BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Introduction to Data Science
Classification and Prediction

- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Classification

predictive — most popular ML task

Classification

- Classification involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as *classes*
- Many practical decision-making tasks can be formulated as classification problems
  - customers who are likely to buy or not buy a particular product in a supermarket
  - people who are at high, medium or low risk of acquiring a certain illness
  - student projects worthy of a distinction, merit, pass or fail grade
  - objects on a radar display which correspond to vehicles, people, buildings or trees
  - people who closely resemble, slightly resemble or do not resemble someone seen committing a crime
  - houses that are likely to rise in value, fall in value or have an unchanged value in 12 months' time
  - people who are at high, medium or low risk of a car accident in the next 12 months
  - people who are likely to vote for each of a number of political parties (or none)
  - the likelihood of rain the next day for a weather forecast (very likely, likely, unlikely, very unlikely).

will buy

will not buy

D M P F

Classification vs. Prediction

- Classification
  - predicts $\boxed{\text{categorical class labels}}$ (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

- Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values

1150 sft

1100 sft 50L

1200 sft 65L

900 sft 43L

Semi Supervised

Supervised vs. Unsupervised Learning

Supervised ML

- 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

Association

- 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

People also talk about more forms of machine learning
https://www.gartner.com/smarterwithgartner/understand-3-key-types-of-machine-learning/
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQk6wDdKFf6owv5L902n1AMs6UB_dN4KiwC6w&usqp=CAU

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision trees, or mathematical formulae

Algo → M/C
Data → M/C
M/C → Model

- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Test input → Model → predict label

Illustrating Classification Task

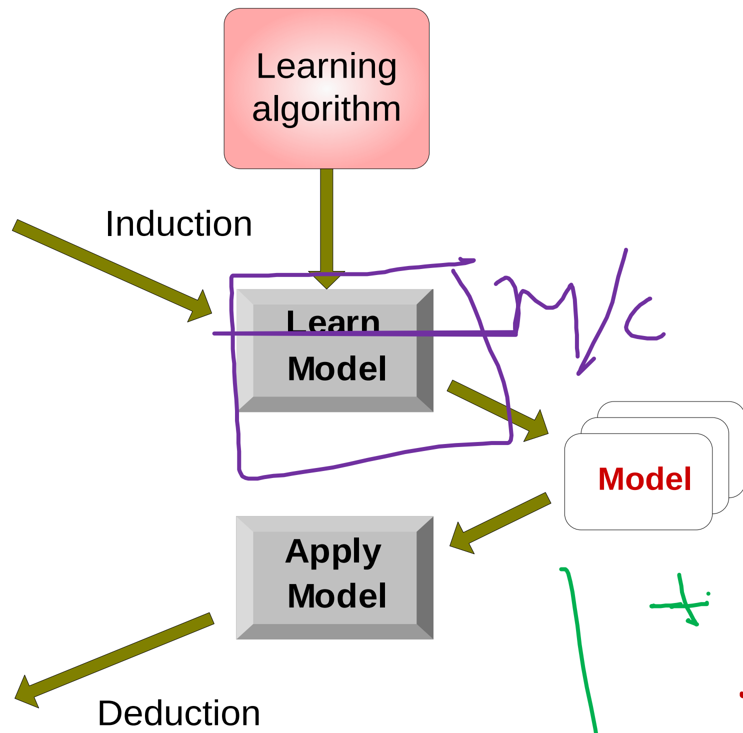
Training data

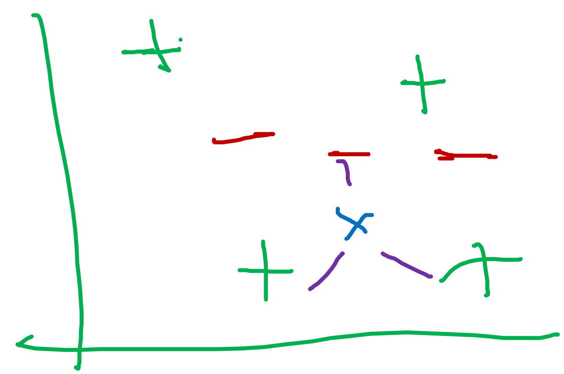
Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

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



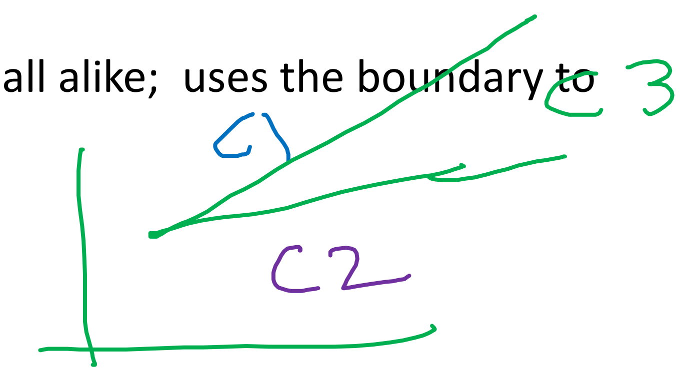


Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Neural Networks
  - computational networks that simulate the decision process in neurons (networks of nerve cell)

ChatGPT Gemini
billions of parameters

- Naïve Bayes and Bayesian Belief Networks
  - uses the *probability theory* to find the most likely of the possible classifications
- Support Vector Machines
  - fits a boundary to a region of points that are all alike; uses the boundary to classify a new point



Lazy vs. Eager Learning

Test Tuple

Training Data

M/c

Label

- Lazy vs. eager learning
  - Lazy learning (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
  - Eager learning (methods discussed in previous slide): Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify

Algorithm

- Lazy: less time in training but more time in predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

Lazy Learner: Instance-Based Methods

- Instance-based learning:
  - Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified
- Typical approaches
  - *k*-nearest neighbor approach
    - Instances represented as points in a Euclidean space.
  - Locally weighted regression
    - Constructs local approximation
  - Case-based reasoning AI
    - Uses symbolic representations and knowledge-based inference

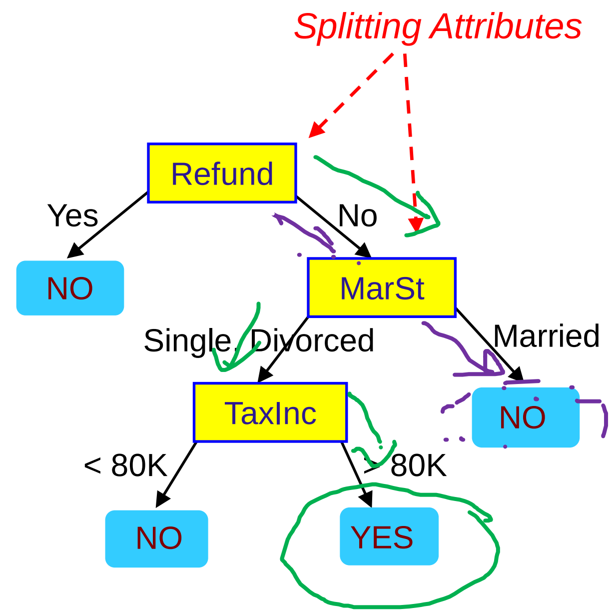
Eager classifier

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

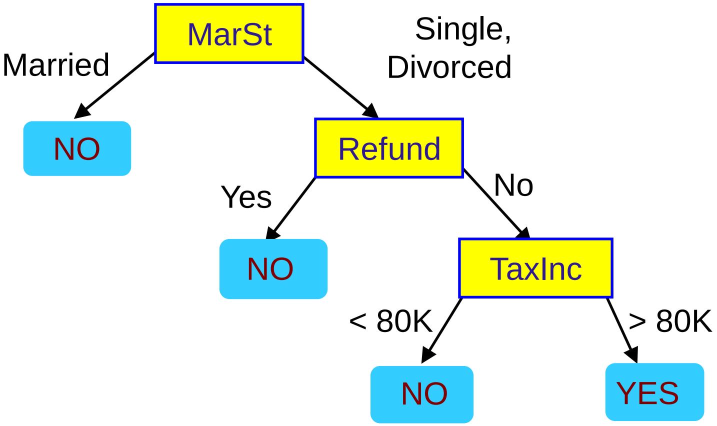


Model: Decision Tree

Another Example of Decision Tree

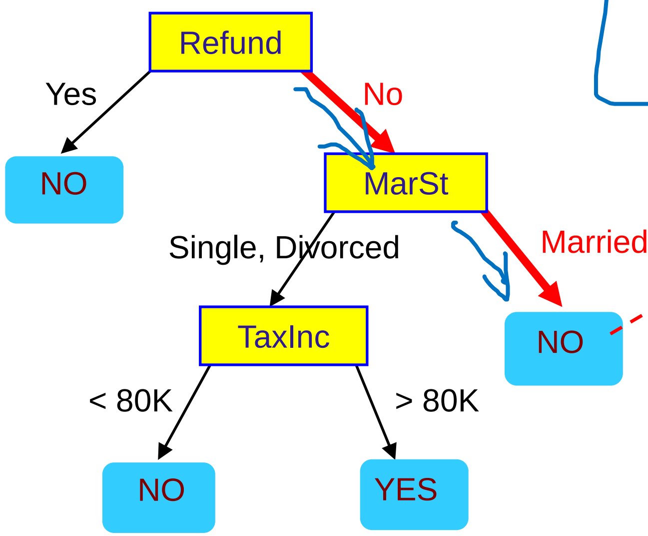
categorical categorical continuous class

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**



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

Apply Model to Test Data



Test Data

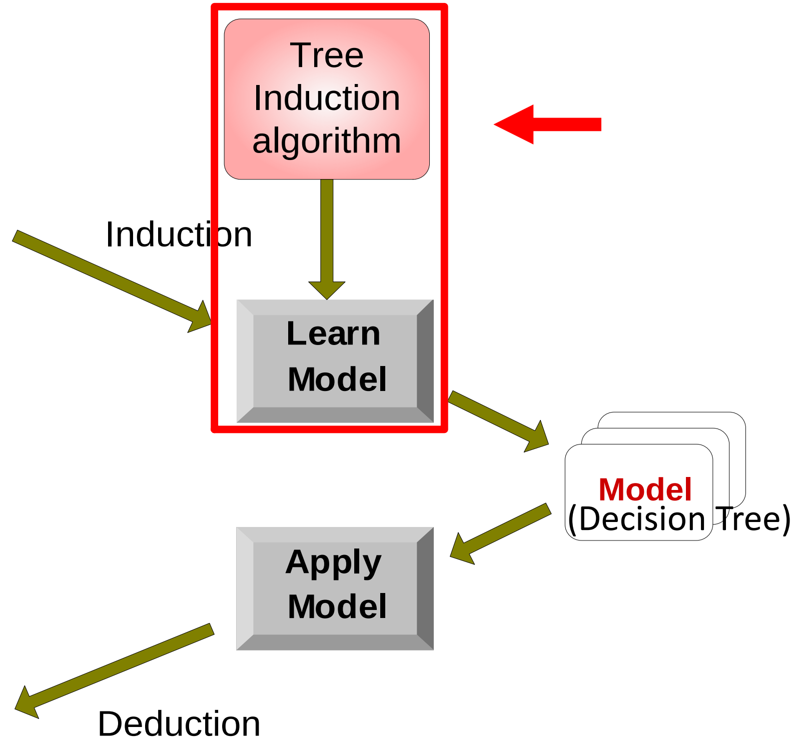
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	? NO

Assign Cheat 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

Issues: Evaluating Classification Methods

>40 ⇒ C
<40 ⇒ M
Income (in K)
Vehicle

- Accuracy
  - classifier accuracy: predicting class label
  - predictor accuracy: guessing value of predicted attributes
- Speed
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability
  - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Explainable AI

$w_1x_1 + w_2x_2 + w_3x_3 = w$

20 M
25 C
30 M
35 M
45 C
50 M
70 C
65 C

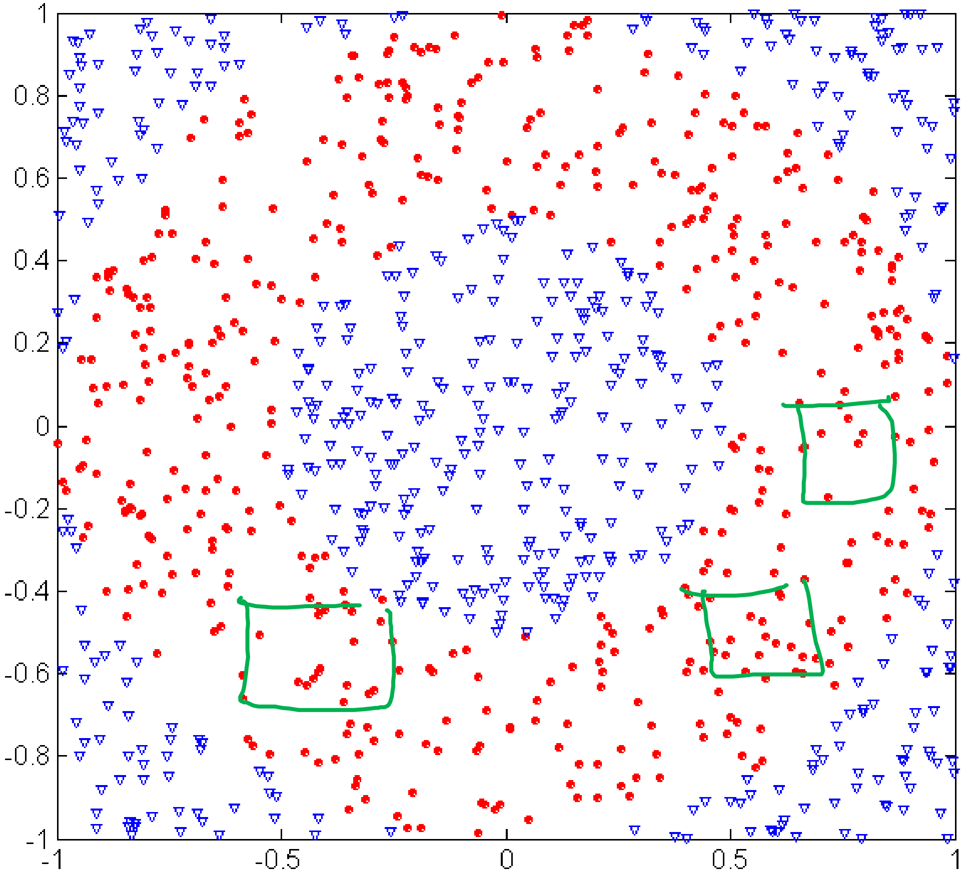
Data may have issues

2-D data 2 classes

Underfitting and Overfitting (Example)

Blue

Red



X_2 ↑

X_1 →

500 circular and 500 triangular data points.

Circular points: Red

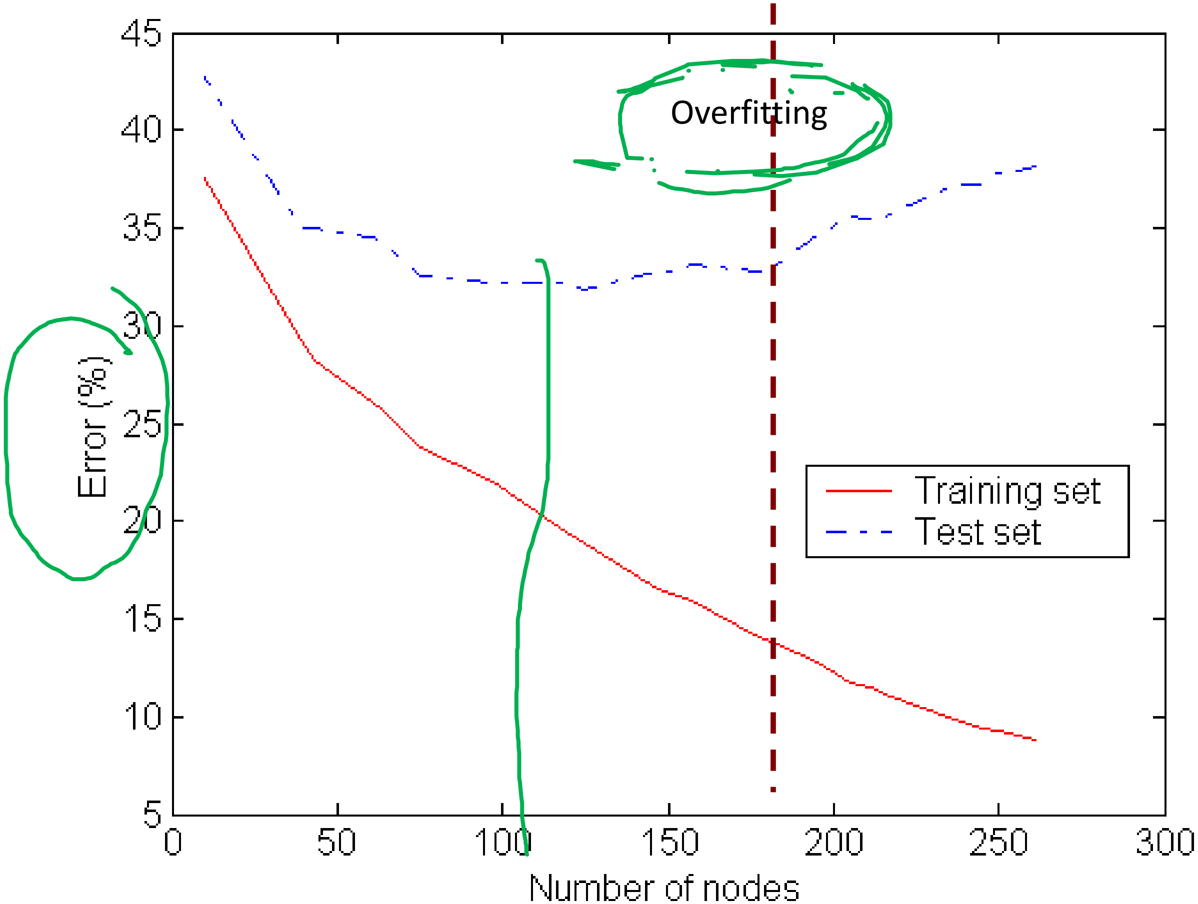
$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

Triangular points: Blue

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

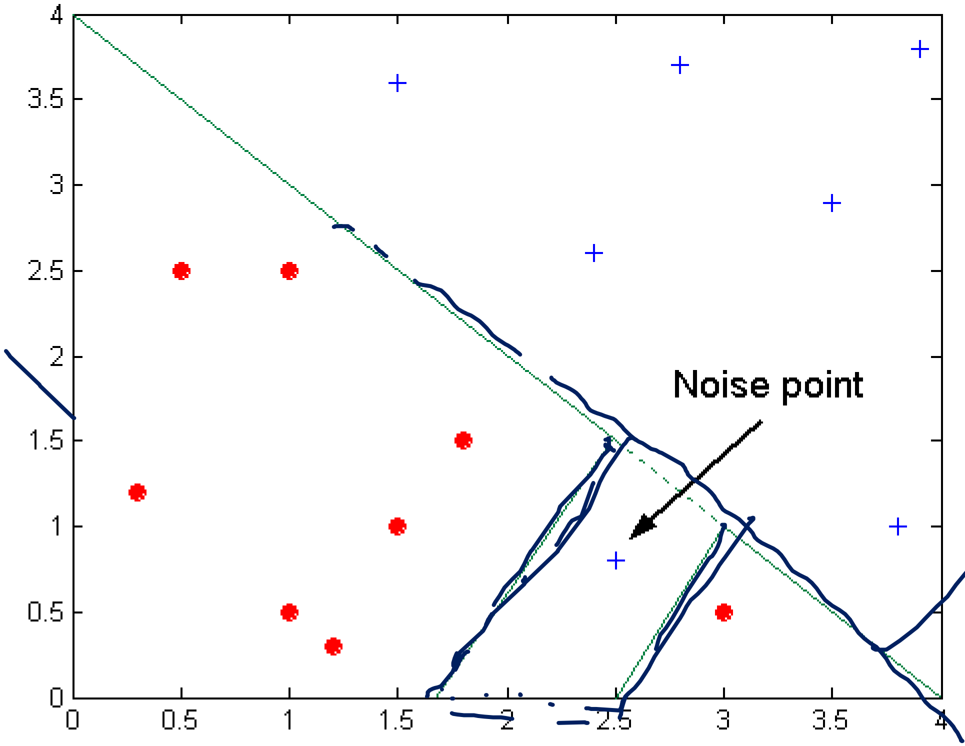
$\text{sqrt}(x_1^2+x_2^2) < 1$

Underfitting and Overfitting



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

Overfitting due to Noise



reason

boundary

Decision boundary is distorted by noise point

Decision Tree Based Classification

- Decision trees are intuitive and frequently used data mining technique for Classification
- For an analyst, they are easy to set up and for a business user they are easy to interpret.
- A decision tree model is a decision flowchart where an attribute is tested in each node and ends in a leaf node where a prediction is made.
- There are many algorithms for decision tree induction such as Hunt's Algorithm, CART, ID3, C4.5, SLIQ,SPRINT

partition train data so that we get purer subsets

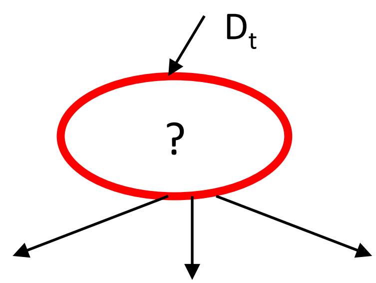
Hunt's Algorithm - Structure

- Hunt's algorithm is among the earliest. More complex algorithms were built upon it.
- It grows a decision tree in a recursive fashion by partitioning the training records into successively purer subsets
- 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.

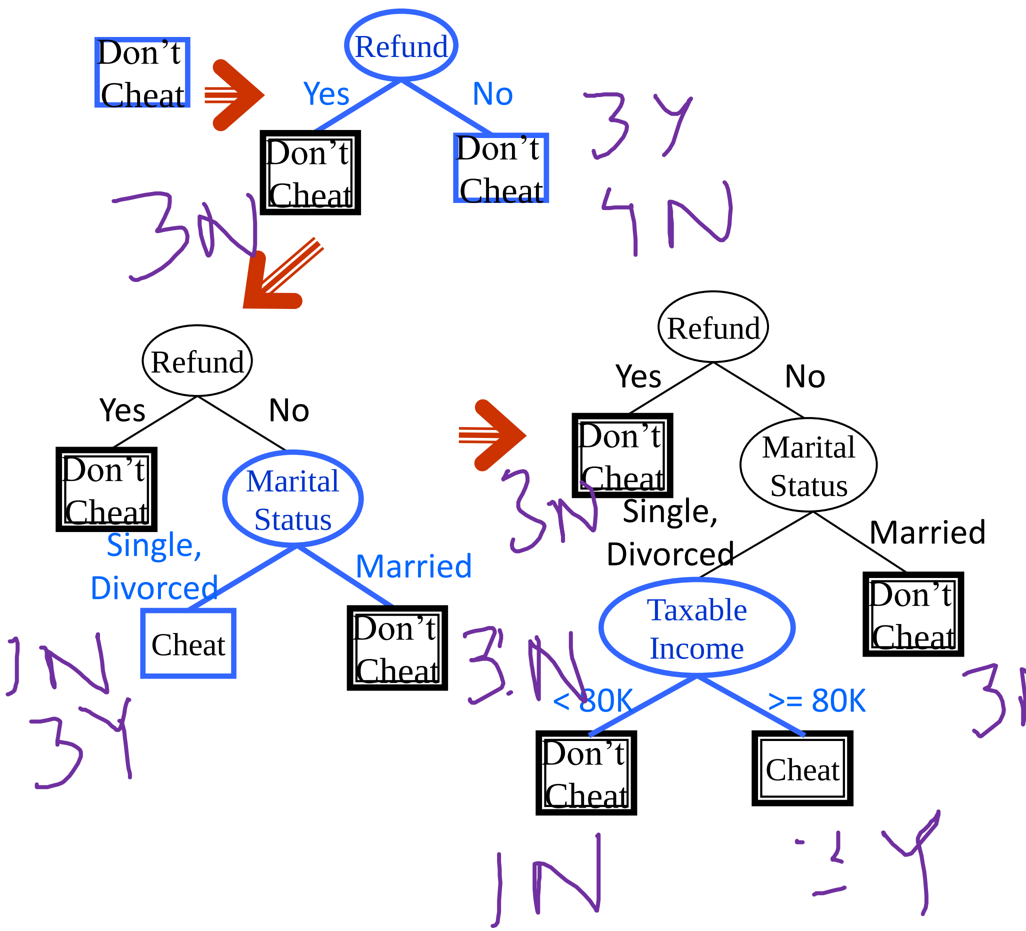
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

pure

impure



Hunt’s Algorithm - Example



Tid	Refund	Marital Status	Taxable Income	Chea
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

At every point

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

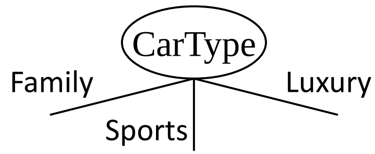
- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to Specify Test Condition?

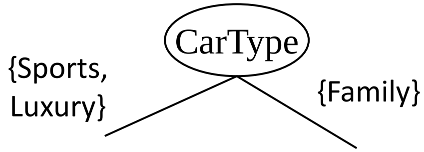
- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

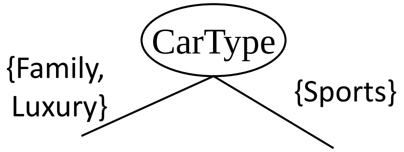
- Multi-way split: Use as many partitions as distinct values.



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

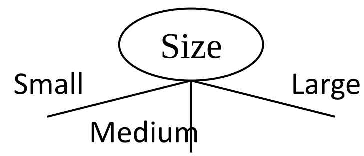


OR



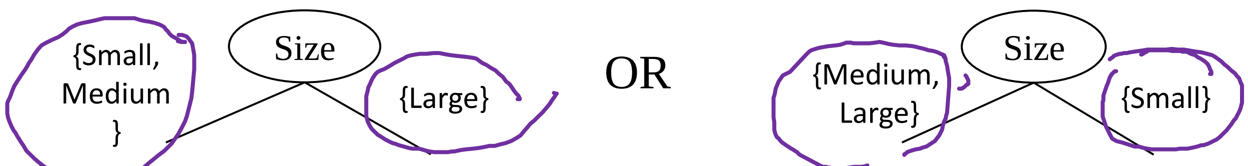
Splitting Based on Ordinal Attributes

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

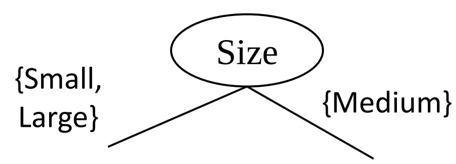


- Binary split: Divides values into two subsets. Need to find optimal partitioning.

more likely



- What about this split?



specific domain based

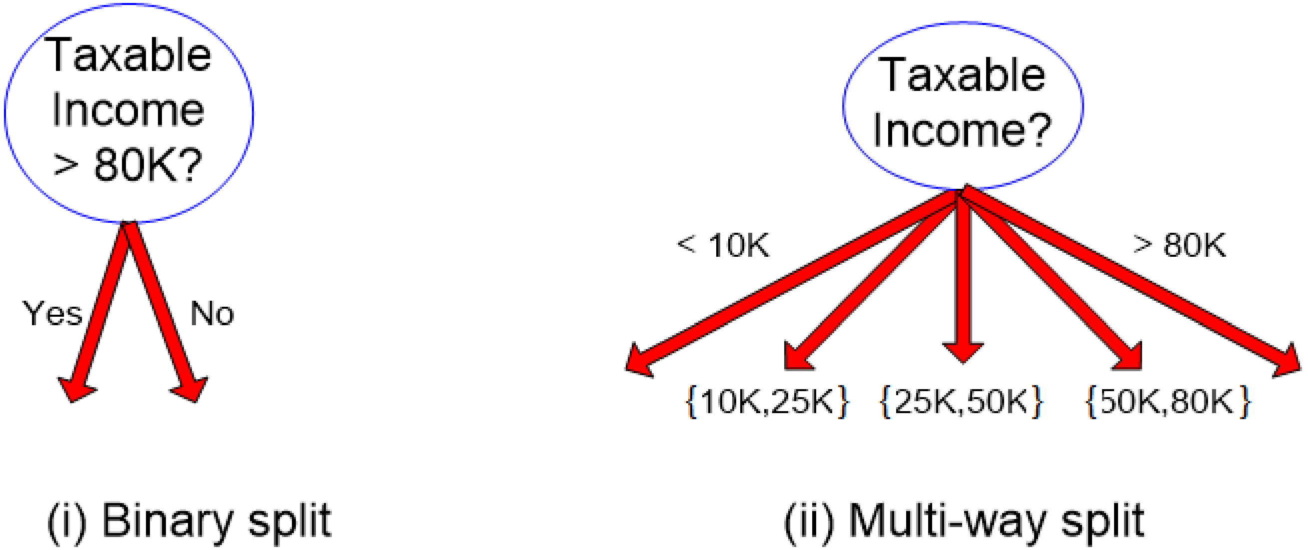
Splitting Based on Continuous Attributes

Income

Low Mid High

- 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 ≥ v)
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



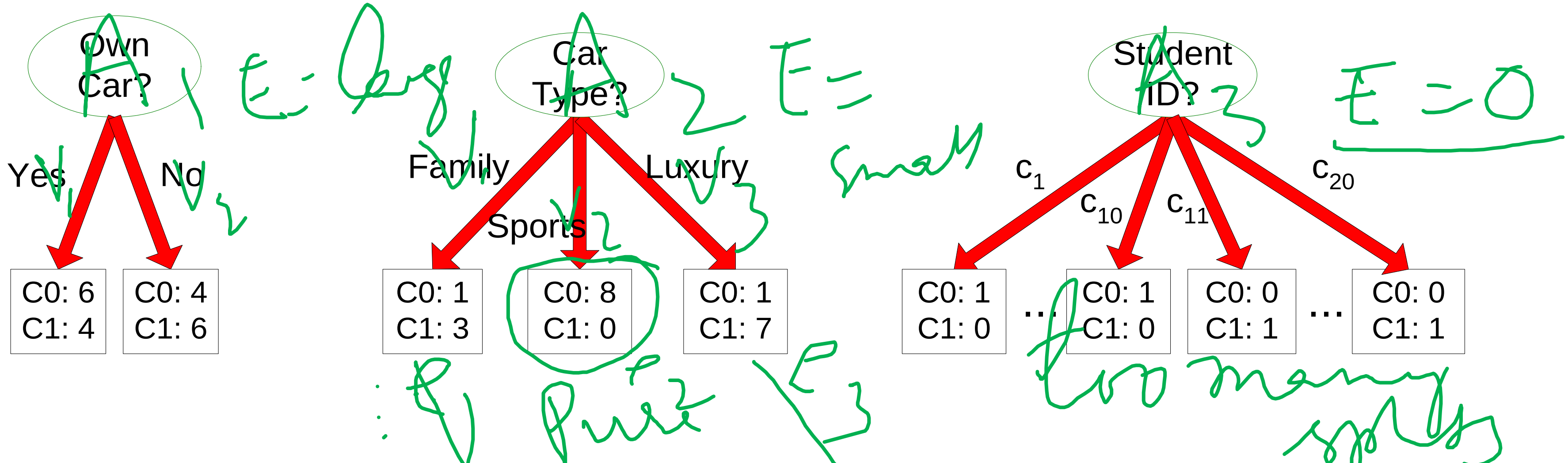
20 training tuples

How to determine the Best Split

$\frac{4}{20} e_1 + \frac{8}{20} e_2 + \frac{8}{20} e_3$ 3 attributes

Before Splitting: 10 records of class 0,
10 records of class 1

completely impure



Which test condition is the best?

nearly pure

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred

- Need a measure of node impurity:

Bad

C0: 5	C0: 9
C1: 5	C1: 1

Good

Non-homogeneous,	Homogeneous,
High degree of impurity	Low degree of impurity

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

Finding the Best Split

D_0 I_0

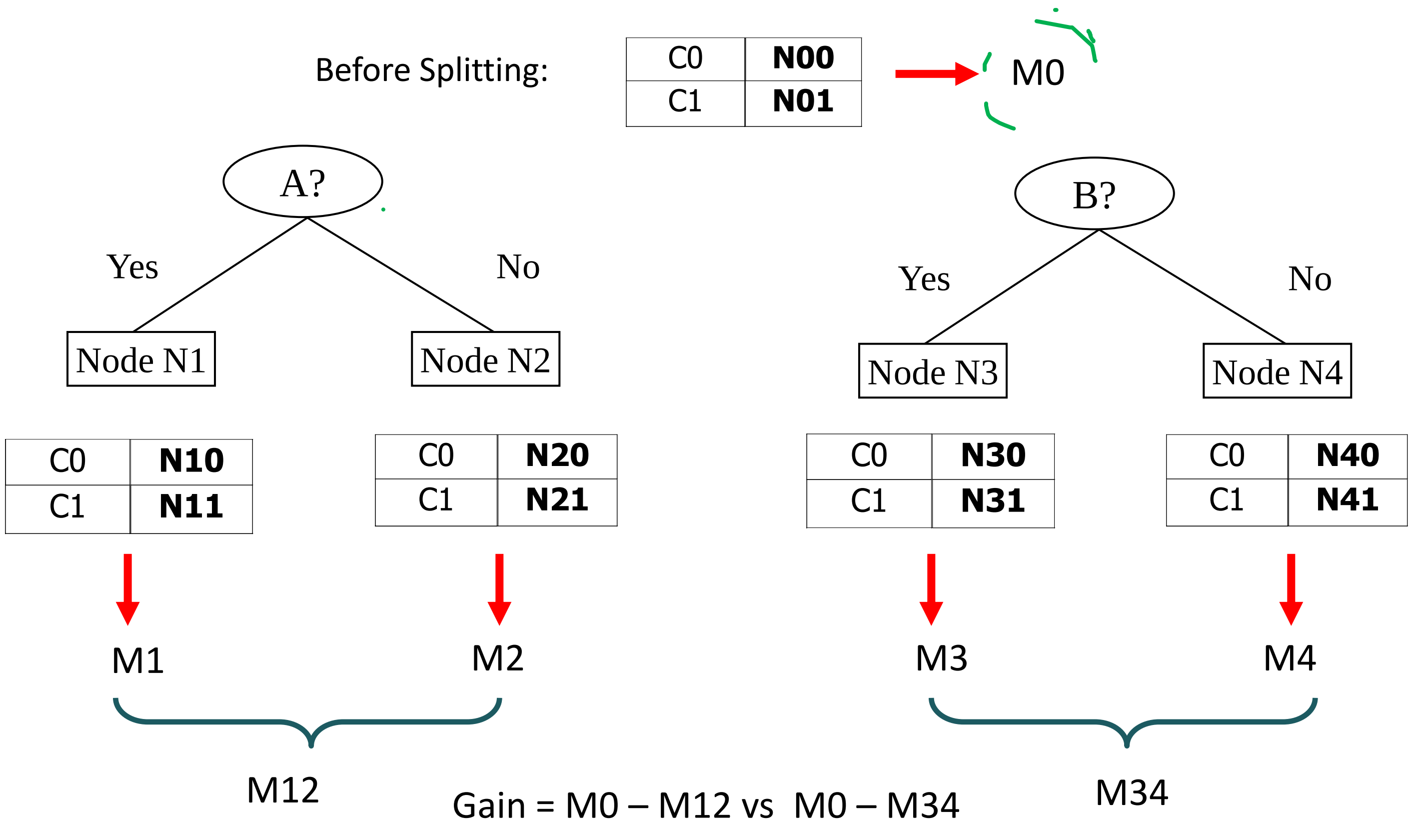
D_1 I_1 D_2 I_2

1. Compute impurity measure (M_0) before splitting
2. Compute impurity measure (M) after splitting
   - Compute impurity measure of each child node
   - M is the weighted impurity of children
3. Choose the attribute test condition that produces the highest gain

$$\text{Gain} = M_0 - M$$

or equivalently, lowest impurity measure after splitting (M)

How to Find the Best Split



Gain = M0 – M12 vs M0 – M34

Measure of Impurity: GINI

$1 - \sum_j p_j^2$

- 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	**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	

$p(C_1) = 1/6$ $p(C_2) = 5/6$

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

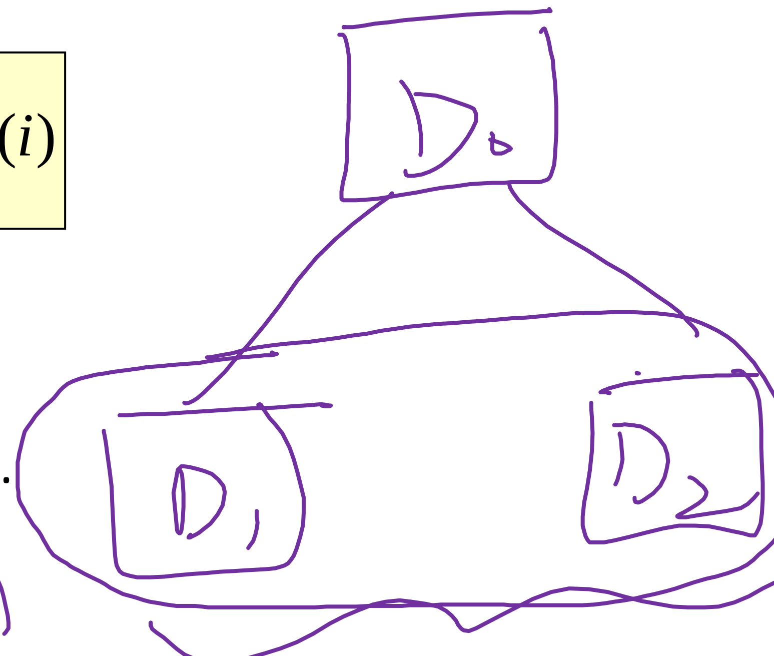
C_1 3
C_2 3

$1 - (3/6)^2 - (3/6)^2 = 0.5$

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,

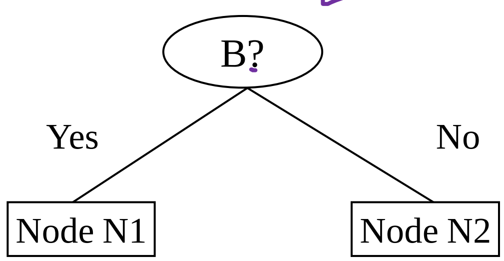
n = number of records at node p.

$$\frac{\|D_1\|}{\|D_1\| + \|D_2\|} * G_1 + \frac{\|D_2\|}{\|D_1\| + \|D_2\|} * G_2$$

Binary Attributes: Computing GINI Index

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

12



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

Gini(N1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.408$

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

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

Gini(Children)
$= 7/12 * 0.408 +$
$\quad 5/12 * 0.32$
$= 0.37$

Weighted Sum

$0.5 - 0.37 = 0.13$

Categorical Attributes: Computing Gini Index

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

Go to

Multi-way split

more complex tree

	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	

less complex tree

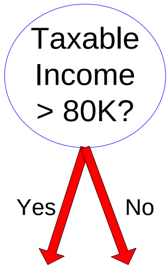
Greatest gain

Income

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



4 < 4N 0Y

6 < 3Y 3N

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	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	

Keep trying multiple split points

0 to 2 8 classes 0-3

Alternative Splitting Criteria

- Entropy at a given node t:

$$Entropy(t) = -\sum_{j} p(j \mid t) \log_2 p(j \mid t)$$

(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

$0 \text{ to } \log_2 n_c$

Examples for computing Entropy

$p(C_1) \times \log_2(p(C_1)) + p(C_2) \times \log_2(p(C_2))$

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

$\log_2 p = \dfrac{\log p}{\log 2}$

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

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

Entropy = – 0 * $\log_2$ 0 – 1 * $\log_2$ 1 = – 0 – 0 = 0

$\log_2 1 = 0$

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

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

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

−2.59   −.26

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

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

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

$1/6 \times 2.59 + 5/6 \times 0.26$

$0.43 + 0.22 = 0.65$

Splitting Criteria 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

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

Classification

Examples for Computing Error

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

0

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

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

Error = 1 – max (0, 1) = 1 – 1 = 0

$1 -$

1/6

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

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

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = $\boxed{1/6}$

1/3

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

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

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = $\boxed{1/3}$

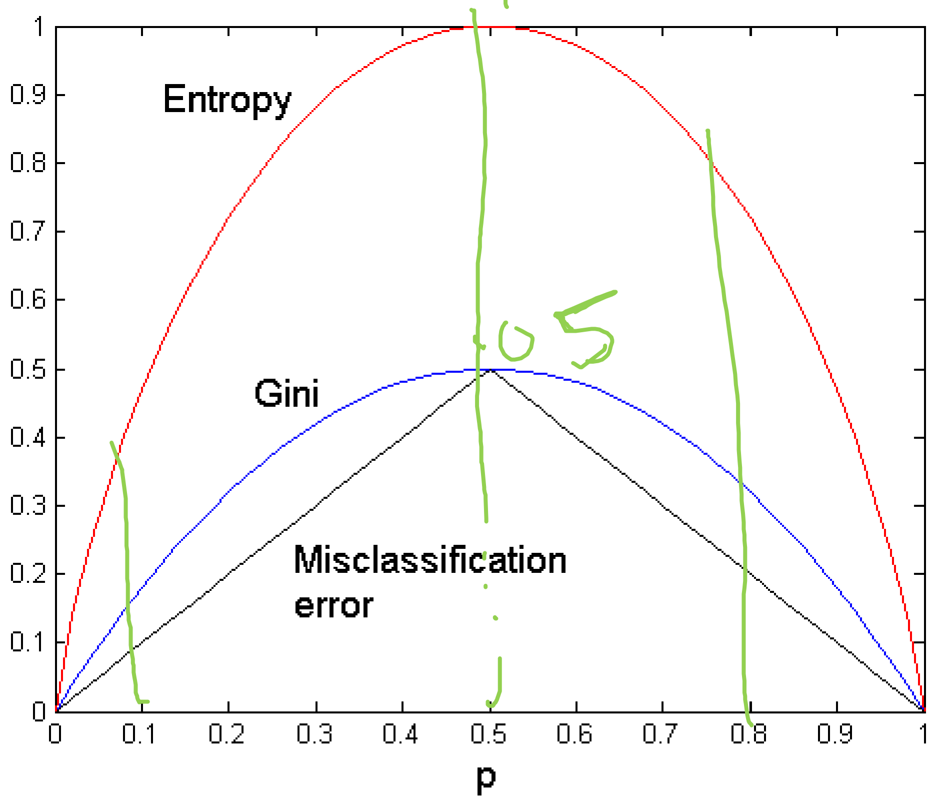
Comparison among Splitting Criteria

For a 2-class problem:

Better

0.1, 0.5

0.3, 0.7



Gain Ratio

Going beyond entropy

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

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

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Attribute Selection: Information Gain

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2)$$
$$= 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

16 paths dividing data equally

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)
  - GainRatio(A) = Gain(A)/SplitInfo(A)

$$\boxed{SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)}$$

$$-\sum_{}^{16} \frac{1}{16} \times \log_2 \frac{1}{16} =$$

- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  - gain_ratio(income) = 0.029/1.557 = 0.019

$$16 \times \frac{1}{16} \times 4$$

- The attribute with the maximum gain ratio is selected as the splitting attribute

$$= 4$$

Refining Decision Tree Model

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

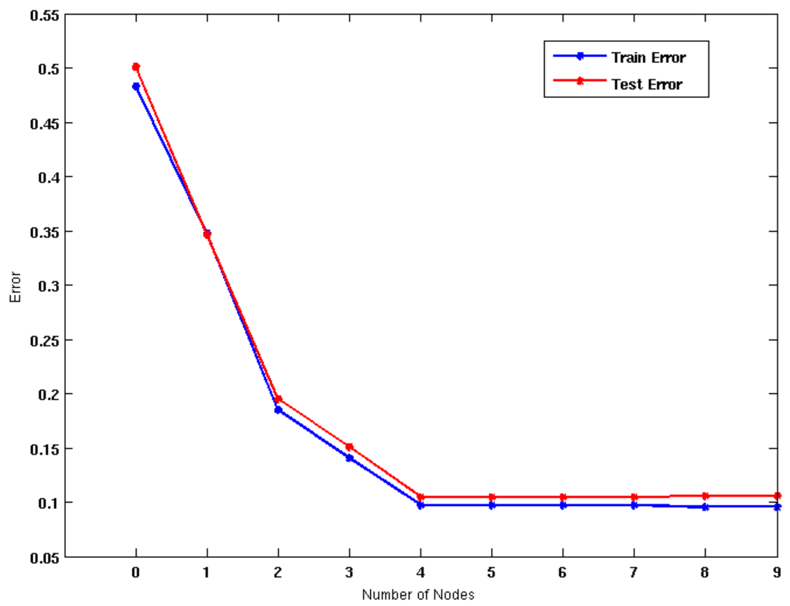
Practical Issues of Classification

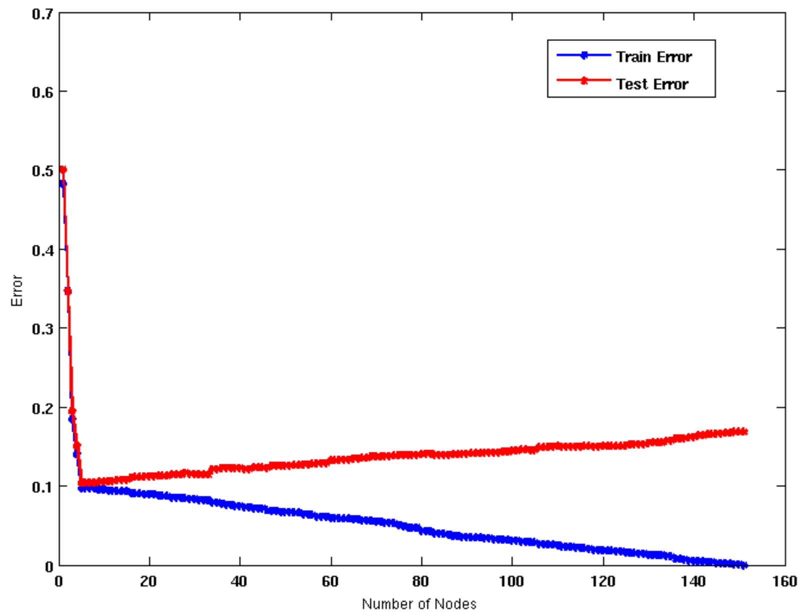
- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Underfitting vs. Overfitting

- Underfitting results in decision trees that are too simple to solve the problem. They may offer superior interpretability.
- 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

Model Overfitting





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

Overfitting: when model is too complex, training error is small but test error is large

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - General 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 (for pre-pruning) :
    - 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 χ^2 test)
    - 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(i.e. expected error of the model on previously unseen records) 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

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

Prescribed Text Books

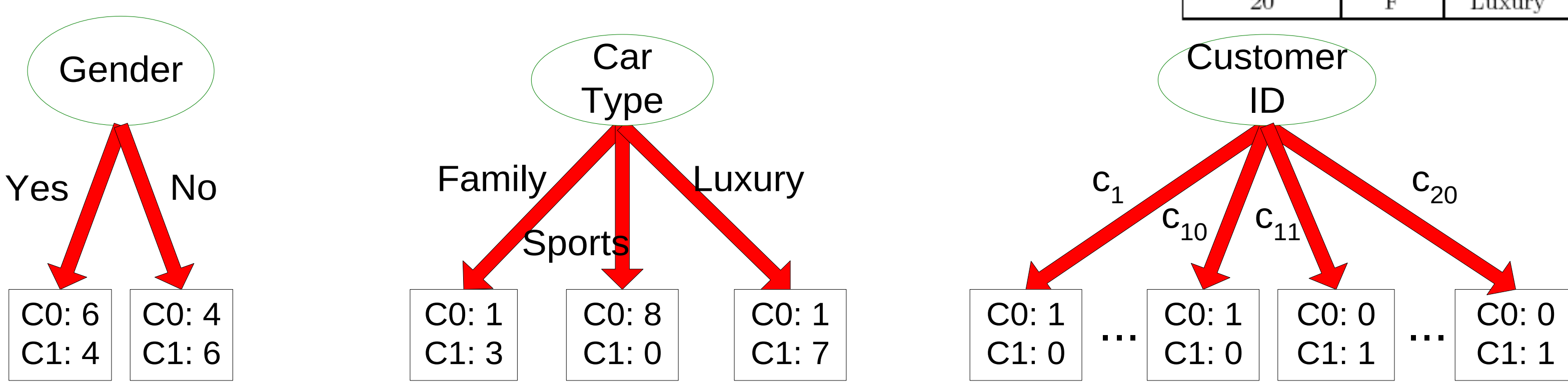
	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T4	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
	Principles of Data Mining, Second Edition by Max Bramer Springer © 2013

Thank You

How to determine the Best Split

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

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?