# Machine Learning: Decision Trees

by Elena Battini Sönmez
İstanbul Bilgi University

- based on: '*Artificial Intelligence, a Modern Approach*', by Stuart Russell and Peter Norvig, Prentice Hall.

# Decision Trees:

1. Def: Decision Trees: vector of attribute values → decision
2. A decision tree reaches its decision by performing a sequence of tests
3. Each internal node of the tree corresponds to a test on the values of one of the input attributes.
4. All branches from a node are labeled with the possible values of the attribute
5. Each leaf node is a decision
6. Boolean classification: positive and negative samples

by Elena Battini Sönmez
Istanbul Bilgi University

# Example: wait at the restaurant?

Let us build a decision tree to decide whether the agent has to wait for a table at a restaurant:

Attributes:

- *Alternate*: whether there is a suitable alternative restaurant nearby
- *Bar*: whether the restaurant has a comfortable bar area to wait in
- *Fri/Sat*: true on Friday and Saturday
- *Hungry*: whether we are hungry
- *Patrons*: how many people are in the restaurant (values: None, Some, Full)
- *Price*: the restaurant price range ($, $$, $$$)
- *Raining*: whether it is raining outside
- *Reservation*: whether we made a reservation
- *Type*: the kind of restaurant (French, Italian, Tai, Burger)
- *Wait Estimate*: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60)

# Example1: Training Set

| Example | Input Attributes | | | | | | | | | | Goal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *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* | $y_{12} = Yes$ |

**Figure 18.3** Examples for the restaurant domain.

# Example:
# Decision Tree (1st solution)

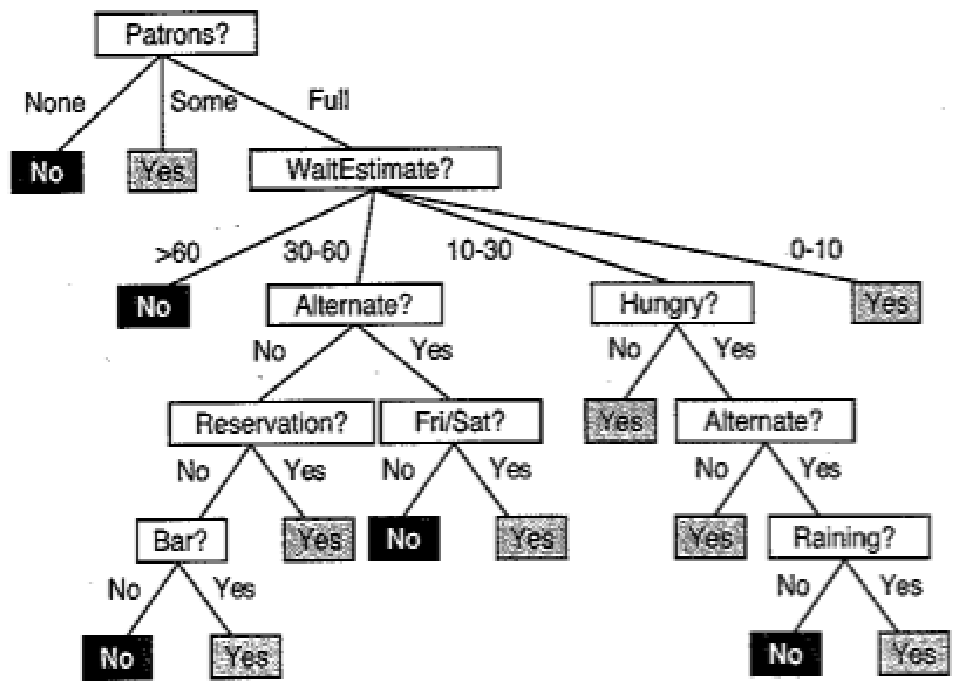


**Figure 18.2** A decision tree for deciding whether to wait for a table.

by Elena Battini Sönmez
Istanbul Bilgi University

# Example: Important attributes

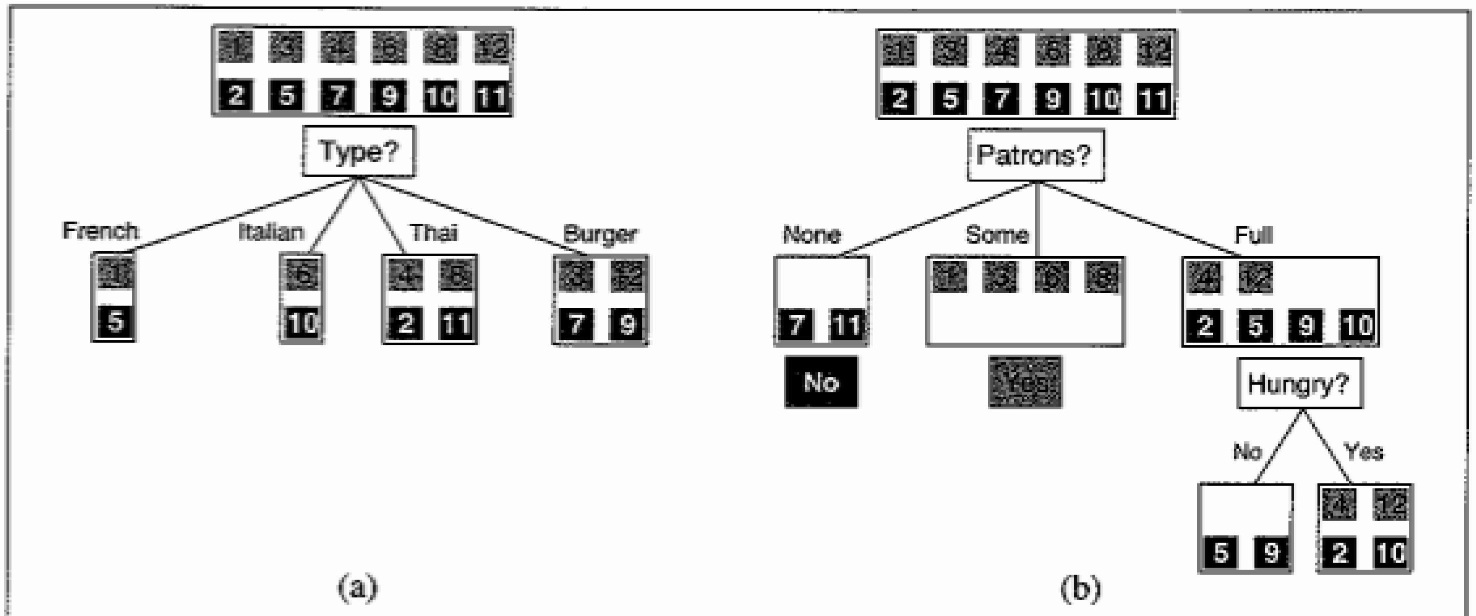


**Figure 18.4** Splitting the examples by testing on attributes. At each node we show the positive (light boxes) and negative (dark boxes) examples remaining. (a) Splitting on *Type* brings us no nearer to distinguishing between positive and negative examples. (b) Splitting on *Patrons* does a good job of separating positive and negative examples. After splitting on *Patrons*, *Hungry* is a fairly good second test.

by Elena Battini Sönmez
Istanbul Bilgi University

# Measure of Impurity: GINI

- Gini index for a given node t:

$$GINI(t) = 1 - \sum_{j} p(j \backslash t)^{2}$$

where p(.) is the relative frequency of class j at node t

- Maximum value of Gini index: non-homogeneous node, high degree of impurity, when records are equally distributed among all classes
- Minimum value of Gini index (0): homogeneous node, low degree of impurity, when all records belong to one class

by Elena Battini Sönmez
Istanbul Bilgi University

# Examples:

$$GINI(t) = 1 - \sum_j p(j \backslash t)^2$$

| 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-P(C1)$^2$-P(C2)$^2$=1-(1/6)$^2$ - (5/6)$^2$=0.278

| C1 | 2 |
|---|---|
| C2 | 4 |

P(C1)=2/6 P(C2)=4/6
Gini=1-P(C1)$^2$-P(C2)$^2$=1-(2/6)$^2$ - (4/6)$^2$=0.444

# Splitting based on Gini index:

When a node t 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 t

The best splitting attribute is the one with minimum $Gini_{split}$

by Elena Battini Sönmez
Istanbul Bilgi University

# Example of calculation:

- Which is the better attribute for the root node of the decision tree of the given example? Let us restrict our interest into the “Type”  and “patrons” attributes?
- Which is the $\text{Gini}_{split}$(type)?
- Which is the $\text{Gini}_{split}$(patrons)?

by Elena Battini Sönmez
Istanbul Bilgi University

# Alternative Splitting Criteria based on Info:

Entropy at a given node t:

$$Entropy(t) = -\sum_{j} p(j\backslash t)\log p\,(j\backslash t)$$

where p(.) is the relative frequency of class j at node t

- It measures the homogeneity of a node
- Maximum value: when records are equally distributed among all classes implying least information
- Minimum value: when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

# Splitting based on info:

Information Gain:

$$Gain_{split} = Entropy(parent\_node) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i)\right)$$

where the parent node is split into k partitions, and $n_i$ is number of records in partition i

- Choose the attribute that maximizes the GAIN
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure

by Elena Battini Sönmez
Istanbul Bilgi University

# Exercises:

- Considering the training set of example1, use the Gini index to build the best decision tree
- Considering the training set of the example1, use the entropy to build the best decision tree

by Elena Battini Sönmez
Istanbul Bilgi University