Decision Trees

Learning from Examples

Asmaa Elbadrawy
PhD, Lecturer
IFT Program, ASU

Arizona State University

Decision Tree is a Supervised Learning algorithm that learns a function f() that maps input values x to a class label y.

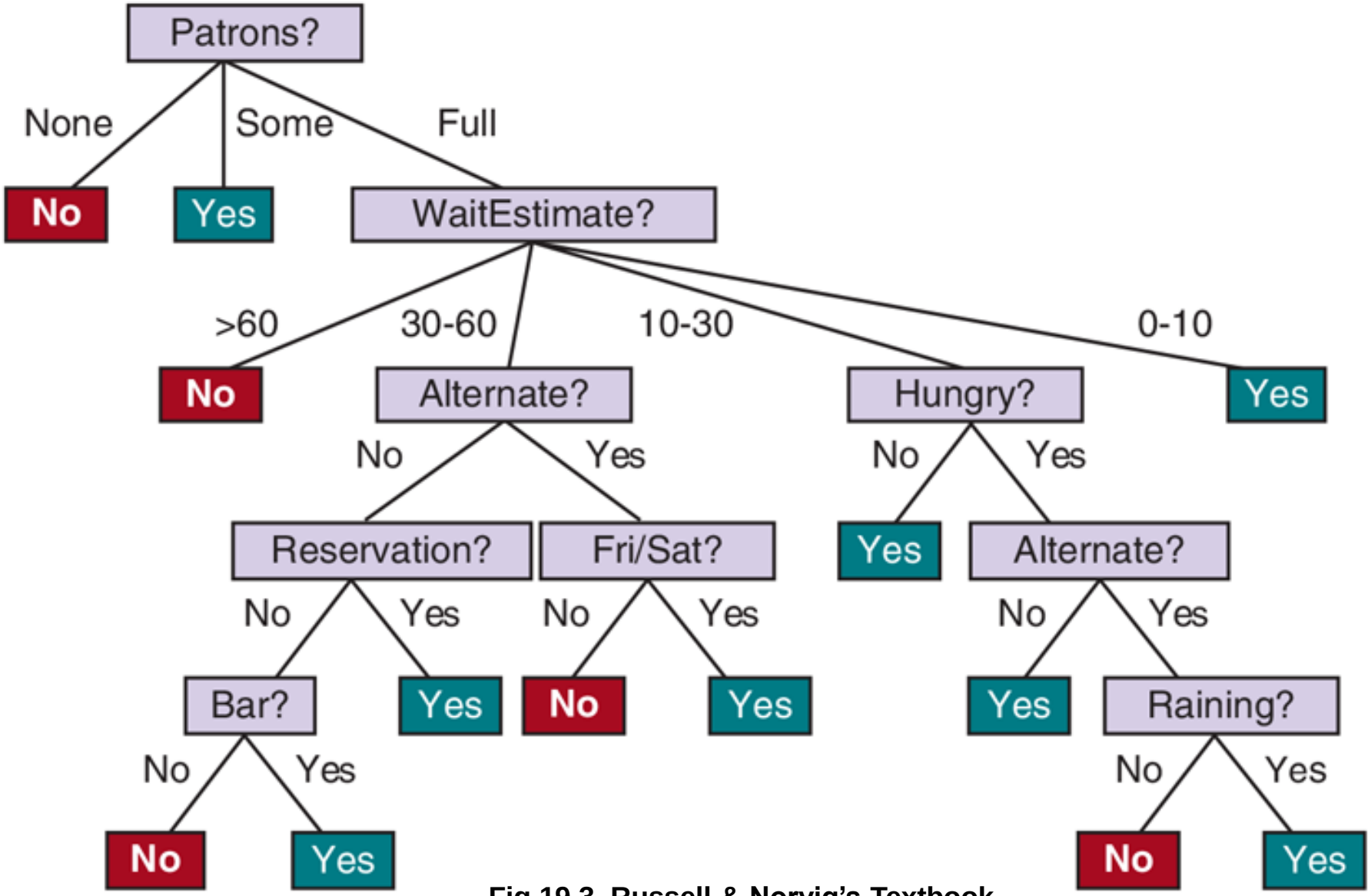


Fig 19.3, Russell & Norvig's Textbook

The function f() is a logical function that carries if/else tests over the attribute values x.

Once the DT model is learned based on the available (training) data, it can be applied to new, unlabeled data to assign it a class label.

DT model finds which attribute values are associated with which class labels.

Repetitively split the data points into groups based on attribute values until pure groups are obtained.

A pure group is one that all (or most) of its data points belong to the same group.

Design Issues

- How to select an attribute to split the data records?

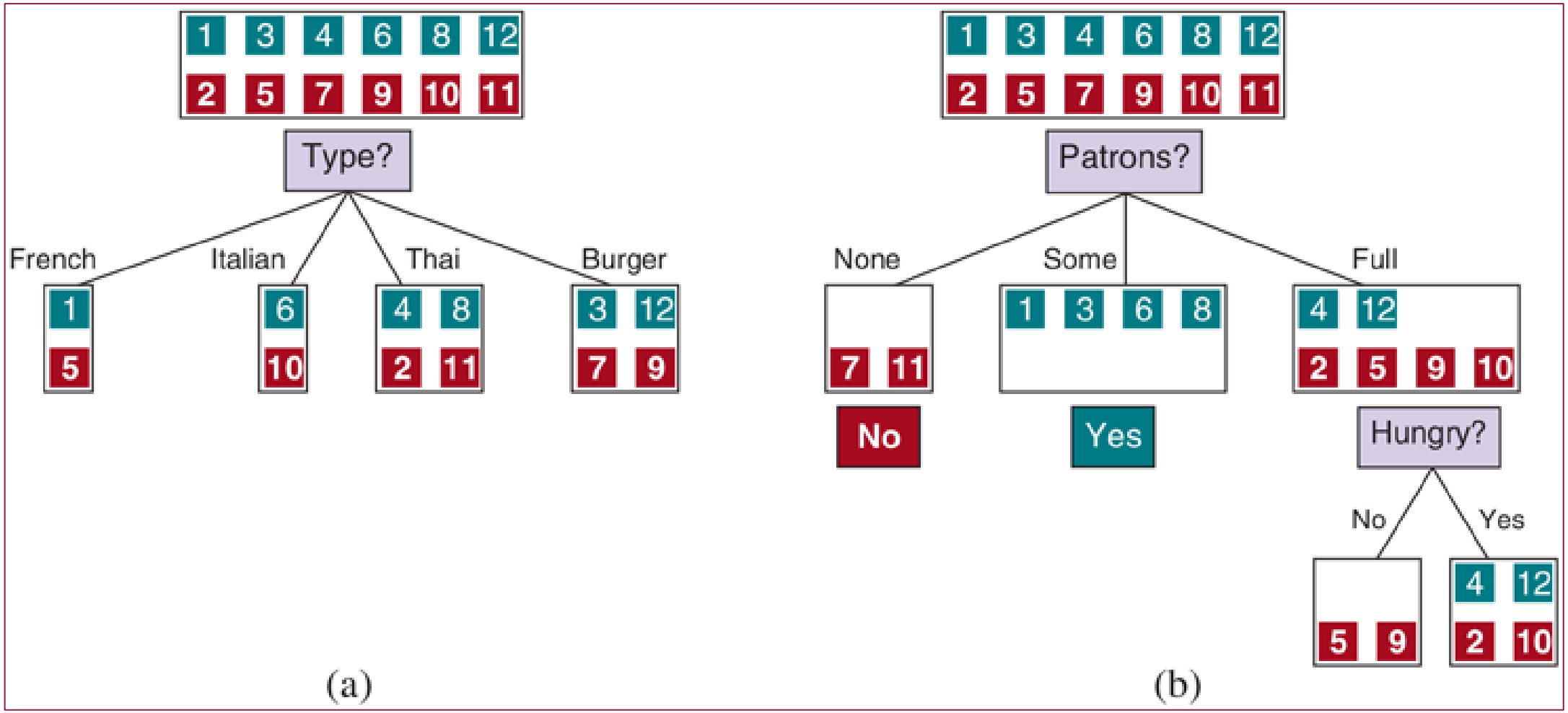


Fig 19.4, Russell & Norvig's Textbook

Select the attribute that gives the purest split!

Measures of Node Impurity

Entropy of a node (t)

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

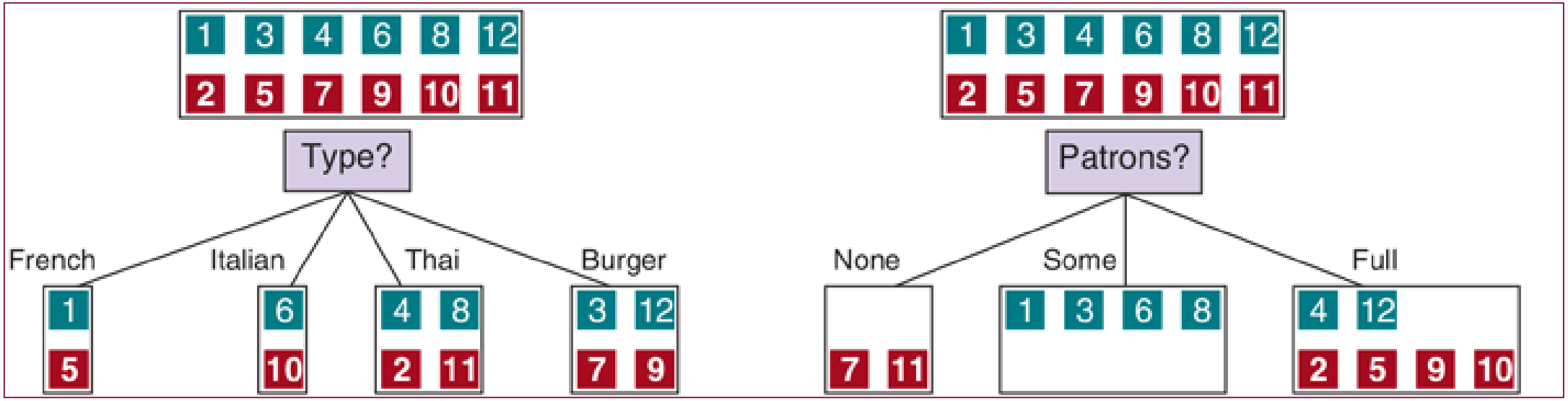
p(j | t) = probability of class j in node t

Max entropy = 1 ➔ Means all classes equally appear at this node ➔ Impure node
Min entropy = 0 ➔ Means only one class appears in this node ➔ Pure node

Any value between 0 & 1 implies some level of impurity.

The lower the entropy, the purer the node.

Measures of Split Impurity



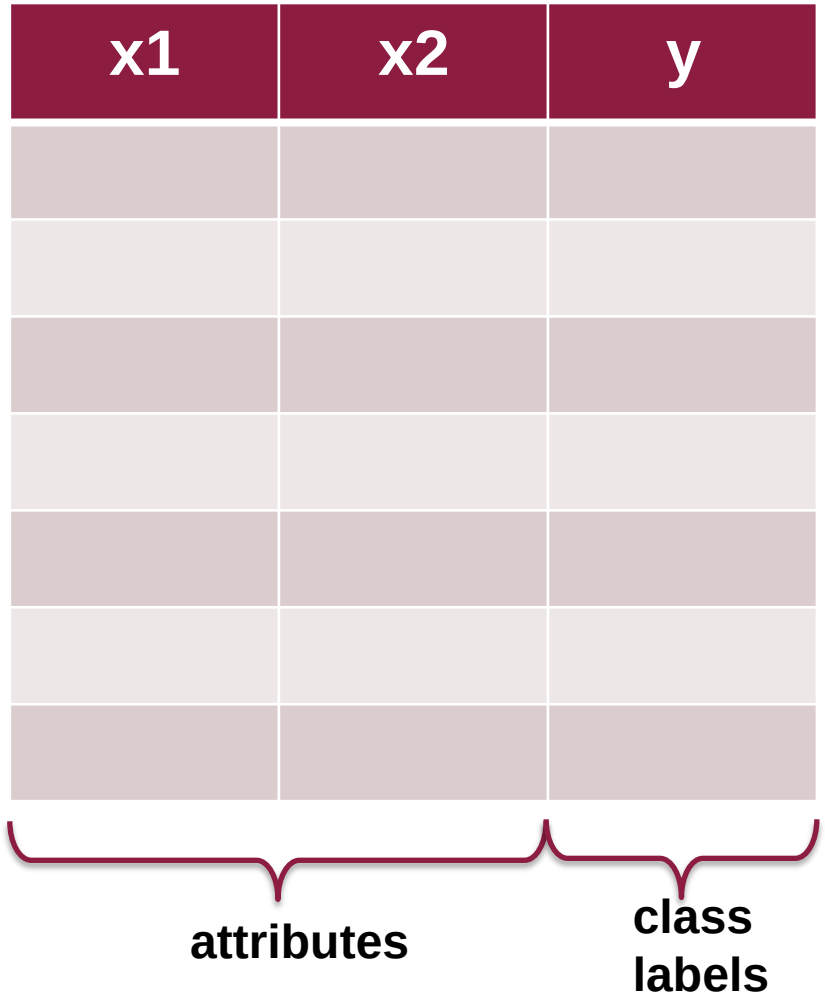
Splitting based on Type gave us 4 nodes!

- Calculate the Entropy for each node
- Take a weighted average to get the split impurity.

Same for Splitting based on Patrons.

DT Full Algorithm

1. Compute split entropy for all attributes.
2. Split the data based on the attribute with the lowest split entropy.
3. For any impure node, repeat steps 1 & 2 until no more impure nodes are there, or no more attributes to split based on.



The Restaurant Waiting Problem

Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*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*	$y_{12} = Yes$

Fig 19.2, Russell & Norvig's Textbook

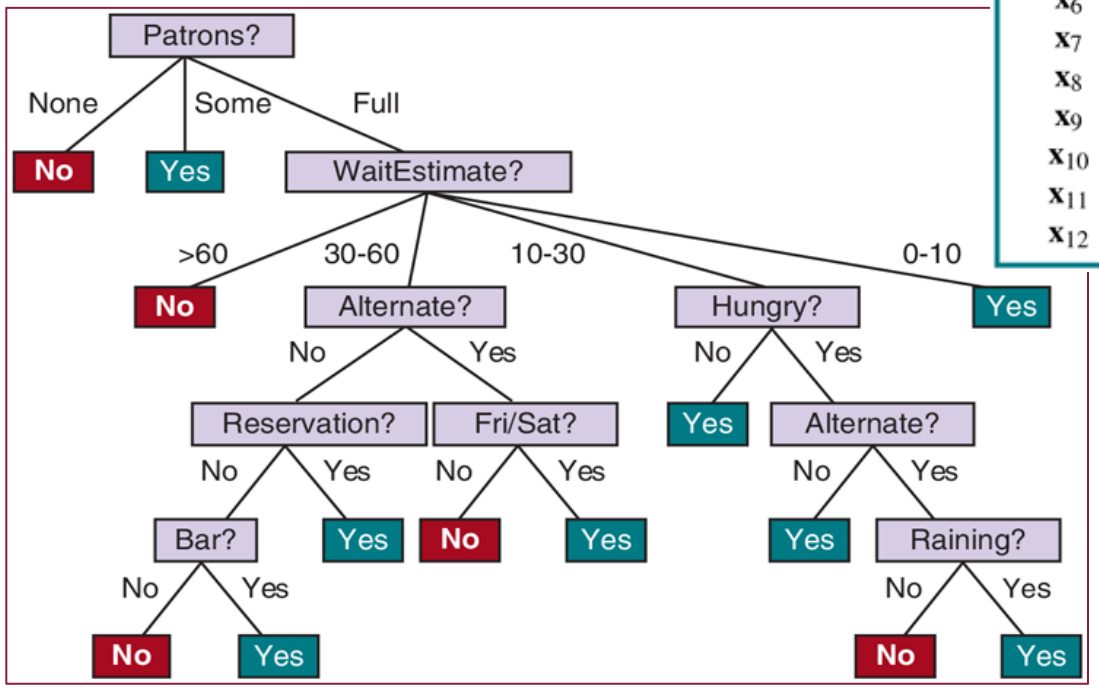


Fig 19.3, Russell & Norvig's Textbook