Decision Trees:

KNN , Naive Bayes , logistic Regress, linear regre, SVM

KNN ↓ (Instance Based method)

Naive Bayes ↓ (probabilistic Method)

logistic Regress, linear regre, SVM → (Geometric, hyperplane)

✧ **Decision Trees:** similar to if... else condition

DT: nested if... else classifier

EDA: Iris dataset
$\hookrightarrow y_i = \{1, 2, 3\}$
(SL, SW, PL, PW)

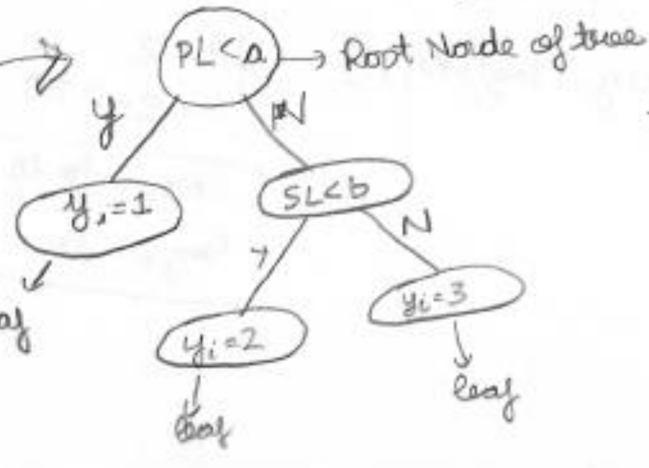
Simple $\left\{ \begin{array}{l} \text{if } PL < 5 \\ \quad \text{then } y_i = 1 \end{array} \right\}$

$x_i < SL, PL, SW, PL >$

model $\left[\begin{array}{l} \text{if } PL < a \\ \quad y_i = 1 \\ \text{else} \\ \quad \left[\begin{array}{l} \text{if } SL < b \\ \quad \text{class}=2 \\ \text{else} \\ \quad \text{class} = 3 \end{array} \right] \end{array} \right]$

nested if... else conditions

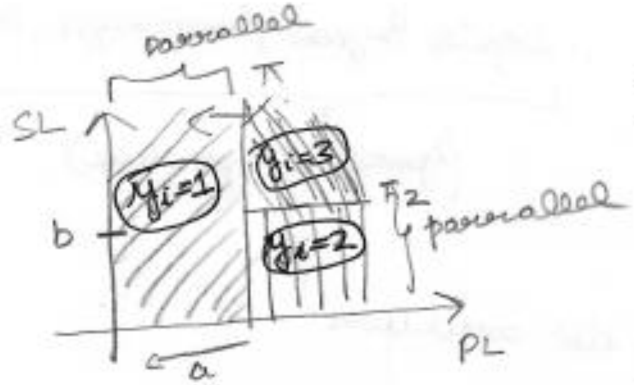
(Tree)

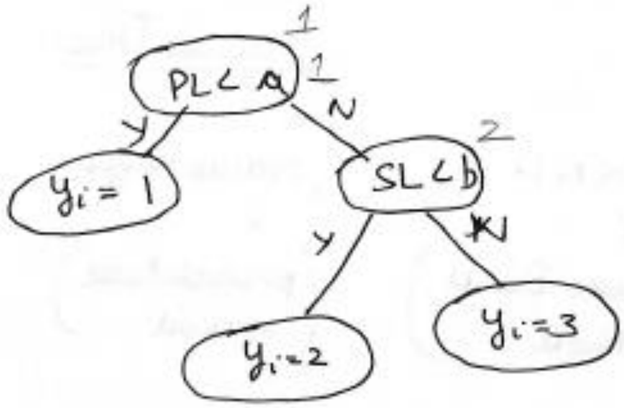


Terminating vertex are called leafs
starting → Root Node
Vertex → node →
leaf nodes → decision node
Internal node → neither root nor leaf.

non leaf node we make decisions.

Geometric Intuition :

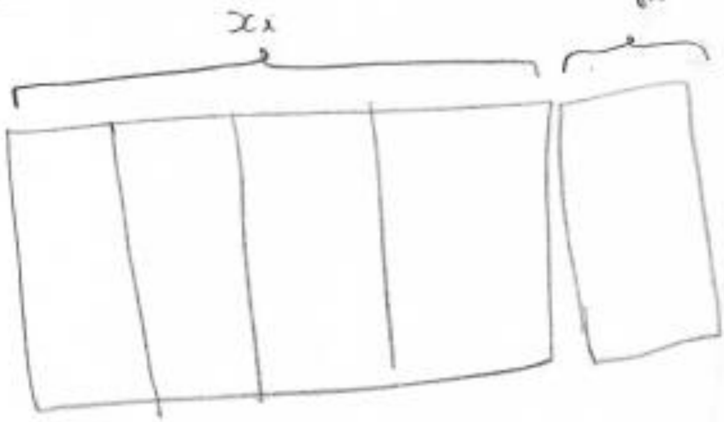




corresponding to decision hyperplane is introduced

NOTE: All of your hyperplanes are axis-parallel in a decision Tree.

Play Tennis Example



Entropy

Given Random variable $Y \rightarrow y_1, y_2, y_3, \dots, y_k$

$$H(Y) = \sum_{i=1}^{K} -P(y_i) \log_b(P(y_i))$$

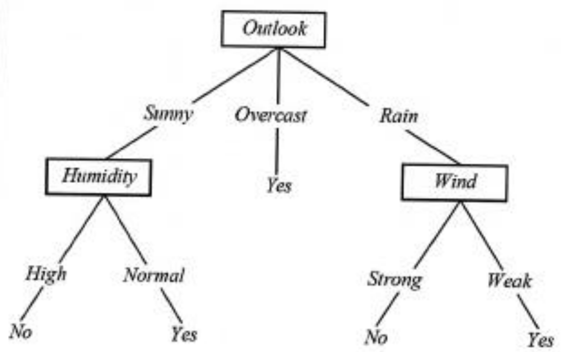
entropy

$\begin{cases} b=2 \\ b=e=2.718 \end{cases}$

$$\boxed{\begin{aligned} \log_2 &= \lg \\ \log_e &= \ln \end{aligned}}$$

Play Tennis Example

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No



$$H(Y) = -\sum_{i=1}^{k} P(y_i) \log (P(y_i))$$
$$P(y_i) = P(Y = y_i)$$

Y: play Tennis

Y_+ , Y_- probability $\begin{cases} P(Y_+) = 9/14 \\ P(Y_-) = 5/14 \end{cases}$

$$H(Y) = -\sum_{i=1}^{k} P(y_i) \log (P(y_i))$$

$$H(Y) = -\underbrace{\boxed{\frac{9}{14}}}_{P(y_+)} \lg\left(\frac{9}{14}\right) - \underbrace{\frac{5}{14}}_{P(y_-)} \lg\left(\frac{5}{14}\right) = 0.94$$

(%age of -ve pts in D)

$\frac{\#\text{+ve pts}}{\text{Total}\#\text{pts}}$ = % age of +ve pts in D

Properties of Entropy $\quad Y \to Y_+\ Y_-$ (2 class, 2 category)

Case 1: $D \to y_+ \to 99\%$, $D \to y_- \to 1\%$

$H(Y) = 0.99 \log 0.99 - 0.01 \log 0.01$
$H(Y) = 0.0801$

Case 2: $D \to y_+ \to 50\%$, $D \to y_- \to 50\%$

$H(Y) = 0.5 \log 0.5 - 0.5 \lg 0.5$
$= 1$

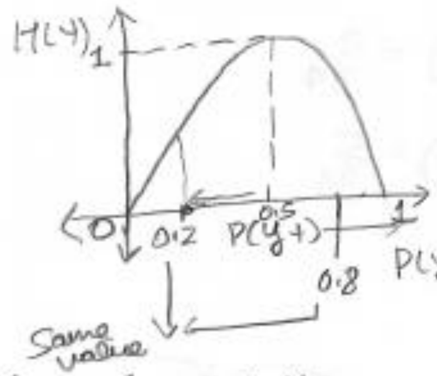
Case 3: $D \to y_+ \to 0\%$, $D \to y_- \to 100\%$

$H(Y) = 0$

NOTE : when both possibility are equally probable that

$H(Y) = 1$ case 2.

One class fully dominates entropy become 0 case 3.



$P(y+) = 1 - P(y)$

$$\begin{cases} 0.2 \leftarrow y_+ \\ 0.8 \leftarrow y_- \end{cases}$$

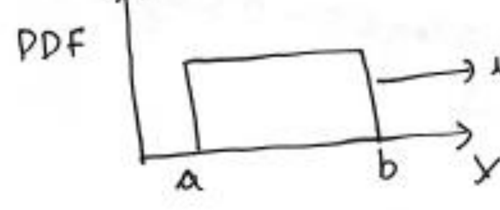
$$\left.\begin{aligned} &= -0.2 \lg 0.2 - 0.8 \lg 0.8 \\ &= -0.8 \lg 0.8 - 0.2 \lg 0.2 \end{aligned}\right\} \text{both are same}$$

$Y \rightarrow \underbrace{y_1, y_2, \dots, y_k}$

equi probable → entropy is maximum

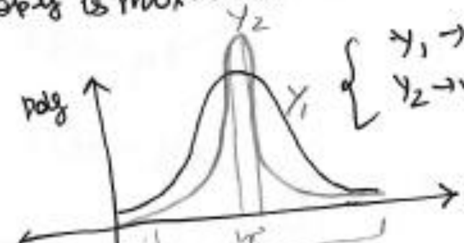
$$\left.\begin{aligned} &y_1 \rightarrow \text{most probable} \\ &y_2, y_3 \dots \rightarrow 0 \end{aligned}\right\} \text{entropy is minimum}$$

PDF | Histogram based



→ uniform distribution

entropy is maximum = 1

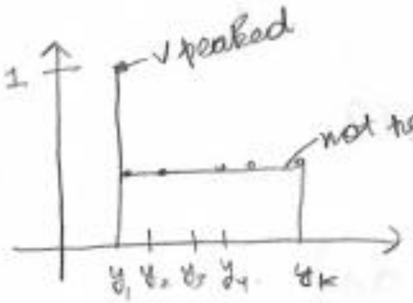


$$\begin{cases} Y_1 \rightarrow \text{less peaked} \\ Y_2 \rightarrow \text{very peaked} \end{cases}$$

$H(Y_2) < H(Y_1)$

more peaked the distribution In case of Y_2 spread is less

On the other hand Y_1 has big spread.



NOTE → The more peaked a distribution is less is its entropy.

⇓

If only one value dominates entropy is less

but if points are spread than entropy is maximum

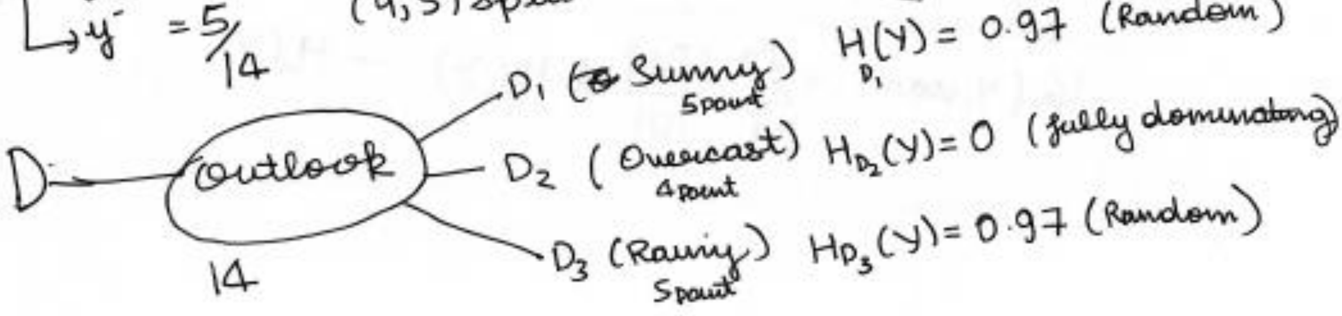
☆ Information Gain

$H(y) = 0.94$

$y \to y^+ = 9/14$

$\quad \to y^- = 5/14$ (9,5) split

$$H_{D_1} = -\frac{3}{5} * \log\left(\frac{3}{5}\right) - \frac{2}{5} * \log\left(\frac{2}{5}\right) = 0.97$$



Information Gain $H_D(y) = 0.94$ → before breaking dataset

$H_{D_1}(y) = 0.97$, $H_{D_2}(y) = 0$, $H_{D_3}(y) = 0.97$

$$IG(Y, outlook) = \left[\left(\frac{5}{14} \times 0.97\right) + \left(\frac{4}{14} * 0\right) + \left(\frac{5}{14} * 0.97\right)\right] - 0.94$$

$$\Downarrow$$

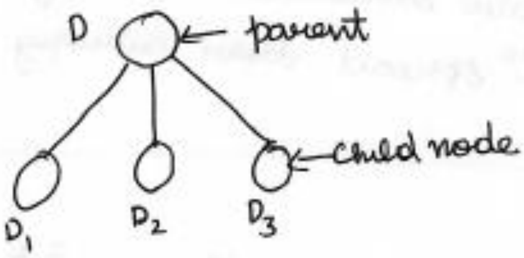
Weighted entropy after D_1, D_2, D_3 $\leftarrow$ $\left(\frac{5}{14} \times 0.97 + \frac{4}{14} * 0 + \frac{5}{14} * 0.97\right)$

$$\downarrow$$

$$= \left(\frac{5}{14} \times 0.97 \times 2\right)$$

$$= \frac{5}{7} \times 0.97 = 0.69$$

$$\boxed{IG(Y, outlook) = \left[\frac{5}{7} \times 0.97 - 0.94\right]}$$



$$H_D(Y) = [H_{D_1}(Y)] * \frac{|D_1|}{|D|} + \cancel{\#\#} [H_{D_2}(Y)] * \frac{|D_2|}{|D|} + [H_{D_3}(Y)] * \frac{|D_3|}{|D|}$$

$Y \xrightarrow{var} y_1, y_2, \ldots, y_k$

$$\boxed{IG(Y, var) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} * H_{D_i}(Y) - H_D(Y)}$$

Gini Impurity ~ similar to Entropy

$$I_G(y) \neq IG(y)$$

Gini Impurity ↓ Information Gain ↓

$$I_G(y) = 1 - \sum_{i=1}^{K} (P(y_i))^2$$

$$Y \to y_+, \; y_-$$

$$Y \to y_1, y_2, y_3, \ldots\ldots, y_K$$

Case I :

$$P(y+) = 0.5$$
$$P(y-) = 0.5$$

$$I_G(Y) = 1 - \sum_{i=1}^{} \left[(0.5)^2 + (0.5)^2\right]$$
$$= 1 - (0.25 + 0.25)$$

Gini Entropy → $I_G(Y) = 0.5$

Entropy → $H(Y) = 1$

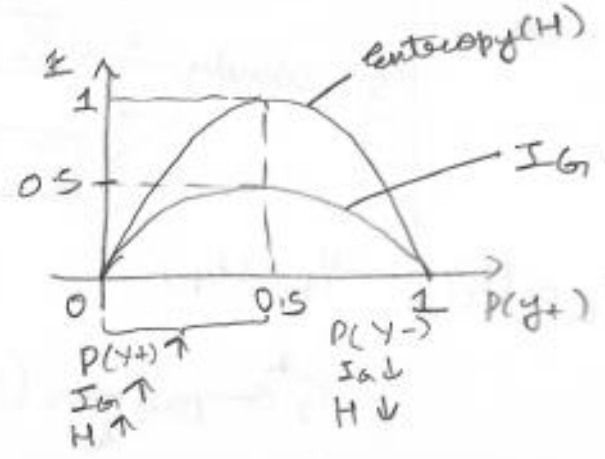
Case II :

$$P(y+) = 1$$
$$P(y-) = 0$$

Gini Entropy → $I_G(y) = 1 - (1^2 + 0^2) = 0$

Entropy → $H(Y) = 0$

2-Category case :- y_+, y_- $\quad P(y_+) = 1 - P(y_-)$



$I_G(y)$ (Gini Entropy)

$1 - \{ (P(y+)^2 + (P(y-))^2 \}$

↓

No log

more computanously efficent to compute Gini Impurity

$H(y)$ (Entropy)

$-P(y+) \log P(y+) - P(y-) \log(P(y-))$

↓

log

has log ~~tres~~ more time complexity

> NOTE: Gini Entropy is used over Entropy due to ~~its~~ computational efficiency

BUILDING A DECISION TREE

We start with Top to down.

① $D \rightarrow 9+, 5-$

$H_0(y) = 0.94$

② Outlook

- S → D_1: [2+, 3−] $H_{D_1}(y)$
- O → D_2: [4+, 0−] $H_{D_2}(y)$
- R → D_3: [2+, 3−] $H_{D_3}(y)$

$H_{D_1}(y) = 0.97$

$H_{D_2}(y) = 0$

$H_{D_3}(y) = 0.97$

Weighted Entropy $= 0.69$

Information Gain $= 0.94 - 0.69$

$IG(y, \text{Outlook}) = 0.25$

③ Temperature

- H → D_1: [2+, 2−] $H_{D_1}(y)$
- m → D_2: [4+, 2−] $H_{D_2}(y)$
- C → D_3: [3+, 1−] $H_{D_3}(y)$

Weighted entropy =

$\frac{4}{14}$

④ Humidity

- H: [3+, 3−]
- N: [6+, 1−]

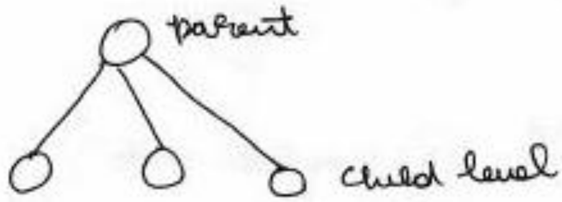
⑤ Windy

- F: [6+, 2−]
- T: [3+, 3−]

$D_g(S, t) = g(t) - P_L\, g(t_L) - P_R\, g(t_R)$

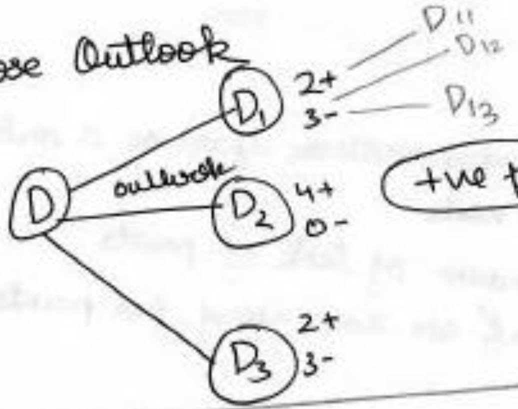
$S^* \leftarrow \arg\max (D_g(S, t))$

IG (y, f) = entropy @ parent-level
-weighted entropy @ child level



$$IG(y, f) = H_D(y) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} * H_{D_i}(y)$$

Choose Outlook



+ve pts → Such a node is called pure node

D → Tree $\quad IG(y, f) = H_D(y) - \sum_{i=1}^{k} \underbrace{\frac{|D_i|}{|D|}} * \underbrace{H_{D_i}(y)}_{\text{Entropy}}$

IG (Y, outlook) = ~~0.69~~ 0.25

IG (Y, Temp) = —

IG (Y, Humidity) =

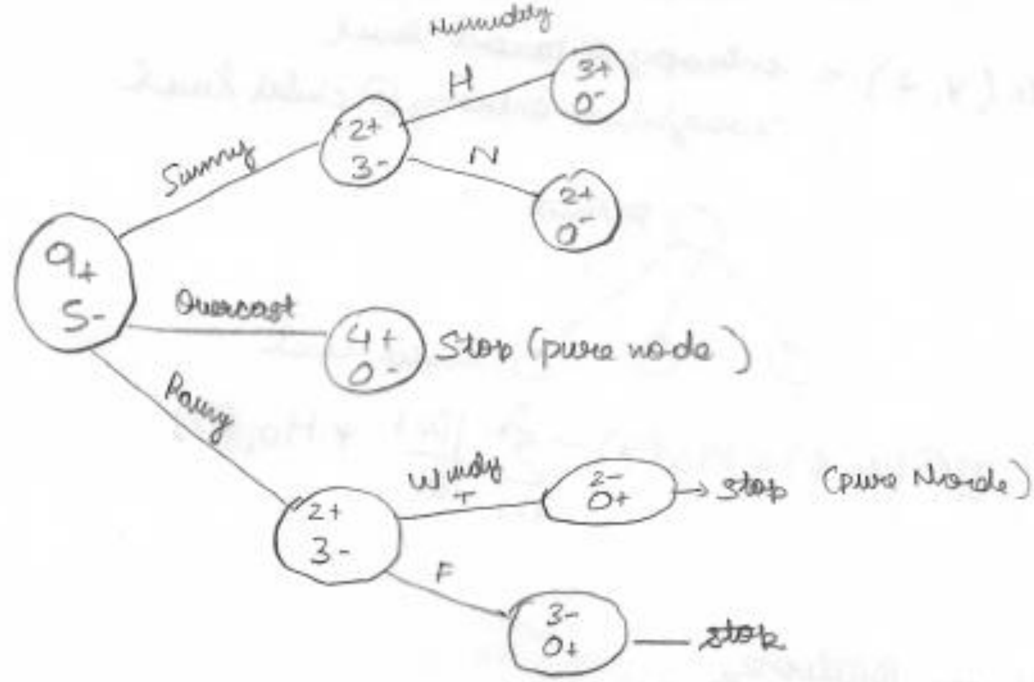
IG (Y, windy) =

NOTE: Pick the one which has most /maximum information gain

IG (Y, f - feature) = (entropy @ parent-level) - (weighted entropy @ child level)

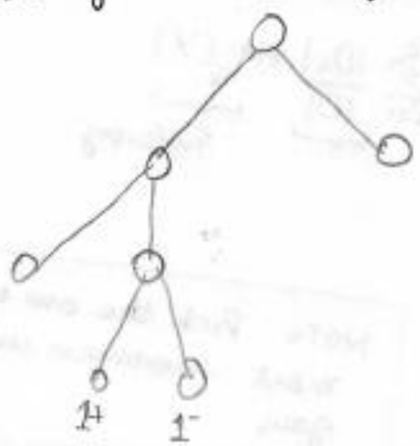
D

D.

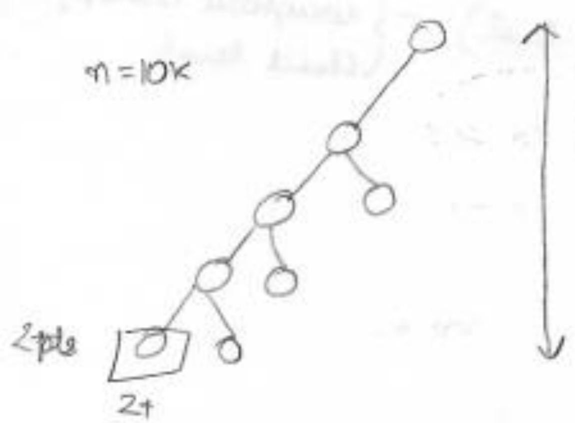


Recursively breaking each node using Information gain as a criteria

1. pure node → stop growing the node
2. can't grow the tree anymore because of lack of points
   we wont grow a node further if we have very few points



3. if we are too deep we stop growing tree.



☆ Depth of the tree ↑; overfitting ↑ (few pts)

Depth of tree is small; underfitting

→ To prevent overfitting and underfitting

Decision Tree → hyperparameter → depth - CV (Cross Validation)

↳ λ logistic reg.

↳ C SVM.

Splitting numerical features

Construct a DT :- Splitting a node → Information Gain

IG → entropy

↳ Gini impurity → computationally efficient

f_1	y
2.2	1
2.6	1
3.5	0
3.8	0
4.6	1
5.3	0

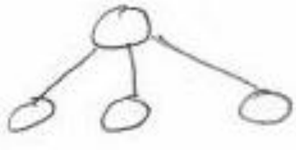
f_1 : numerical :

↳ Integer

↳ real valued

Split based on categorical variables

f_2 : 3-categories



① Sort the numerical feature (in ascending order)

$f_1 < 2.2$ (one possible condition)

at most n possible conditions:

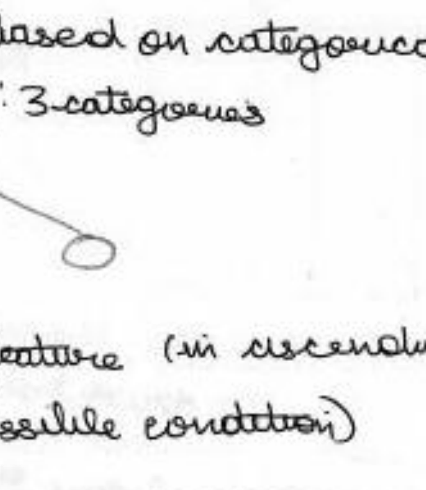
$f_2 < 2.6$

$f_1 < 3.5$

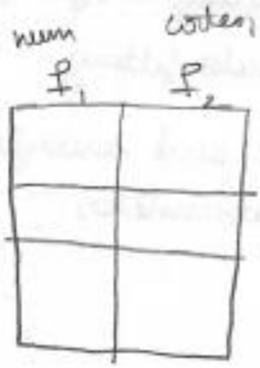
$f_1 < 3.8$

$f_1 < 4.6$

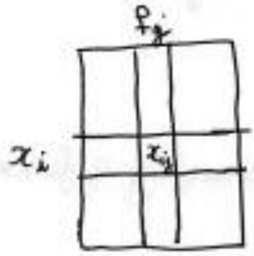
$f_1 < 5.3$



We calculate information gain for each n and compute maximum information gain



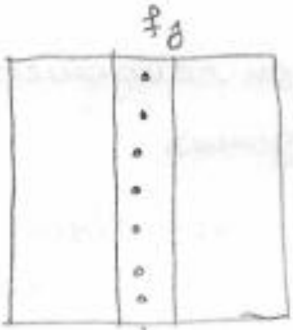
Feature standardization



$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

μ_j, σ_j } mean and standard deviation

In case of decision tree → Not a distance base method



Threshold

↳ sort this column

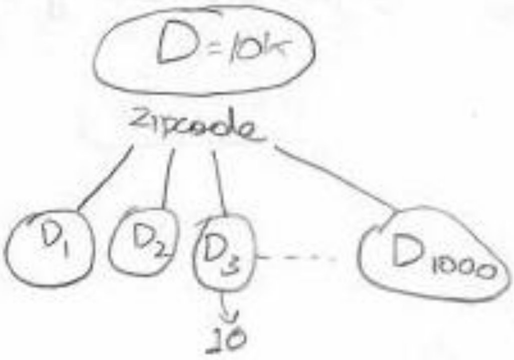
(In decision tree it is dependent on order)

NOTE

> No need to perform feature standardization in decision tree as it is not concerned with distance but it with order.

categorical features with many categories

pincode / zipcode → 1000's



hack (feature engineering)

Pincode $y_i = \{0, 1\}$

Pincode / Zipcode → Categorical

↓ convert it

numerical feature

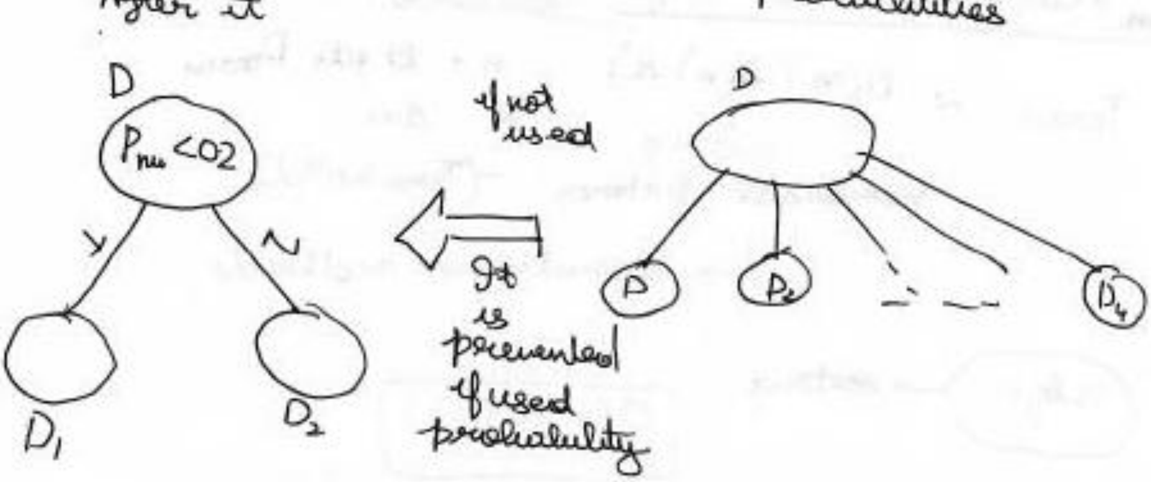
↓

$P(y_i=1 | P_j)$

For example $P(y_i = 1 | P_j)$

20 times p_j occurs
19 times $y_i = 1$
$P(y_i = 1 | P_j) = \frac{19}{20}$
After it

Conversion from Categorical to numerical feature using conditional probabilities



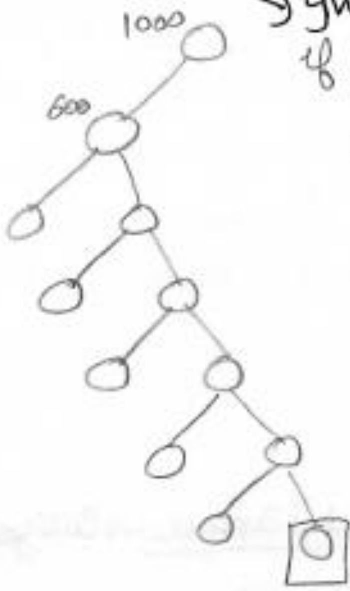
Overfitting and underfitting

depth ↑ :→ possibility of having very few pts @ a leaf node ↑

→ Interpretability of model also decreases

if ( ) & ( ) & ( ) & (( )

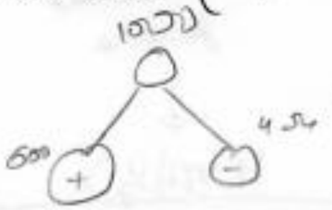
$\boxed{x_q \to y_q}$



2 pts - noisy

overfitting my model to noise

If depth is low (depth = 1)



underfitting

So right depth need to be discovered using cross validation

☆ Train & Run Time complexity

Train ~ $O(n\,(\lg n)\,d)$ — evaluate IG

sorting

n = # pts D_{Train}

d = dim.

numerical features: – (Threshold)

↳ algorithmic methods

$n \lg n$ → sorting

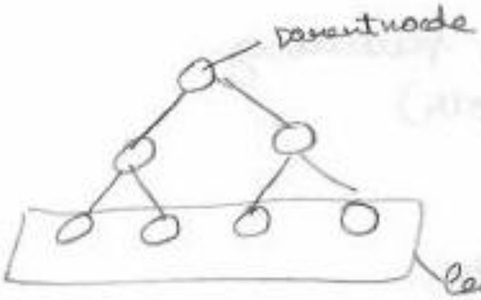
$\boxed{O(n \lg(n))}$

~~Also T~~

After Training :-

a) Runtime Space :- Store my D.Tree

$\boxed{x_q \to y_q}$ ↳ if else } nested if else



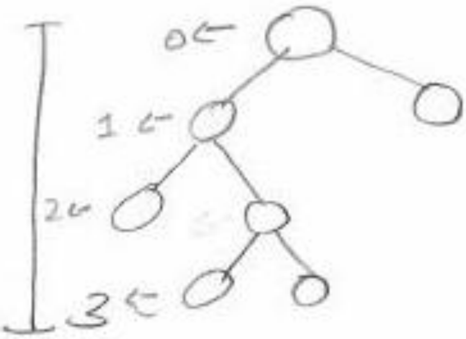
Σ # internal-nodes
\+
leaf nodes.

Space complexity O(#nodes)

depth = 5 or 10 $\boxed{\text{depth} \uparrow \Rightarrow \text{interpretability} \downarrow}$

Run time space :- resonable

run time complexity $x_q \to y_q$



K = max - depth of any leaf nodes

atmost 3 comparisions.

O(3).

DT :- Super good when you have large data.

dimensionality is small

low latency → O(depth)

Regression Using DT

Pros of Decision Trees

- Interpretable : humans can understand decisions
- easily handles irrelevant attributes (gain = 0)
- can handle missing data
- very compact : # nodes << D after pruning
- very fast at testing time : O(depth)

Cons :

- only axis aligned splits of data
- exponentially many possible trees

Example Decision tree

Book Data mining Pag 338

Examp 8.1

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

$$Info_{age}(D) = \left[\frac{5}{14} \times \left(\frac{-2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\left(\frac{-4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14} \times \left(\frac{-3}{5}\log_2\frac{3}{5} - 2\log_2\frac{2}{5}\right)\right]$$

$$= 0.694 \text{ bits}$$

Hence the Information Gain

$$Gain(age) = Info(D) - Info_{age}(D)$$
$$= 0.940 - 0.694$$
$$= 0.246 \text{ bits}$$

Similarly,

Info Gain(Income) = 0.029 bits

Info Gain (Students) = 0.151 bits

Info Gain (credit Rating) = 0.048

Info. Gain for age is maximum so it is selected