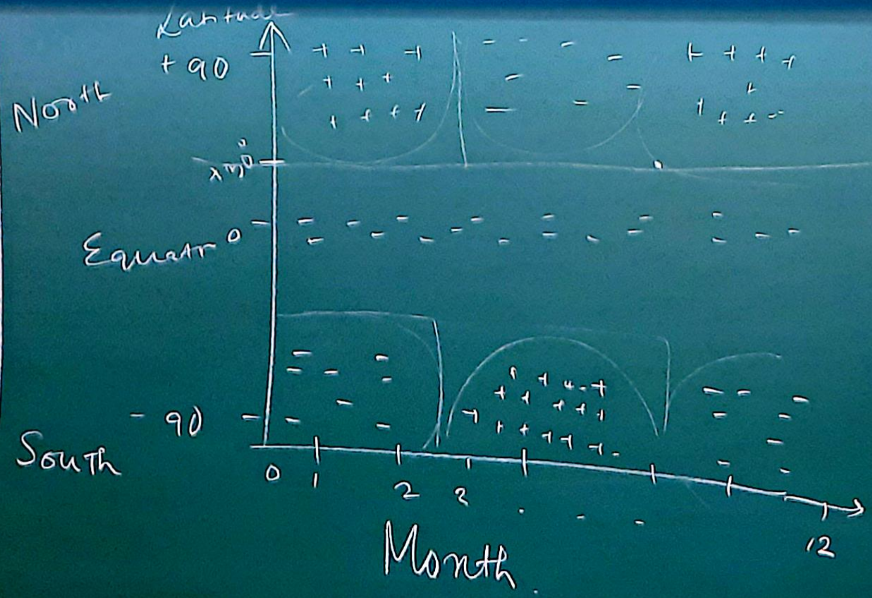
Decision Trees

1. Non. linear Model

Ch - 3 Tom Mitchell

Binary clas

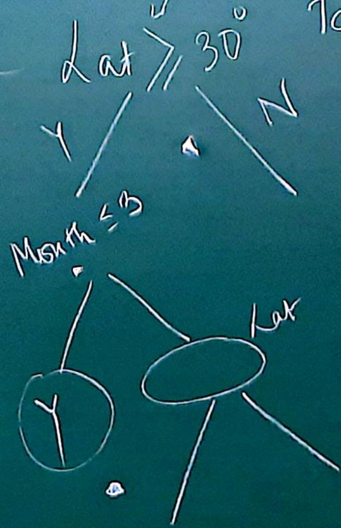


Greedy Method

Top-down,

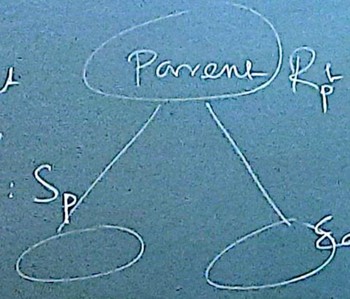
Recursive

Partitioning



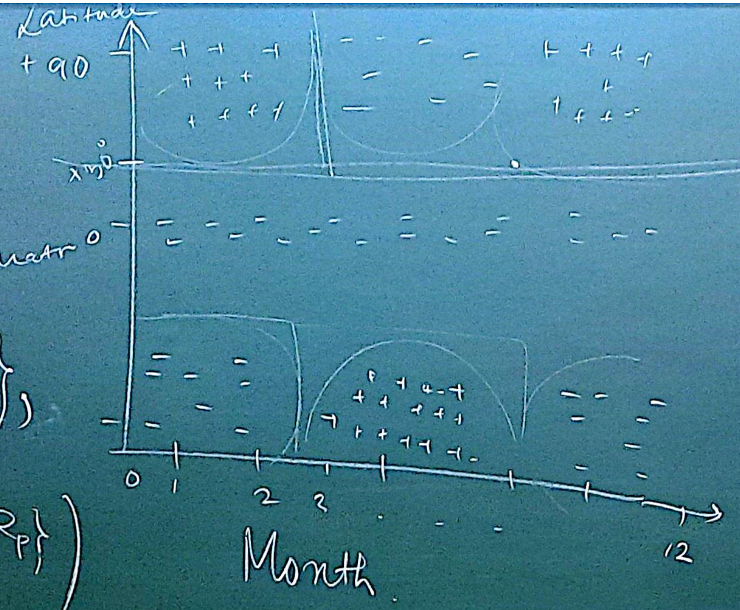
Region of Parent : R_p

Looking for a split : S_p



$S_p(i, t) = \left(\{x \mid x_i < t;\ x \in R_p\},\ \{x \mid x_i \geq t;\ x \in R_p\} \right)$

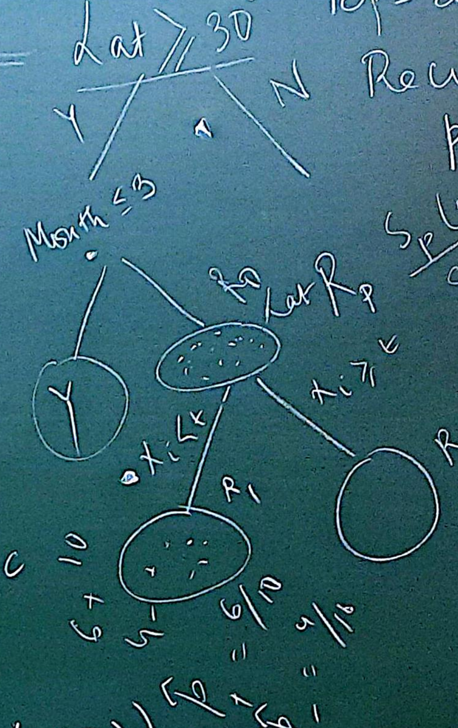
i-th feature   threshold



Greedy Method

Top-down,

Recursive partitioning



$\frac{S_p(i,t)}{S_p(j,t')}$

Loss $L(R)$ = misclassification loss

Given class c, define $\hat{p}_c$ to be the proportion of examples in R belong to c

$L_{R_1} = 1 - \max_c \hat{p}_c$

Come up with a split s.t.

$\max \{ L(R_p) - (L(R_1) + L(R_2)) \}$

S_1

1000

900+
100−

R_1 R_2

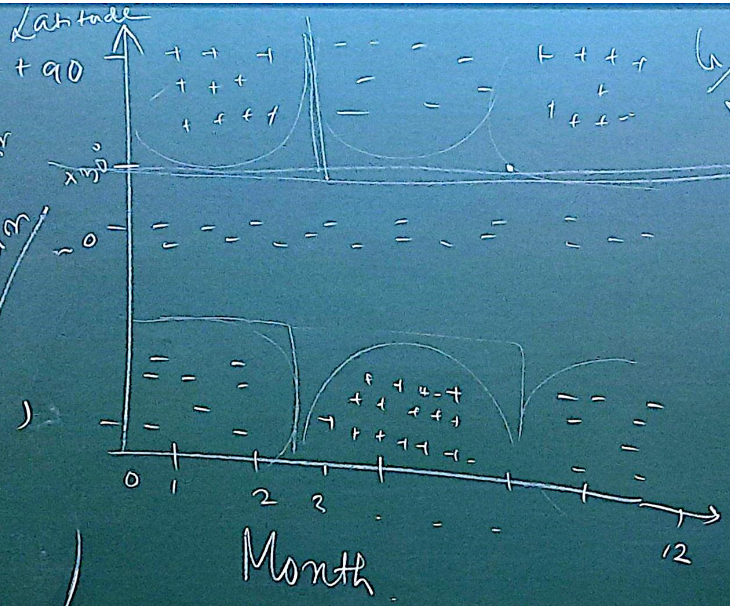
R_1	R_2
700+	200+
100−	0−
100	0

S_2

1000

900+	500+
100−	0−
100	0

Information Gain



Greedy Method
Top-down,
Recursive
Partitioning

Lat $\geq 30^\circ$

Y N

Month ≤ 3

Y

Lat R_p

$x_i < t$ $\quad$ $x_i > t$

R_1 $\quad$ R_2

$$\frac{S_p(j,t)}{S_p(j,t')}$$

Loss $L(R)$ = misclassification loss

Given class c, define $\hat{p}_c$ to be the proportion of examples in R belong to c

$$L_{R_1} = 1 - \max_c \hat{p}_c$$

Come up with a split s.t.

$$\max \{ L(R_p) - (L(R_1) + L(R_2)) \}$$

1 Information Gain

Shannon's Entropy

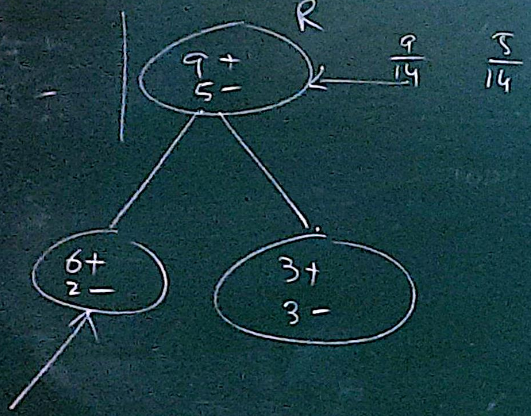
$- p_H \log p_H - p_T \log p_T$ $\quad X = \{ \; i \; \}$

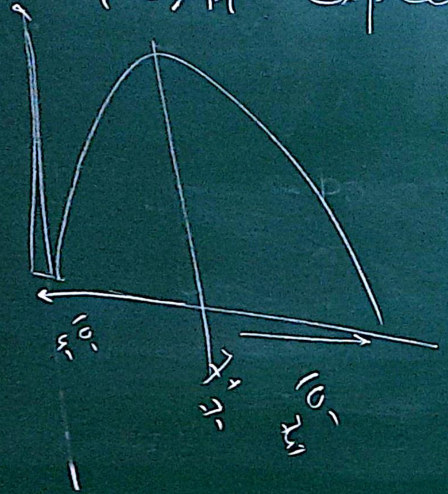
$- \sum p_i \log p_i$

p_i = prob of the occurrence of the outcome i

$\log p_i$ = min. no of bits required to encode the outcome i

$p_i \log p_i$ = expected no of bits req. to encode the outcomes of the event.





1 Information Gain

$-p_H \log p_H - p_T \log p_T$

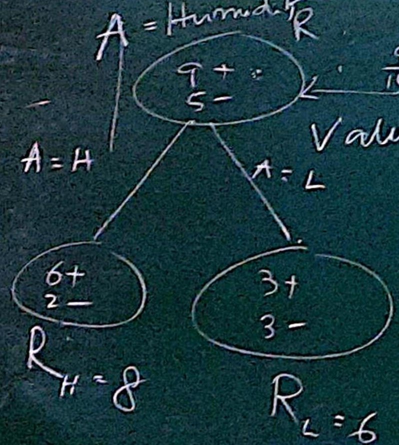
$-\sum p_i \log p_i$

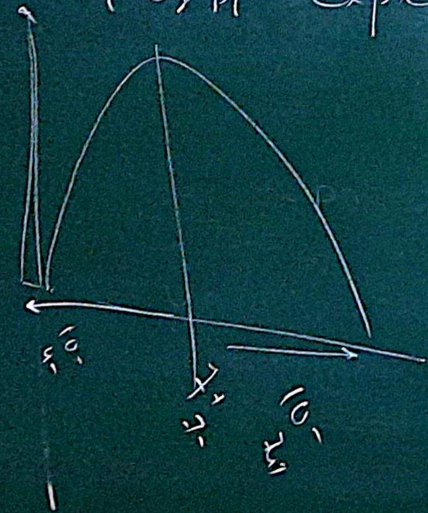
p_i = prob of the occ

$\log p_i$ = min. no of

$p_i \log p_i$ = expected no

Shannon's Entropy



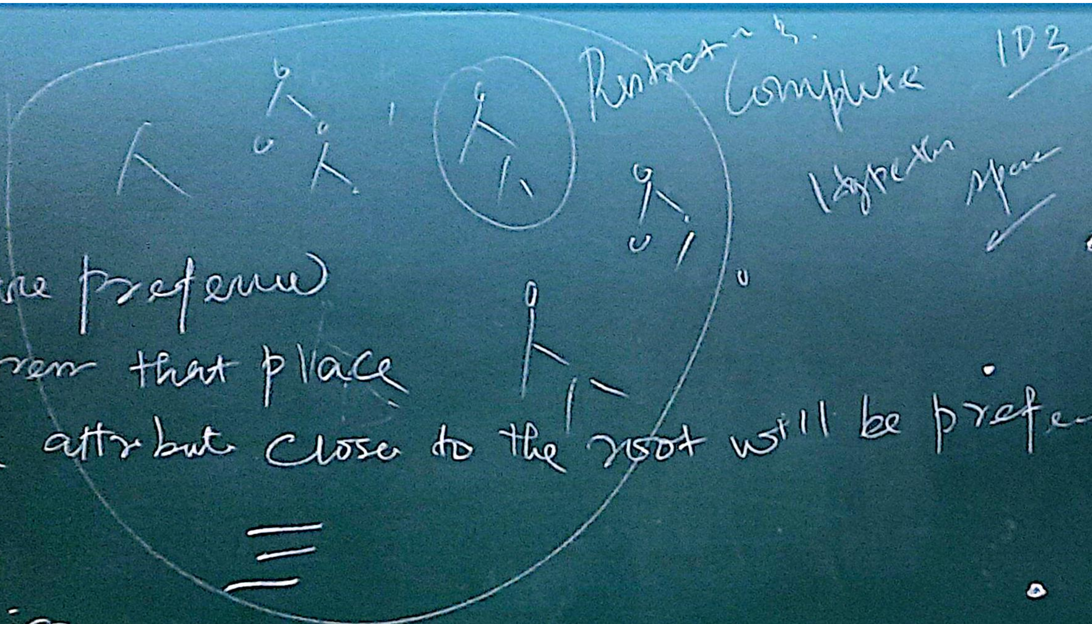


$$Gain(R, A) = \left[Entropy(R) - \sum_{v \in Value(A) = \{H, L\}} \frac{|R_v|}{|R|} Entropy(R_v) \right]$$

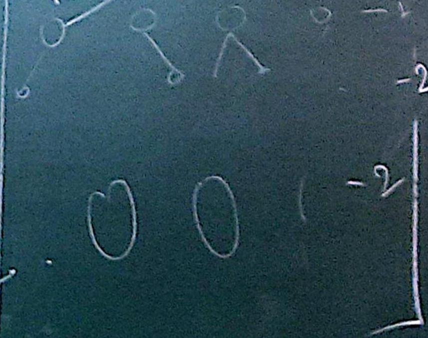
A_1 A_2 .. A_m

Preference Bias

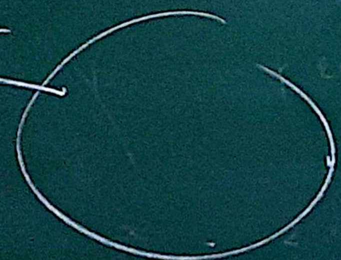
Shorter trees are preferred over longer trees. Trees that place high info Gain attribute close to the root will be preferred.



BFS-Tree



Candidate Elimination



Occam's Razor

Overfitting

$\mathcal{H}$

$h_1 \in \mathcal{H}$

$Loss_{test}(h_1)$

$Loss_{train}(h_1)$

$h_2 \in \mathcal{H}$

$Loss_{train}(h_1) < Loss_{train}(h_2)$

$Loss_{test}(h_1) > Loss_{test}(h_2)$