CS 412 Intro. to Data Mining

Chapter 8. Classification: Basic Concepts

Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017

Chapter 8. Classification: Basic Concepts

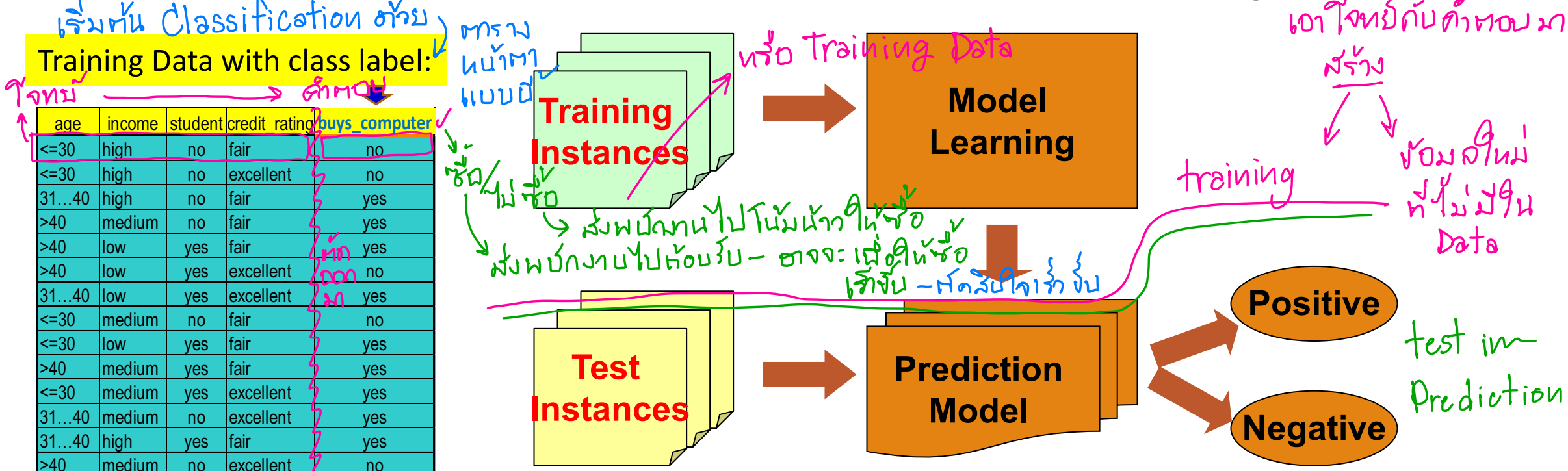
- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Linear Classifier
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Additional Concepts on Classification
- Summary

Supervised vs. Unsupervised Learning (1)

- Supervised learning (classification)
  - Supervision: The training data such as observations or measurements are accompanied by **labels** indicating the classes which they belong to
  - New data is classified based on the models built from the training set

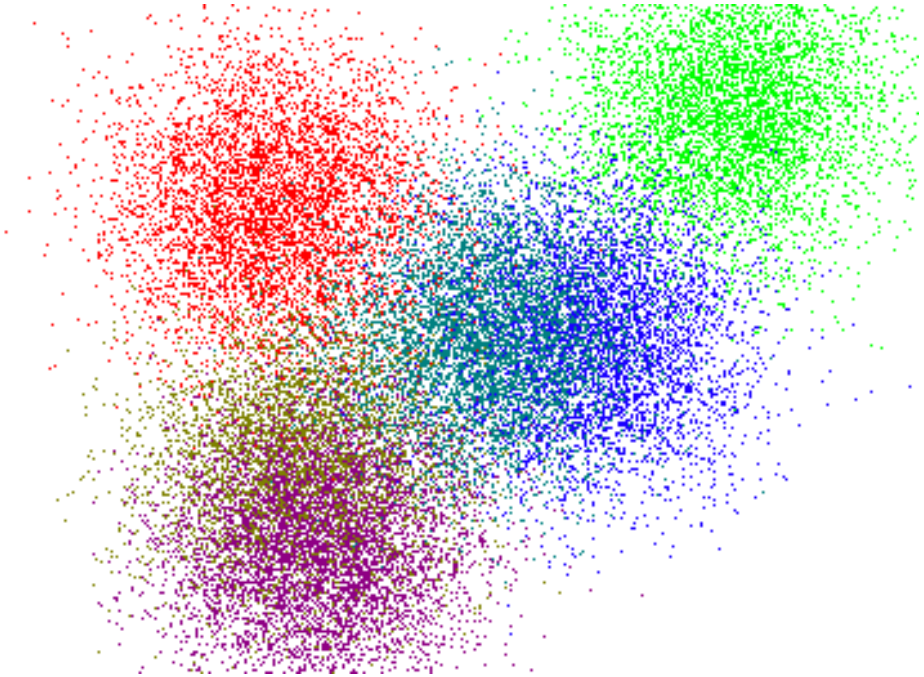
Training Data with class label:

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



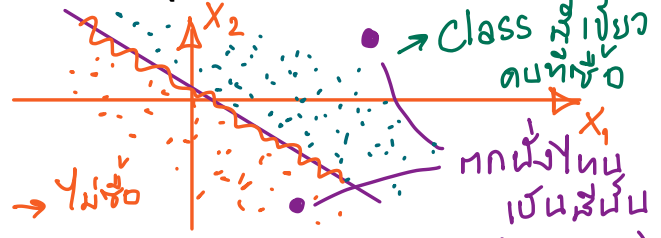
Supervised vs. Unsupervised Learning (2)

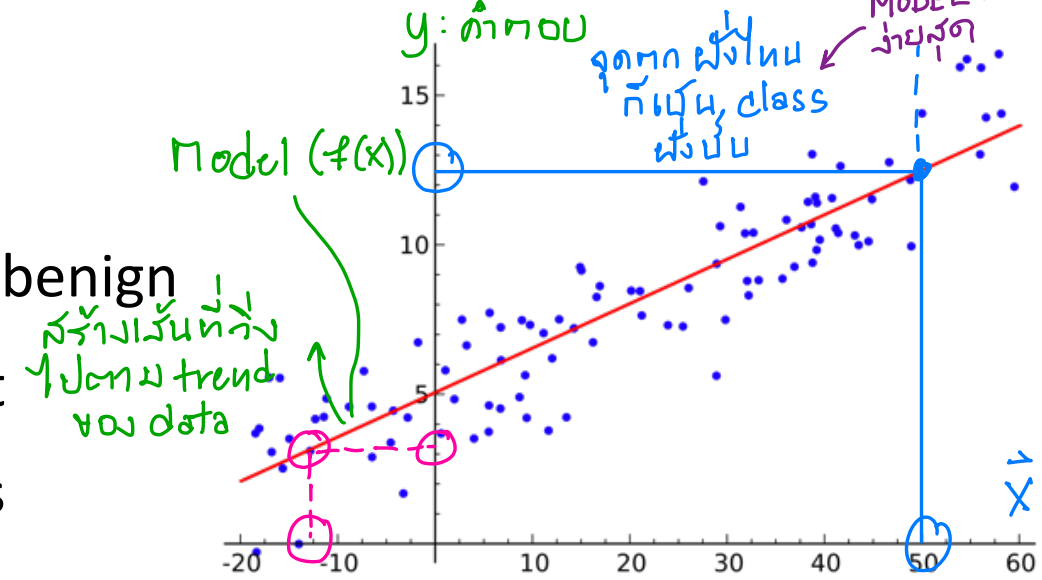
- Unsupervised learning (clustering)
  - The class labels of training data are unknown
  - Given a set of observations or measurements, establish the possible existence of classes or clusters in the data



Prediction Problems: Classification vs. Numeric Prediction

- Classification
  - Predict categorical class labels (discrete or nominal)
  - Construct a model based on the training set and the **class labels** (the values in a classifying attribute) and use it in classifying new data
- Numeric prediction
  - Model continuous-valued functions (i.e., predict unknown or missing values)
- Typical applications of classification
  - Credit/loan approval
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is





Classification—Model Construction, Validation and Testing

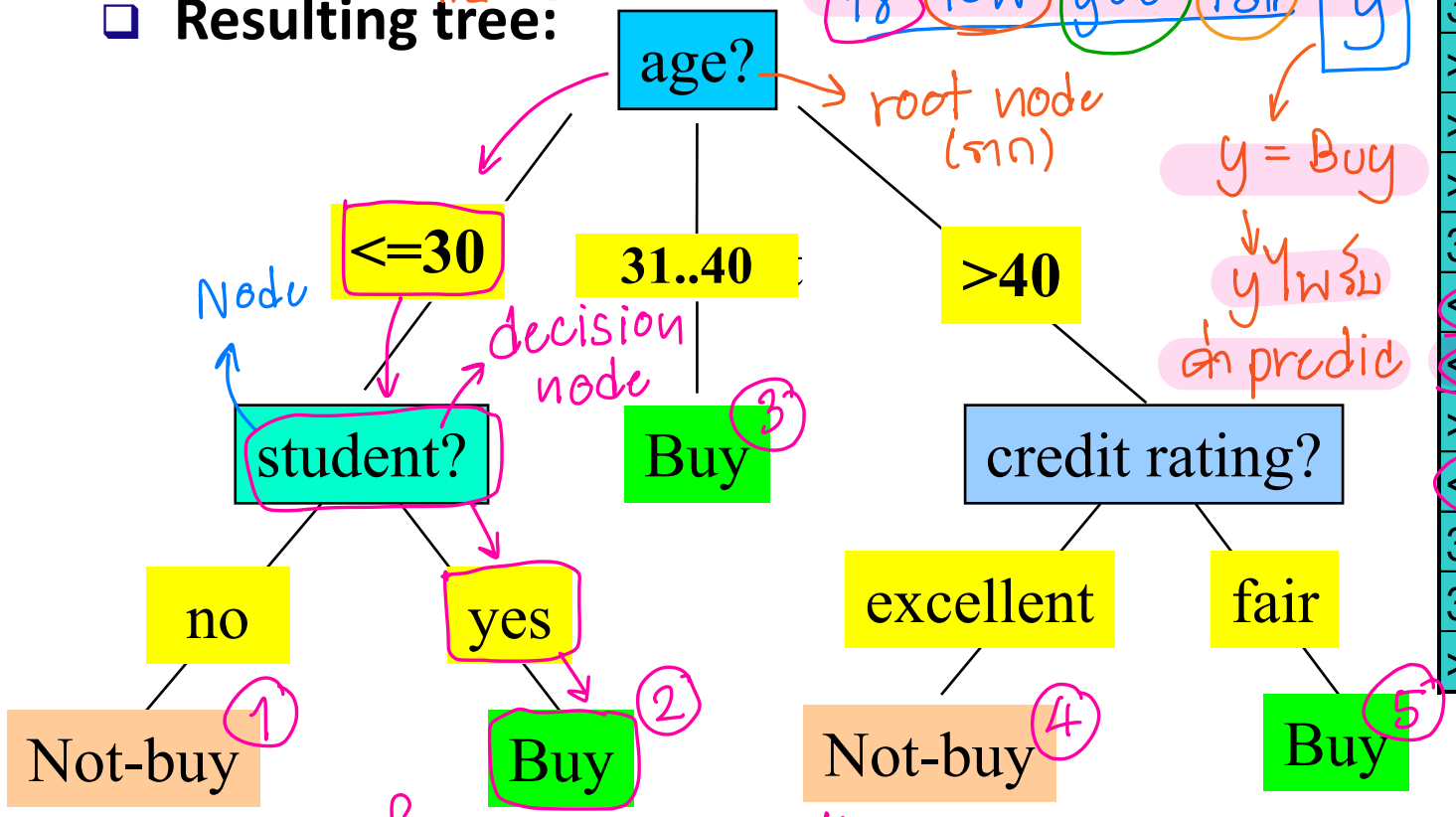
- **Model construction**
  - Each sample is assumed to belong to a predefined class (shown by the **class label**)
  - The set of samples used for model construction is **training set**
  - Model: Represented as decision trees, rules, mathematical formulas, or other forms
- **Model Validation and Testing**:
  - **Test:** Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - *Accuracy:* % of test set samples that are correctly classified by the model
    - Test set is independent of training set
  - **Validation**: If *the test set* is used to select or refine models, it is called **validation** (or development) **(test) set**
- **Model Deployment:** If the accuracy is acceptable, use the model to classify new data

Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Linear Classifier
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Additional Concepts on Classification
- Summary

Decision Tree Induction: An Example

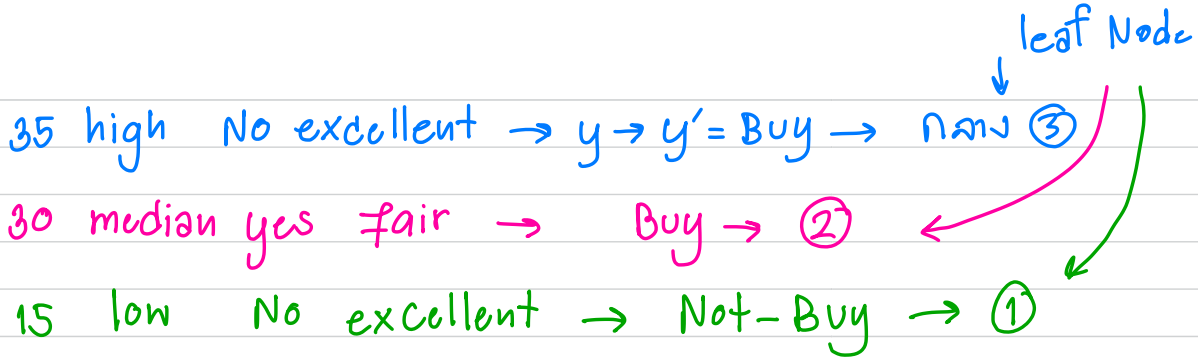
- **Decision tree construction:**
  - A top-down, recursive, divide-and-conquer process
- **Resulting tree:**



Training data set: Who buys computer?

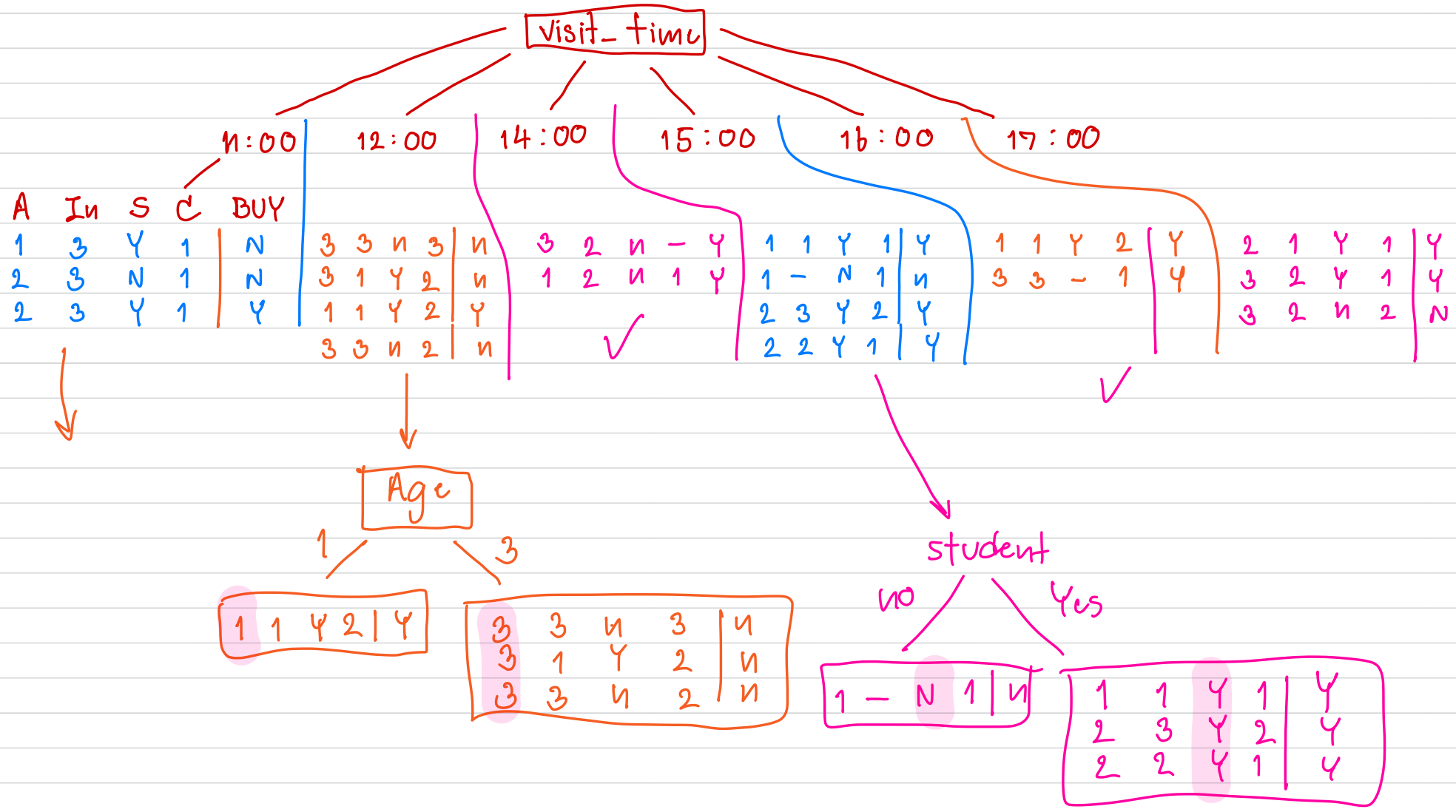
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent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

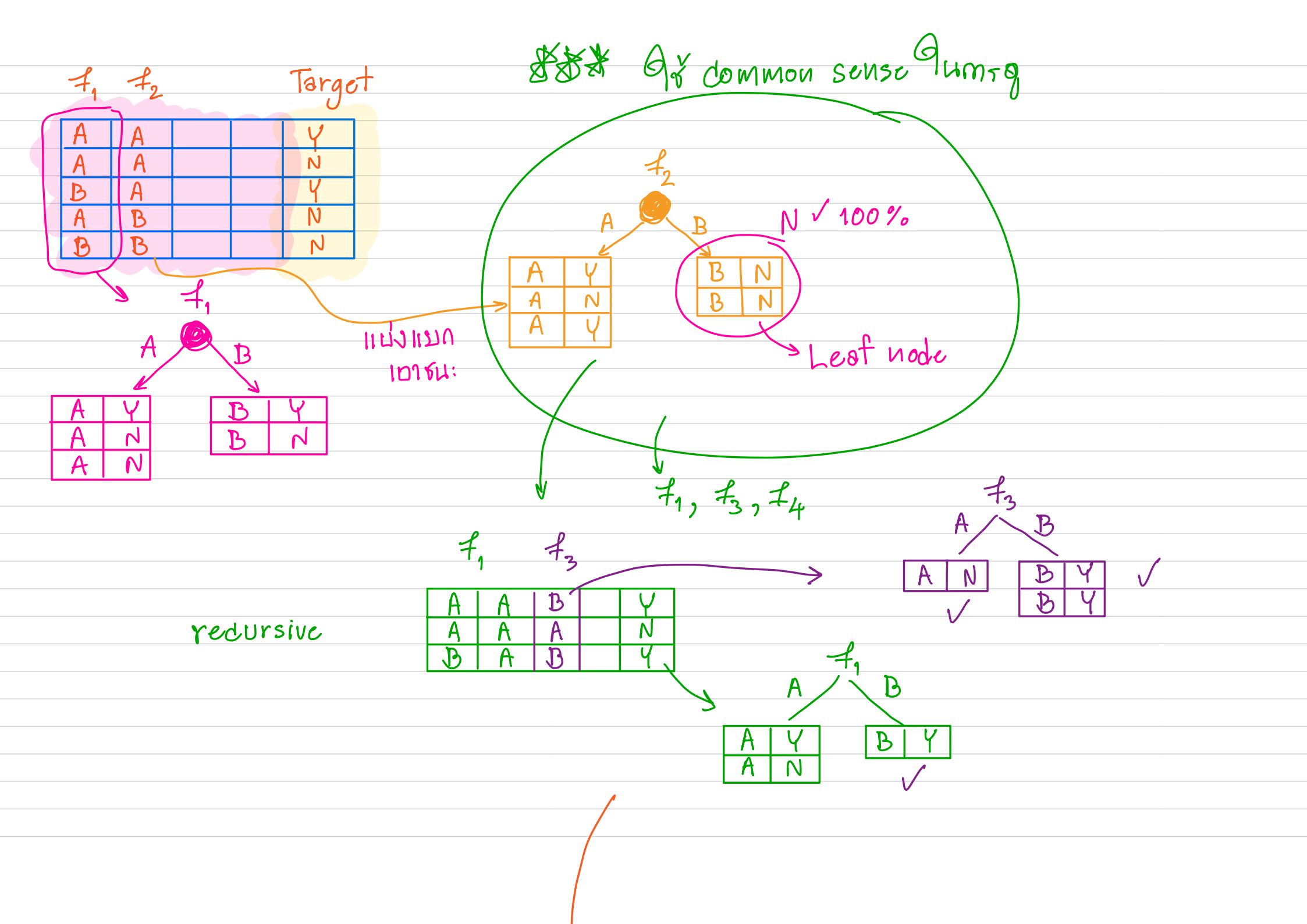
Note: The data set is adapted from "Playing Tennis" example of R. Quinlan

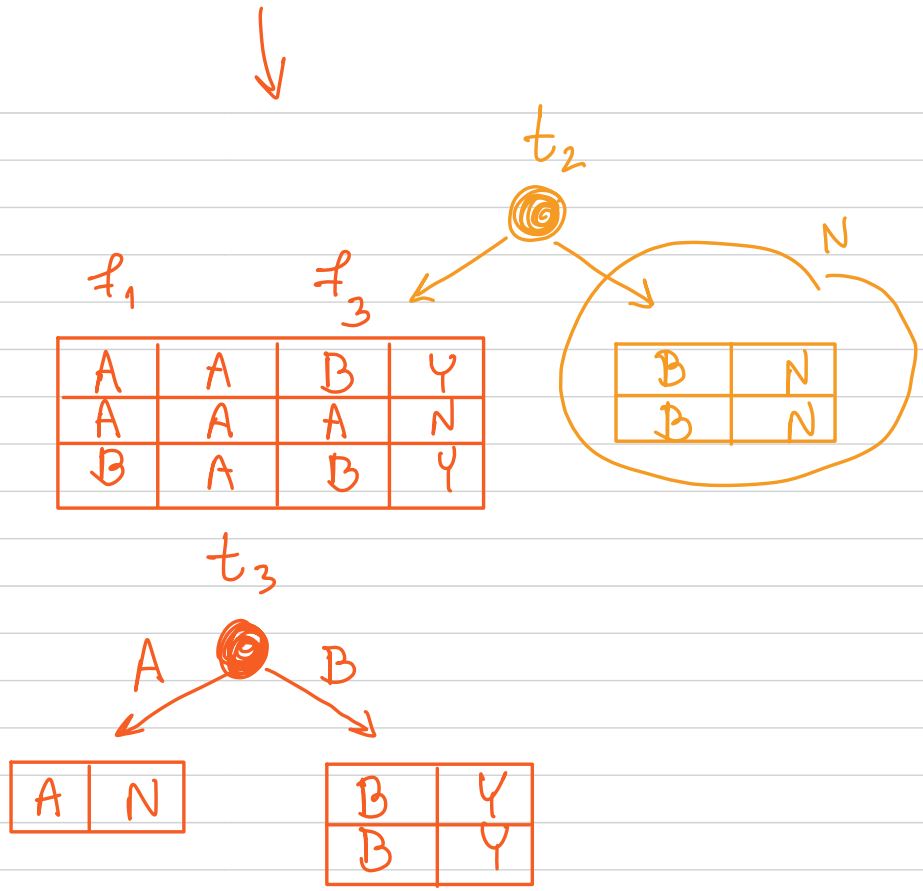


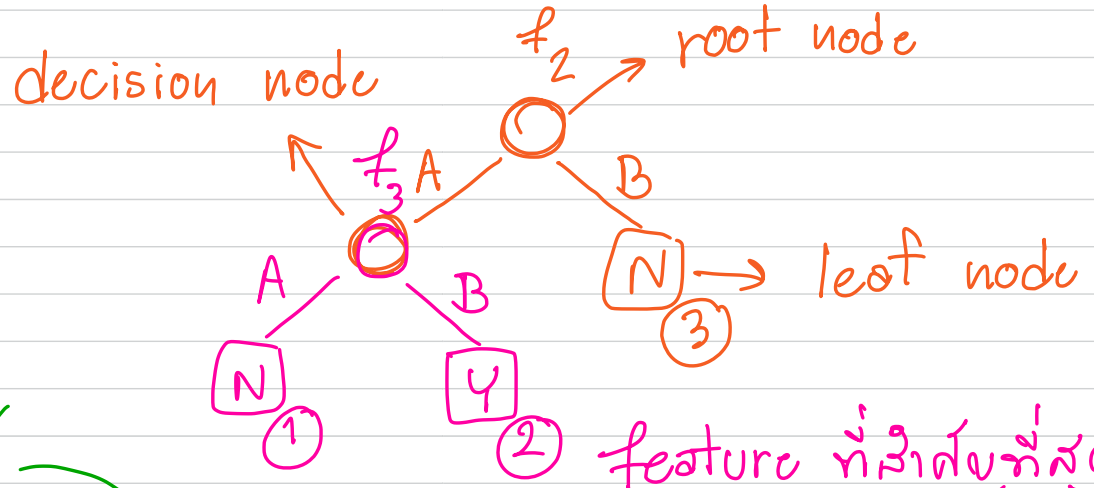
✦ เลือก root node → Data ที่แบ่งข้อมูลได้ดีที่สุด

1	<=30	low	no	fair	no
2	31...40	medium	yes	ex	yes
3	>40	high			


Visit_time
11:00
12:00
14:00
15:00
16:00
17:00
A In S C BUY
1 3 Y 1 N
2 3 N 1 N
2 3 Y 1 Y
3 3 N 3 N
3 1 Y 2 N
1 1 Y 2 Y
3 3 N 2 N
3 2 N - Y
1 2 N 1 Y
1 1 Y 1 Y
1 - N 1 N
2 3 Y 2 Y
2 2 Y 1 Y
1 1 Y 2 Y
3 3 - 1 Y
2 1 Y 1 Y
3 2 Y 1 Y
3 2 N 2 N
Age
1
3
1 1 Y 2 Y
3 3 N 3 N
3 1 Y 2 N
3 3 N 2 N
student
No
Yes
1 - N 1 N
1 1 Y 1 Y
2 3 Y 2 Y
2 2 Y 1 Y


f1 f2 Target
A A Y
A A N
B A Y
A B N
B B N
f1
A B
A Y
A N
A N
B Y
B N
แบ่งแยกเอาธน:
*** ดูที่ common sense ด้วย
f2
A B
A Y
A N
A Y
B N
B N
N ✓ 100%
Leaf node
f1, f3, f4
f1 f3
A A B Y
A A A N
B A B Y
redursive
f3
A B
A N
B Y
B Y
✓
✓
f1
A B
A Y
A N
B Y
✓


t_2
t_1
t_3
N
A
B
Y
N
A
N
B
Y



ใช้ feature ซ้ำไม่ได้

กรณี ถ้าไม่ป f_3

f_1 → A Y 50% N 50%
B N 100%

feature ที่สำคัญที่สุด

ดูจาก root node

ข้อมูล

Top-down

ดูภาพรวม

recursive

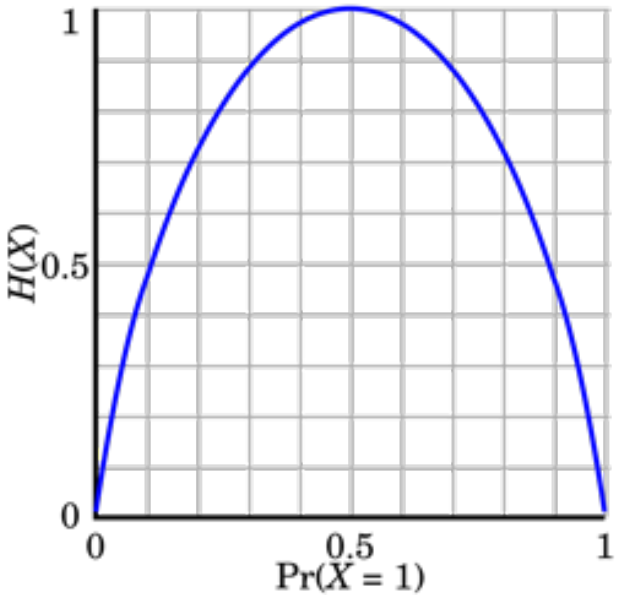
From Entropy to Info Gain: A Brief Review of Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random number
  - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, y_2, \ldots, y_m\}$

$$H(Y) = -\sum_{i=1}^{m} p_i \log(p_i) \quad where\ p_i = P(Y = y_i)$$

  - Interpretation
    - Higher entropy → higher uncertainty
    - Lower entropy → lower uncertainty
- Conditional entropy

$$H(Y|X) = \sum_{x} p(x) H(Y|X = x)$$



m = 2

Information Gain: An Attribute Selection Measure

- Select the attribute with the highest information gain (used in typical decision tree induction algorithm: ID3/C4.5)
- 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:

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

class yes มา sum รวมกับ class No

class ตามน่าจะเป็นที่จะเกิด class ที่ m

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

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

✓ Feature

แบ่งข้อมูลได้ดี

- Information gained by branching on attribute A

อยากได้

ค่าเยอะๆ → เลือกเป็น root node

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

Node ที่ดี ↳ มี

มากที่สุด

ถูกแบ่งโดย feature A แล้ว

Example: Attribute Selection with Information Gain

- Class P: buys_computer = "yes" = 9
- Class N: buys_computer = "no" = 5

} 14

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

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

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

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

feature

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

Hence

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

Similarly, we can get

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

entropy ↗ ความวุ่นวาย ↘ ยุ่งยาก ↓ แบ่งยาก

เลือกมาเป็น root node

Example: Attribute Selection with Information Gain

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

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

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

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

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

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

Hence

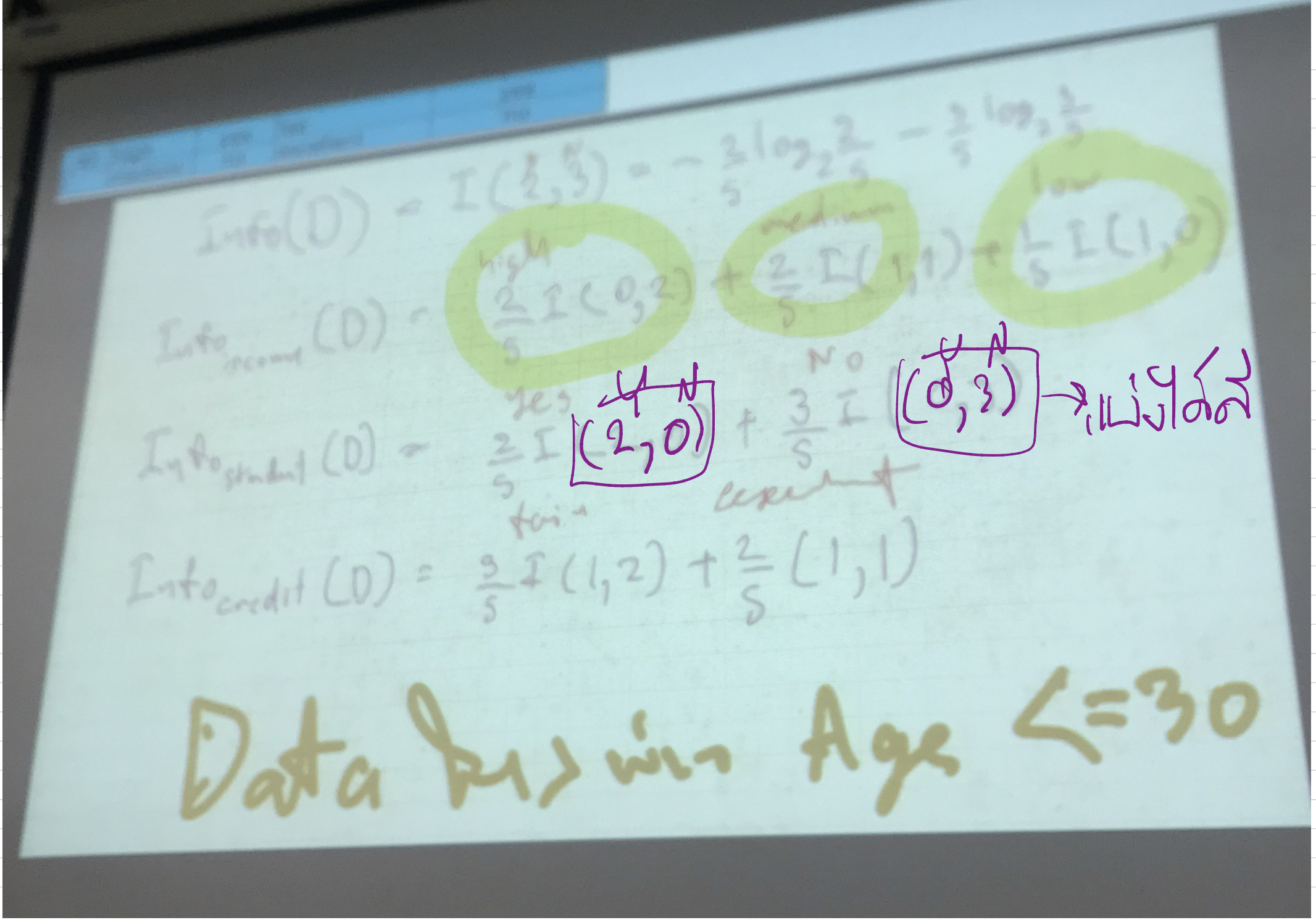
$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly, we can get

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$


Info(D) = I(9,5) = −9/14 log2 9/14 − 5/14 log2 5/14
Info_income(D) = 4/14 I(3,1) + 6/14 I(4,2) + 4/14 I(2,2)
high
medium
low
yes
no
Info_student(D) = 7/14 I(6,1) + 7/14 I(3,4)
(6,1)
(3,4)
Info_credit(D) = 8/14 I(6,2) + 6/14 I(3,3)
Data ให้ผล Age <=30

no

Age	Income	Credit_rating	buys_computer
<=30	high	fair	no
<=30	high	excellent	no
>30	high	fair	yes
<=30	low	fair	yes
<=30	low	fair	no
>30	low	excellent	yes
>30	low	excellent	no

yes

Age	Income	Credit_rating	buys_computer
<=30	high	fair	yes
>30	low	excellent	no
>30	high	excellent	yes
<=30	low	fair	yes
<=30	low	fair	yes
<=30	low	excellent	yes
>30	high	fair	yes

Student : Yes

Age	Income	Credit_rating	buys_computer
<=30	high	fair	no
<=30	high	excellent	no
>30	high	fair	yes
<=30	low	fair	yes
<=30	low	fair	no
>30	low	excellent	yes
>30	low	excellent	no

Student : No

Age	Income	Credit_rating	buys_computer
<=30	high	fair	yes
>30	low	excellent	no
>30	high	excellent	yes
<=30	low	fair	yes
<=30	low	fair	yes
<=30	low	excellent	yes
>30	high	fair	yes

ตาราง Student : no

$$Info_{student:no}(D) = I(3,4) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right) = 0.985$$

$$Info_{age}(D) = \frac{4}{7}I(1,3) + \frac{3}{7}I(2,1) = \frac{4}{7}(0.811) + \frac{3}{7}(0.918)$$
$$= 0.857$$

age	p_i	n_i	p_i+n_i	$I(p_i+n_i)$
<= 30	1	3	4	
> 30	2	1	3	

$$I(1,3) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2\left(\frac{3}{4}\right) = 0.811$$

$$I(2,1) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 0.918$$

$$Info_{income}(D) = \frac{3}{7}I(1,2) + \frac{4}{7}I(2,2) = \frac{3}{7}(0.918) + \frac{4}{7}(1)$$
$$= 0.965$$

income	p_i	n_i	p_i+n_i	$I(p_i+n_i)$
high	1	2	3	
low	2	2	4	

$$I(1,2) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

$$I(2,2) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

$$Info_{credit\_rating}(D) = \frac{3}{7}I(1,2) + \frac{4}{7}I(2,2) = \frac{3}{7}(0.918) + \frac{4}{7}(1)$$
$$= 0.965$$

$$Gain(age) = Info_{student:no} - Info_{age}(D)$$
$$= 0.985 - 0.857 = 0.128$$

$$Gain(income) = Info_{student:no} - Info_{income}(D)$$
$$= 0.985 - 0.965 = 0.02$$

$$Gain(credit\_rating) = Info_{student:no} - Info_{credit\_rating}(D)$$
$$= 0.985 - 0.965 = 0.02$$ ※

credit_rating	p_i	n_i	p_i+n_i	$I(p_i+n_i)$
excellent	1	2	3	
fair	2	2	4	

$$I(1,2) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

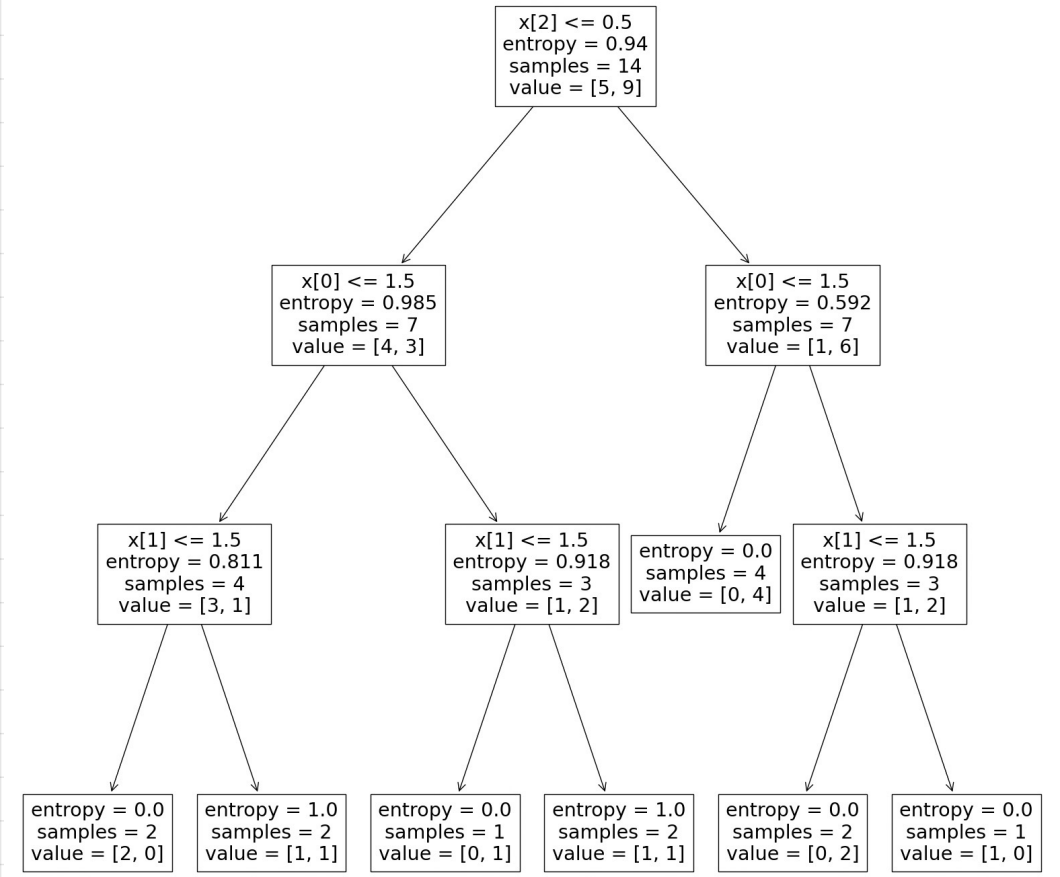
$$I(2,2) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

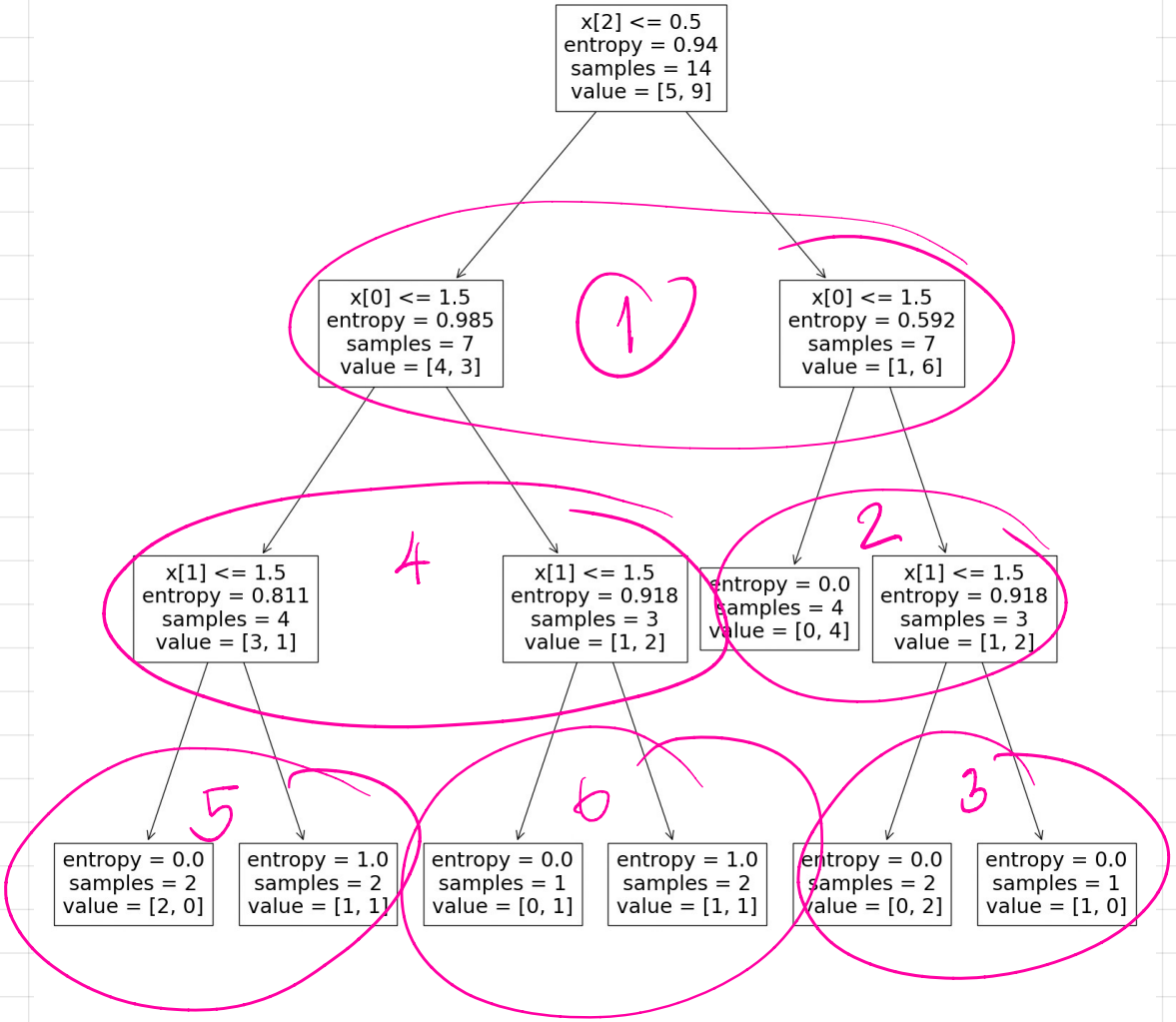
Student : No

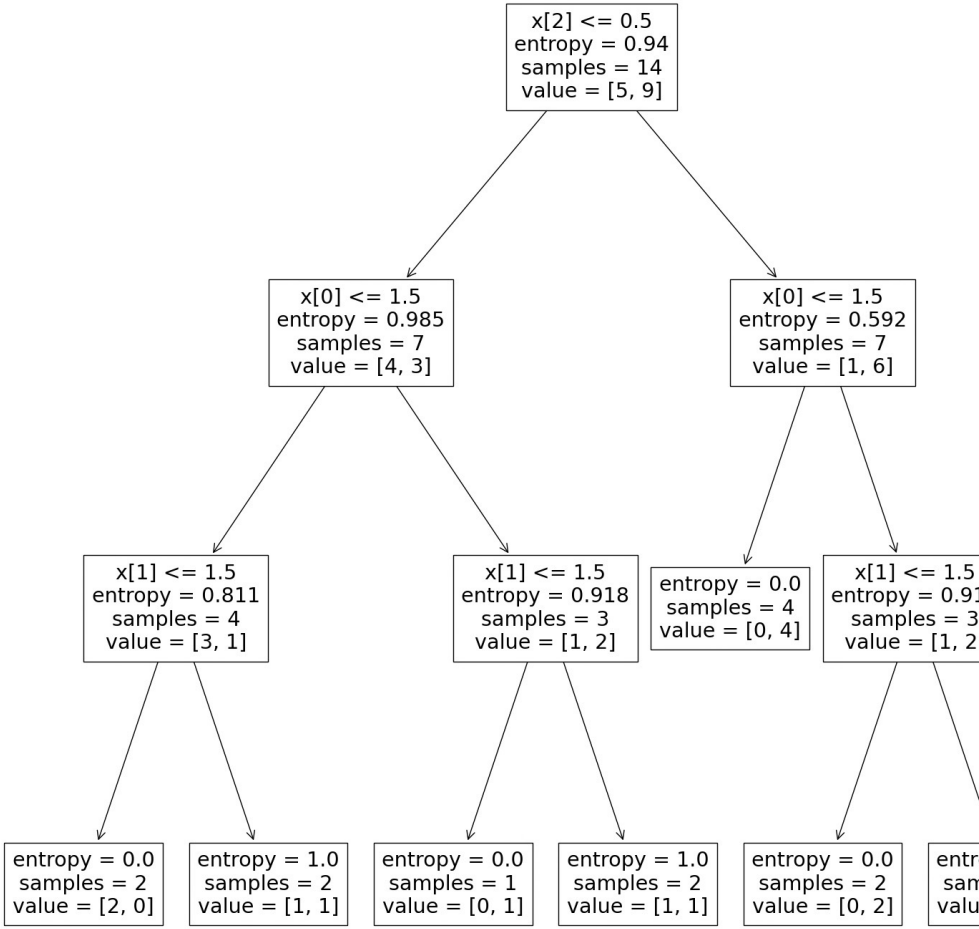
$$Info_{student:no}(D) = I(3,4) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right)$$
$$= 0.985$$

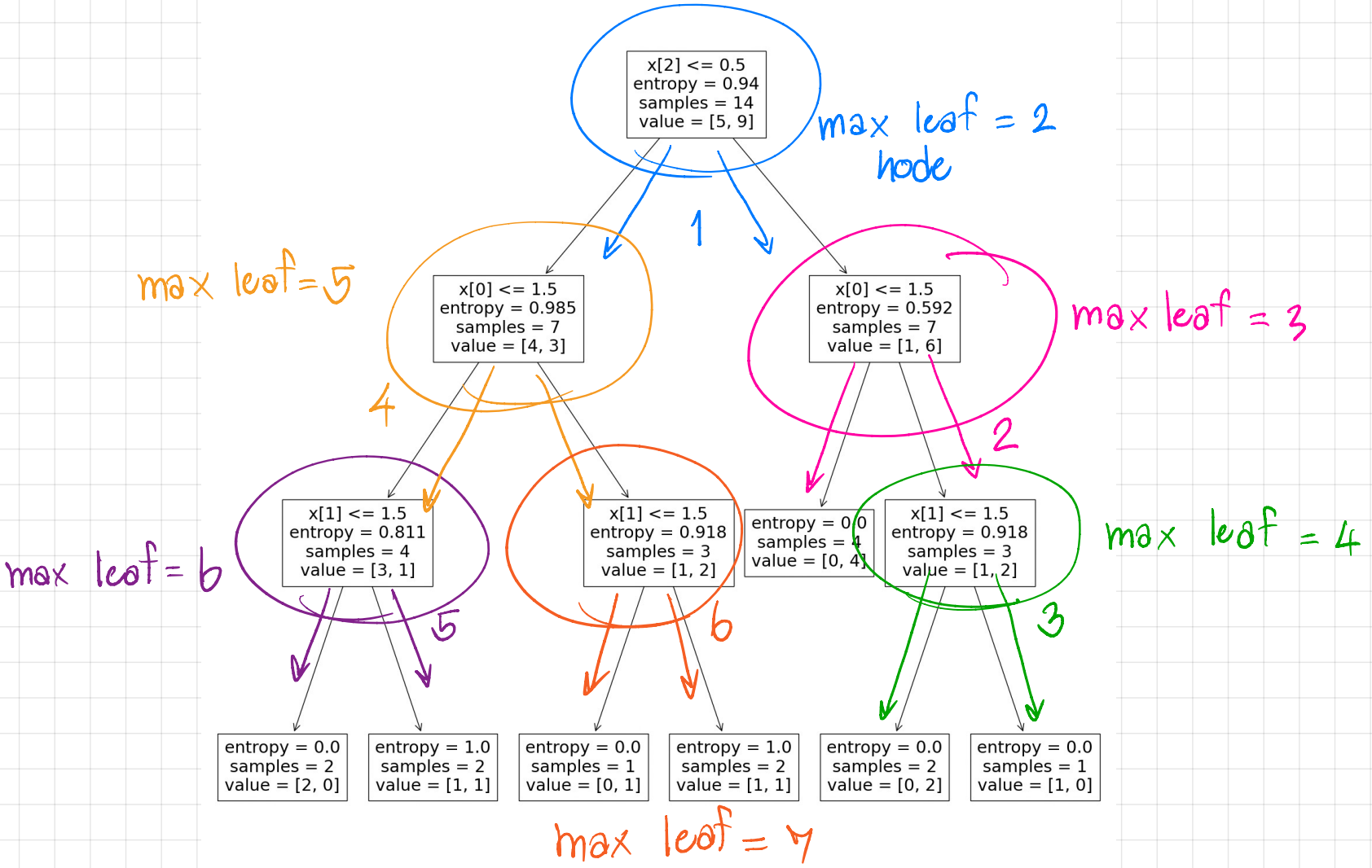
$$Info_{age}(D) = \frac{4}{7}I(1,3) + \frac{3}{7}I(2,1)$$
$$= \frac{4}{7}\left[-\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2-\left(\frac{3}{4}\right)\right] + \frac{3}{7}\left[-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right]$$
$$= \frac{4}{7}(0.811) + \frac{3}{7}(0.918) \quad = 0.857$$

$$Info_{income}(D) = \frac{3}{7}I(1,2) + \frac{4}{7}I(2,2)$$
$$= \frac{3}{7}\left[-\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right] + \frac{4}{7}\left[-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right]$$
$$= \frac{3}{7}(0.918) +$$


x[2] <= 0.5
entropy = 0.94
samples = 14
value = [5, 9]
x[0] <= 1.5
entropy = 0.985
samples = 7
value = [4, 3]
x[0] <= 1.5
entropy = 0.592
samples = 7
value = [1, 6]
x[1] <= 1.5
entropy = 0.811
samples = 4
value = [3, 1]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 4
value = [0, 4]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 2
value = [0, 2]
entropy = 0.0
samples = 1
value = [1, 0]


x[2] <= 0.5
entropy = 0.94
samples = 14
value = [5, 9]
x[0] <= 1.5
entropy = 0.985
samples = 7
value = [4, 3]
x[0] <= 1.5
entropy = 0.592
samples = 7
value = [1, 6]
x[1] <= 1.5
entropy = 0.811
samples = 4
value = [3, 1]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 4
value = [0, 4]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 2
value = [0, 2]
entropy = 0.0
samples = 1
value = [1, 0]
1
2
3
4
5
6


x[2] <= 0.5
entropy = 0.94
samples = 14
value = [5, 9]
x[0] <= 1.5
entropy = 0.985
samples = 7
value = [4, 3]
x[0] <= 1.5
entropy = 0.592
samples = 7
value = [1, 6]
x[1] <= 1.5
entropy = 0.811
samples = 4
value = [3, 1]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 4
value = [0, 4]
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 2
value = [0, 2]

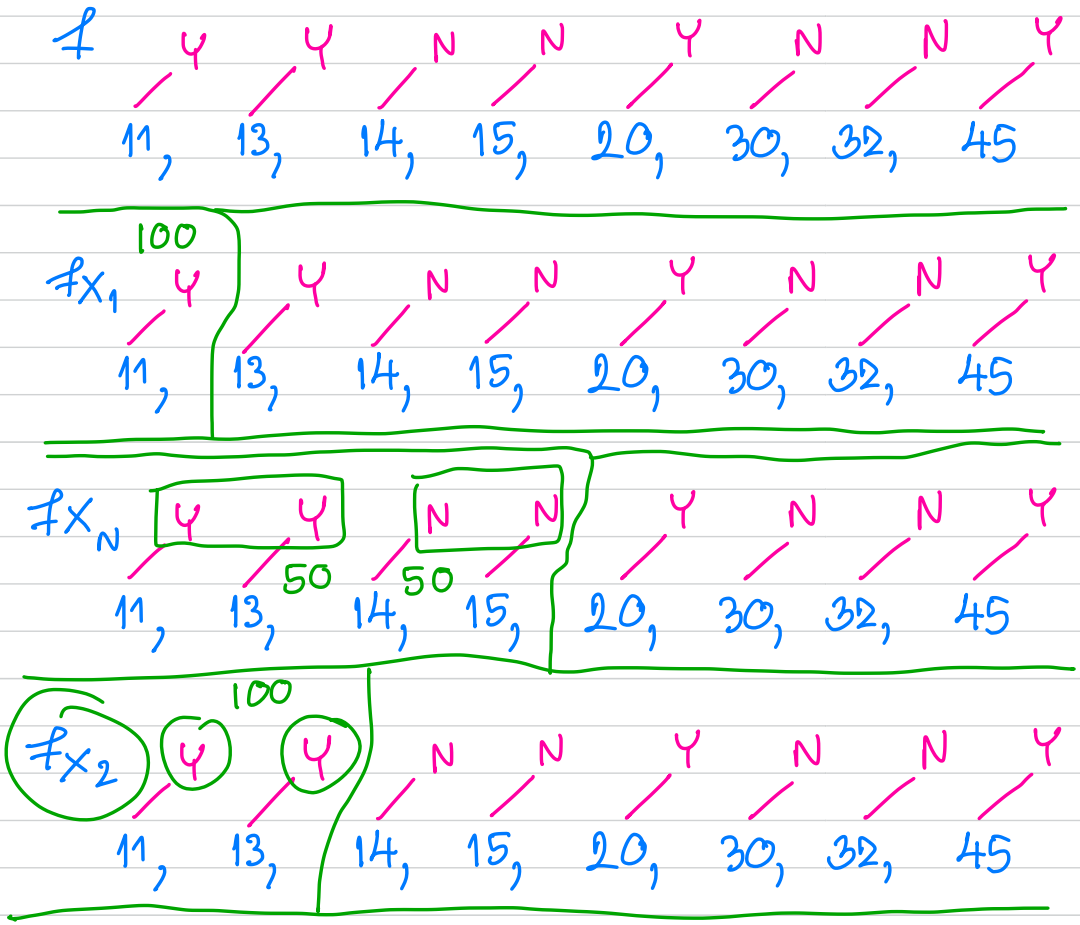

x[2] <= 0.5
entropy = 0.94
samples = 14
value = [5, 9]
max leaf = 2
node
1
max leaf=5
x[0] <= 1.5
entropy = 0.985
samples = 7
value = [4, 3]
x[0] <= 1.5
entropy = 0.592
samples = 7
value = [1, 6]
max leaf = 3
4
2
max leaf = 6
x[1] <= 1.5
entropy = 0.811
samples = 4
value = [3, 1]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
entropy = 0.0
samples = 4
value = [0, 4]
x[1] <= 1.5
entropy = 0.918
samples = 3
value = [1, 2]
max leaf = 4
5
6
3
entropy = 0.0
samples = 2
value = [2, 0]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 1
value = [0, 1]
entropy = 1.0
samples = 2
value = [1, 1]
entropy = 0.0
samples = 2
value = [0, 2]
entropy = 0.0
samples = 1
value = [1, 0]
max leaf = 7

Decision Tree Induction: Algorithm

- Basic algorithm
  - Tree is constructed in a **top-down, recursive, divide-and-conquer manner**
  - At start, all the training examples are at the root
  - Examples are partitioned recursively based on selected attributes
  - On each node, attributes are selected based on the training examples on that node, and a heuristic or statistical measure (e.g., **information gain**)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning
  - There are no samples left
- Prediction
  - **Majority voting** is employed for classifying the leaf

How to Handle Continuous-Valued Attributes?

- Method 1: Discretize continuous values and treat them as categorical values
  - E.g., age: < 20, 20..30, 30..40, 40..50, > 50
- Method 2: Determine the ***best split point*** for continuous-valued attribute A
  - Sort the value A in increasing order:, e.g. 15, 18, 21, 22, 24, 25, 29, 31, …
  - *Possible split point:* the midpoint between *each pair of adjacent values*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
    - e.g., (15+18/2 = 16.5, 19.5, 21.5, 23, 24.5, 27, 30, …
  - The point with the *maximum information gain* for A is selected as the **split-point** for A
- Split: Based on split point P
  - The set of tuples in D satisfying A ≤ P vs. those with A > P



8 ค่า → 8 entropy → ดีที่สุด

Y = pure น่าจะดีที่สุด

กรณี → ถ้าเกิดค่าเป็น continuous

Gain Ratio: A Refined Measure for Attribute Selection

- Information gain measure is biased towards attributes with a large number of values
- Gain ratio: Overcomes the problem (as a normalization to information gain)

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

  - GainRatio(A) = Gain(A)/SplitInfo(A)
  - The attribute with the maximum gain ratio is selected as the splitting attribute
  - Gain ratio is used in a popular algorithm C4.5 (a successor of ID3) by R. Quinlan
- Example
  - $\text{SplitInfo}_{\text{income}}(\text{D}) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.557$
  - GainRatio(income) = 0.029/1.557 = 0.019

Another Measure: Gini Index

- Gini index: Used in CART, and also in IBM IntelligentMiner
- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as
  - $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$
    - p_j is the relative frequency of class j in D
- If a data set D is split on A into two subsets D_1 and D_2, the $gini$ index $gini(D)$ is defined as
  - $gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$
- Reduction in Impurity:
  - $\Delta gini(A) = gini(D) - gini_A(D)$
- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Computation of Gini Index

- Example: D has 9 tuples in buys_computer = “yes” and 5 in “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2
  - $gini_{income \in \{low,medium\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

$$= Gini_{income \in \{high\}}(D)$$

  - $Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450
  - Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Three Attribute Selection Measures

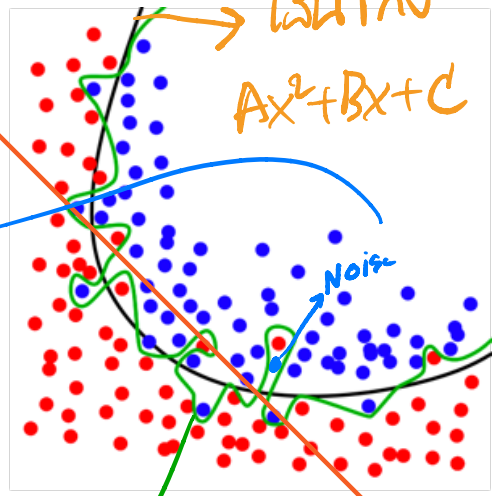
- The three measures, in general, return good results but
  - **Information gain**:
    - biased towards multivalued attributes
  - **Gain ratio**:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - **Gini index**:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Other Attribute Selection Measures

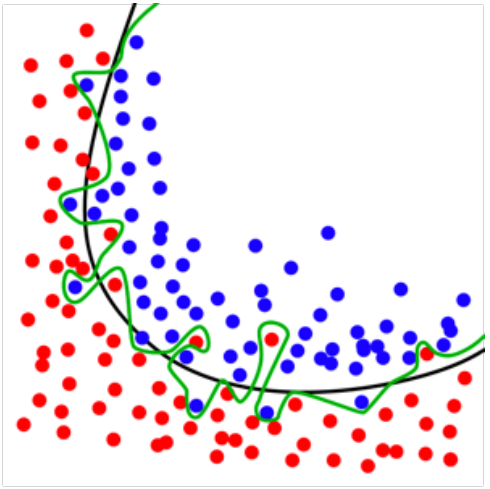
- Minimal Description Length (MDL) principle
  - Philosophy: The simplest solution is preferred
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear combination of attributes
- There are many other measures proposed in research and applications
  - E.g., G-statistics, C-SEP
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"



Overfitting and Tree Pruning



- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

CODE

PACKAGE : Sklearn Classification

1. Input
2. Define Model ← condition จำนวนต้นไม้ ความลึกของต้นไม้ ความโค้งเกิน กำหนด / จำกัดได้
3. Training (.fit (X, Y)
4. test - จำแนก (.predic ($\vec{x}$) → $\hat{y}$

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why is decision tree induction popular?
  - Relatively fast learning speed
  - Convertible to simple and easy to understand classification rules
  - Easy to be adapted to database system implementations (e.g., using SQL)
  - Comparable classification accuracy with other methods
- **RainForest** (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
  - Builds an AVC-list (attribute, value, class label)

RainForest: A Scalable Classification Framework

- The criteria that determine the quality of the tree can be computed separately
  - Builds an AVC-list: **AVC (Attribute, Value, Class_label)**
- **AVC-set** (of an attribute *X*)
  - Projection of training dataset onto the attribute *X* and class label where counts of individual class label are aggregated
- **AVC-group** (of a node *n*)
  - Set of AVC-sets of all predictor attributes at the node *n*

age	income	student	credit_rating	comp
<=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

The Training Data

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *Income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

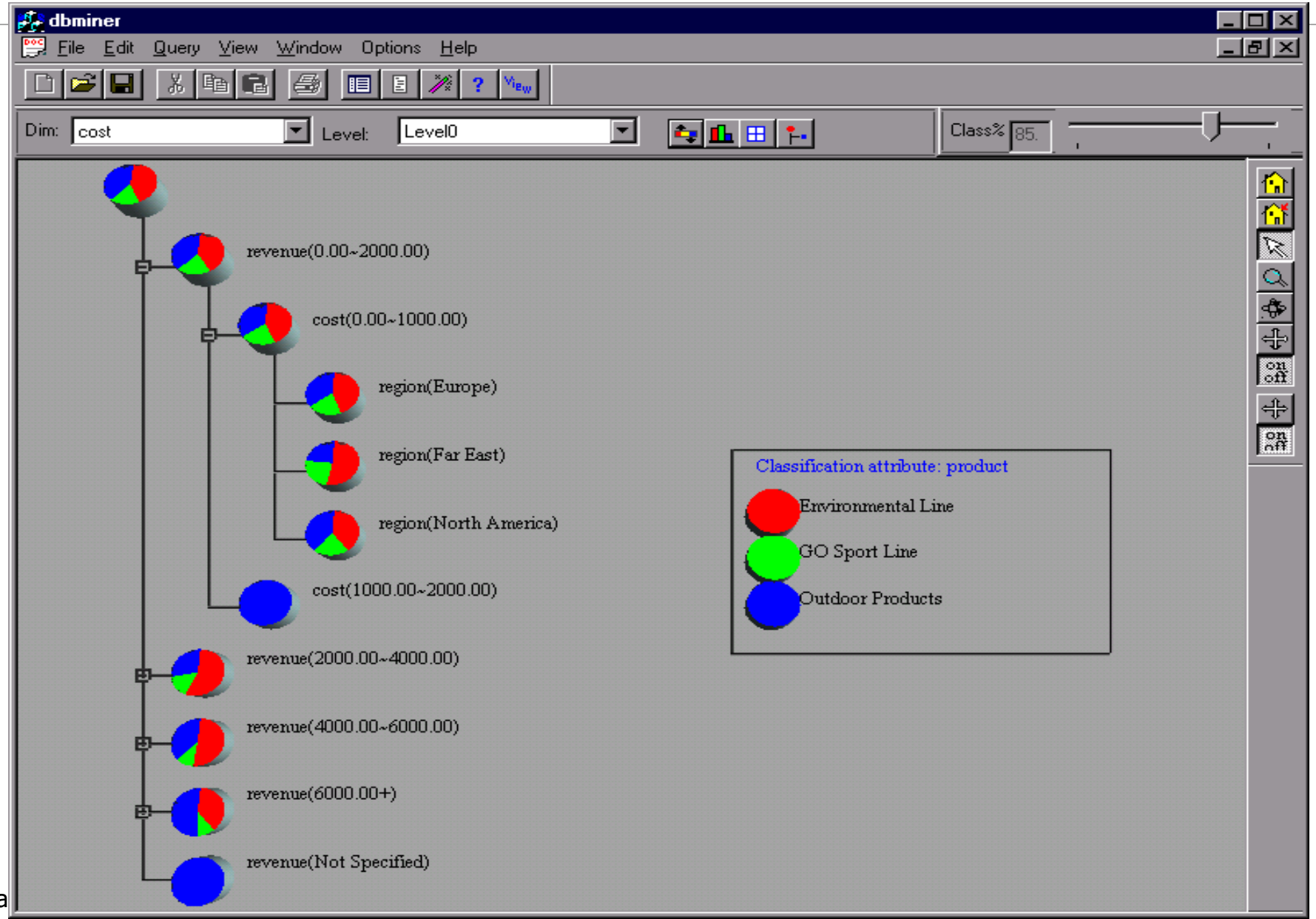
student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

AVC-set on *Credit_Rating*

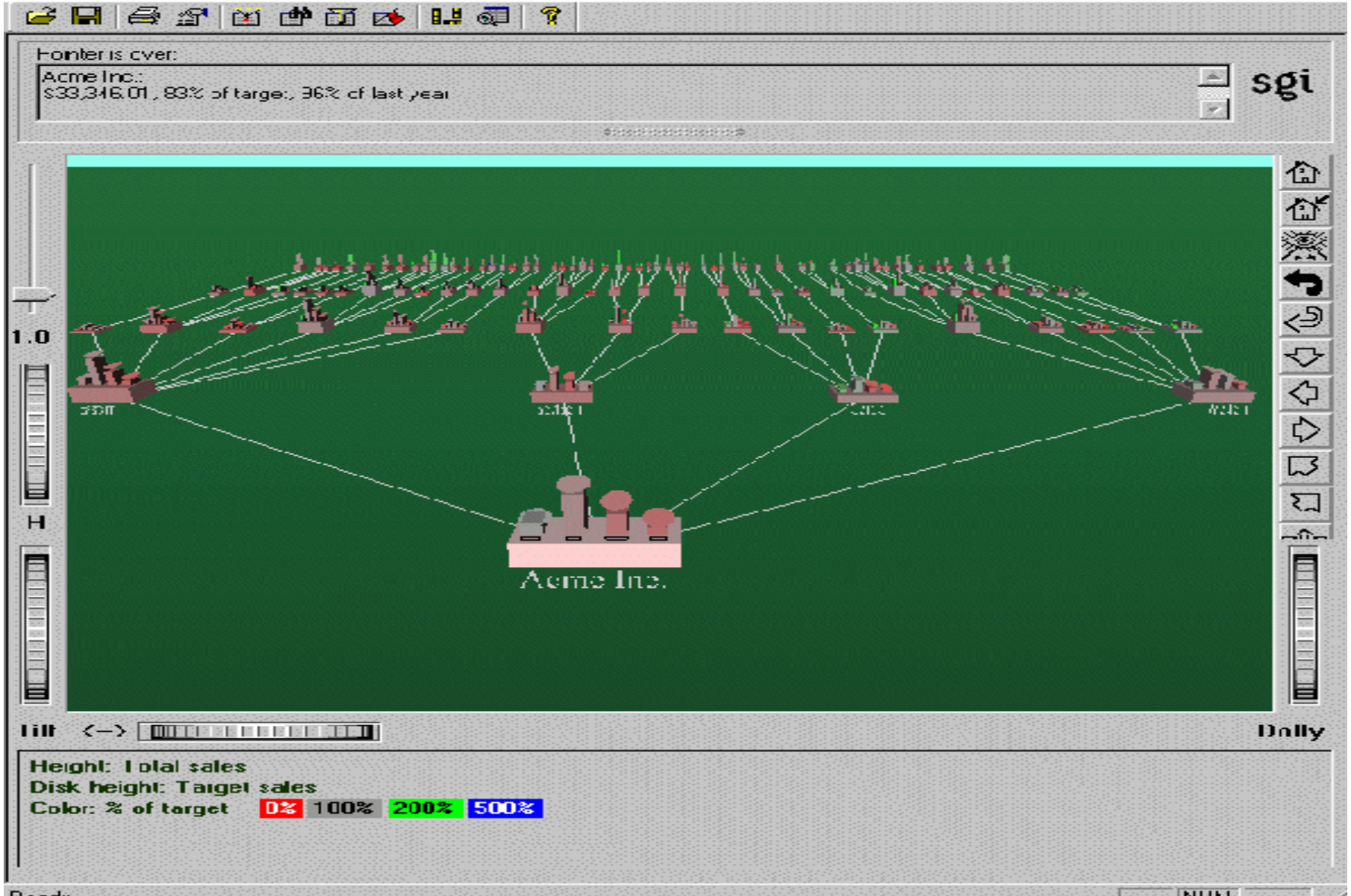
Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

Its AVC Sets

Presentation of Classification Results

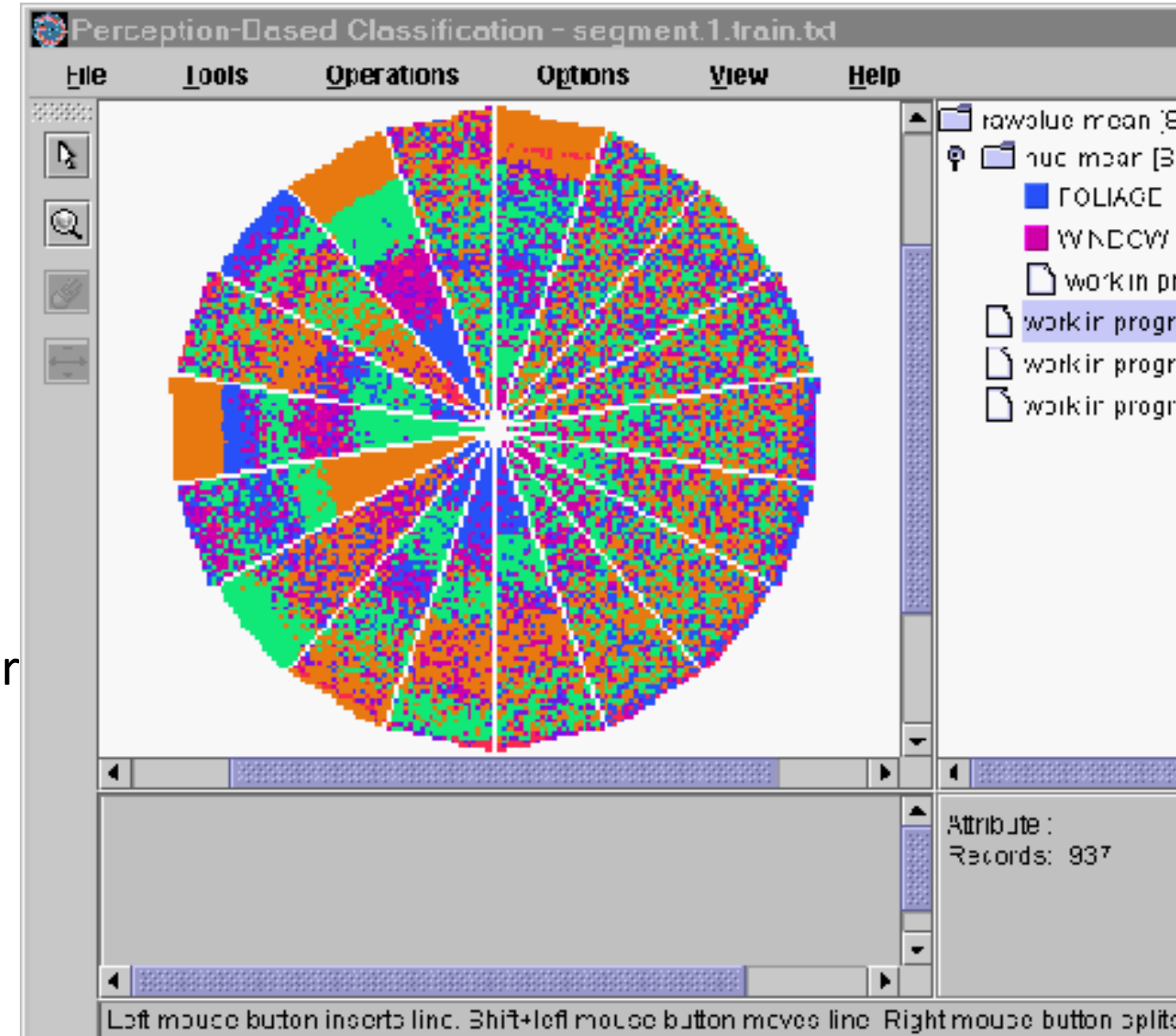


Visualization of a Decision Tree (in SGI/MineSet 3.0)



Interactive Visual Mining by Perception-Based Classification (PBC)

- Perception-based classifier (PCB): developed at Univ. of Munich (1999)
- One color represents one class label
- One pie represents one attribute (or variable)
- The pie with random spread implies weak classification power
- The pie with clearly partitioned color strips implies good classification power
- One can select a good attribute and regenerate new pie charts for classification at the subsequent levels



CS412-Fall 2017: Midterm Statistics

Range	Count
0-30	1
30-40	4
40-50	20
50-60	35
60-70	45
70-80	50
80-90	32
90-100	13

Mean	68.64
Median	69.5
1st quartile	57.75
3rd quartile	79.5