Decision Tree

Training / Learning from examples.

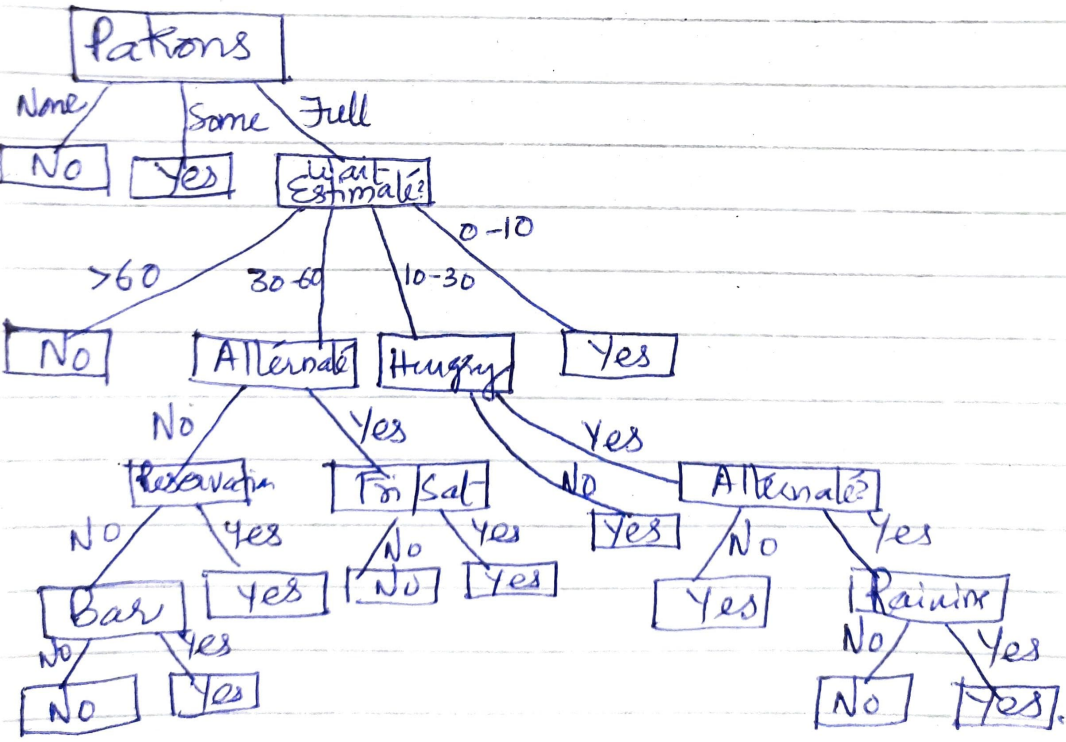
Example problem :. Restaurant waiting problem.

A decision tree is a representation of a function that maps a vector of attribute values to a single output value called a "decision". A DT reaches its decision by performing a sequence of tests, starting at the root and following the appropriate branch until a leaf node is found. Each internal node in the tree corresponds to a test of the value of one of the input attributes, the branches from the node are labelled with the possible values of the attribute, and the leaf nodes

Example	Input Attributes										O/P
	Alt	Bar	Fri	Hungry	Patrons	Price	Rain	Res	Type	Est	
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	y_1 Yes
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	y_2
x_3	No	Yes	No	No	Some	$	No	No	Burger	0-10	y_3
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10-30	y_4
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	y_5
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	y_6
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	y_7
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	y_8
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	y_9
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	y_{10} No
x_{11}	No	No	No	No	None	$	No	No	Thai	0-10	y_{11} No
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	y_{12} Yes

specify what value is to be returned by the function. Input and output values can be discrete or continuous; and a decision tree having discrete i/o pairs is called a Boolean DT. Here the i/p is discrete and the output is True or False.

A sample DT to decide whether we wait for a table or not.



Learning decision trees from examples.

* Adopts a greedy divide-and-conquer strategy to find a DT consistent with the training examples which is as small as possible.
* Done by choosing the most important attribute and then recursively solve the smaller sub-problems that are defined by the possible results of the test.
* To measure the importance of an attribute, a DT uses "Information Gain" as a measure, which is defined in terms of entropy.
* Entropy is a fundamental quantity in Information Theory coined by Shannon and Weaver in 1949;

* Entropy is a measure of uncertainty of a random variable.
* The entropy of a random variable V with possible values v_k, having probability $P(v_k)$ is defined as:-

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)}$$

$$= -\sum_k P(v_k) \log_2 P(v_k)$$

eg:- Entropy of a fair coin flip:-

$$H(\text{fair coin}) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5)$$

$$= \underline{\underline{1 \text{ bit}}}$$

* Entropy of a Boolean random variable that is true with prob. q

$$B(q) = -(q \log_2 q + (1-q) \log_2 (1-q))$$

* Apply these concepts in DT problem, with a training set having p positive examples and n negative examples. Then the output variable on the whole training set has an entropy H(output):

$$H(\text{output}) = B\left(\frac{p}{p+n}\right)$$

* Restaurant Waiting problem

$$p = n = 6 \Rightarrow H(\text{output}) = B\left(\frac{6}{6+6}\right) = B(0.5)$$

$$\therefore B(0.5) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5)$$

$$= \underline{1 \text{ bit}}$$

* The result of a test on an attribute A will give some information and thereby reducing the overall entropy by some amount.
* Measure this reduction by looking at the entropy before and after the attribute test.

* An attribute A with d distinct values divides the training set E into subsets $E_1, \ldots E_d$.

* Each E_k subset has p_k positive and n_k negative values.

* If we go along that branch, we have $B(p_k / p_{k+n_k})$ bits of info to answer the question.

* Entropy after testing A

$$\text{Remainder}(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right)$$

∴ Info gain from the attribute test on A is the expected reduction in entropy

$$\text{Gain}(A) = B\left(\frac{p}{p+n}\right) - \text{Remainder}(A)$$

Gain(Patrons)

$$= 1 - \left[\frac{2}{12} B\left(\frac{0}{2}\right) + \frac{4}{12} B\left(\frac{4}{4}\right) + \frac{6}{12} B\left(\frac{2}{6}\right)\right]$$

$$= \underline{0.541 \text{ bits}}$$

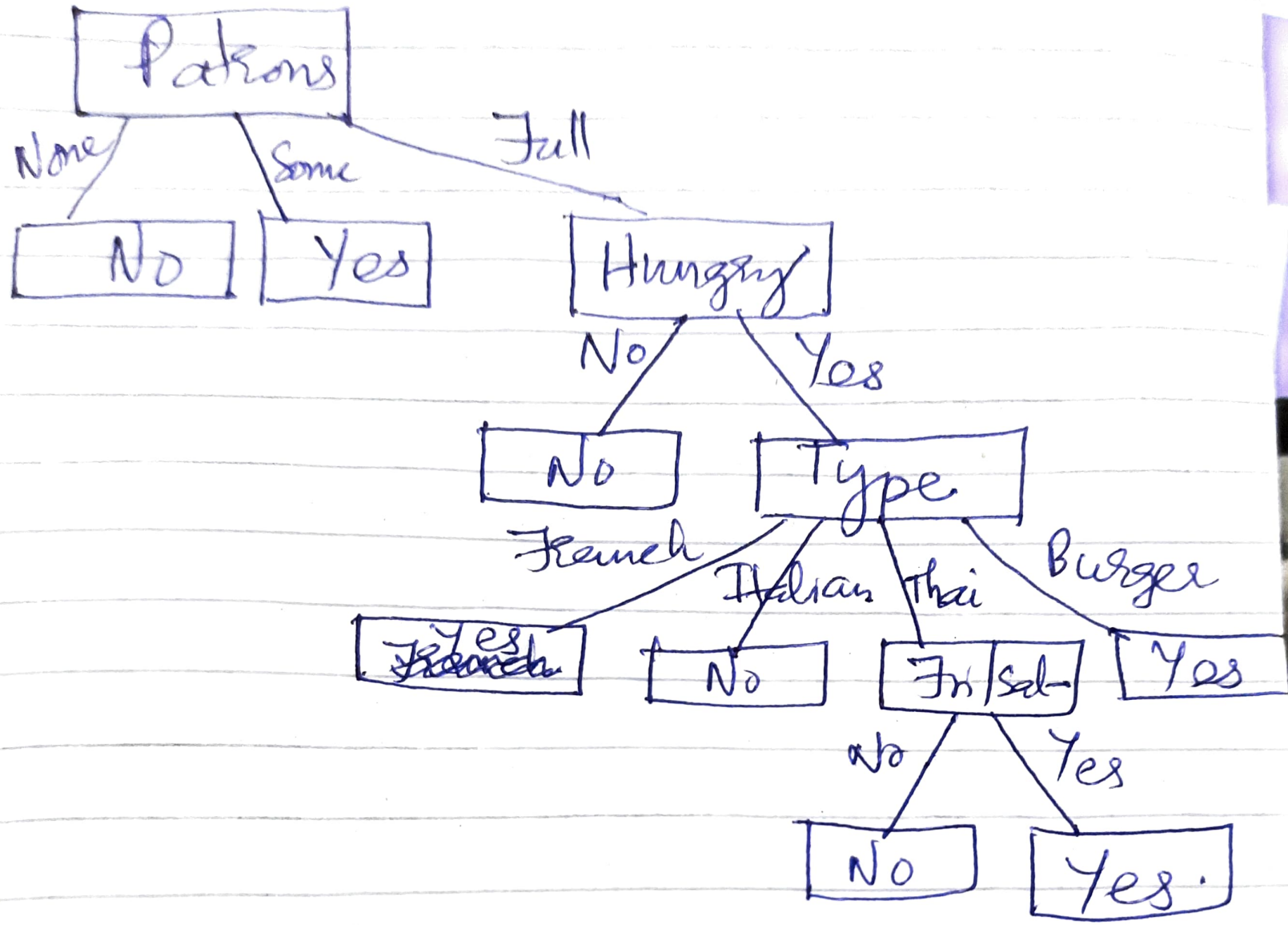
$$\text{Gain (Type)} = 1 - \left[\frac{2}{12} B\left(\frac{1}{2}\right) + \frac{2}{12} B\left(\frac{1}{2}\right) + \frac{4}{12} B\left(\frac{2}{4}\right) + \frac{4}{12} B\left(\frac{2}{4}\right)\right]$$

$$= \underline{0 \text{ bits}}$$

Hence "Patron" is more important attribute than "Type".

Also among all the available attributes, Patron has the max. info gain and is chosen as the root.

Final Tree


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