \text{Info.} &= -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1 \\ \text{Error} &= 1 - \max[3/6, 3/6] = 0.5 \end{aligned}$$

Measures for Selecting the Best Split: Information Gain

**Best selecting is determined by the degree of impurity of the child nodes*

- S contains s_i records of class C_i for i = {1, …, m}
- Information: impurity measure required to classify a given sample (small is better)

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

- Entropy: summation of information values of subsets partitioned by attribute A (small is better)

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj})$$

- Information Gain: information before splitting – entropy (large is better, how impurity decreased by attribute A)

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

Select the attribute with the highest information gain

Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

(1) Splitting Based on Information Gain

- 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 (splitting by 5(low entropy) is better than splitting by 2)
  - To overcome, do adjustment by considering the ratio of sample numbers of each node to the entire samples(전체 중 이 노드가 차지하는 비중)
  - Gain Ratio for C4.5, etc.
  - GINI Index for CART, SLIQ, SPRINT, etc.

Splitting Based on INFO...

- Gain Ratio:

$$GainRATIO = \frac{GAIN}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

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

Measure of Impurity: GINI

- Gini Index for a given node t :

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

(NOTE: $p(j \mid t)$ 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	**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	

Lower is better

Examples for computing GINI

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

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

(2) 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,

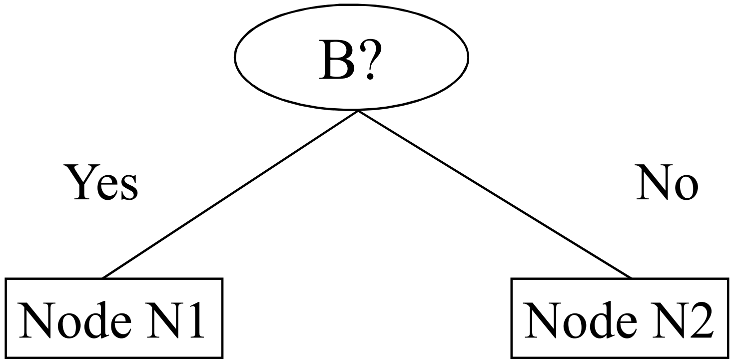
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

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
  - If you pick B as an attribute →



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

Gini(N1)
= 1 – $(5/6)^2$ – $(2/6)^2$
= 0.194

Gini(N2)
= 1 – $(1/6)^2$ – $(4/6)^2$
= 0.528

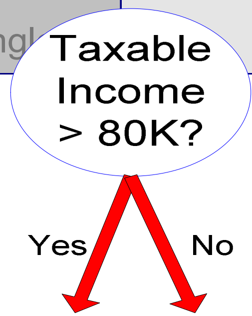
	N1	**N2**
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

Gini (Children)
= 7/12 * 0.194 +
5/12 * 0.528
= 0.333

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 ≥ 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	Singl	[illegible]	**Yes**



(3) Splitting based on Classification Error

- Classification 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

Examples for Computing Error

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

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

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

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

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

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

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

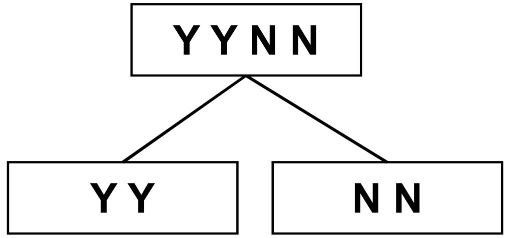
Tree Induction

- 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 there is no remaining attribute for further partitioning

- Early termination (to be discussed later)
  – Become to be huge → Rather negative effect

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

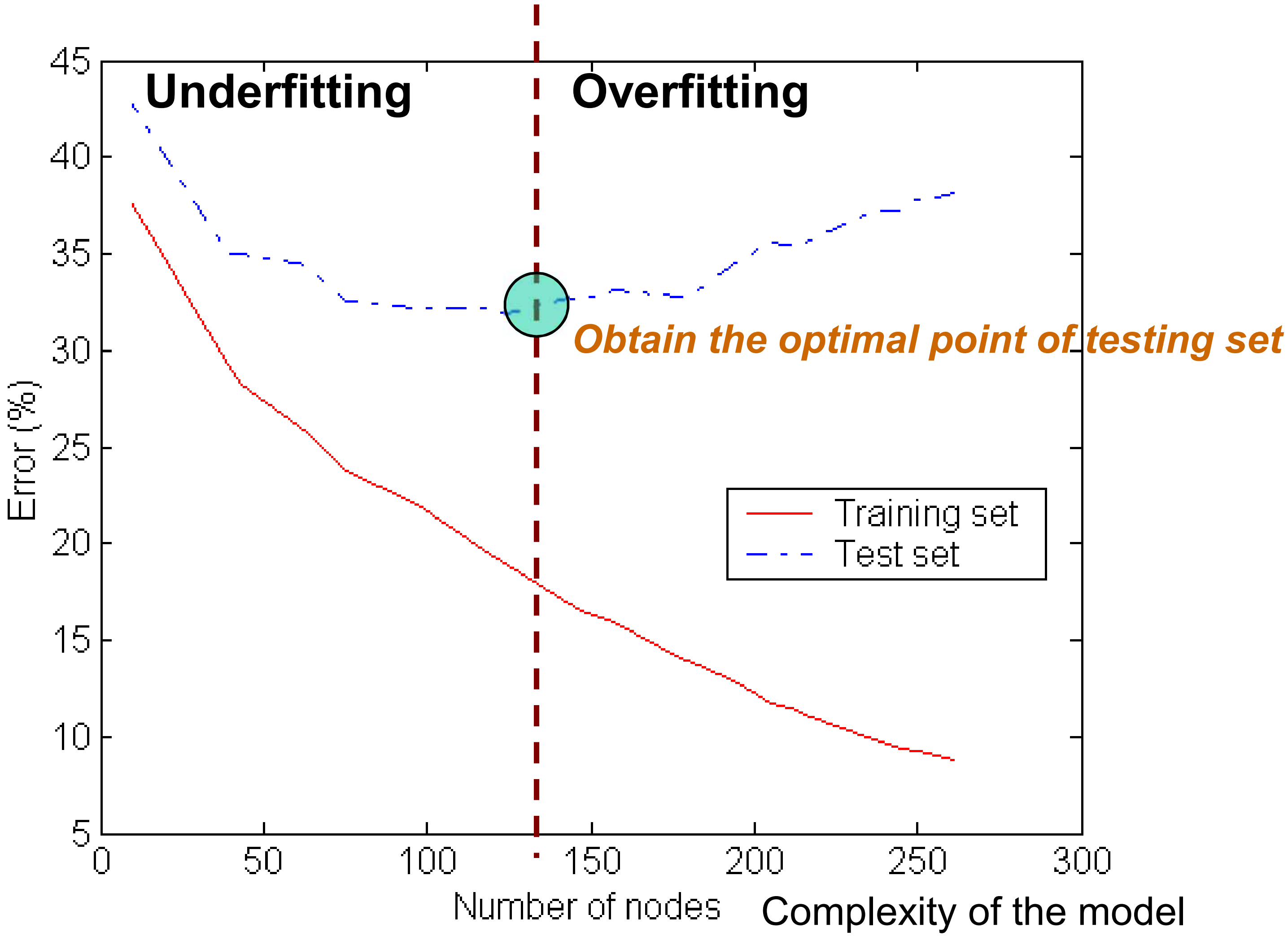
Example: C4.5

- Uses information gain
- Sorts continuous attributes at each node
- Unsuitable for large datasets
- You can download the software from:
  http://www.rulequest.com/Personal/
- See5: commercial version (free demo available):
  http://www.rulequest.com

Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

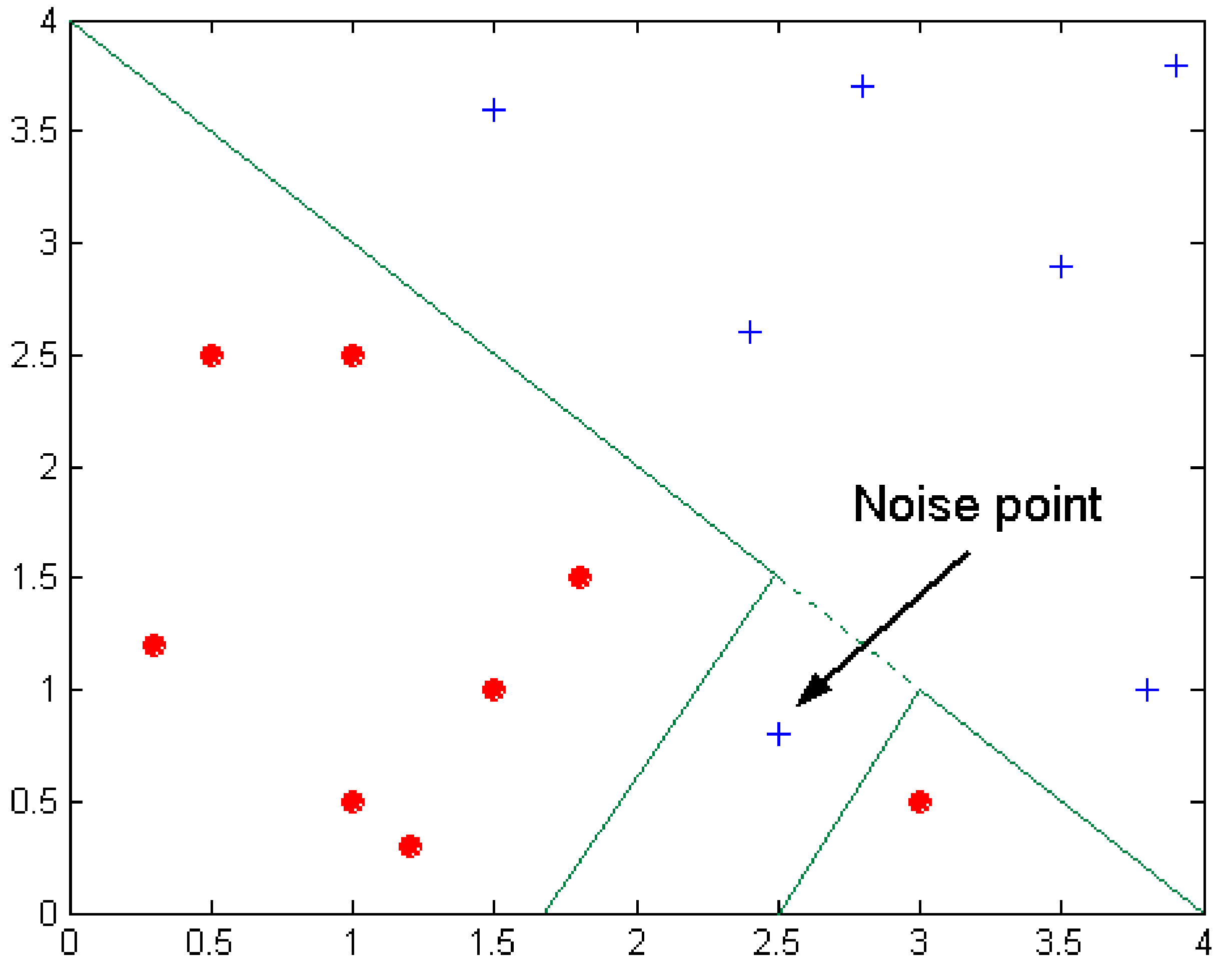
Underfitting and Overfitting



Underfitting: too simple → both training and test errors are large
Overfitting: too complex → generalization errors (errors on testing) are large (data fragmentation)

Occam's Razor: Given two models of similar generalization errors, <u>the simpler model</u> is preferred.

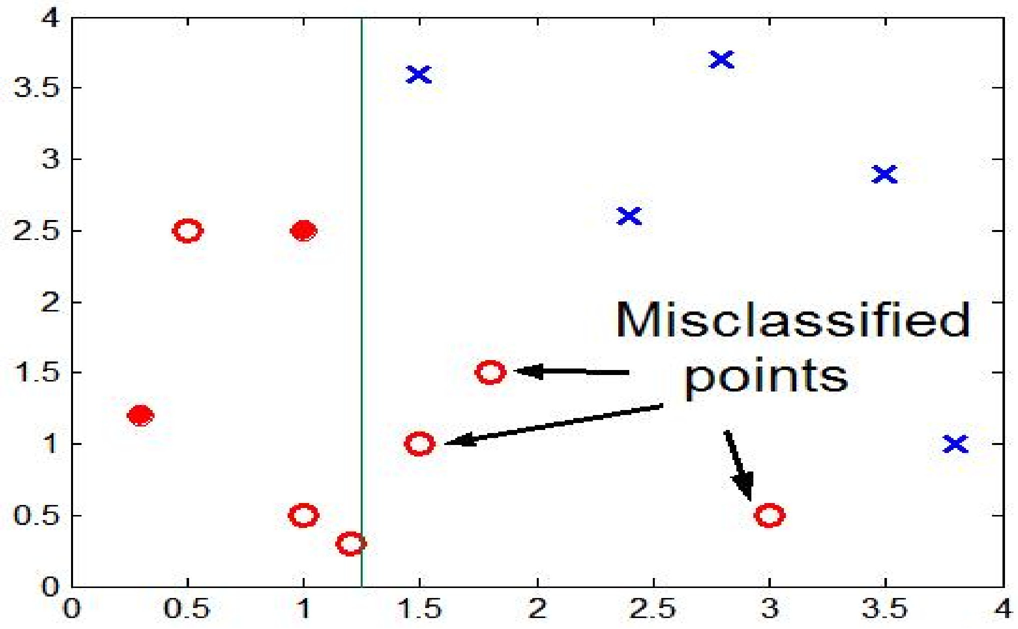
Overfitting due to Noise



Create tree models with noise samples

(Decision boundary is distorted by noise point)

Overfitting due to Insufficient Examples



Tree models are development with insufficient samples

(Lack of data points makes it difficult to predict correctly the class labels of a region)

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 there is no remaining attribute for further partitioning
  - 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)
    - Stop if the tree reaches generalization error threshold (pessimistic error estimate, Minimum Description Length Principle, etc.)

Optimistic: training error = testing error | Pessimistic: training error < testing error

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 or impurity 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

Handling Missing Attribute Values

- Missing values affect decision tree construction in three different ways
  - Affects when impurity measures are computed
  - Affects how to assign instance with missing value to child nodes
  - Affects how a test instance with missing value is classified

Computing Impurity Measure

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
10	?	Single	90K	Yes

Missing value

Before Splitting:

Information(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:

Information(Refund=Yes) = 0

Information(Refund=No)
= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

Entropy(Refund)
= 0.3 (0) + 0.6 (0.9183) = 0.551

Gain = 0.9 × (0.8813 – 0.551) = 0.3303

Do not consider missing values and adjust with weight (0.9)

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

Refund

Yes

Class=Yes	0
Class=No	3

No

Class=Yes	2
Class=No	4

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

Refund

Yes

Class=Yes	0 + 3/9
Class=No	3

No

Class=Yes	2 + 6/9
Class=No	4

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

Model Evaluation

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

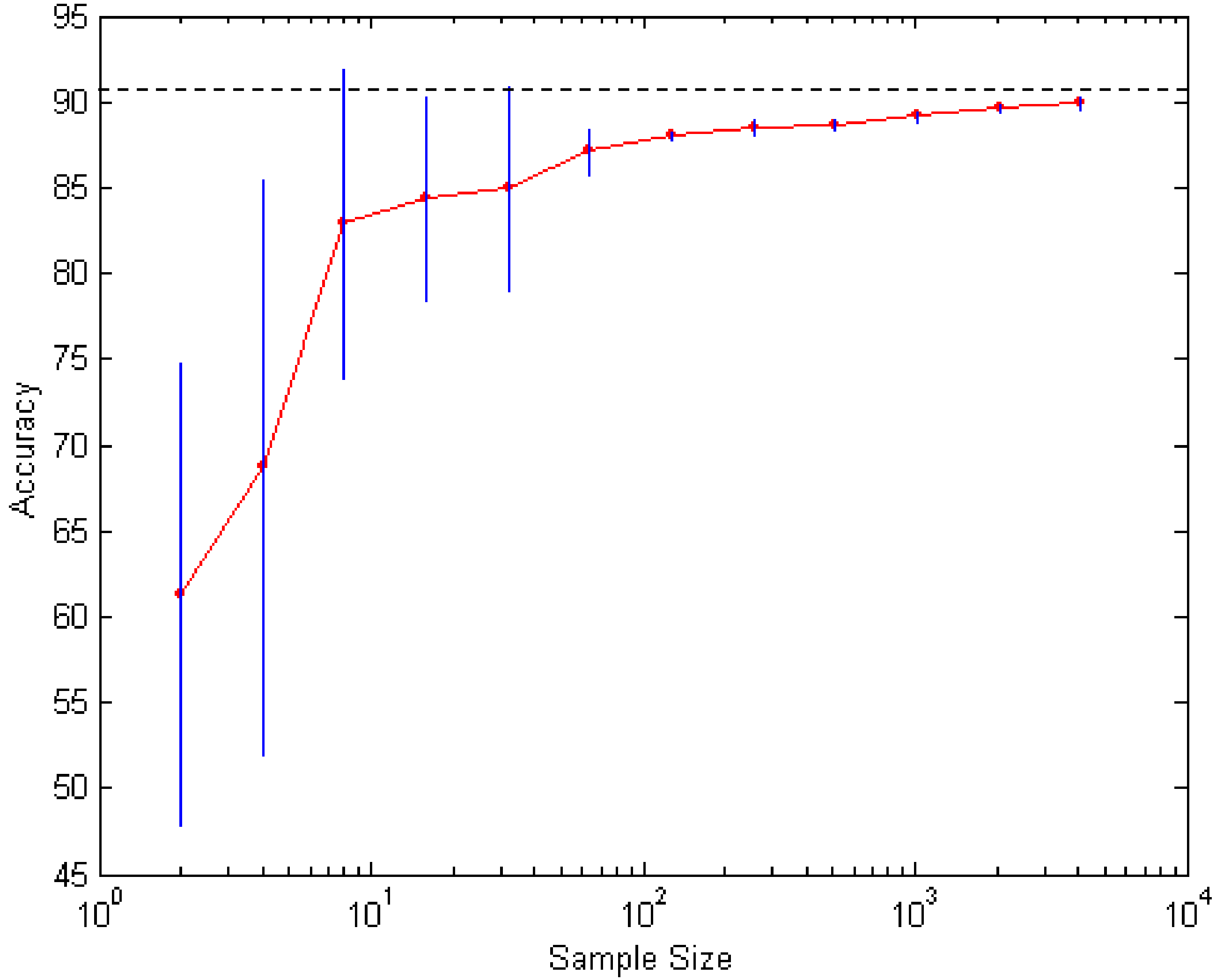
Model Evaluation

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

Methods for Performance Evaluation

- Classification performance of a model may depend on other factors besides the learning algorithms
  - Class distribution
  - Sampling approaches
  - Size of training and test sets

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
  - Arithmetic sampling (Langley et. al.) 1000, 1500, 2000, 2500, 3000, …
  - Geometric sampling (Provost et. al.) 1000, 2000, 4000, 8000, 16000, …

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

Methods of Estimation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing (independent each other)
- Random subsampling
  - Repeated holdout

Limited testing samples, results are affected by the distribution

- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one

  (10-fold: training 90%, testing 10%, then repeat this 10 times)

Good generalization and representativeness, but high computation and still small # of testing samples

- Bootstrap
  - Previous: no replacement → no duplicate records
  - Sampling with replacement
  - Probability to be selected to Bootstrap : $1-(1-1/N)^N \approx 1-e^{-1} = 0.632$
  - Records not included in the bootstrap sample → test set

Model Evaluation

- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- 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
ACTUAL CLASS Class=No	c	d

a: TP (true positive)

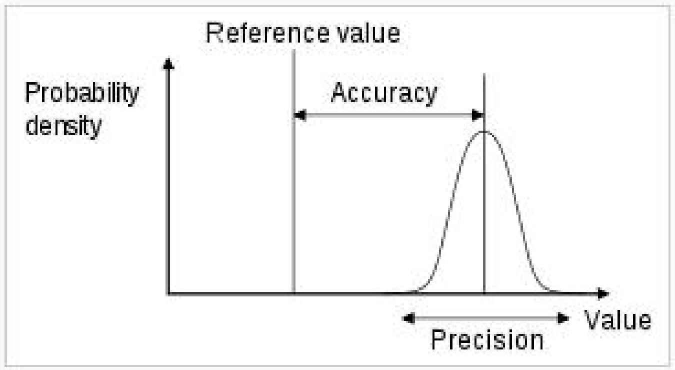
b: FN (false negative)

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)



Accuracy is the proximity of measurement results to the true value; precision, the repeatability, or reproducibility of the measurement

- Most widely-used metric:

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

Precision (Positive Prediction Value) = TP / (TP + FP)
A retrieved document is relevant

Recall (Sensitivity) = TP / (TP + FN)
A relevant document is retrieved

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 (e.g., safety accident)

Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i

higher cost → higher concern

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	+	-
	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 80%

Cost = -1*150 + 100* 40 + 1*60

= 3910

Accuracy = 90%

Cost = 4255

Model Evaluation

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

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis) → answer is P
- Performance of each classifier represented as a point on the ROC curve

$$TPR = \frac{TP}{TP + FN}$$

TPR

P(Actually True 중 Predict as True)

$$FPR = \frac{FP}{TN + FP}$$

FPR

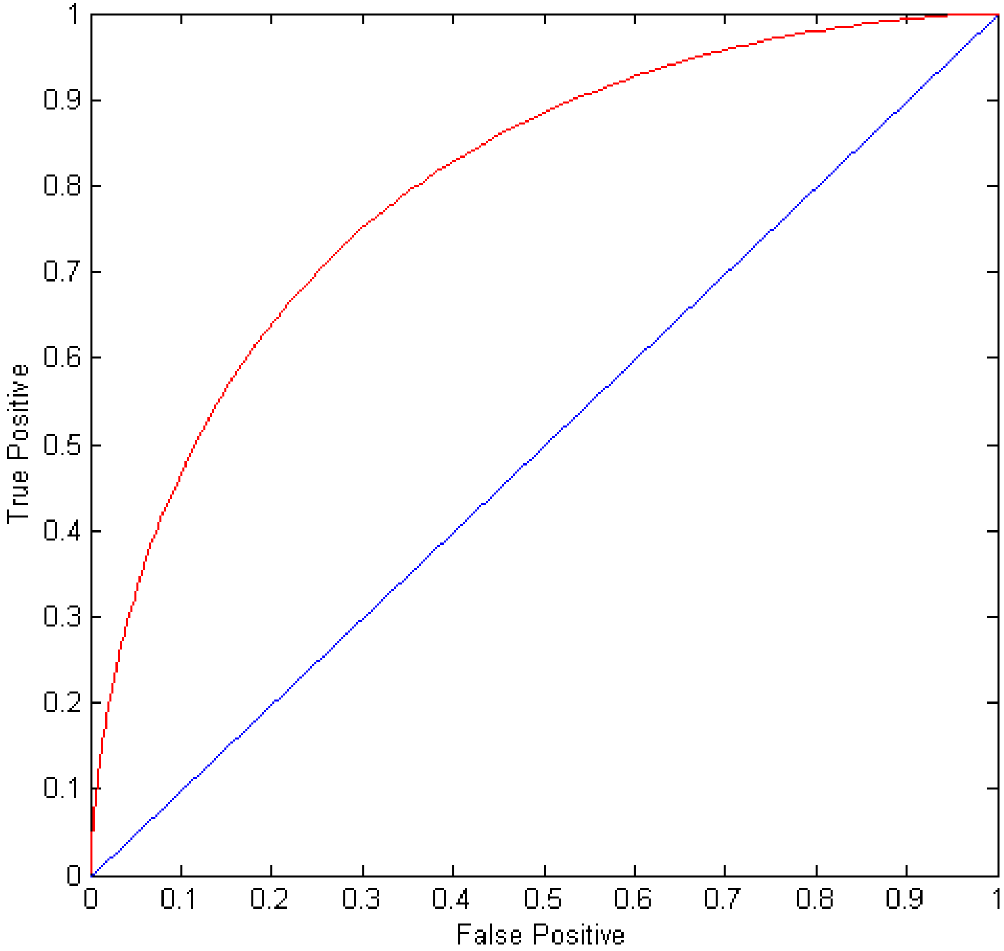
P(Actually False 중 Predict as True)

ROC Curve

(TPR,FPR):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal
- (0,1): worst

- Diagonal line:
  - Random guessing(reference line)
  - Below diagonal line:
    - prediction is opposite of the true class



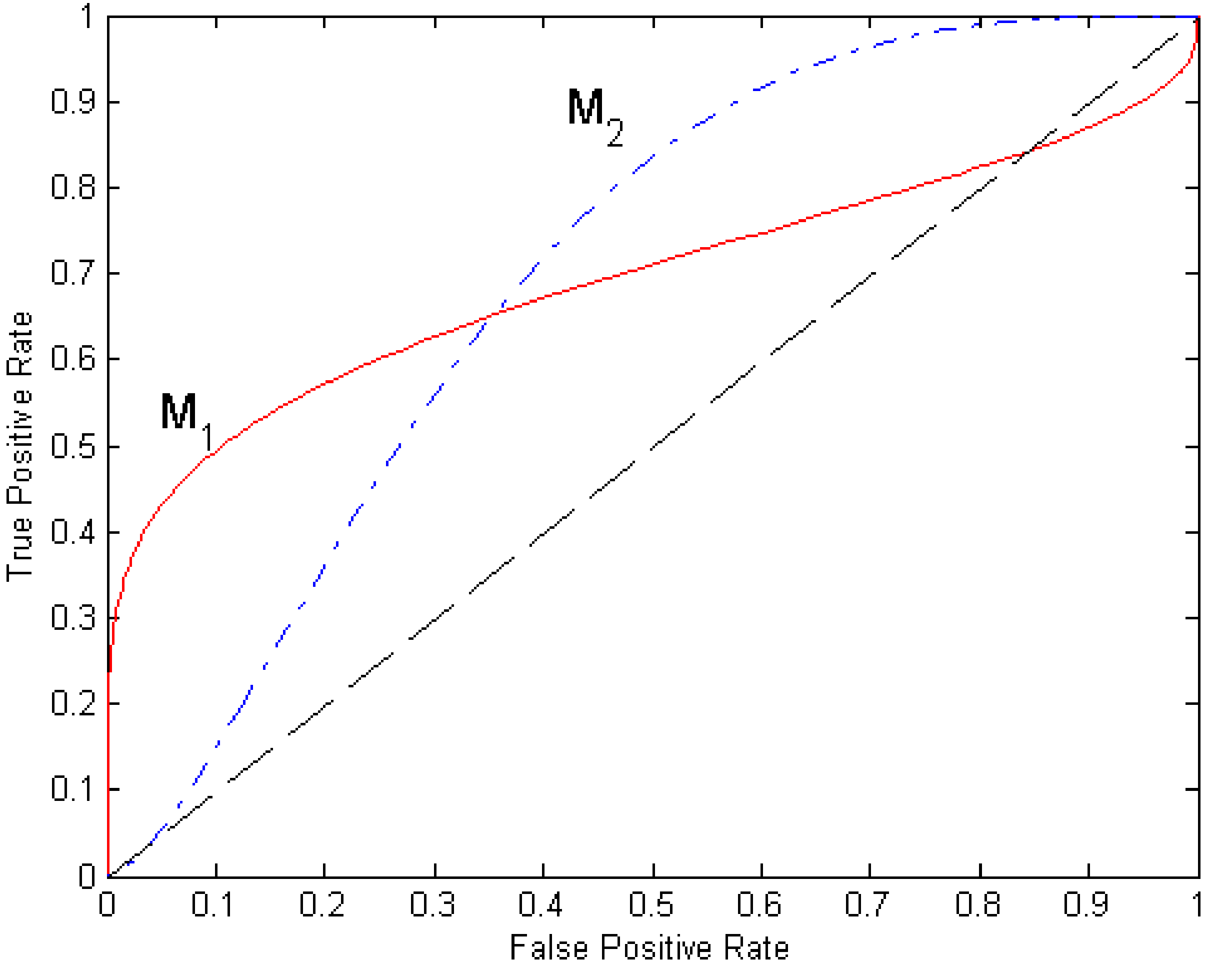
TPR

P(Actually True 중 Predict as True)

FPR

P(Actually False 중 Predict as True)

Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR

Test of Significance

- Given two models:
  - Model M1: accuracy = 85%, tested on 30 instances
  - Model M2: accuracy = 75%, tested on 5000 instances
- Can we say M1 is better than M2?
  - How much confidence can we place on accuracy of M1 and M2?
  - Can the difference in performance measure be explained as a result of random fluctuations in the test set?

Comparing Performance of 2 Models

- Given the models M1 and M2, which is better?
  - M1 is tested on D1 (size=n1), found error rate = e_1
  - M2 is tested on D2 (size=n2), found error rate = e_2
  - Assume D1 and D2 are independent
  - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1)$$
$$e_2 \sim N(\mu_2, \sigma_2)$$

Comparing Performance of 2 Models

- To test if performance difference is statistically significant: d = e1 – e2
  – Since D1 and D2 are independent, their variance adds up:

$$\hat{\sigma}_d^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$
$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  – At (1-α) confidence level,

$$d_t = d \pm Z_{\alpha/2}\hat{\sigma}_d$$

For this Example

2-sided (tailed) test: deviations of the estimated parameter in either direction are considered (normal distribution)

1-sided: only deviations in one direction are considered (p-value)

- Given: M1: n1 = 30, e1 = 0.15
  M2: n2 = 5000, e2 = 0.25
- d = |e2 – e1| = 0.1 (2-sided test)

$$\hat{\sigma}_d^2 = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

- At 95% confidence level, $Z_{\alpha/2}=1.96$

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

[-0.028, 0.228]

=> Interval contains 0 => difference may not be statistically significant