Learning Agent: Supervised Learning (kNN, Naïve Bayes, Decision Tree)

Informatics Engineering Study Program
School of Electrical Engineering and Informatics

Institute of Technology Bandung

Review Supervised Learning

- Learning Agent
- Feedback vs learning type
- Supervised learning:

Learning Method (Supervised Learning)

- **Example-based classifier:**
  - k-Nearest Neighbor:
    - Instance-based learning
    - Lazy learner (not eager learner)
- **Numeric/quantitative:**
  - probabilistic classifier, linear classifier, SVM, regression, artificial neural network, Naïve Bayes
  - Naïve Bayes:
    - Probabilistic classifier : P(Y|x)
- **Nonnumeric/symbolic:**
  - Decision tree learning, decision rule classifier

Contoh: Decision Tree Model

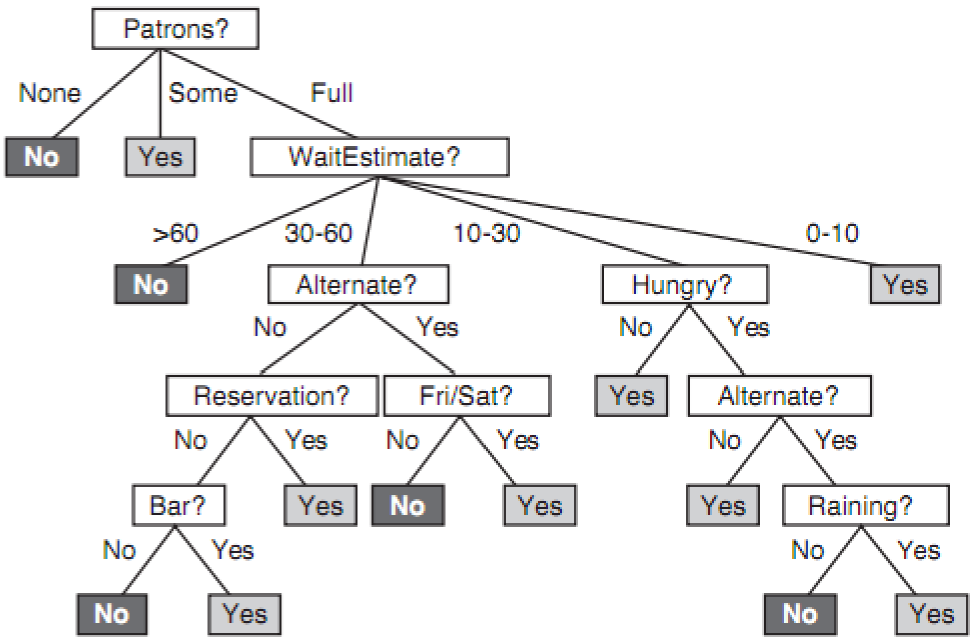


Figure 18.2 **FILES: figures/restaurant-tree.eps (Tue Nov 3 16:23:29 2009).** A decision tree for deciding whether to wait for a table.

Contoh: Neural Network

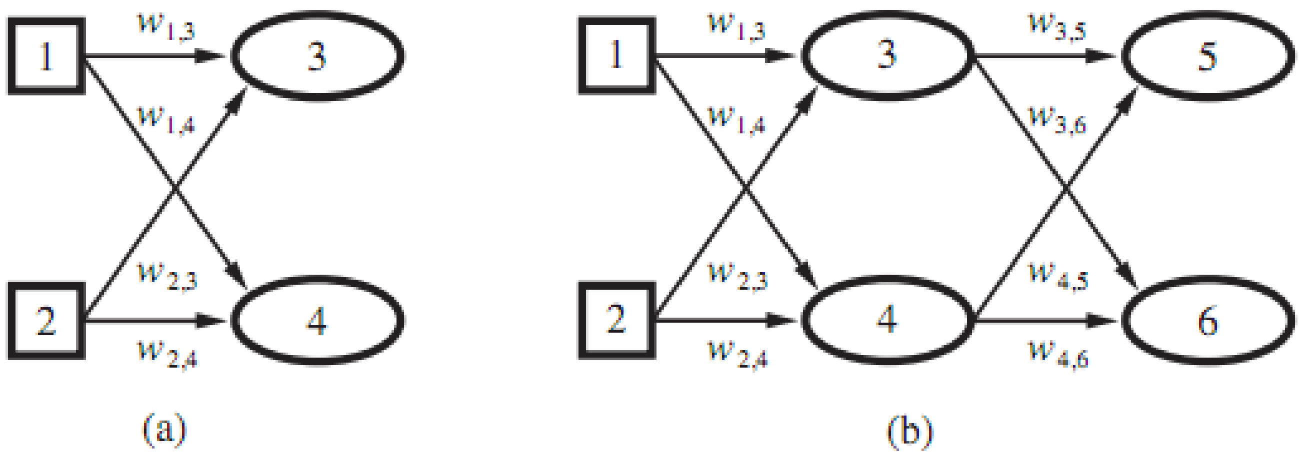


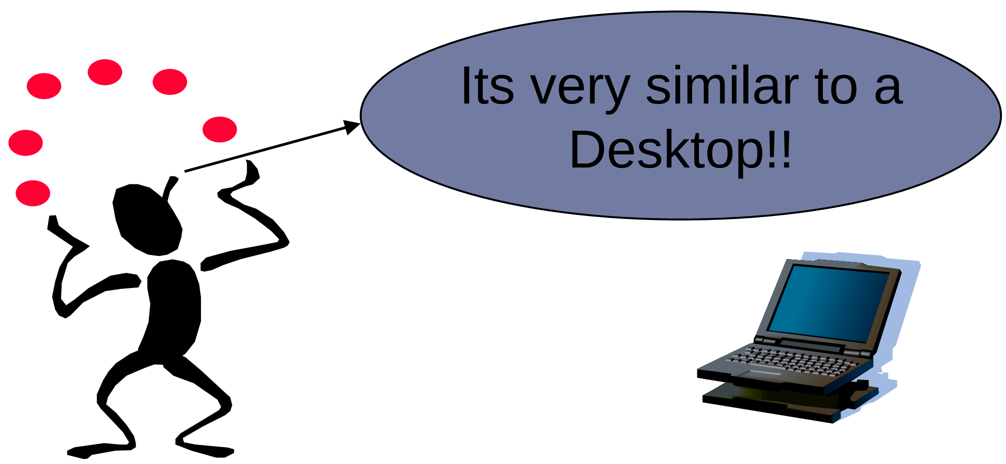
Figure 18.20 FILES: figures/neural-net.eps (Wed Nov 4 11:08:22 2009). (a) A perceptron network with two inputs and two output units. (b) A neural network with two inputs, one hidden layer of two units, and one output unit. Not shown are the dummy inputs and their associated weights.

Contoh: Support Vector Machines

```
svm_type c_svc
kernel_type rbf
gamma 0.017241379310344827
nr_class 2
total_sv 402
rho -0.8300595301464558
label 0 2
probA -5.460987117018979
probB 2.0721783740735344
nr_sv 222 180
SV
1.0 1:0.0 2:1.0 3:0.0 4:1.0 5:1.0 
27:0.0 28:0.0 29:0.0 30:0.0 31:0.0
53:0.0 54:0.0 55:0.0 56:0.0 57:0.0
1.0 1:0.0 2:1.0 3:0.0 4:1.0 5:2.0 
27:0.0 28:0.0 29:1.0 30:0.0 31:0.0
```

k-Nearest Neighbour

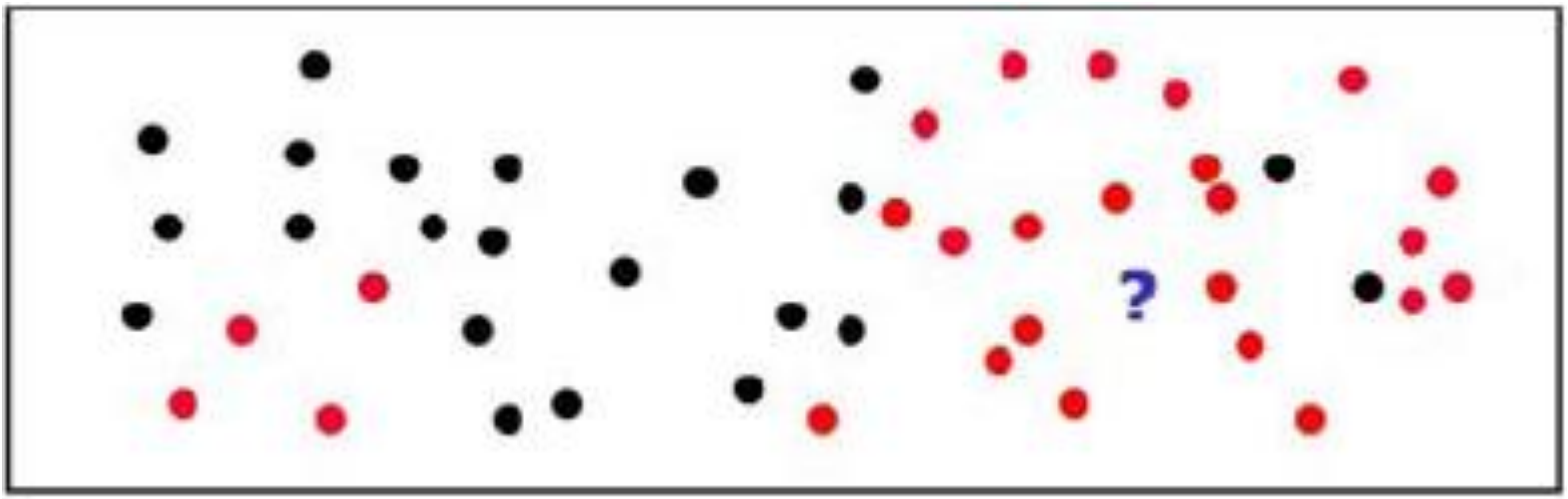
Instance-based Learning



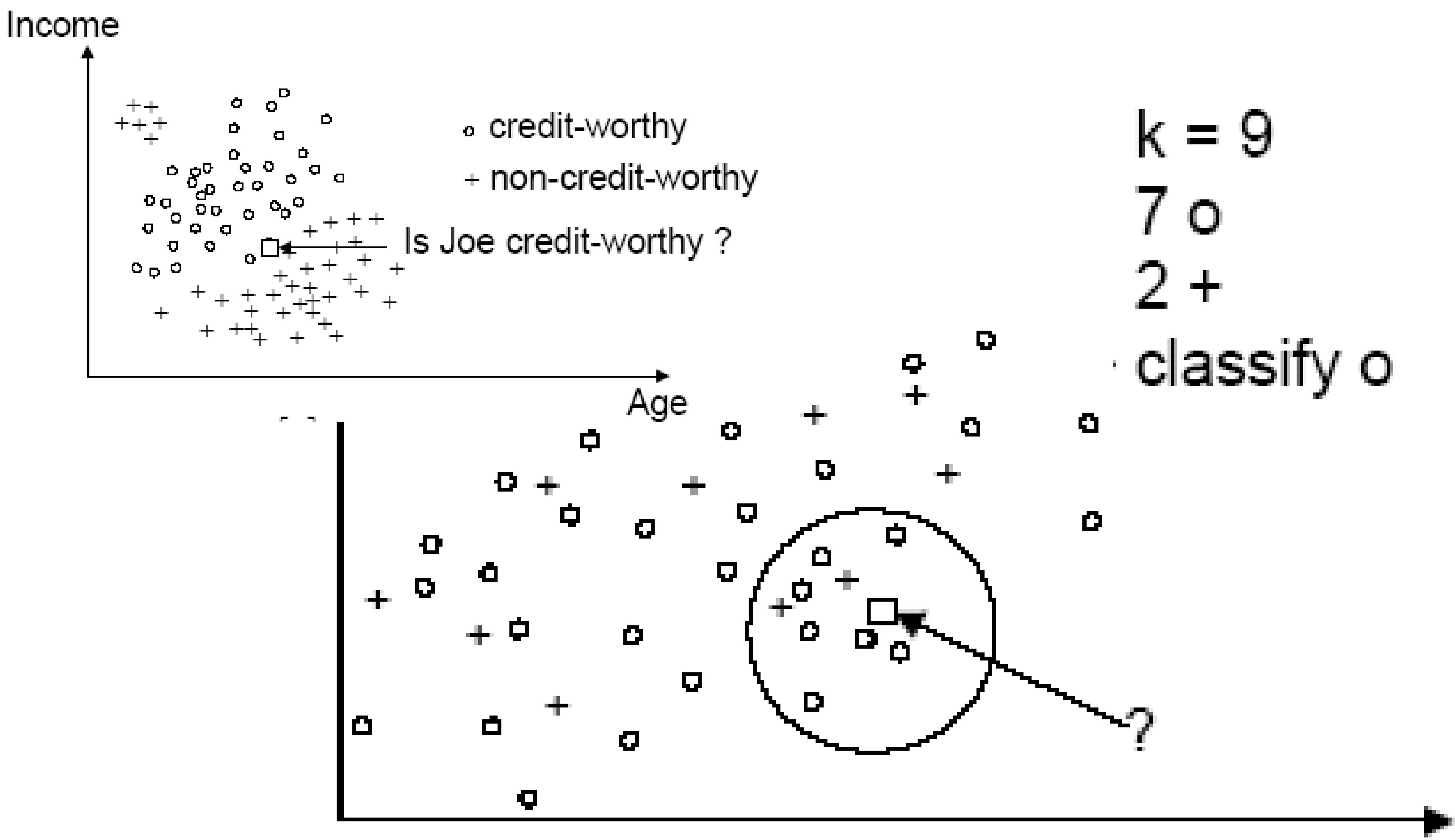
K-Nearest Neighbor Learning (Dipanjan Chakraborty)

K-Nearest Neighbor

- Menyimpan semua data
- Input baru → kelas dari data terdekat



Contoh k-NN



k = 9

7 o

2 +

classify o

Contoh Data set: Play Tennis

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes

outlook	temp.	humidity	windy	play
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

outlook	temp.	humidity	windy	play
sunny	cool	high	true	?

Naïve Bayes

Naïve Bayes

$$v_{NB} = \arg\max_{v_j \in \{yes,no\}} P(v_j) \prod_i P(a_i|v_j)$$

- $P(v_j)$: probabilitas kelas v_j
- $P(a_i|v_j)$: probabilitas atribut a_i pada v_j

Contoh Data set: Play Tennis

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Frekuensi setiap nilai atribut

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								

Model Probabilitas

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Klasifikasi Unseen Example

outlook	temp.	humidity	windy	play
sunny	cool	high	true	?

$$
\begin{aligned}
v_{NB} &= \arg\max_{v_j \in \{yes,no\}} P(v_j) \prod_i P(a_i|v_j) \\
&= \arg\max_{v_j \in \{yes,no\}} P(v_j) P(\text{outlook} = \text{sunny}|v_j) P(\text{temp} = \text{cool}|v_j) \\
&\quad P(\text{humidity} = \text{high}|v_j) P(\text{windy} = \text{true}|v_j)
\end{aligned}
$$

1. Kalikan probabilitas semua atribut untuk setiap kelas
2. Hasil 1 dikalikan dengan probabilitas setiap kelas
3. Klasifikasi: kelas dengan probabilitas maksimum

Proses Klasifikasi

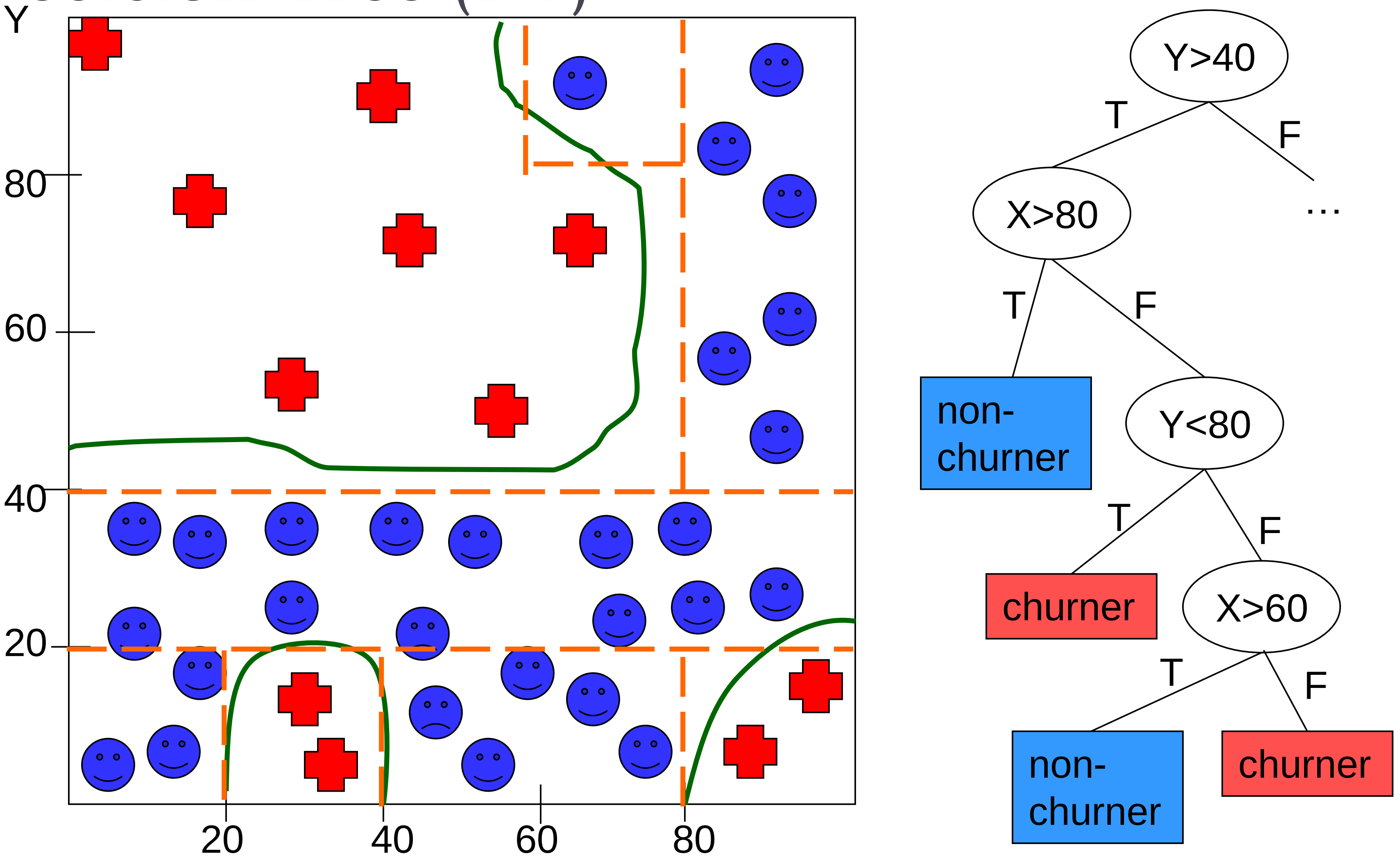
$$P(\text{play} = \text{yes}) = 9/14 \quad P(\text{play} = \text{no}) = 5/14$$

$$P(\text{yes})P(\text{sunny}|\text{yes})P(\text{cool}|\text{yes})P(\text{high}|\text{yes})P(\text{true}|\text{yes})$$
$$= 9/14 \cdot 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.0053$$
$$P(\text{no})P(\text{sunny}|\text{no})P(\text{cool}|\text{no})P(\text{high}|\text{no})P(\text{true}|\text{no})$$
$$= 5/14 \cdot 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0206$$

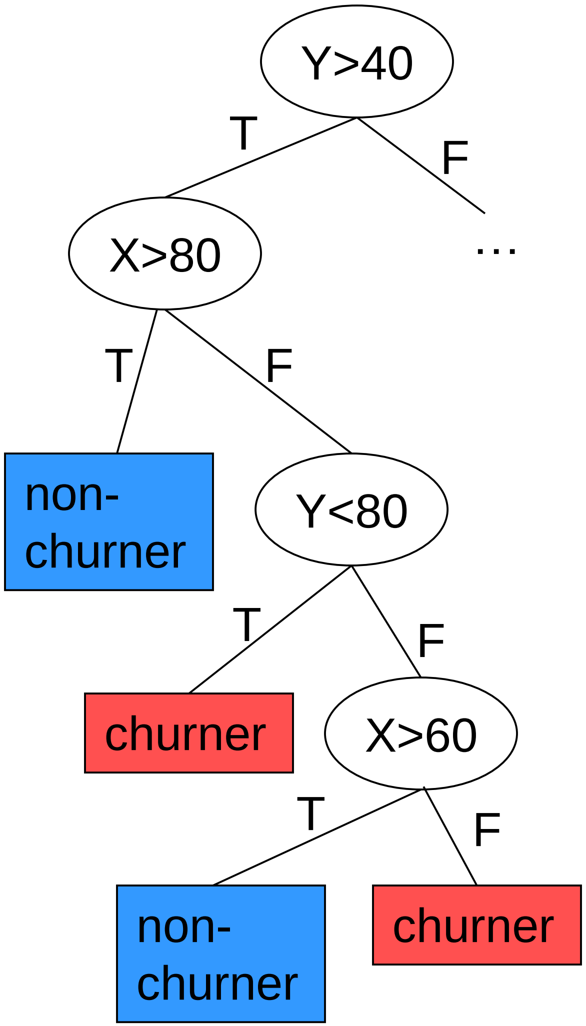
$$\begin{aligned} v_{NB} &= \underset{v_j \in \{\text{yes},\text{no}\}}{\arg\max} P(v_j)P(\text{sunny}|v_j)P(\text{cool}|v_j)P(\text{high}|v_j)P(\text{true}|v_j) \\ &= \text{no} \end{aligned}$$

Decision Tree

Decision Tree (DT)



Decision Tree: Pengetahuan



IF Y>40 and X>80 THEN non-churner

…

Will I wait for a table ?

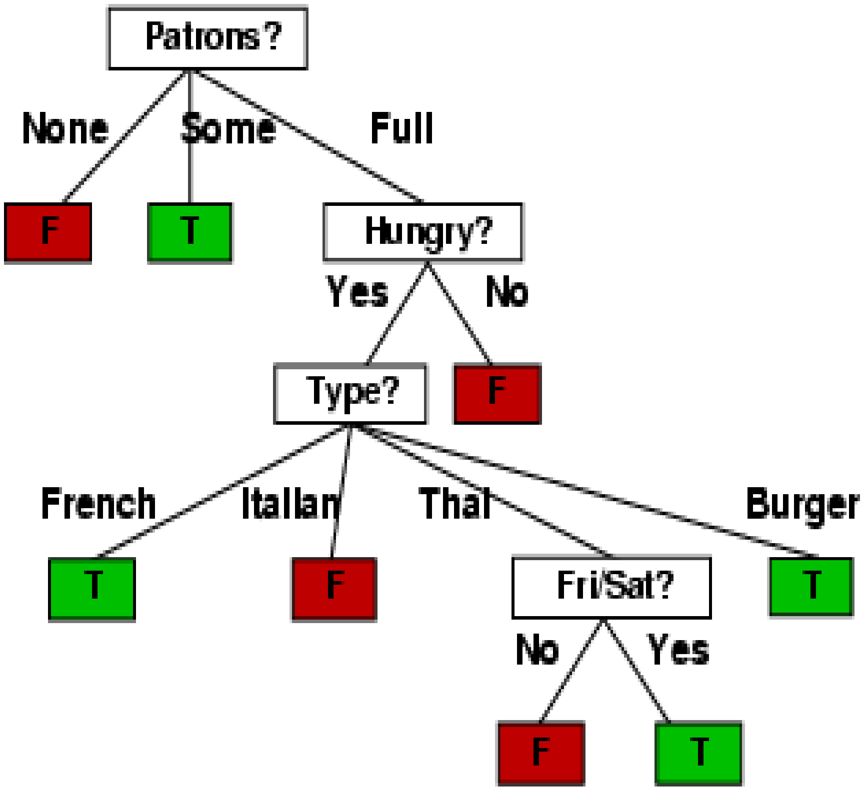
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Alt: whether there is a suitable alternative restaurant nearby
Bar: whether the restaurant has a comfortable bar area to wait in
Fri: true on Fridays and Saturdays
Hun: whether we are hungry
Pat: how many people are in the restaurant
Price: price range
Raining: whether it is raining outside
Reservation: whether we made a reservation

Decision tree learning

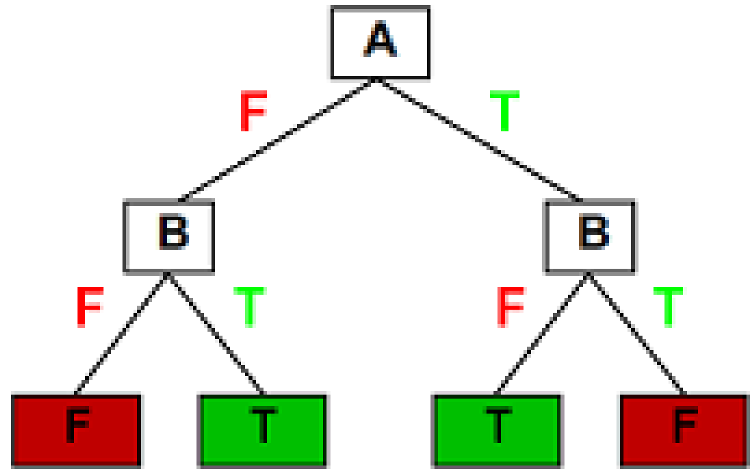
- Data training digunakan untuk membangun decision tree
- Simpul: pertanyaan
- Cabang: jawaban yang mungkin

Problem: decide whether to wait for a table at a restaurant

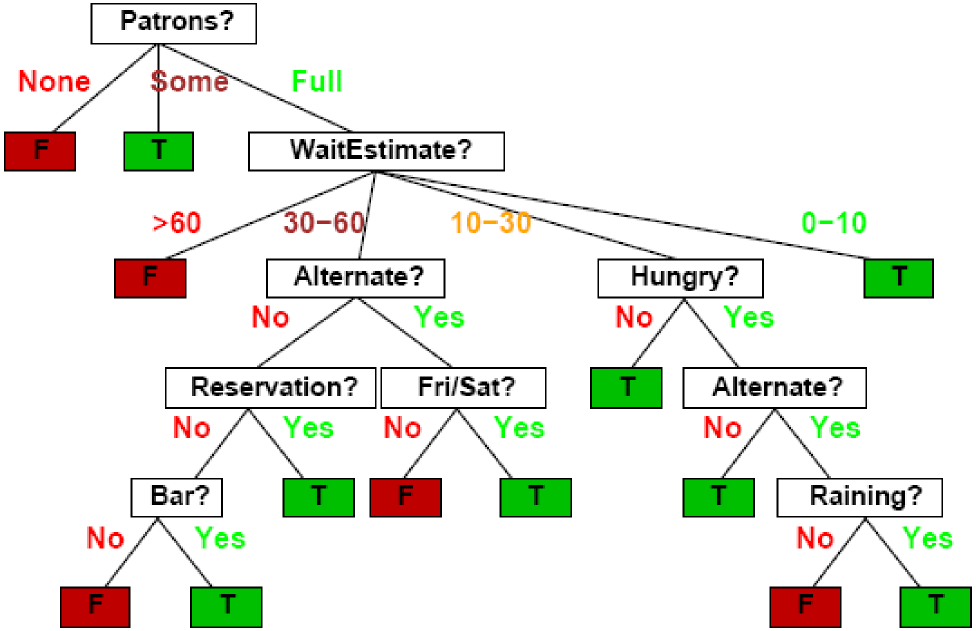


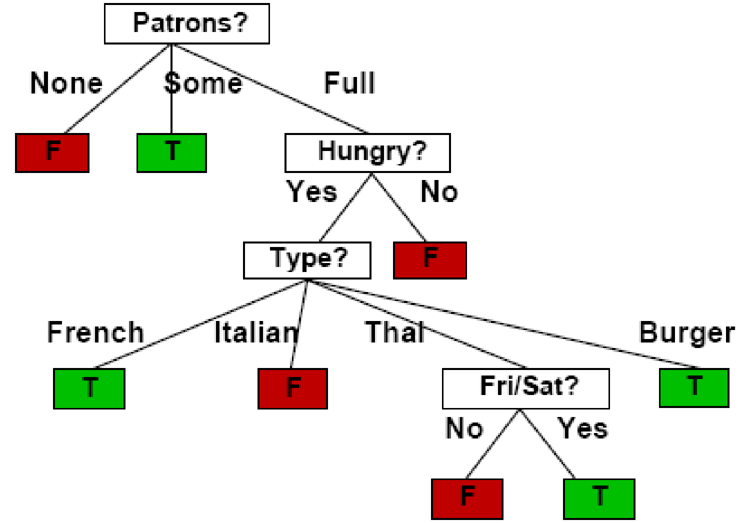
Decision Tree vs Truth Table

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- 1 row truth table → 1 path DT
- Data training: 1 examples → 1 path ?
- N boolean atribut → 2^n rows/path → 2^{2^n} distinct decision tree
- No generalization for unseen data





Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Decision tree learning (2)

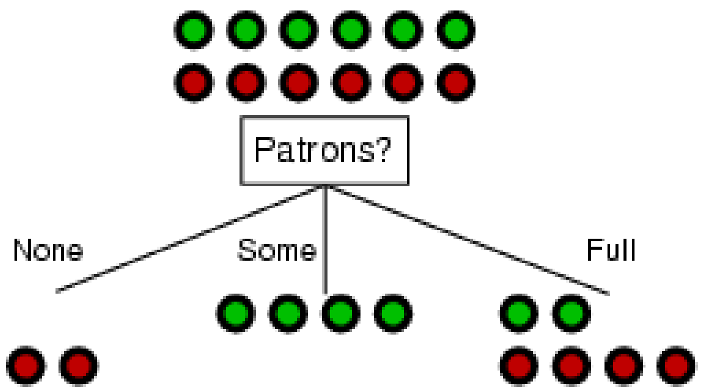
- Aim: find a small tree consistent with the training examples
- Idea: (recursively) choose "most significant" attribute as root of (sub)tree

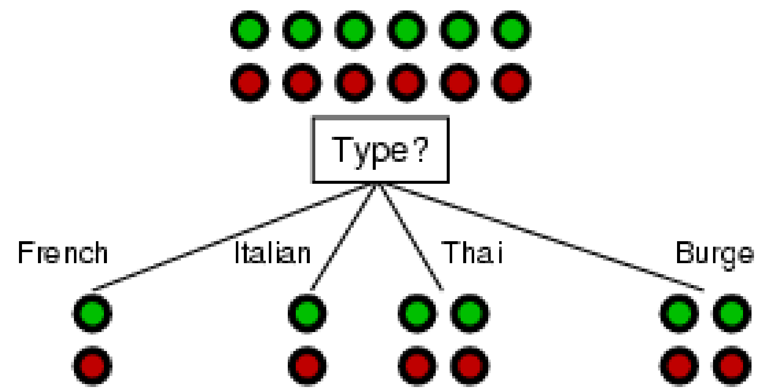
```
function DTL(examples, attributes, default) returns a decision tree

    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Choosing an attribute

- Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"





- *Patrons?* is a better choice

Using information theory

- To implement `Choose-Attribute` in the DTL algorithm
- Information Content (Entropy):

$$I(P(v_1), \dots, P(v_n)) = \Sigma_{i=1} -P(v_i) \log_2 P(v_i)$$

- For a training set containing *p* positive examples and *n* negative examples:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information gain

- A chosen attribute A divides the training set E into subsets E_1, … , E_v according to their values for A, where A has v distinct values.

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Information Gain (IG) or reduction in entropy from the attribute test:

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

- Choose the attribute with the largest IG

Representasi Example

- Attribute-based representations
- Examples described by attribute values

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Information gain

For the training set, $p = n = 6$, *I(6/12, 6/12)* = *1* bit

Consider the attributes *Patrons* and *Type* (and others too):

$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .0541 \text{bits}$$

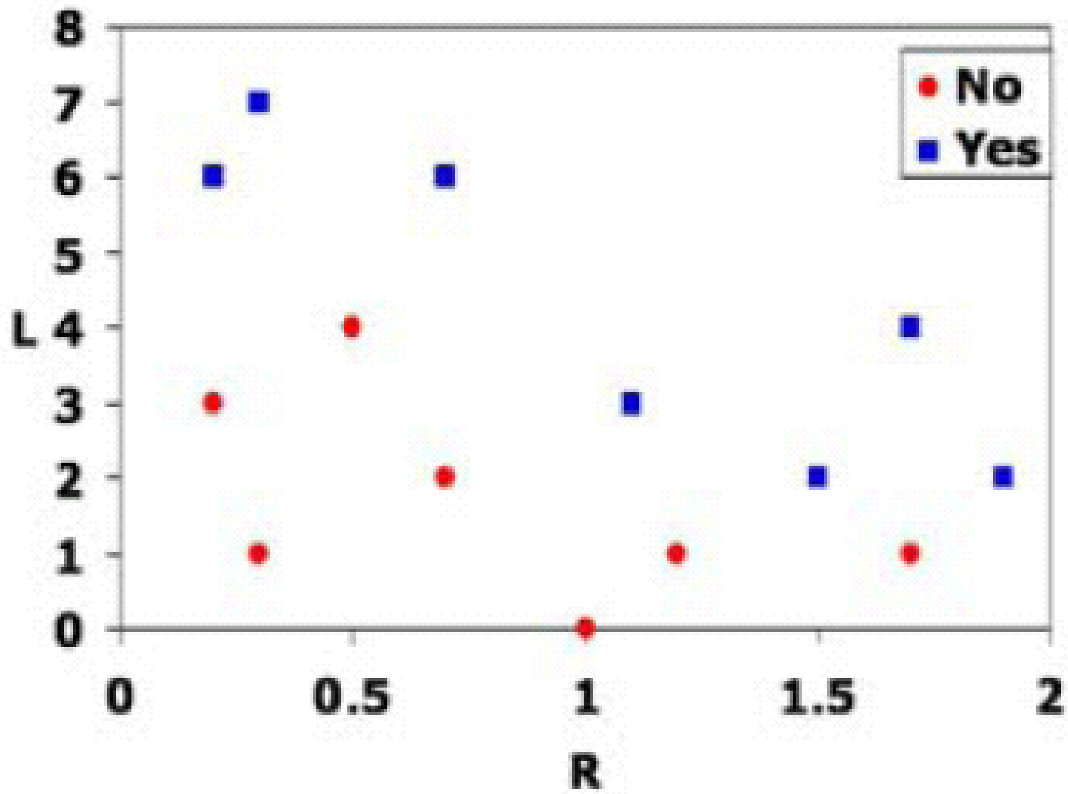
$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$

Patrons has the highest IG of all attributes and so is chosen by the DTL algorithm as the root

Mengapa DT Learning ?

- Mudah diimplementasikan
- Hipotesis yang dihasilkan mudah dipahami
- Efisien

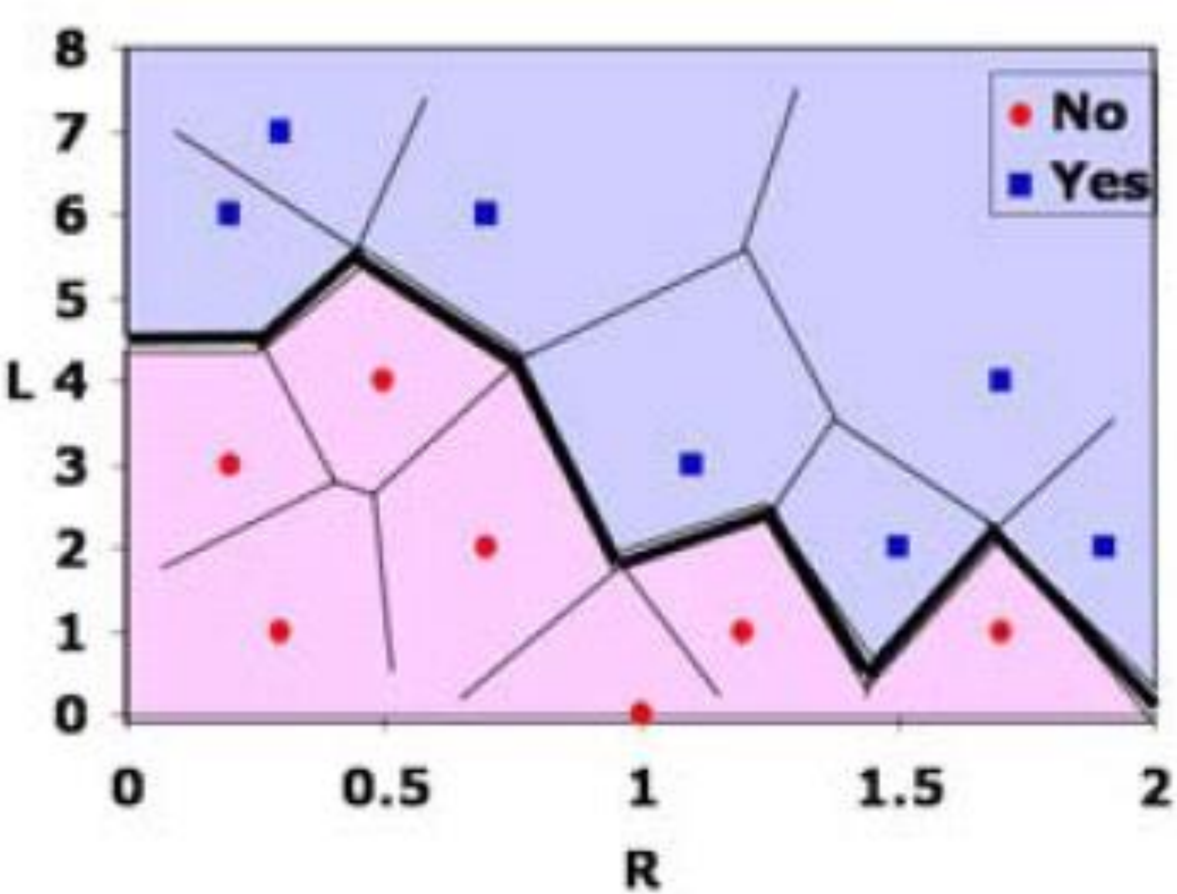
Contoh Bankruptcy Dataset



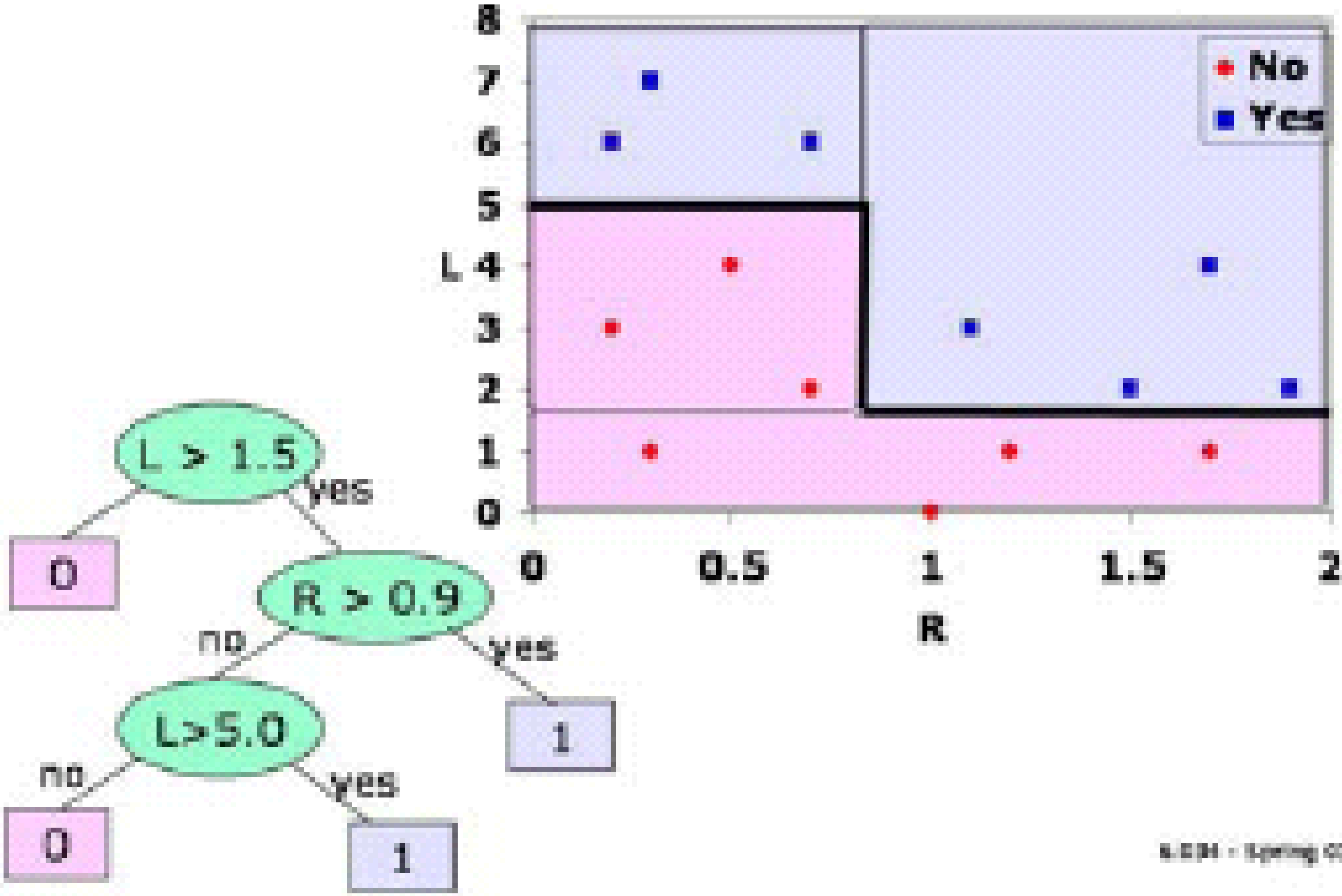
R: Rasio pendapatan-pengeluaran

L: jumlah pembayaran tagihan kartu kredit yang telat pada tahun lalu

Hipotesis 1-Nearest Neighbor



Hipotesis Decision Tree



THANK YOU