CSC242: Introduction to Artificial Intelligence

Lecture 4.1

Please put away all electronic devices

Learning

Learning

Learning

- "gives computers the ability to learn without being explicitly programmed" (Samuel, 1959)
- "… agents that can improve their behavior through diligent study of their own experiences" (Russell & Norvig)
- Improving one's performance on future tasks based on observations

Why Learn?

Why Learn?

- Can't anticipate (or store) all possible situations that the agent might find themselves in

Why Learn?

- Can't anticipate (or store) all possible situations that the agent might find themselves in
- Cannot anticipate (or store) all changes that might occur over time

Why Learn?

- Can't anticipate (or store) all possible situations that the agent might find themselves in
- Cannot anticipate (or store) all changes that might occur over time
- Don't know how to program it other than by some kind of learning!

Why Learn?

- Can't anticipate (or store) all possible situations that the agent might find themselves in
- Cannot anticipate (or store) all changes that might occur over time
- Don't know how to program it other than by some kind of learning!

Another tool in your toolbox

What To Learn?

What To Learn?

- Inferring properties of the world (state) from perceptions
- Choosing which action to do in a given state
- Results of possible actions
- How the world evolves in time
- Utilities of states

What To Learn?

- Inferring properties of the world (state) from perceptions
- Choosing which action to do in a given state
- Results of possible actions
- How the world evolves in time
- Utilities of states

Function Learning

$$y = f(\mathbf{x})$$

Function Learning

- There is some function $y = f(\mathbf{x})$
- We don't know f
- We want to learn a function h that approximates the true function f

Hypothesis

Function Learning

- There is some function $y = f(x)$
- We don't know f
- We want to learn a function h that approximates the true function f
- Learning is a search through the space of possible hypotheses for one that will perform well on the data

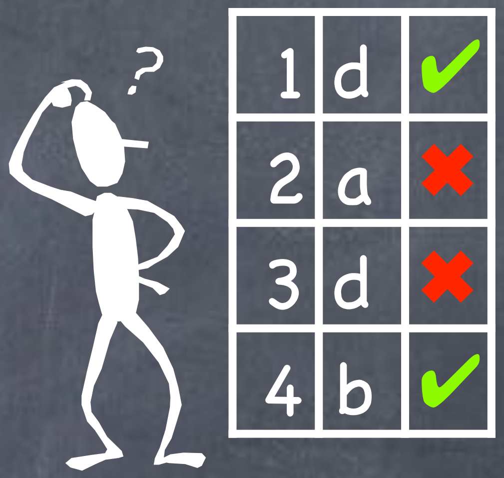
Types of Learning

Types of Learning

Unsupervised
(no feedback)

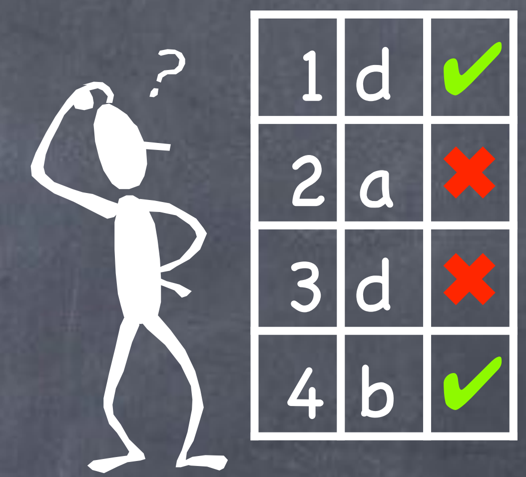
Types of Learning

Unsupervised
(no feedback)



Supervised
(labelled
examples)

Types of Learning

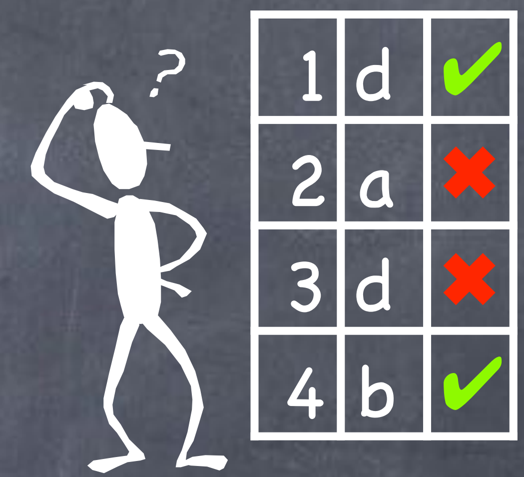


Unsupervised (no feedback)

Semi-supervised

Supervised (labelled examples)

Types of Learning



Unsupervised
(no feedback)

Semi-supervised

Supervised
(labelled
examples)

Reinforcement
(feedback is reward)

Classification

Classification

classification |ˌklasəfəˈkāSH(ə)n|
noun
the action or process of classifying something according to shared qualities or characteristics: *the classification of disease according to symptoms*.
• a category into which something is put.

classify |ˈklasəˌfī|
verb (**classifies**, **classifying**, **classified**) *[with object]*
arrange (a group of people or things) in classes or categories according to shared qualities or characteristics: *mountain peaks are classified according to their shape*.
- assign (someone or something) to a particular class or category: *elements are usually* ***classified as*** *metals or nonmetals*.

From Latin *classis* ‘**division**.’

Classification

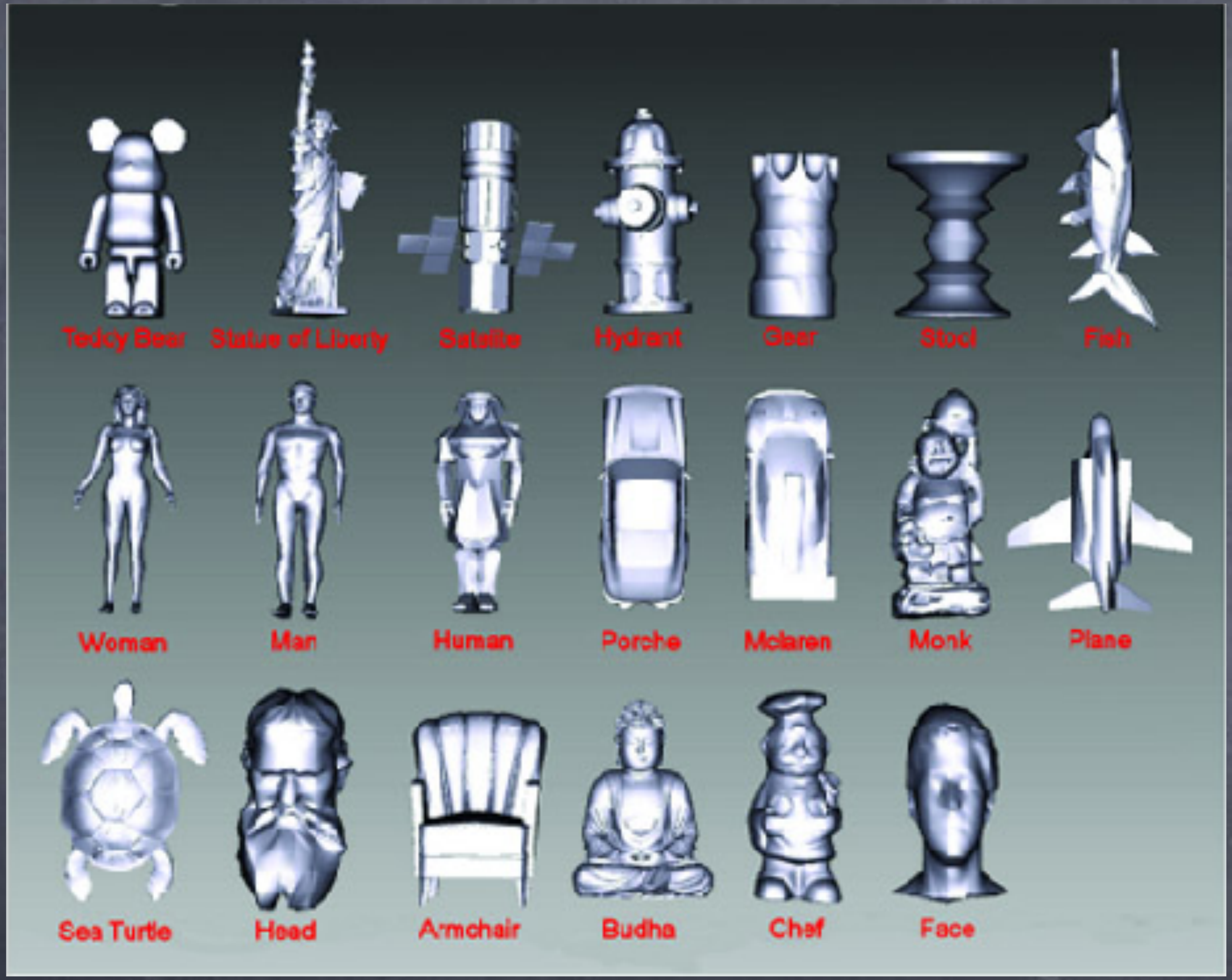
- Objects represented by set of attributes or features
  - Factored representation!
- Input is vector $\mathbf{x}$ of values for the attributes
- Output $y = f(\mathbf{x})$ is one of a finite set of values (classes, categories, labels, ...)
  - Boolean classification: two values

Classification

Input: x = $\langle x_1,\ x_2,\ x_3,\ \ldots,\ x_k \rangle$

Output: $y = \{\ 0,\ 1,\ 2,\ \ldots,\ 9\ \}$

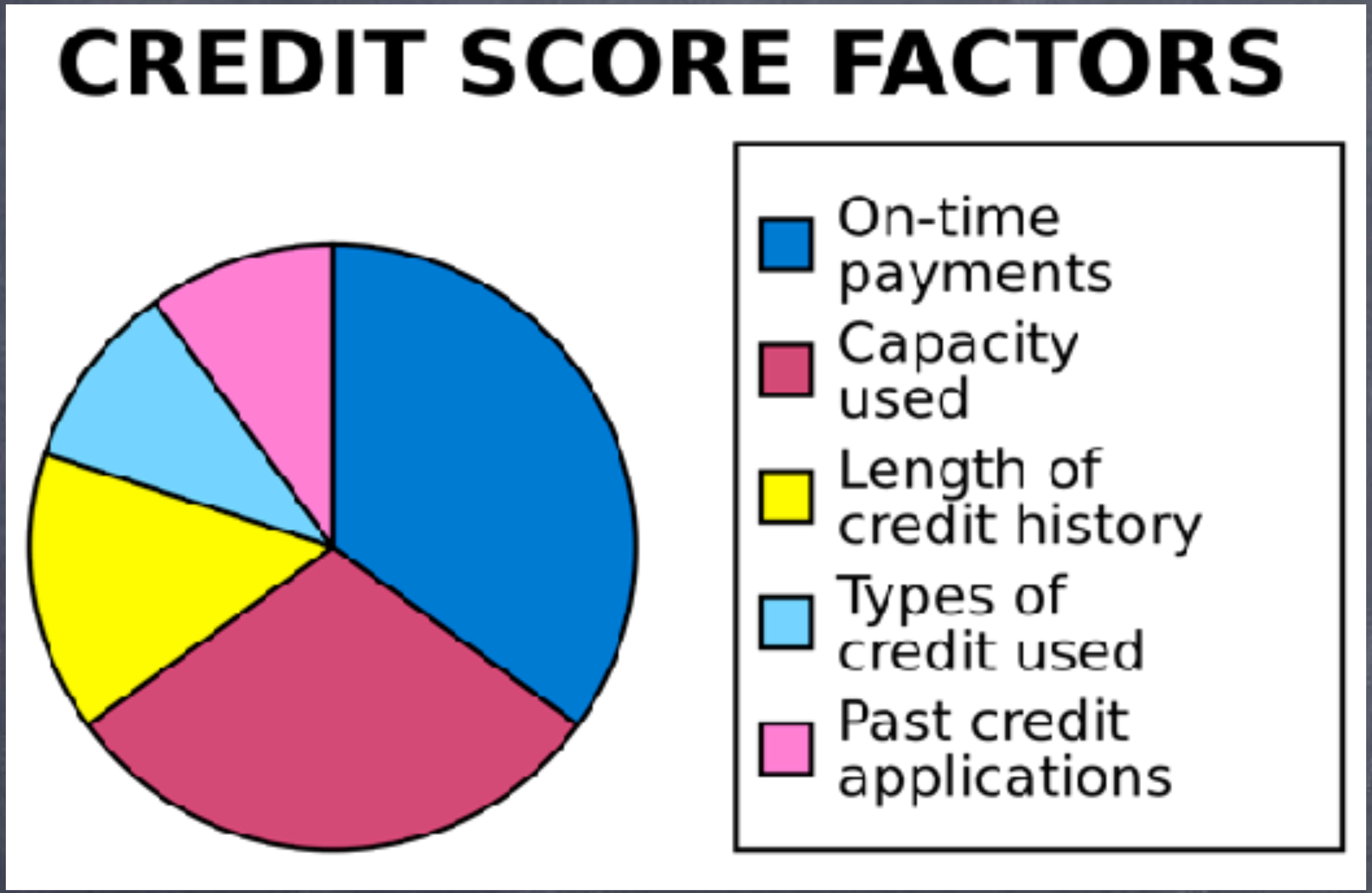
Classification



Input: **x** = $\langle x_1,\ x_2,\ x_3,\ \ldots,\ x_k \rangle$

Output: $y = \{\ \mathit{Teddy},\ \mathit{Liberty},\ \mathit{Satellite},\ \ldots,\ \mathit{Face}\ \}$

Classification



Input: x = $\langle x_1, x_2, x_3, \ldots, x_k \rangle$

Output: $y = \{\ Yes, No\ \}$

Classification

Input: x = $\langle x_1,\ x_2,\ x_3,\ \ldots,\ x_k \rangle$

Output: $y = \{\ Buy,\ Hold,\ Sell\ \}$

Classification

- Output $y = f(x)$ is one of a finite set of values (classes, categories, ...)
  - Boolean classification: yes/no or true/false
- Input is vector x of values for the attributes
  - Factored representation

AIMA Dining

- Going out to dinner in SF
- Restaurants often busy; sometimes have to wait for a table
- Decision: Given the current situation, do we wait or go somewhere else?
  - Can we automate this?

Attributes (Features)

$Alternate$: is there a suitable alternative nearby

Bar: does it have a comfy bar

$FriSat$: is it a Friday or Saturday

$Hungry$: are we hungry

$Patrons$: $None$, $Some$, $Full$

$Price$: \$,\$\$,\$\$\$

$Raining$: is it raining outside

$Reservation$: do we have a reservation

$Type$: $French$, $Italian$, $Thai$, $burger$, ...

$WaitEstimate$: 0-10, 10-30, 30-60, $>$60

Attributes (Features)

$$\mathbf{x} = \langle Alternate, Bar, FriSat, Hungry, Patrons, Price, Raining, Reservation, Type, WaitEstimate \rangle$$

$$\langle \textit{Yes, No, No, Yes, Some, \$\$\$, No, Yes, French, 0-10} \rangle$$

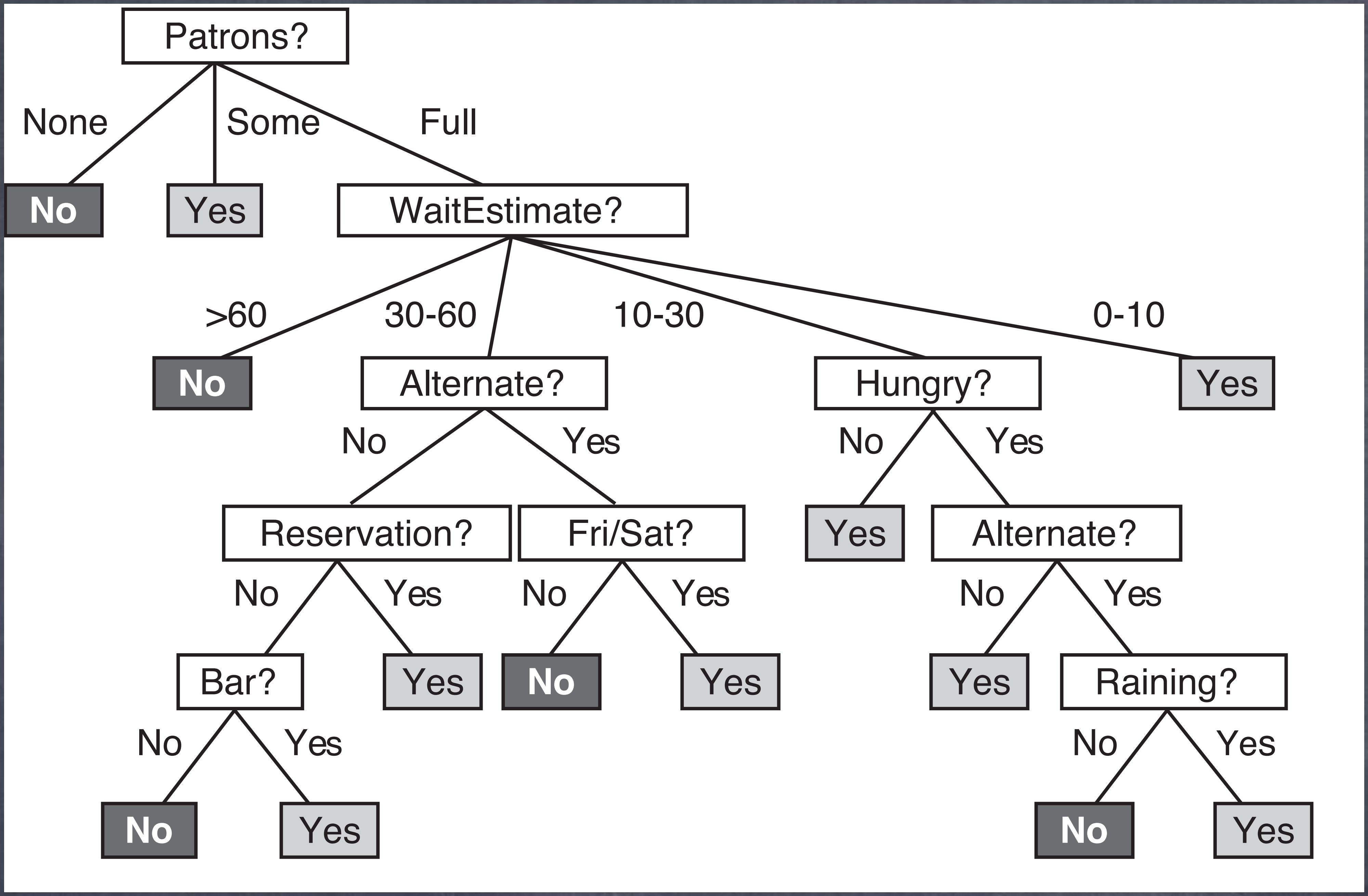
AIMA Dining

- Going out to dinner in SF
- Restaurants often busy; sometimes have to wait for a table
- Decision: Given the current situation, do we wait or go somewhere else?
  - Can we automate this?

Wikimedia

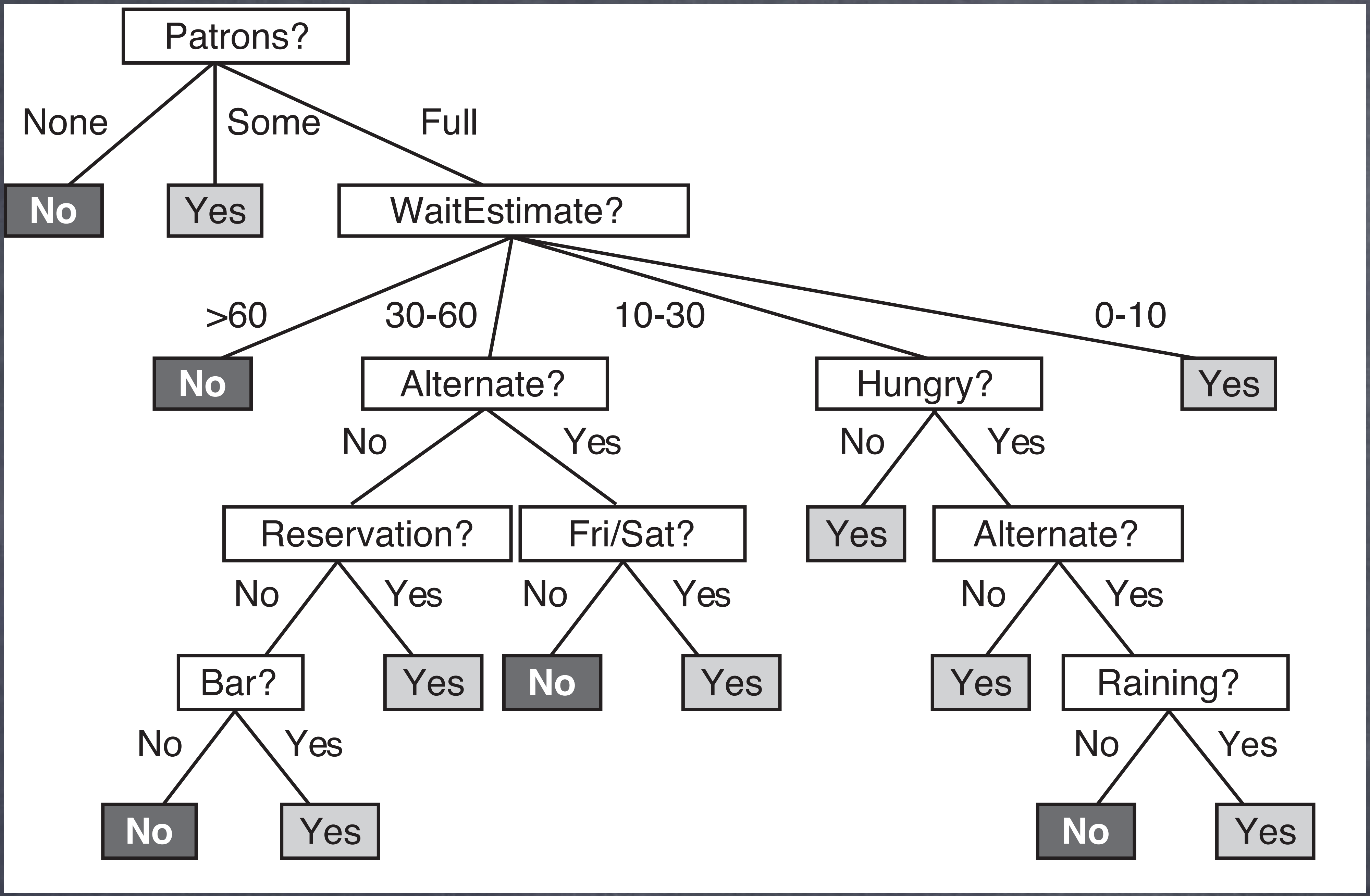
Stuart Russell's Rules

- If the host/hostess says you'll have to wait:
  - Then if there's no one in the restaurant you don't want to be there either;
  - But if there are a few people but it's not full, then you should wait
  - Otherwise you need to consider how long he/she told you the wait would be
    - …


Patrons?
None
Some
Full
No
Yes
WaitEstimate?
>60
30-60
10-30
0-10
No
Alternate?
Hungry?
Yes
No
Yes
No
Yes
Reservation?
Fri/Sat?
Yes
Alternate?
No
Yes
No
Yes
No
Yes
Bar?
Yes
No
Yes
Yes
Raining?
No
Yes
No
Yes
No
Yes
No
Yes

Decision Tree

- Each non-leaf node in the tree represents a test on a single attribute
- Children of the node are labelled with the possible values of the attribute
- Each path represents a series of tests, and the leaf node gives the value of the function when the input passes those tests


Patrons?
None
Some
Full
No
Yes
WaitEstimate?
>60
30-60
10-30
0-10
No
Alternate?
Hungry?
Yes
No
Yes
No
Yes
Reservation?
Fri/Sat?
Yes
Alternate?
No
Yes
No
Yes
No
Yes
Bar?
Yes
No
Yes
Yes
Raining?
No
Yes
No
Yes
No
Yes
No
Yes

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*\$\$\$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*\$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*\$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*\$\$\$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*\$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*\$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$\$\$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*\$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = yes$

Inducing Decision Trees From Examples

- Examples: (x, y) where x is a vector of values for the input attributes and y is a single Boolean value

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = yes$

Inducing Decision Trees From Examples

- Examples: (x, y)
- Want a shallow tree (short paths, fewer tests)
- Greedy algorithm (AIMA Fig 18.5)
  - Always test the most important attribute first
  - Because it makes the most difference to the classification of an example

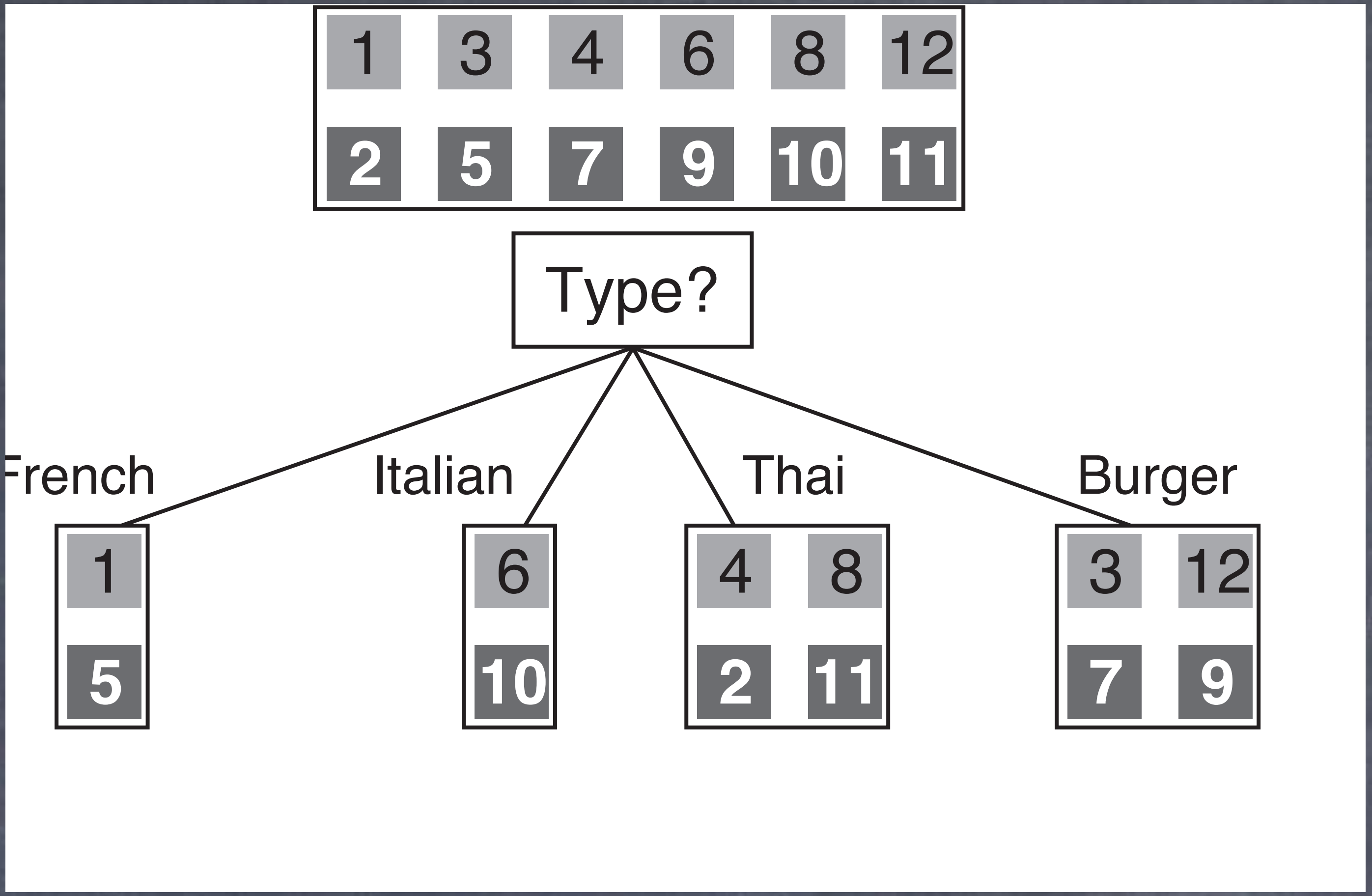
	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	>60	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	>60	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = yes$

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	y_1=*yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	y_2=*no*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	y_3=*yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	y_4=*yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	>60	y_5=*no*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	y_6=*yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	y_7=*no*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	y_8=*yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	>60	y_9=*no*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	y_{10}=*no*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	y_{11}=*no*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	y_{12}=*yes*

Examples

Positive

Negative



	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	y_1=*yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	y_2=*no*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	y_3=*yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	y_4=*yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	y_5=*no*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	y_6=*yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	y_7=*no*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	y_8=*yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	y_9=*no*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	y_{10}=*no*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	y_{11}=*no*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	y_{12}=*yes*

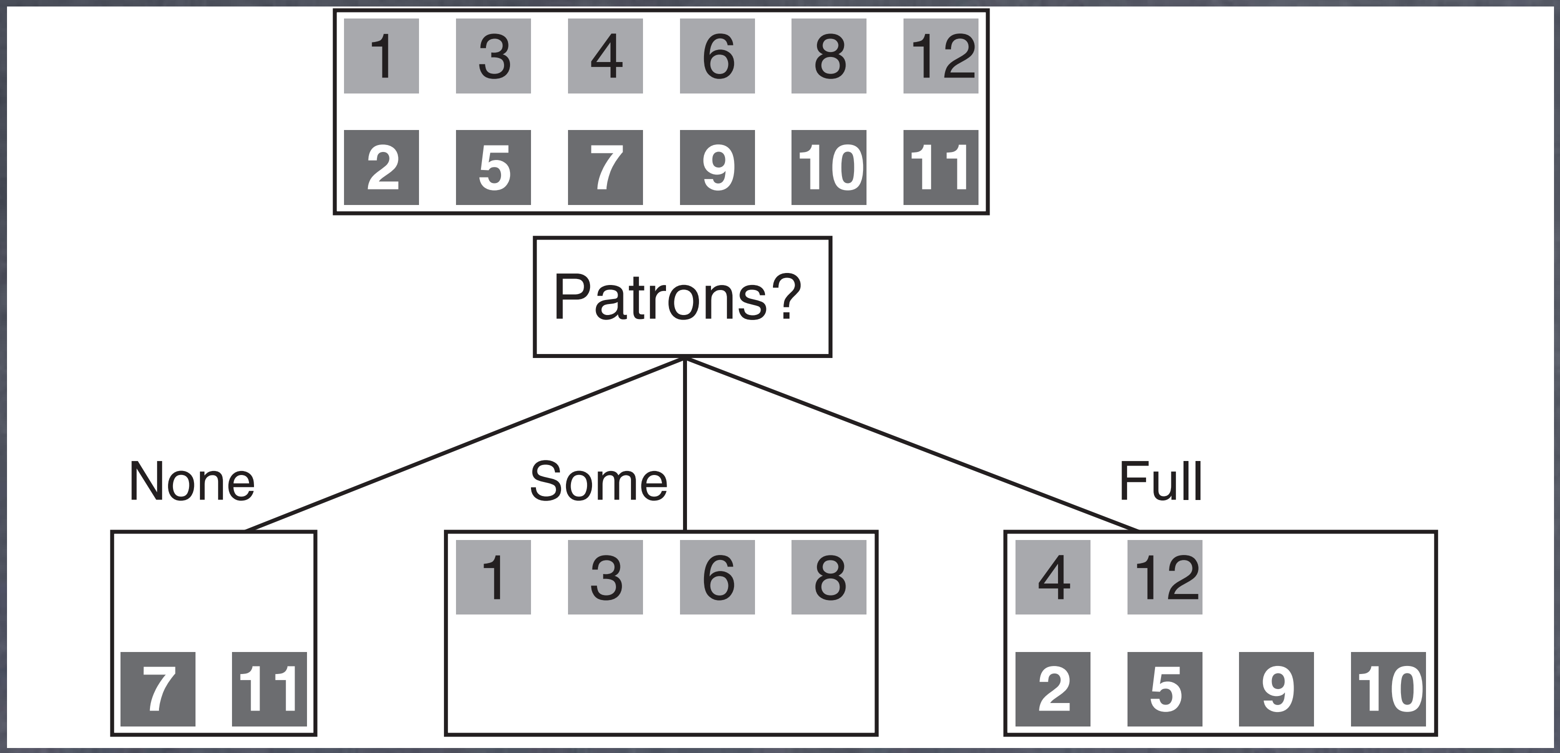
	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = yes$

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*\$\$\$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*\$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*\$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*\$\$\$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*\$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*\$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$\$\$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*\$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = yes$

Examples

Positive

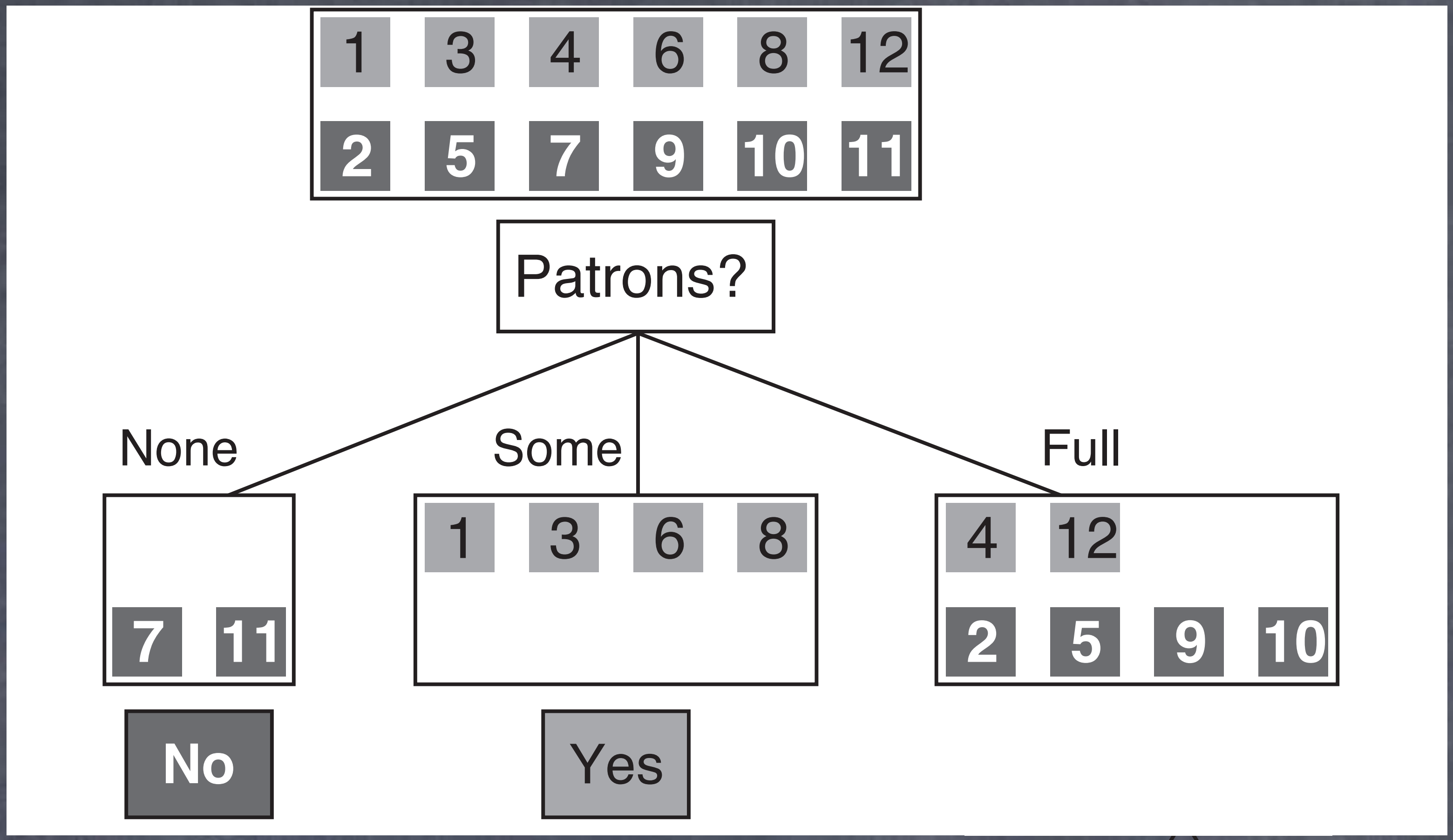
Negative



Examples

Positive

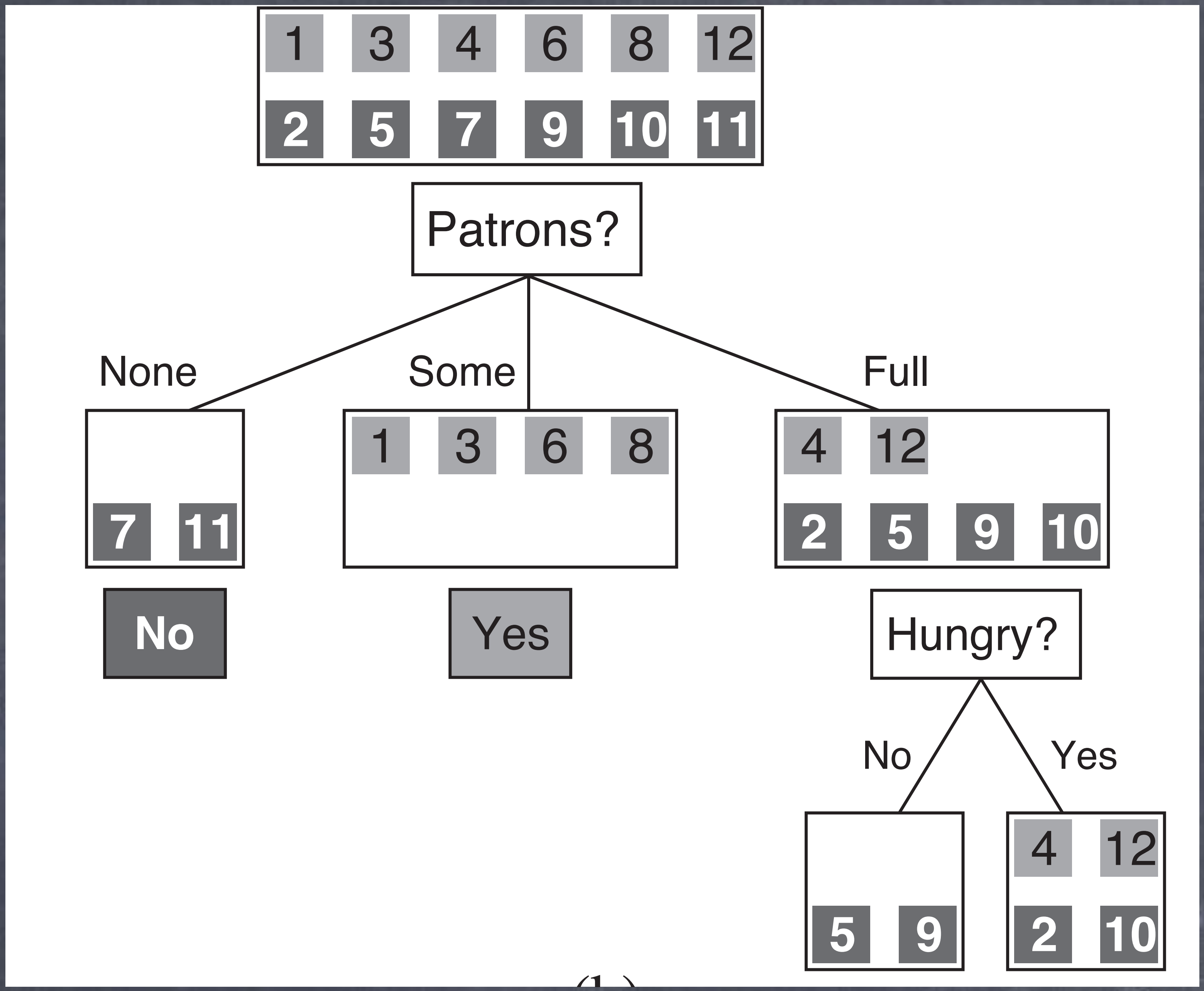
Negative



Examples

Positive

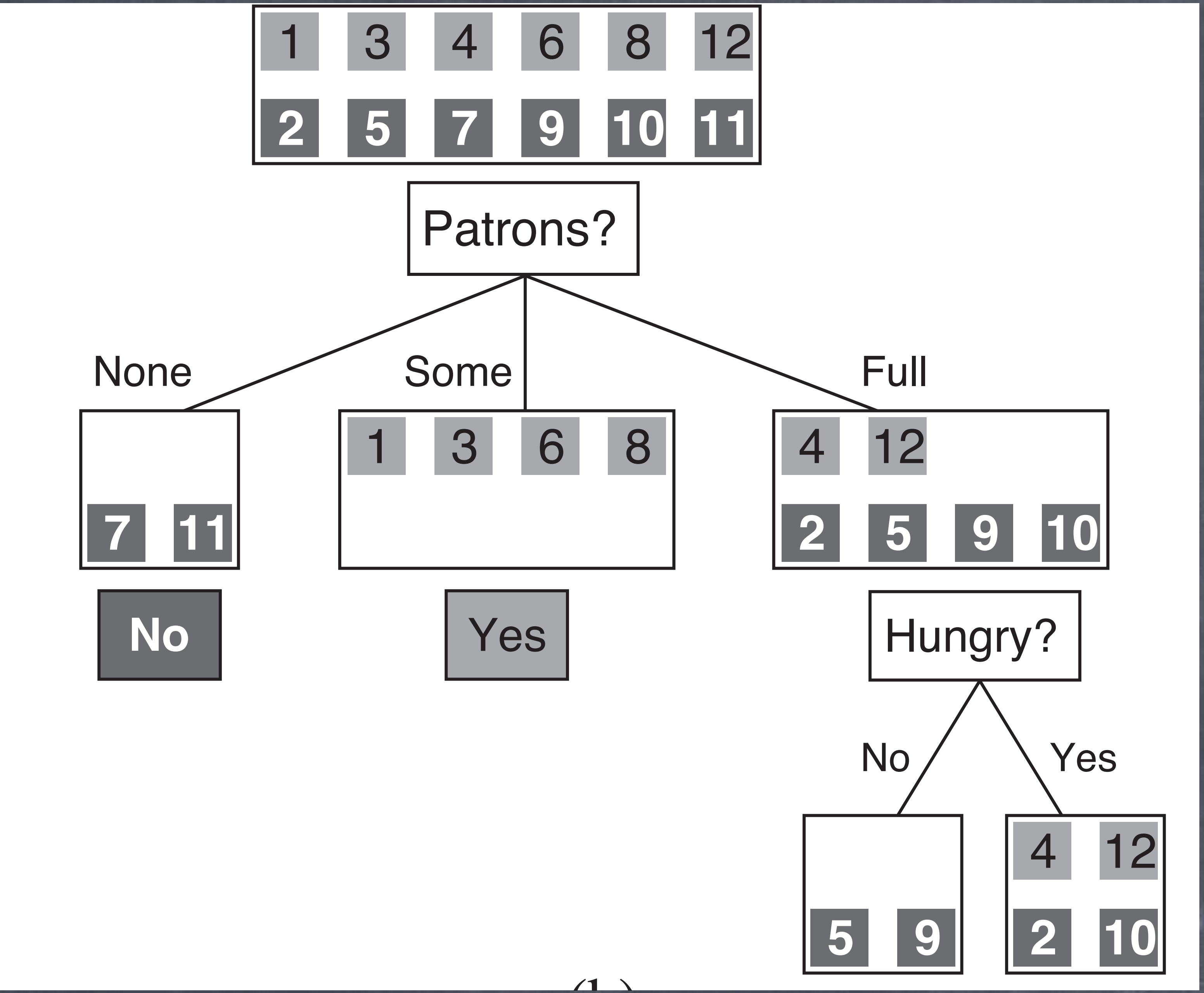
Negative



Examples

Positive

Negative

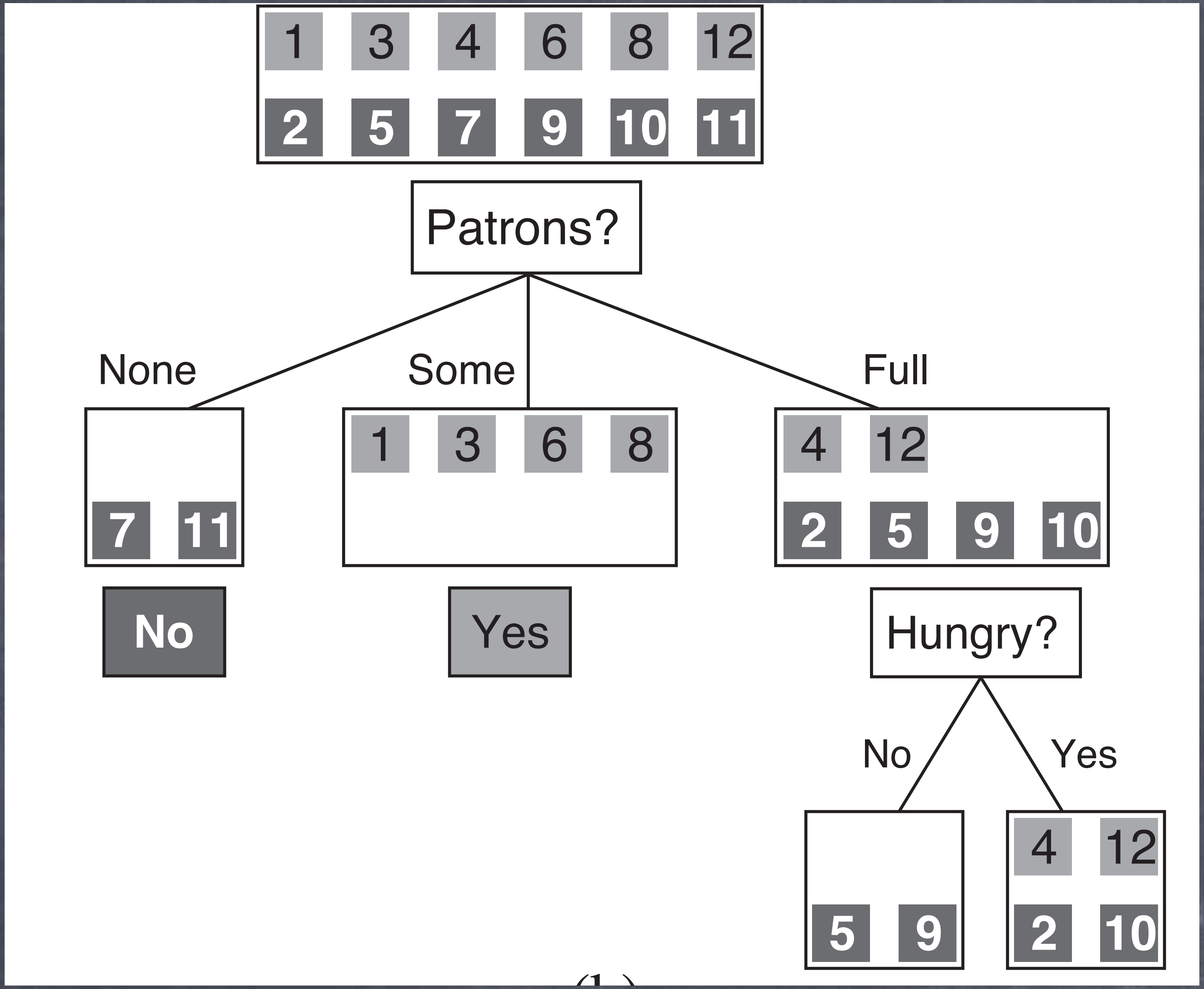


Keep splitting until...

Examples

Positive

Negative



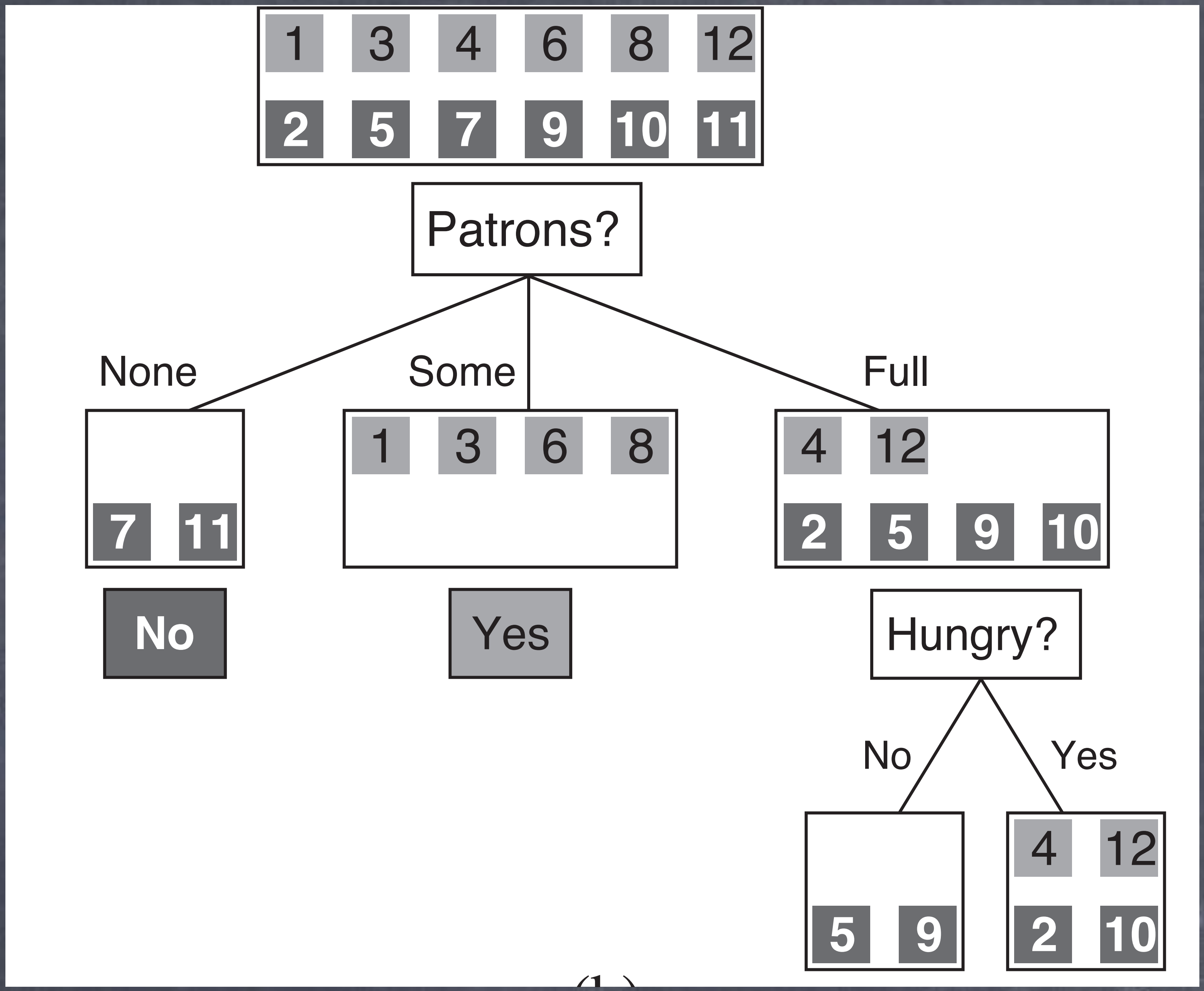
Keep splitting until...

No examples left: Decision tree classifies perfectly

Examples

Positive

Negative

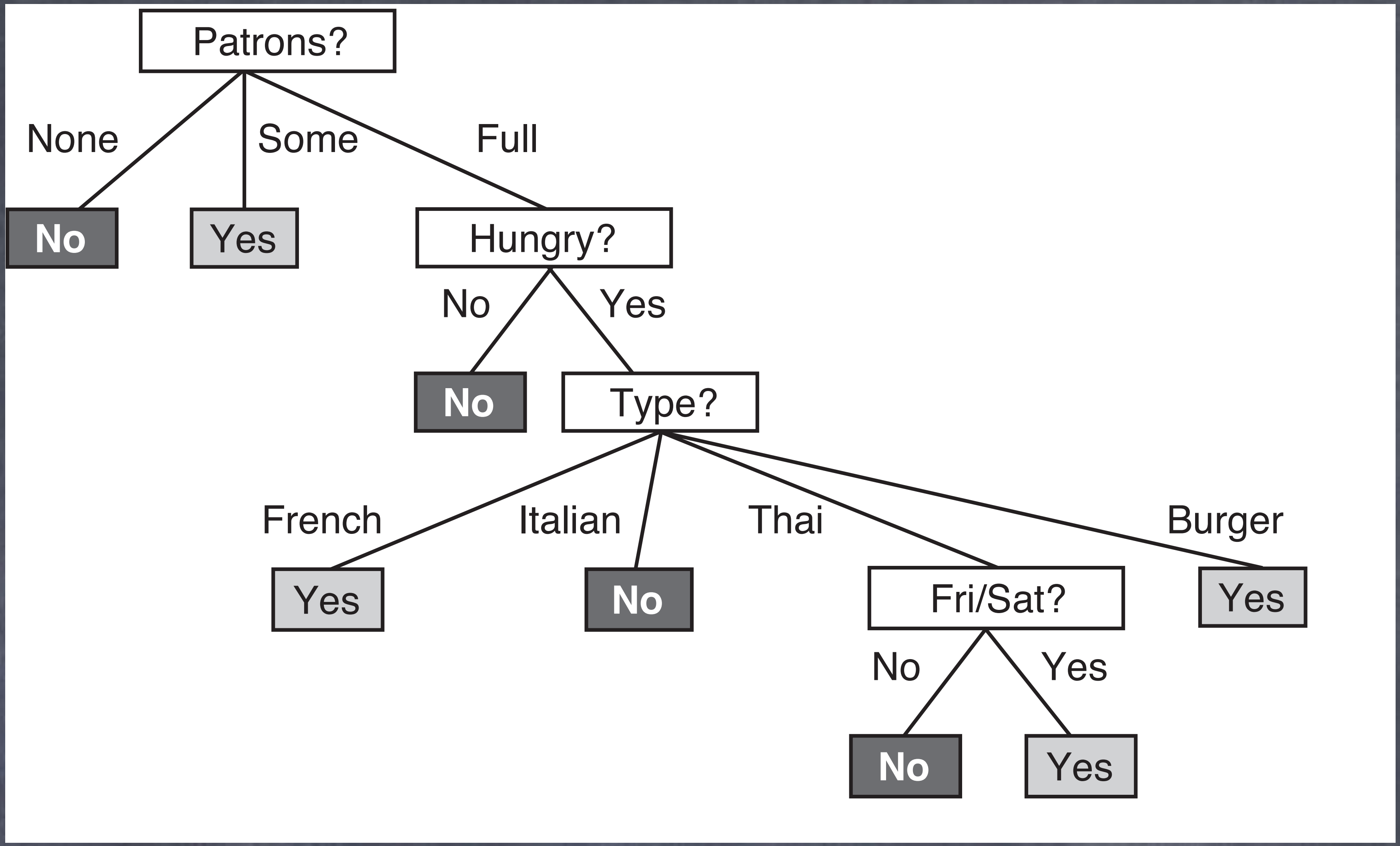


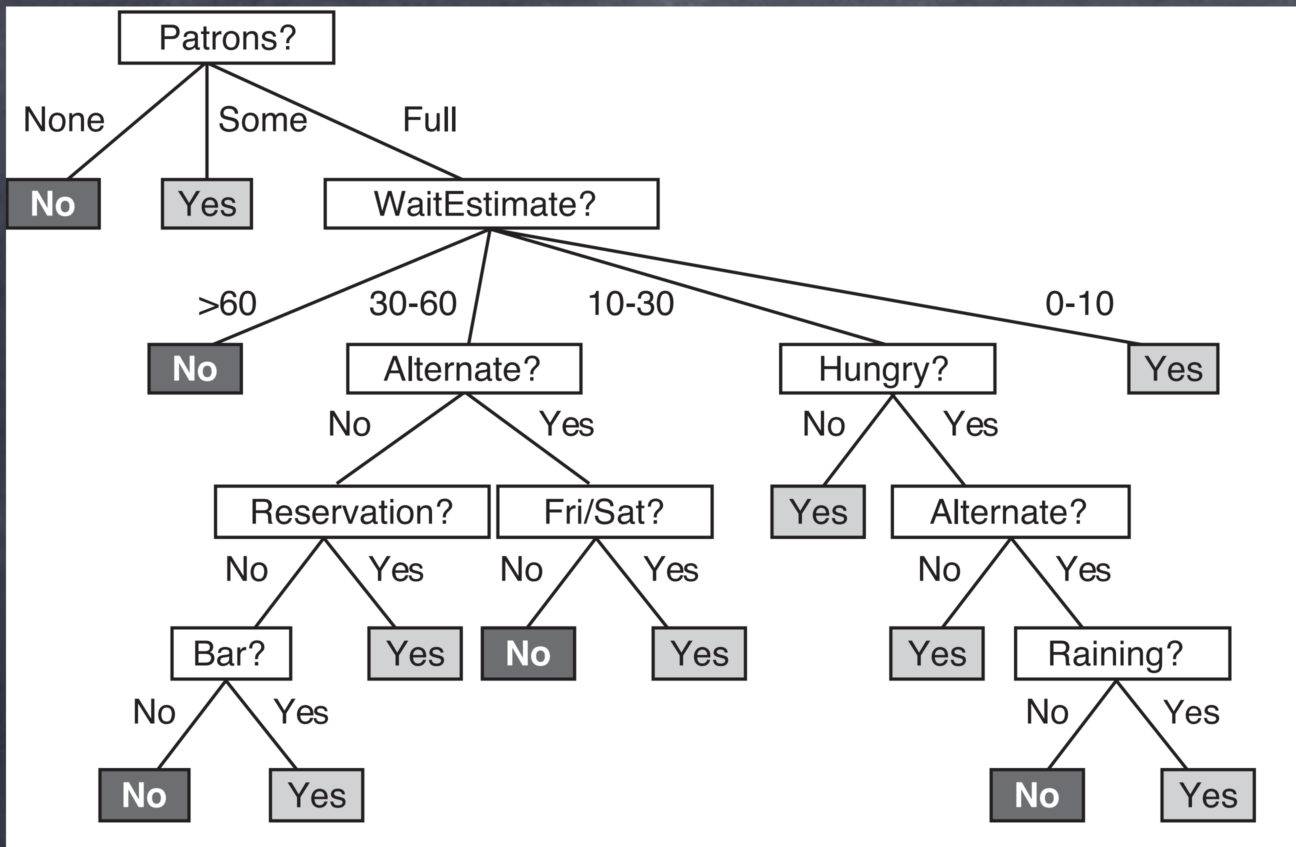
Keep splitting until...

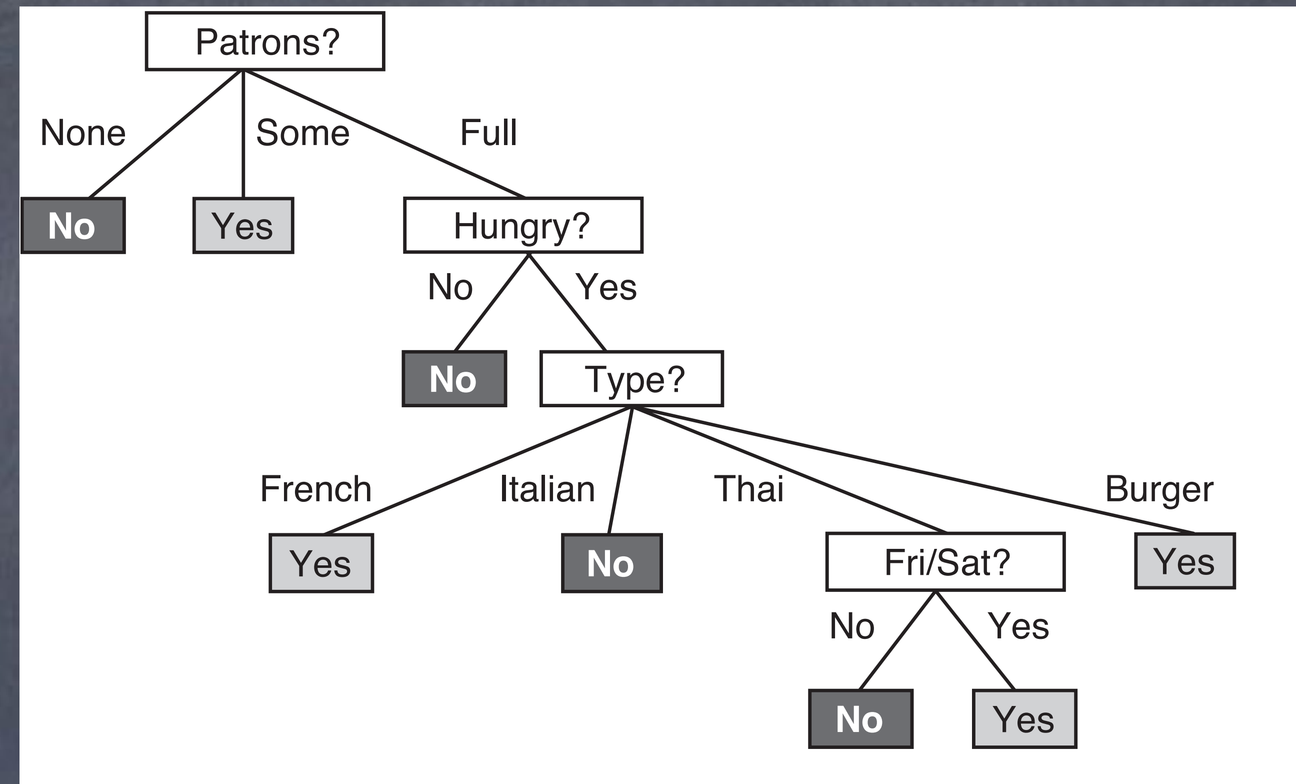
No attributes left: Some examples have the same description (attribute values) but different classifications (outputs)

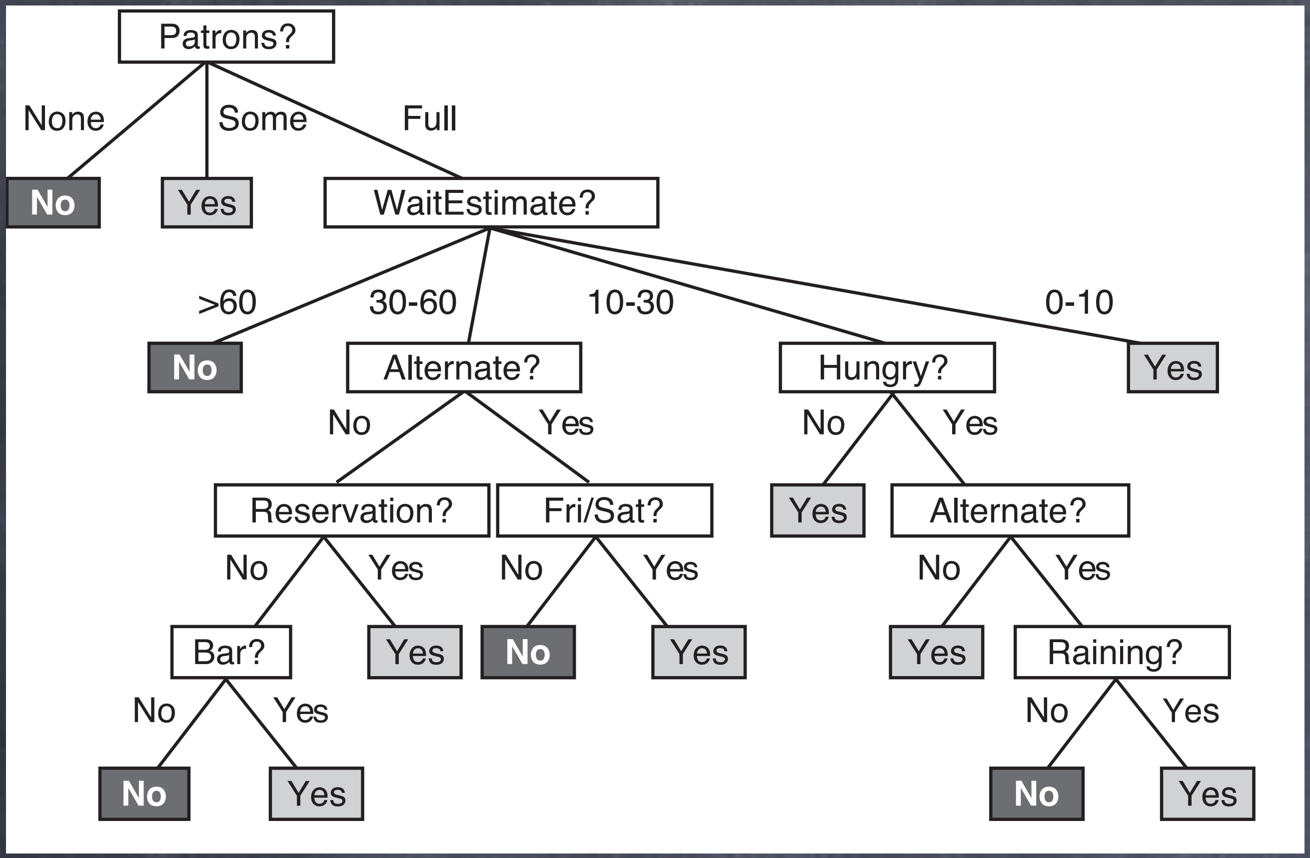
Sources of Error

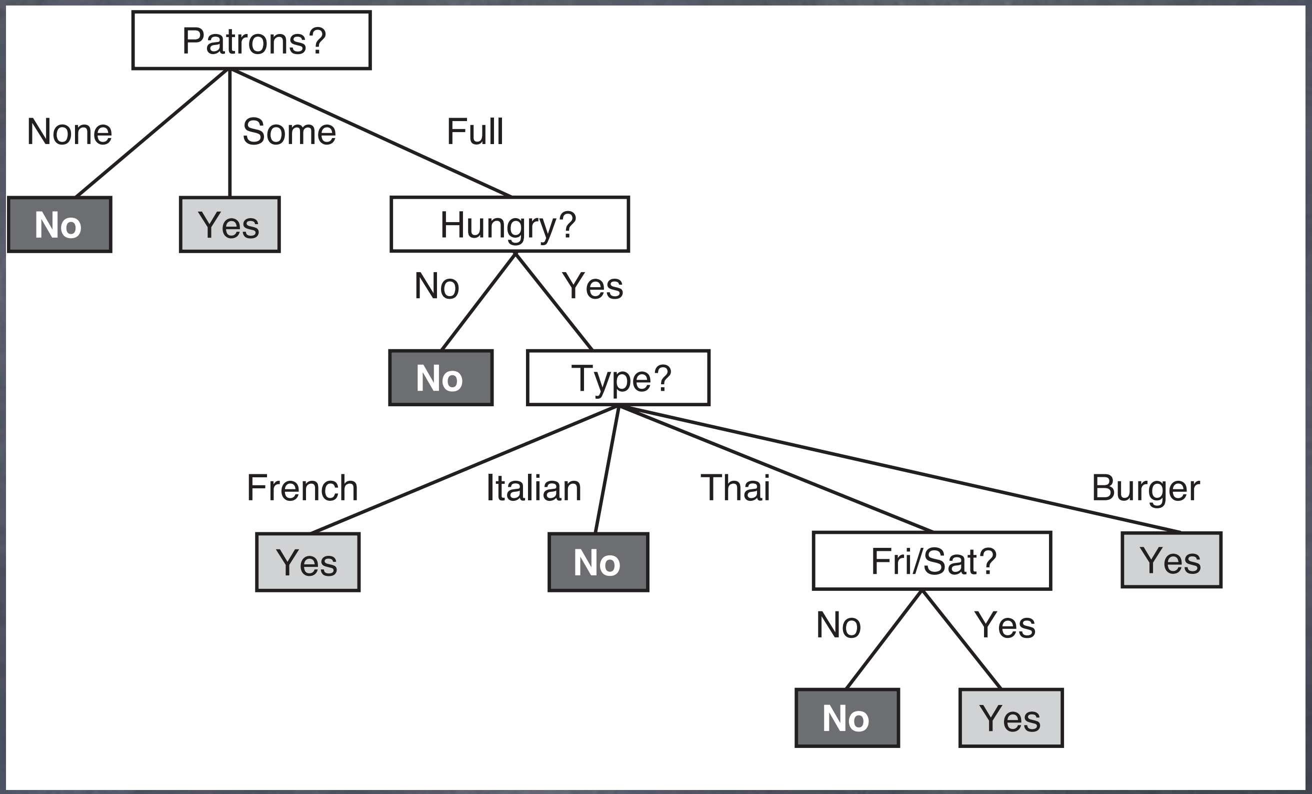
- Examples have same description in terms of input attributes but different classification results
  - Error or noise in the data
  - Nondeterministic domain
  - We can't observe the attribute that would distinguish the examples


Patrons?
None
Some
Full
No
Yes
Hungry?
No
Yes
No
Type?
French
Italian
Thai
Burger
Yes
No
Fri/Sat?
Yes
No
Yes
No
Yes


Patrons?
None
Some
Full
No
Yes
WaitEstimate?
>60
30-60
10-30
0-10
No
Alternate?
Hungry?
Yes
No
Yes
No
Yes
Reservation?
Fri/Sat?
Yes
Alternate?
No
Yes
No
Yes
No
Yes
Bar?
Yes
No
Yes
Yes
Raining?
No
Yes
No
Yes
No
Yes
No
Yes


Patrons?
None
Some
Full
No
Yes
Hungry?
No
Yes
No
Type?
French
Italian
Thai
Burger
Yes
No
Fri/Sat?
Yes
No
Yes
No
Yes



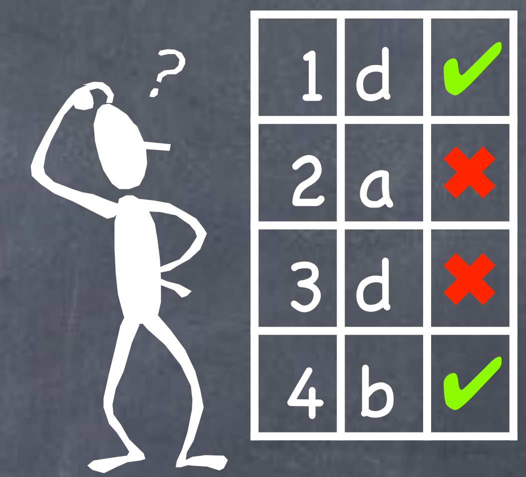


Learning lets the data speak for itself

Decision Trees

- Represent sequence of tests that lead to a decision
- Compact representation of how to make the decision
- Can be learned from examples
- Decision trees have explanatory power!

Types of Learning



Unsupervised
(no feedback)

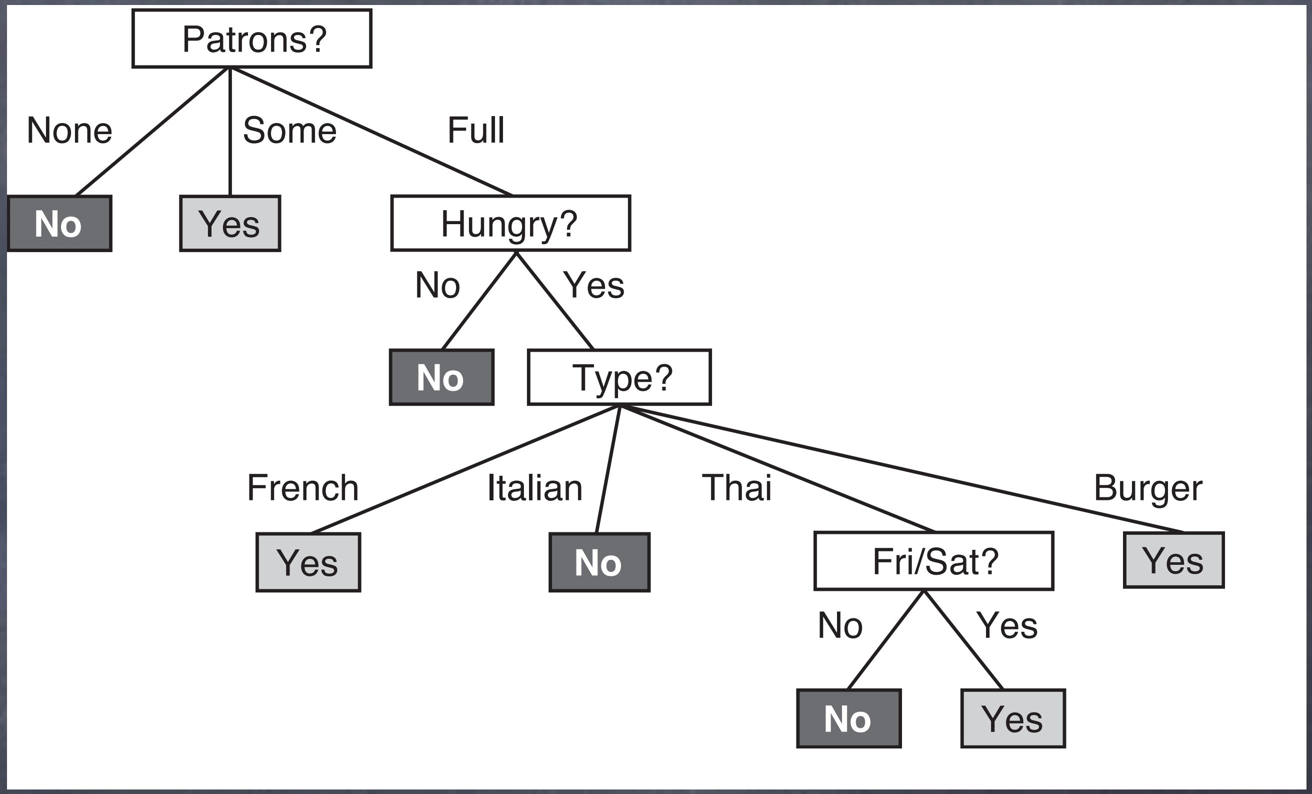
Semi-supervised

Supervised
(labelled
examples)

Reinforcement
(feedback is reward)

Supervised Learning

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = ye$



Supervised Learning

- Given a training set of N example input-output pairs:

$$(\mathbf{x}_1,\ y_1),\ (\mathbf{x}_2,\ y_2),\ \ldots,\ (\mathbf{x}_\mathrm{N},\ y_\mathrm{N})$$

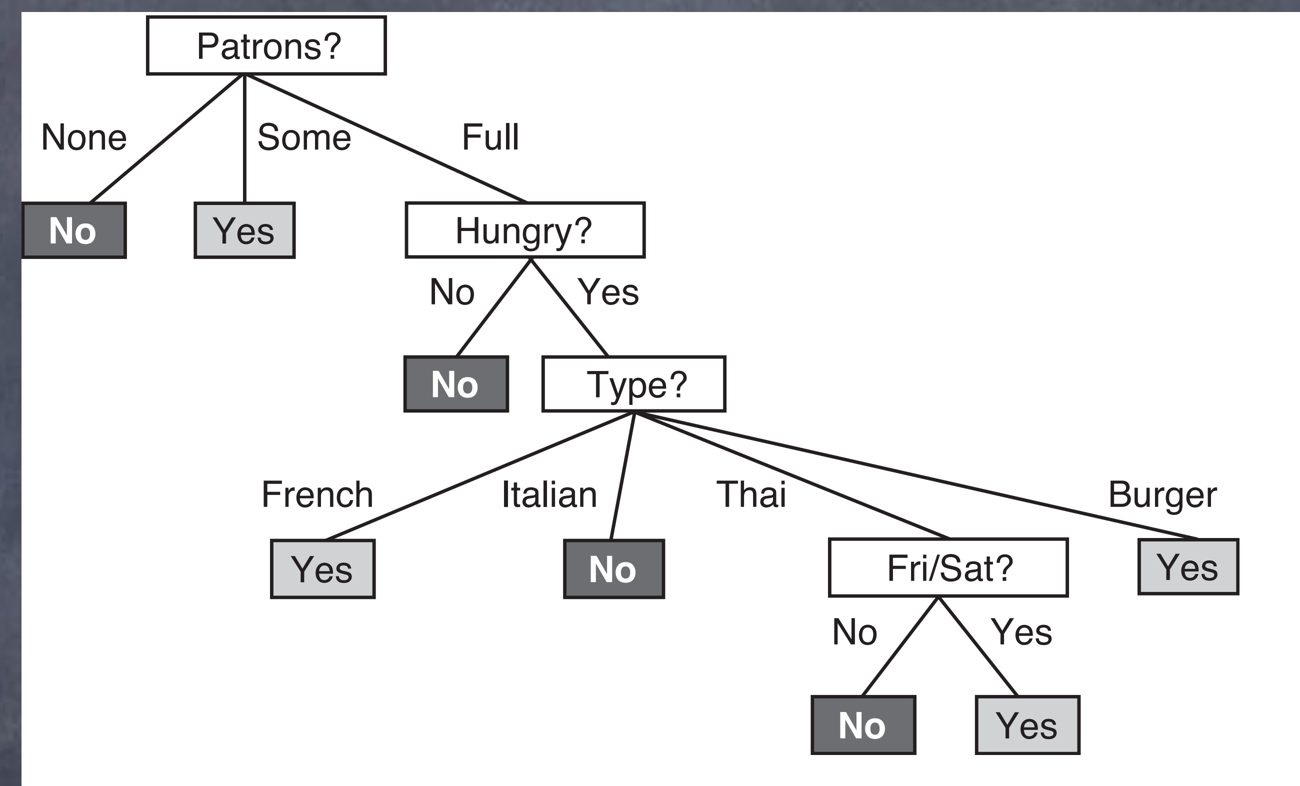
Training Data

where each $y_j = f(\mathbf{x}_j)$

- Discover function h that approximates f
- Search through the space of possible hypotheses for one that will perform well

Supervised Learning

	Input Attributes										Will Wait
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0-10*	$y_1 = yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 = no$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 = yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 = yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = no$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 = yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 = no$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 = yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = no$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} = no$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} = no$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} = ye$



Hypotheses: decision trees

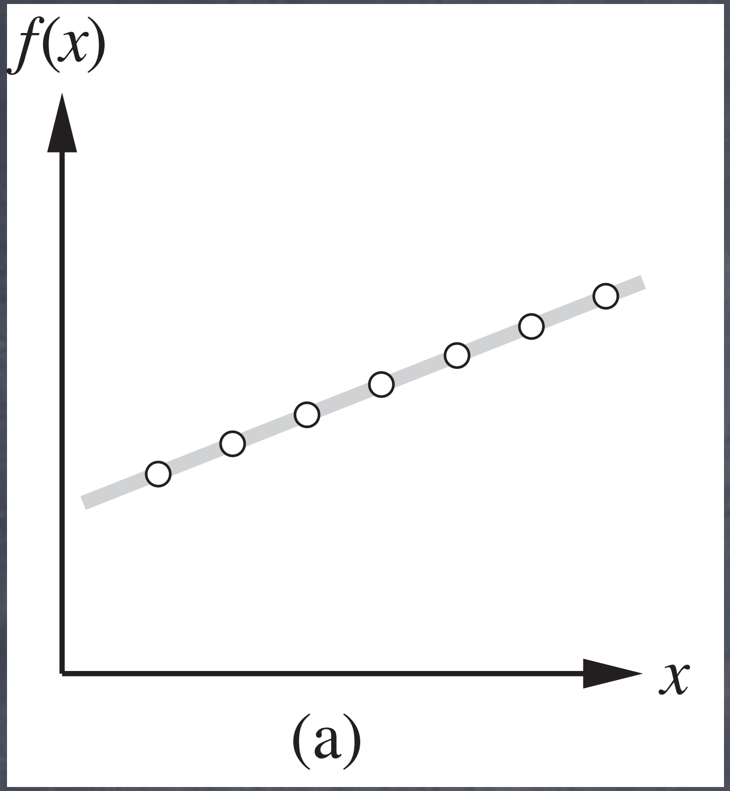
Supervised Learning

- Given a training set of N example input-output pairs:
  $$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$
  where each $y_j = f(\mathbf{x}_j)$
- Discover function h that approximates f
- Search through the space of possible hypotheses for one that will perform well

Hypothesis Space

Hypothesis Space

- Decision trees (Boolean formulas)
- Linear functions $y = mx + b$
- Polynomials (of some degree)
- Java programs?
- Turing machines?

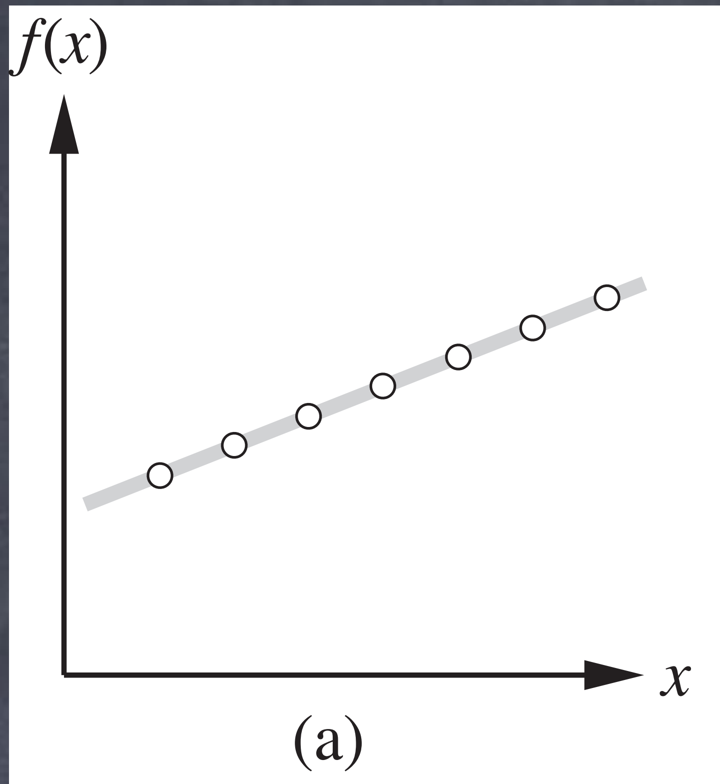


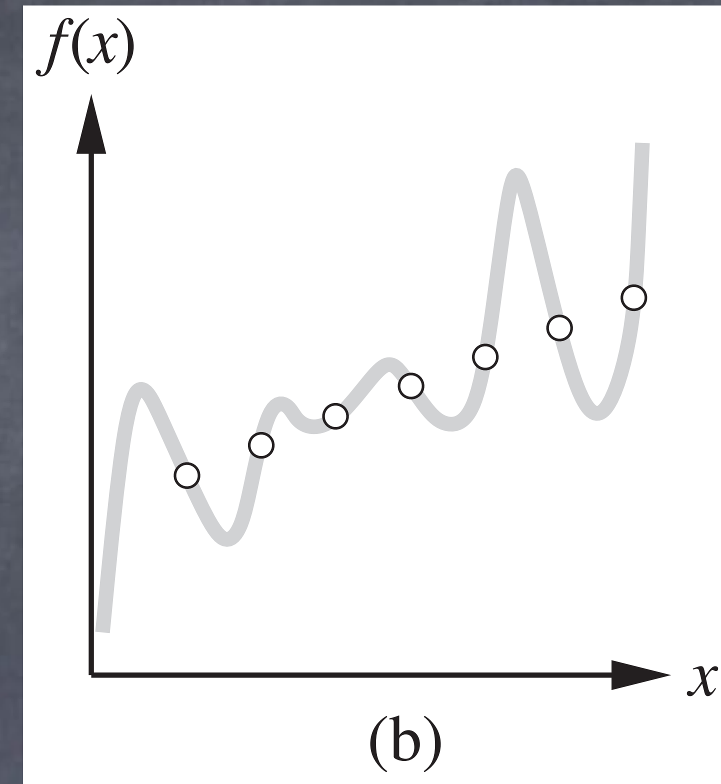
Hypothesis space:

$$y = mx + b$$

Hypothesis:

$$y = -0.4x + 3$$





Hypothesis space:

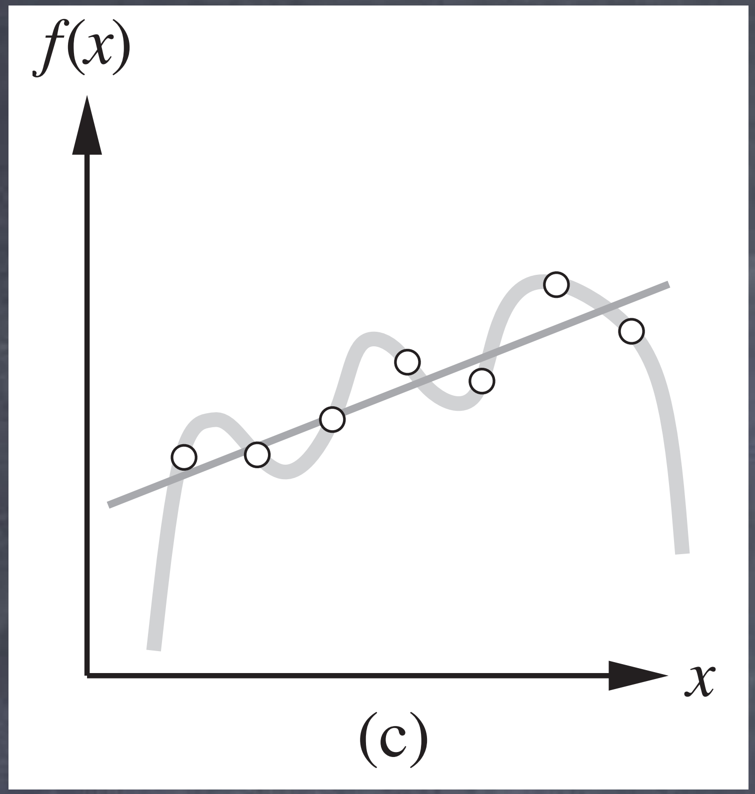
$$y = mx + b$$

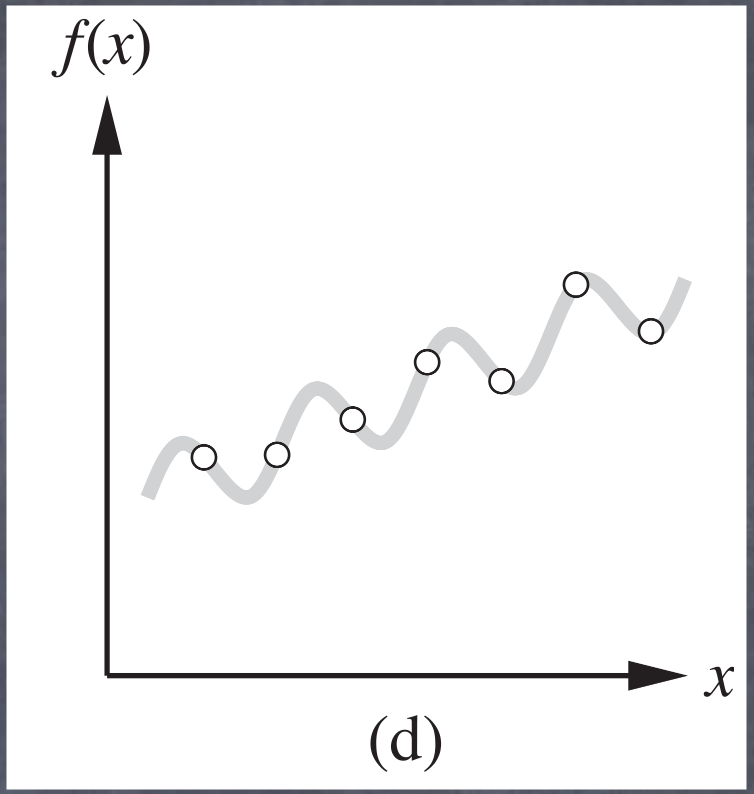
$$y = c_7 x^7 + c_6 x^6 + \ldots + c_1 x + c_0$$

$$= \sum_{i=0}^{7} c_i x^i$$

Hypothesis:

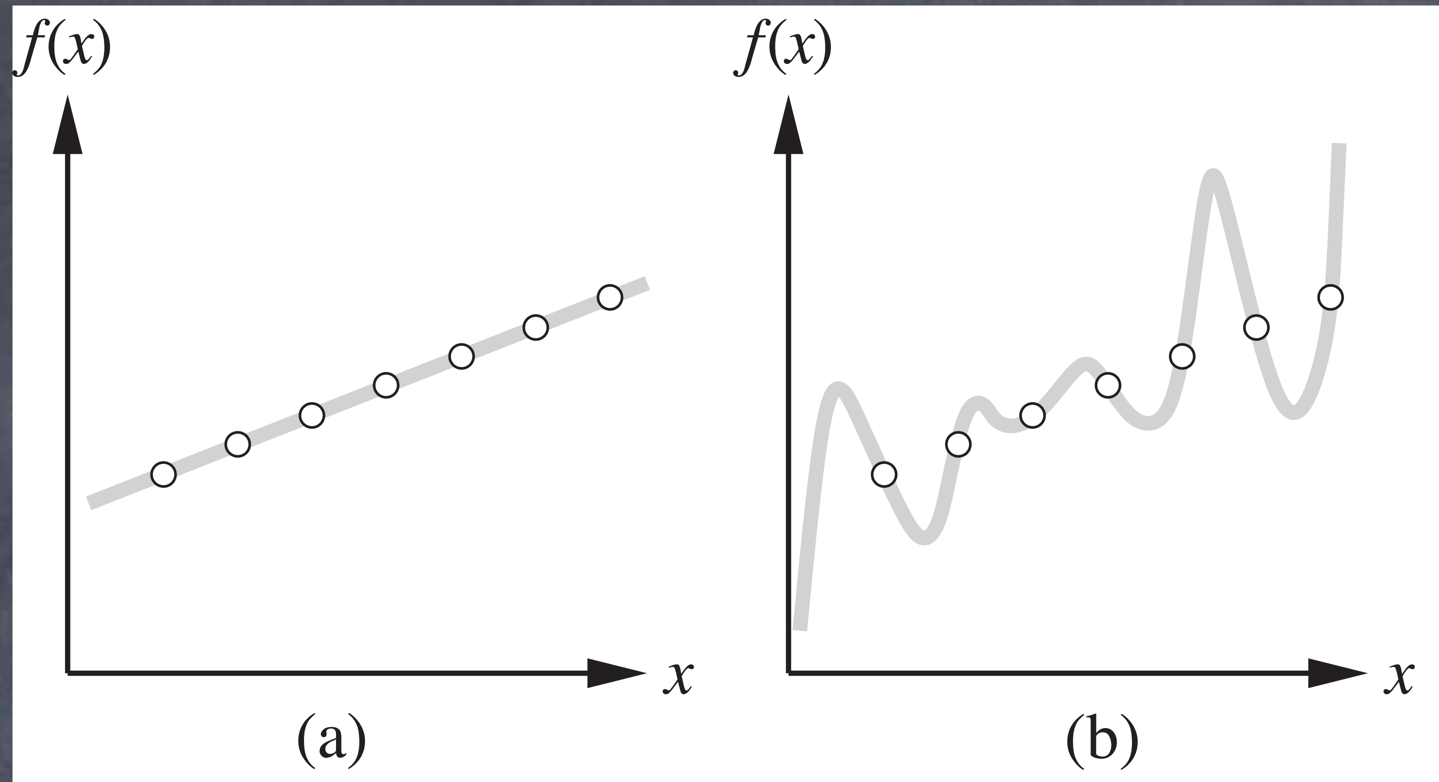
$$y = -0.4x + 3$$

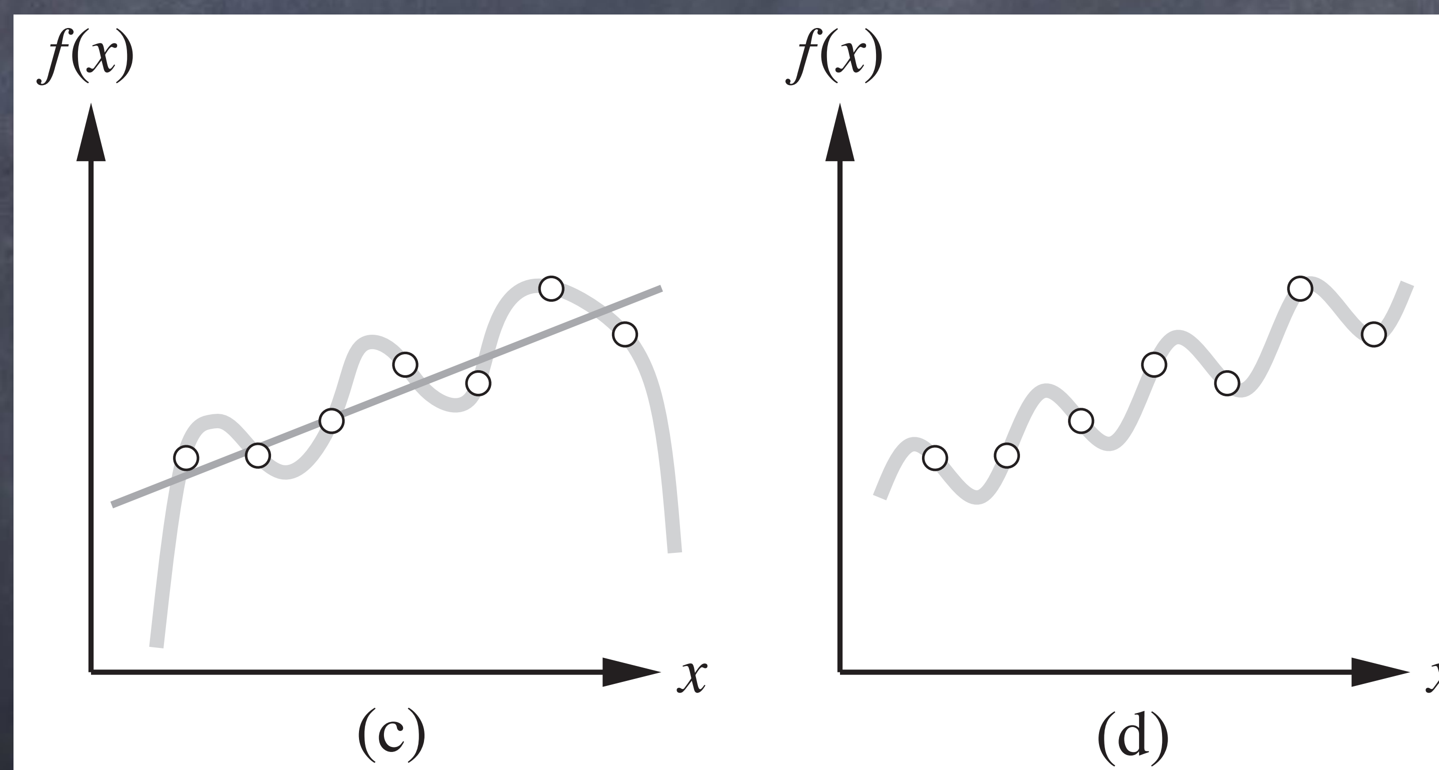




$$y = c_6 x^6 + c_5 x^5 \ldots + c_1 x + c_0$$

$$ax + b + c \sin(x)$$


f(x)
x
(a)
f(x)
x
(b)


f(x)
x
(c)
f(x)
x
(d)

Evaluating Hypotheses

- Accuracy: fits the data
- Generalization: predicts outputs for unseen inputs
- Simplicity and “searchability”

Evaluating Accuracy

Evaluating Accuracy

- Training set: $(\mathbf{x}_1,\ y_1),\ (\mathbf{x}_2,\ y_2),\ \ldots,\ (\mathbf{x}_N,\ y_N)$
- Test accuracy of h by comparing $h(\mathbf{x}_j)$ to y_j

Training Data

$\mathbf{x}$	y
1	3
2	6
4	12
5	15
7	21

$$h(\mathbf{x}) = ?$$

Testing Data

$\mathbf{x}$	y
2	6
7	21

$$f(\mathbf{x}) = y$$

$$h(\mathbf{x}) = y?$$

Evaluating Accuracy

- Training set: $(\mathbf{x}_1,\ y_1),\ (\mathbf{x}_2,\ y_2),\ \ldots,\ (\mathbf{x}_N,\ y_N)$
- Test accuracy of h by comparing $h(\mathbf{x}_j)$ to y_j

Generalization

- A hypothesis (function) generalizes well if it correctly predicts the value of y for novel examples x

Cross-Validation

- Randomly split data into training and testing (in some proportion)
  - Hold out test data during training
- Doesn't use all data for training

k-Fold Cross-Validation

- Divide data into k equal subsets
- Perform k rounds of learning
  - Leave out 1 subset ($1/k$ of the data) each round; use for testing that round
- Average test scores over k rounds

Evaluating Hypotheses

- Accuracy: fits the data
- Generalization: predicts outputs for unseen inputs
- Simplicity and "searchability"

Simplicity

- Hypothesis space easier to search
- Less likely to memorize the data
- Easier to compute
  - During learning
  - During decision-making

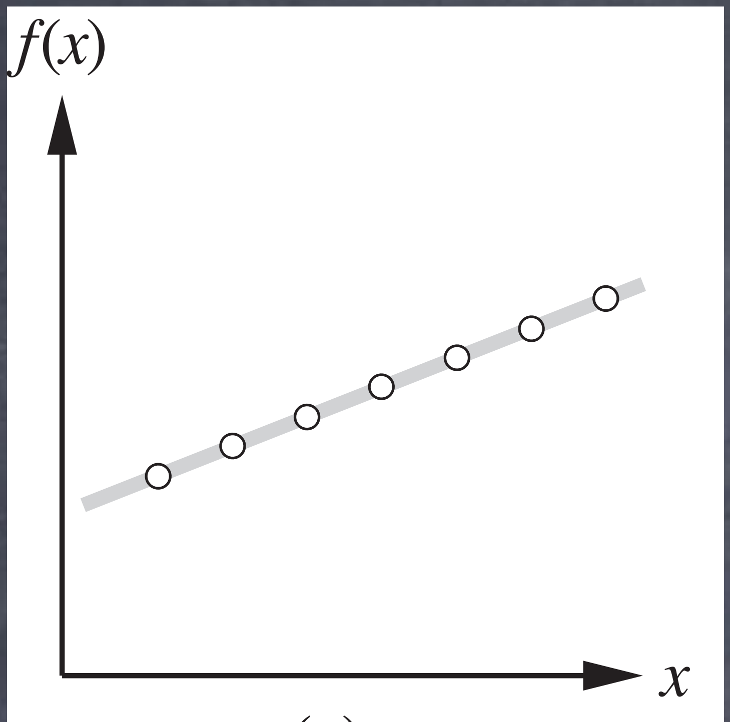
Occam's Razor

William of Occam (or Ockham)

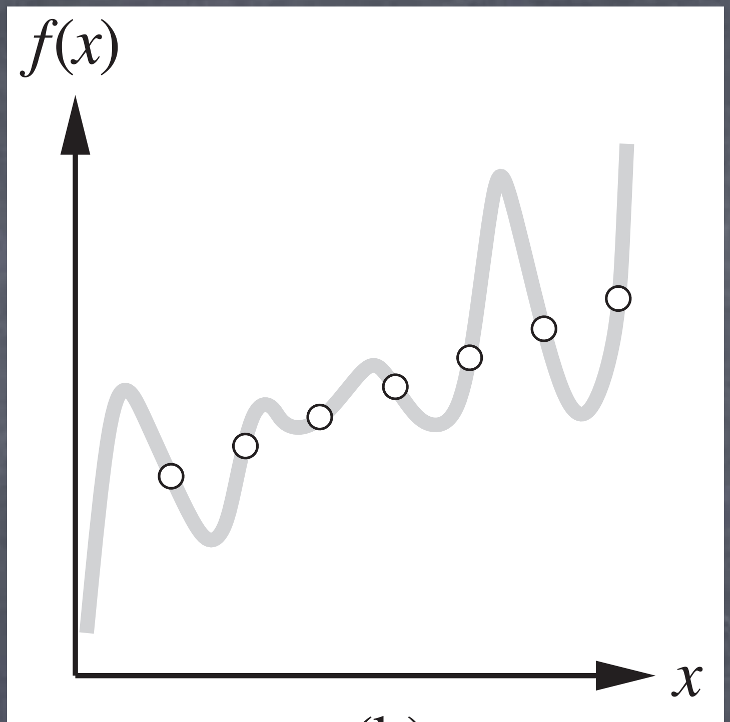
14th c.

Evaluating Hypotheses

- Accuracy: fits the data
- Generalization: predicts outputs for unseen inputs
- Simplicity and "searchability"



(a)



(b)

Hypothesis space:

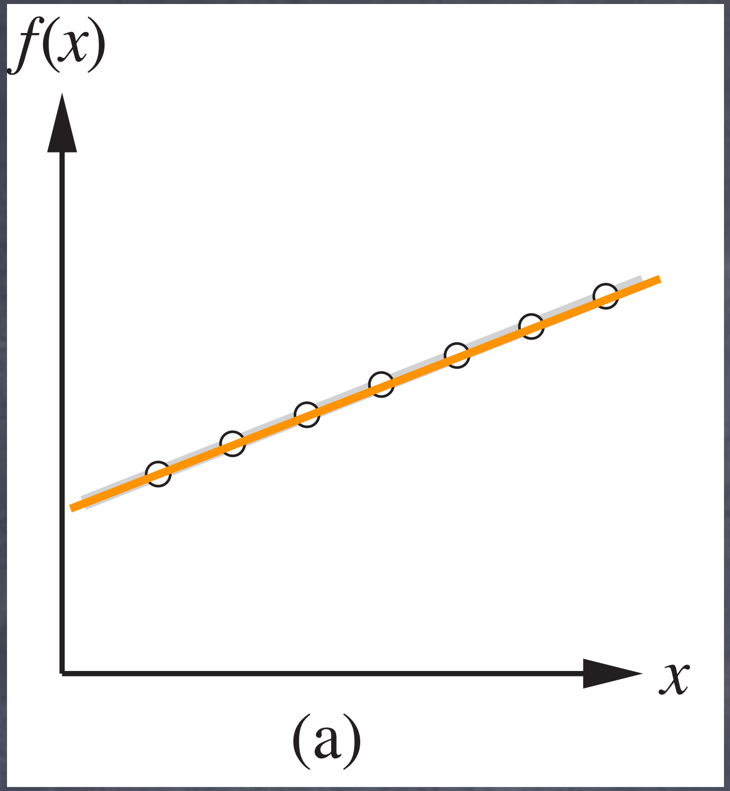
$$y = mx + b$$

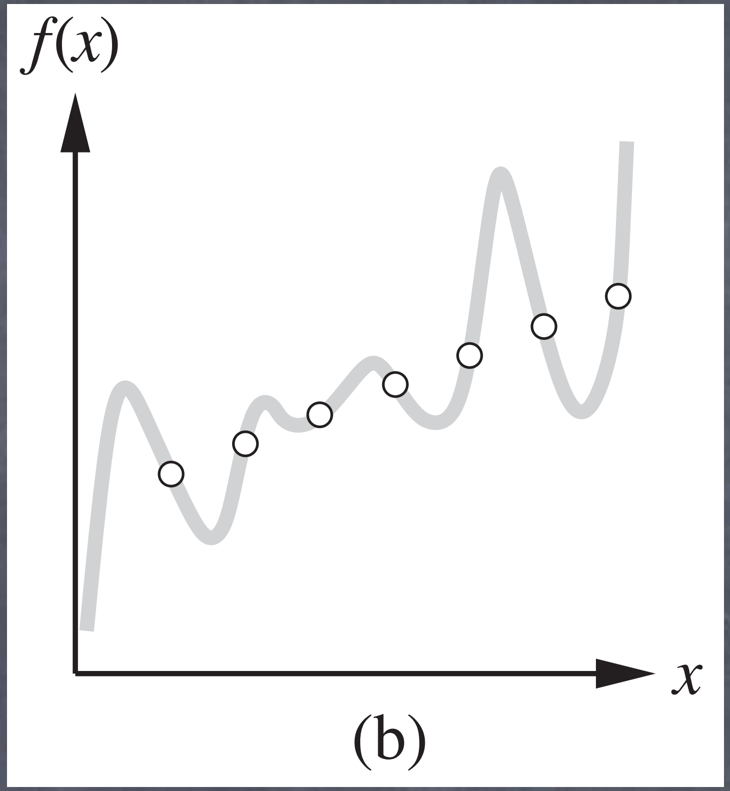
$$y = c_7x^7 + c_6x^6 + \ldots + c_1x + c_0$$

$$= \sum_{i=0}^{7} c_i x^i$$

Hypothesis:

$$y = -0.4x + 3$$





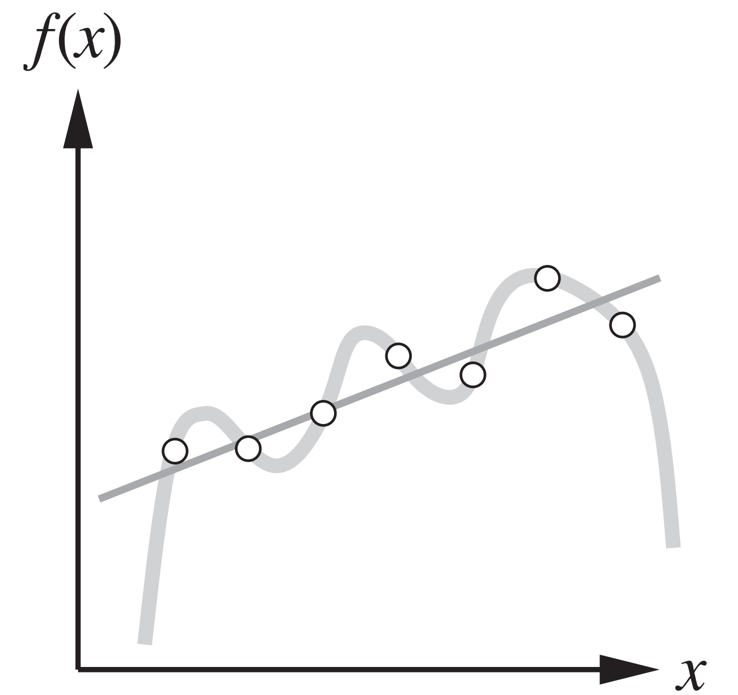
Hypothesis space:

$$y = mx + b$$

$$y = c_7 x^7 + c_6 x^6 + \ldots + c_1 x + c_0 = \sum_{i=0}^{7} c_i x^i$$

Hypothesis:

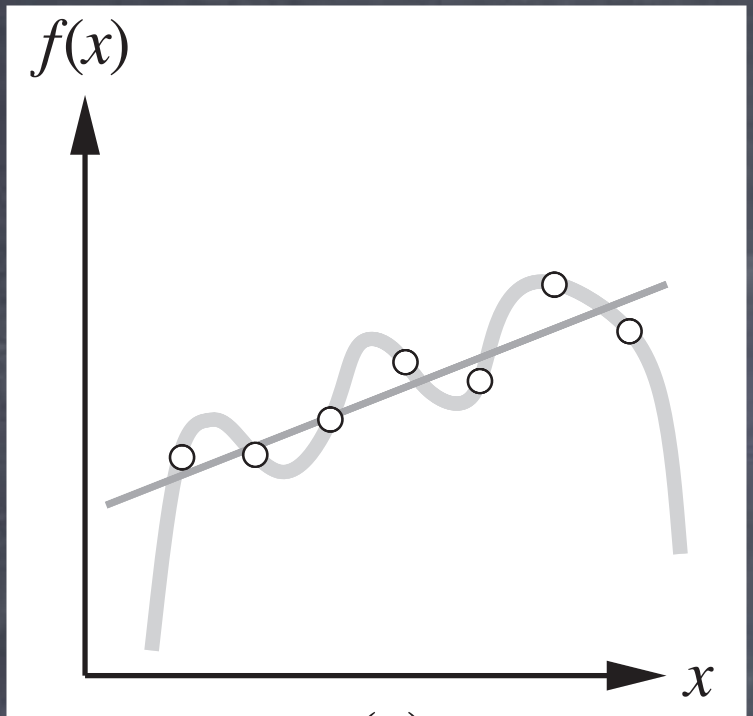
$$y = -0.4x + 3$$



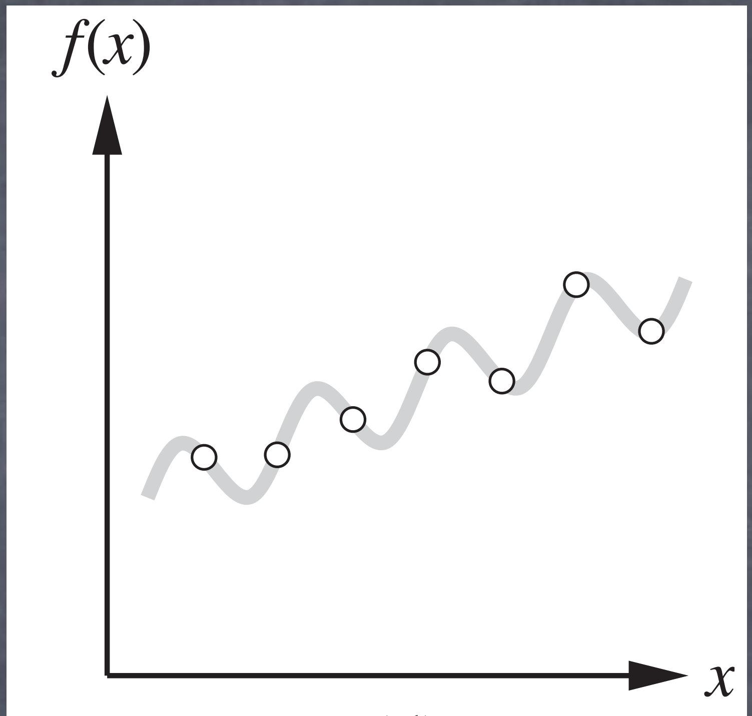
(c)

$$y = c_6 x^6 + c_5 x^5 \ldots + c_1 x + c_0$$

$$y = mx + b$$



(c)



(d)

$$y = c_6 x^6 + c_5 x^5 \ldots + c_1 x + c_0$$

$$ax + b + c\sin(x)$$

Evaluating Hypotheses

- Accuracy: fits the data
- Generalization: predicts outputs for unseen inputs
- Simplicity and "searchability"

Learning

- Unsupervised -> Supervised
  - Supervised: learn from examples
- Decision Tree Learning
- Hypothesis Space
- Evaluating Hypotheses
  - Accuracy, Generalization, Simplicity

For Next Time:

AIMA 18.6