Decision Trees (DT) → if -- else

KNN, Naive Bayes, Logistic regression, LinearRe, SVM (kernel)

- KNN → instance based Method
- Naive Bayes → probabilistic method
- Logistic regression, LinearRe, SVM → geometric, hyperplane

DT: nested if -- else classifier

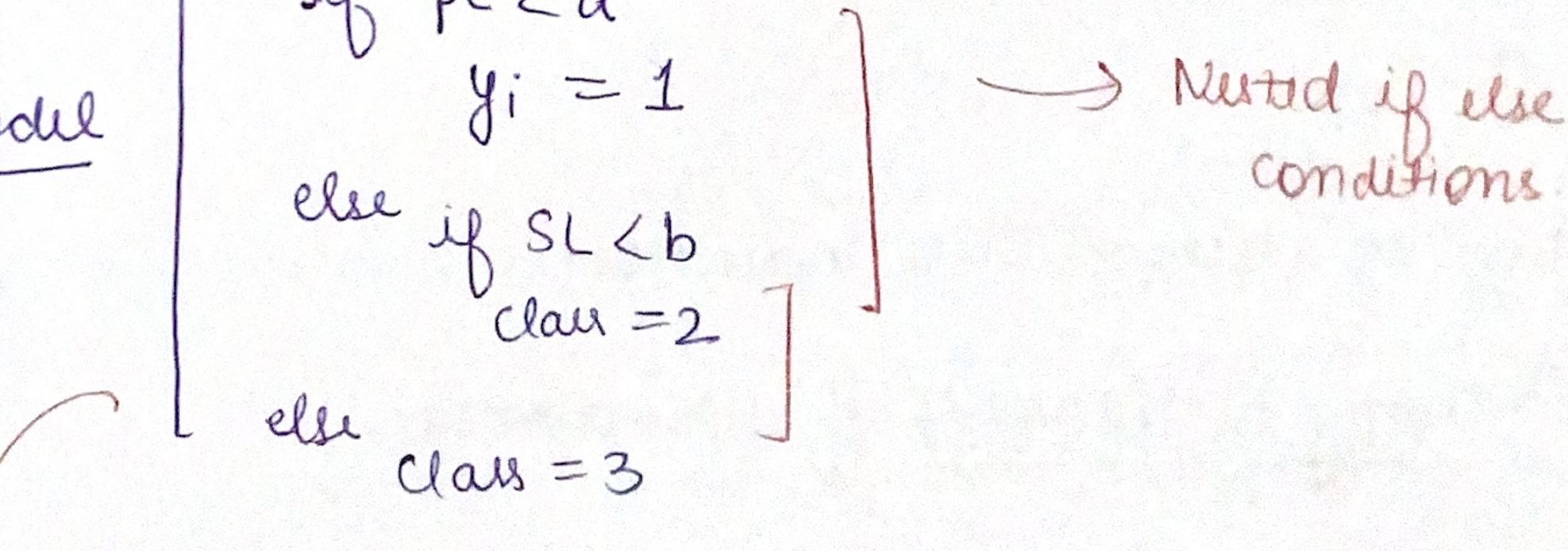
EDA: Iris Dataset
- ↳ y_i [1, 2, 3]
- (SL, SW, PL, PW)

Simple { if PL < 5 then $y_i = 1$ }

Model [x_i = < SL, SW, PL, PW >
if PL < a
$y_i = 1$
else if SL < b
class = 2
else
class = 3] → Nested if else conditions

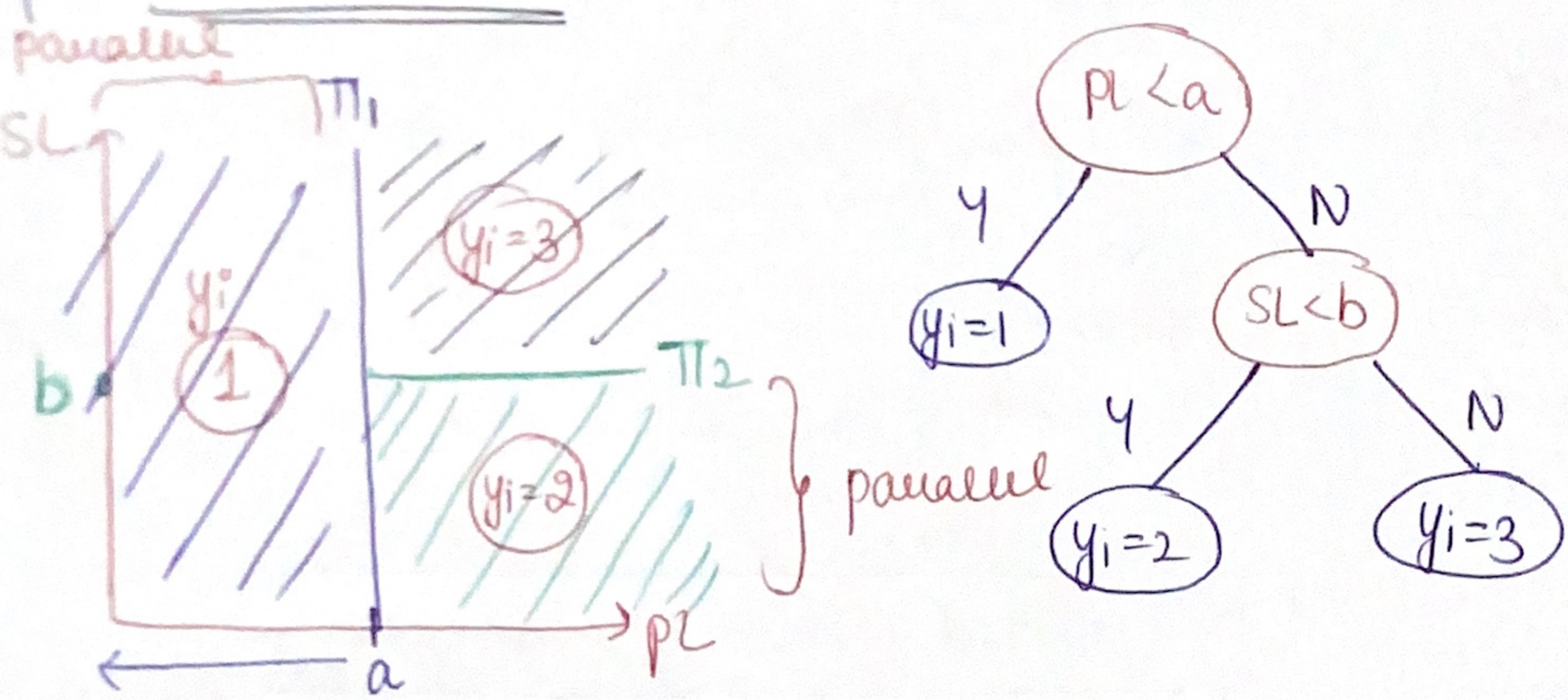
Diagram (Tree)

(Tree)



Geometric Interpretation

D·T: set of axis parallel hyperplanes



decision$_1$:- π_1

decision 2:- π_2

all of your hyperplanes are <u>axis parallel</u>

II to any one axis

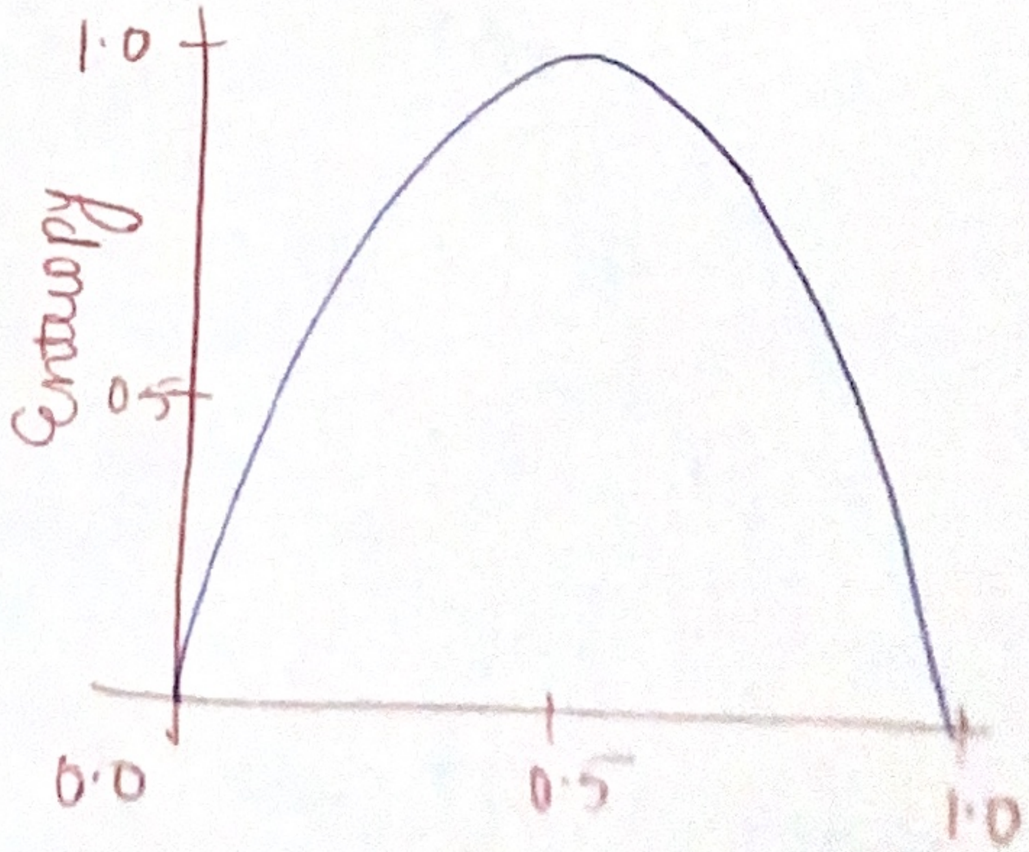
D·T:- Nested if else → programmatic

axis parallel hyperplanes → geometric

Entropy

The key to decision tree induction is notion of entropy

Entropy = measure of randomness



Obs: Entropy is max if we have 50-50% of split among +ve and -ve values.

Obs: Entropy is zero if we have all +ve or all -ve examples

Entropy

- Let $D = \{(\bar{x}_1, y_1), \ldots, (\bar{x}_l, y_l)\} \subseteq A^n \times \{+1, -1\}$

  A dataset D of l labeled examples. Each eg has features $\bar{x} \in A^n$ and binary label $y \in \{+1, -1\}$

- l_+ = number of positive examples in D

  l_- = number of -ve " " "

  Total $l = l_+ + l_-$

$p_+ = \frac{l_+}{l}$ $p_- = \frac{l_-}{l}$ are class probabilities.

$$\text{Entropy}(D) = -\frac{l_+}{l}\log_2\left(\frac{l_+}{l}\right) - \frac{l_-}{l}\log_2\left(\frac{l_-}{l}\right)$$

$$D = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Building a DT: Entropy

$D_{train} \longrightarrow DT$

RV $Y \longrightarrow y_1, y_2, y_3 \cdots y_k$

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

entropy

$b = 2$ ✓ ou $b = e = 2.718$

$P(y_i)$

$$= -\frac{9}{14} \lg\left(\frac{9}{14}\right) - \frac{5}{14} \lg\left(\frac{5}{14}\right) = 0.94$$

$\log_2 = \lg$

$\log_e = \ln$

$\frac{\#\text{-ve pts}}{\#\text{-ve pts}}$ → $P(y+)$ → $P(y-)$ → % age of -ve pts in D

Properties: $Y \rightarrow y+, y-$ (2 Class)

Case 1 $Q \rightarrow y+ \rightarrow 99\%$

$\rightarrow y- \rightarrow 1\%$

$H(Y) = -0.99 \lg 0.99 - 0.01 \lg 0.01$

$= 0.0801$

Case 2 $Q \rightarrow y+ \rightarrow 50\%$

$\rightarrow y- \rightarrow 50\%$

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

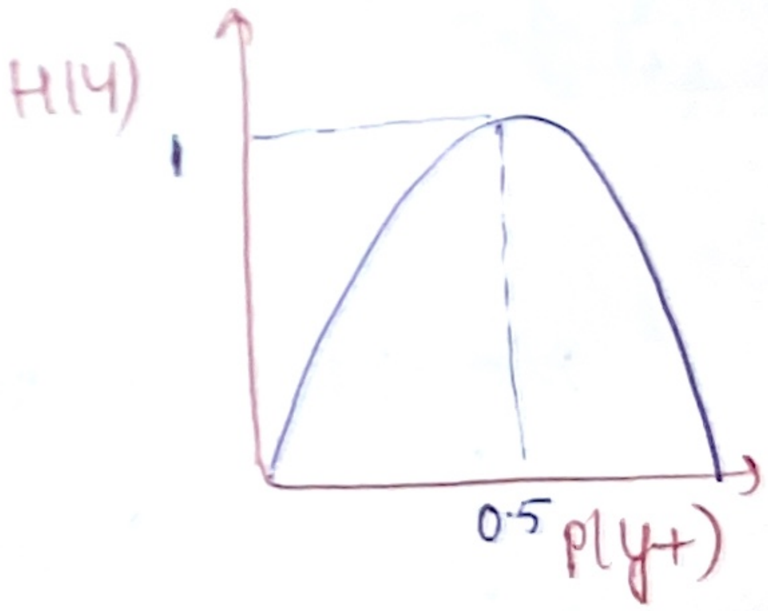
$= 1$

Case 3 $y+ \rightarrow 0\%$

$y- \rightarrow 100\%$

$H(Y) = 0$

2-Class case



$$p(y_+) = 1 - p(y_i)$$

$y \rightarrow y_1, y_2 \text{ -- } y_k$

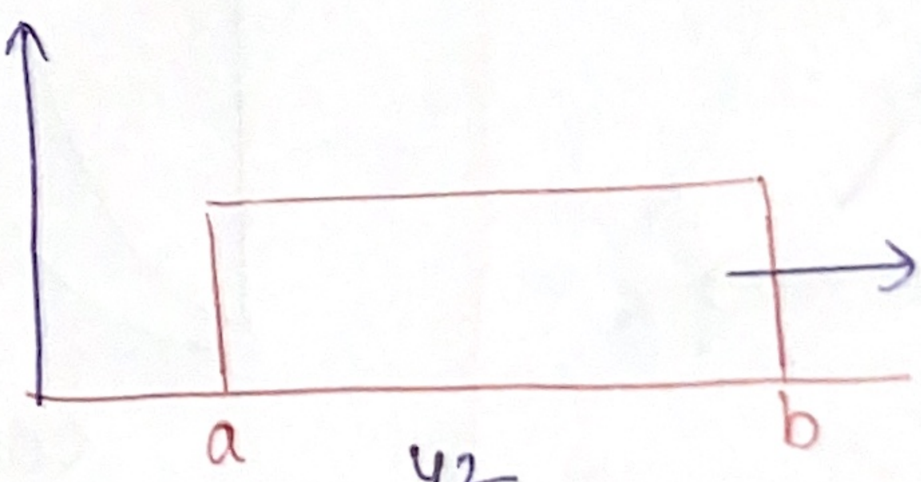
equi-probable → entropy is maximum

$y_1 \rightarrow$ most probable } entropy is minimum

$y_2, y_3 \text{ -- } \rightarrow 0$

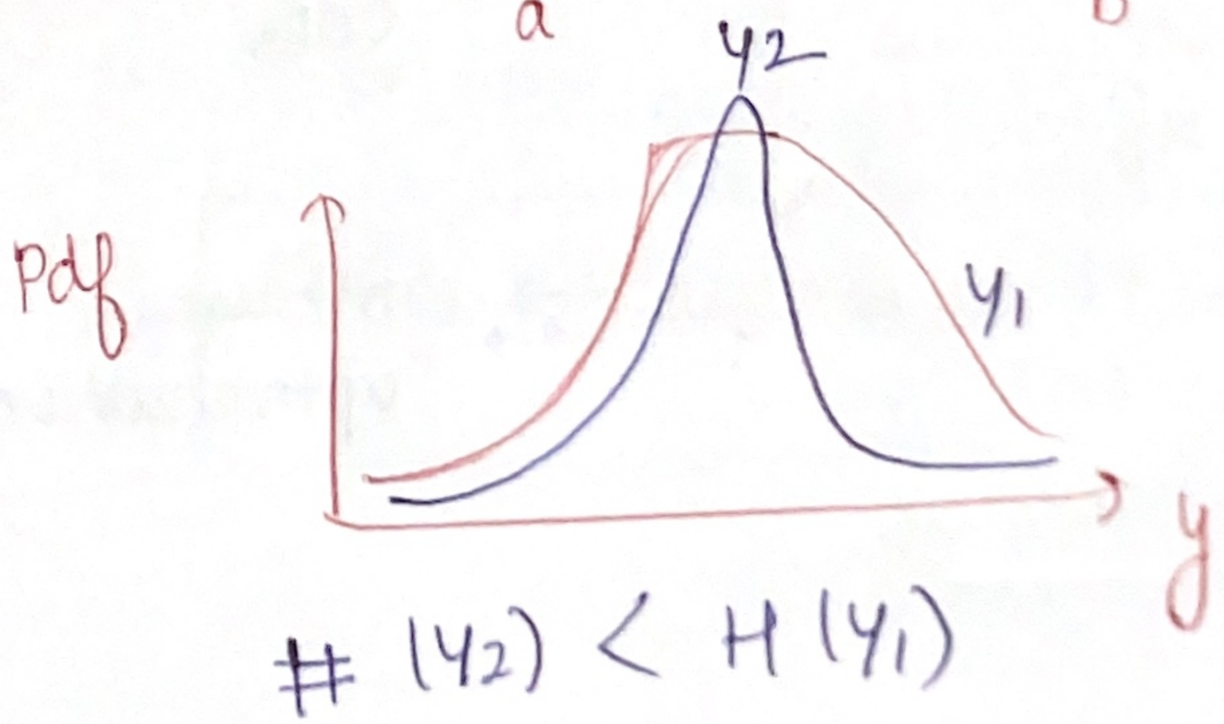
0.94

PDF



→ uniform dist

(entropy max = 1)

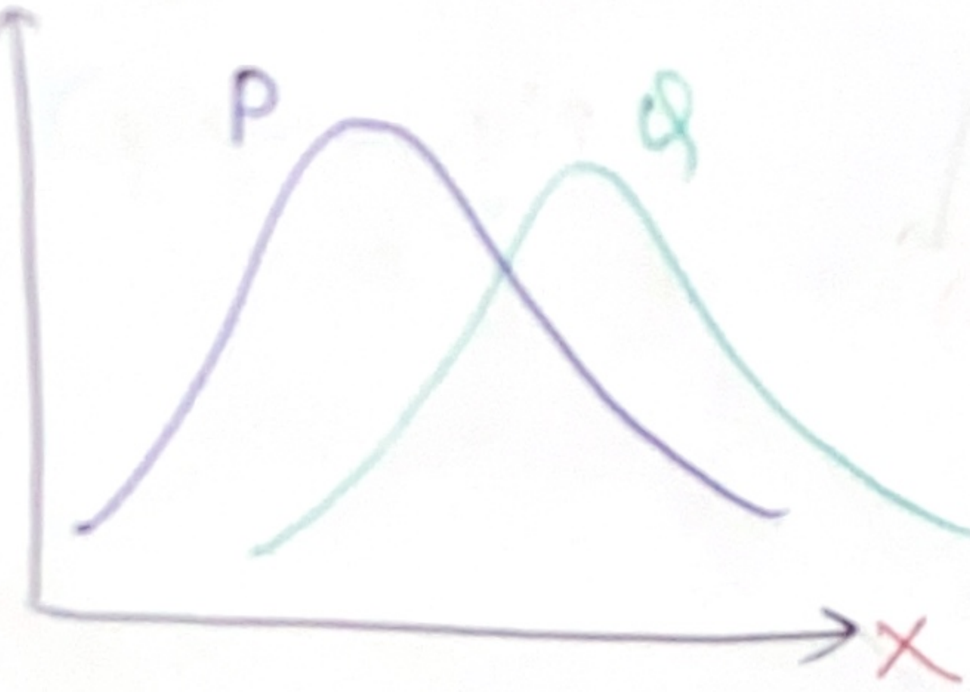


$y_1 \rightarrow$ less peaked

$y_2 \rightarrow$ peaked

$\# (y_2) < H(y_1)$

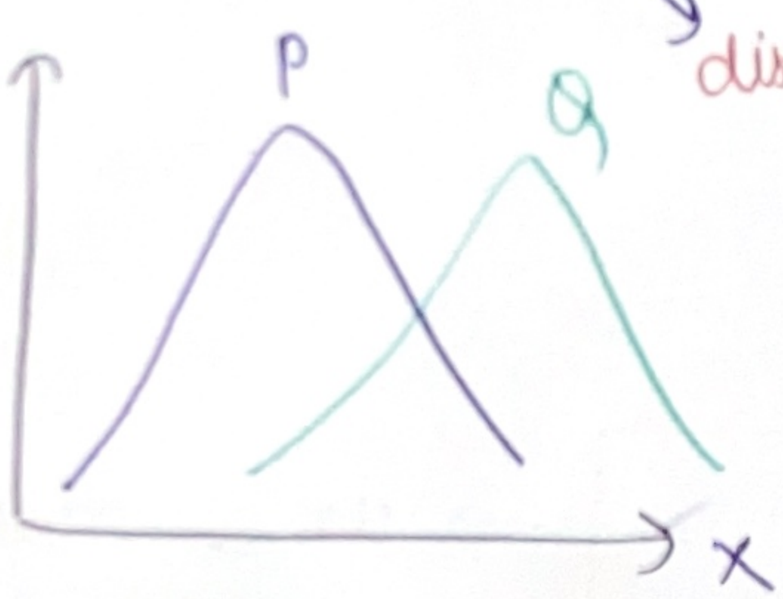
Kullback-Leibler (KL) Divergence



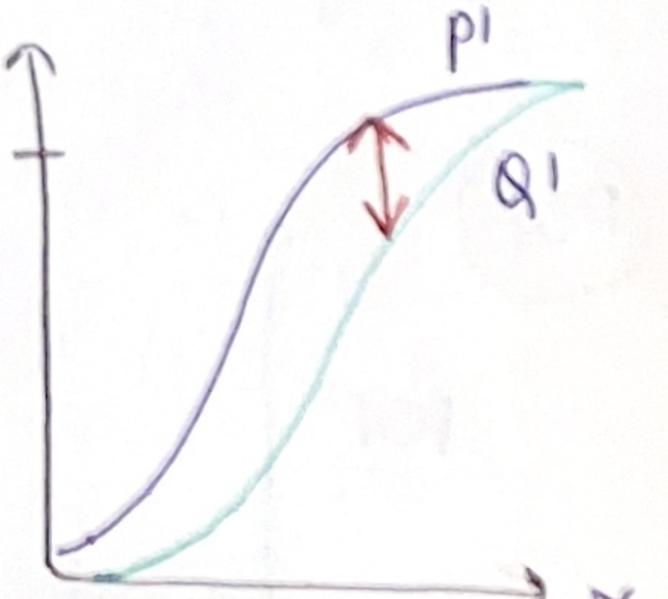
dist (P,Q) → Small
↘ large

must be differentiable to use as loss function

One Idea: KS- Statistic = $\sup\{|P'(x) - Q'(x)|\}$



dist(P,Q)



derivative → can't use in optimization probl

Info-Theoretic measure

$$D_{KL}(P||Q) = \sum_{x} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

KL Divergence

discrete r.v.

or

continuous r.v

$$\int_{x} P(x) \log\left(\frac{P(x)}{Q||x}\right)$$

a.k.a relative entropy

$$D_{KL}(P||Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

$\log\left(\frac{a}{b}\right) = \log a - \log b$

diff

$$= \sum_x P(x) \log P(x) - \sum_x P(x) \log(Q(x))$$

Information Gain (IG)

$$IG(Y, \text{Outlook}) = \underbrace{\left(\frac{5}{14} \times 0.97\right) + \left(\frac{\cancel{4}}{14} \times 0\right) + \left(\frac{5}{14} \times 0.97\right)}_{\text{weighted entropy after } D_1, D_2, D_3} - 0.94$$

$$= \left(\frac{5}{7} \times 0.97\right) - 0.94 = IG$$

① $Y \xrightarrow{\text{var}} y_1, y_2 \cdots y_k$ ($D_1, D_2, \ldots D_K$)

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

Gini Impurity ~ Similar to Entropy

$I_G(Y) \neq I_G(Y)$

↙ gini Impurity

$Y \rightarrow y_1, y_2 \cdots y_k$

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

$Y \rightarrow y+$
$\quad\quad y-$

Case 1 $P(y+) = 0.5$

$P(y-) = 0.5$

$I_G(Y) = 1 - (0.25 + 0.025)$

$= 0.5$

$H(Y) = 1$

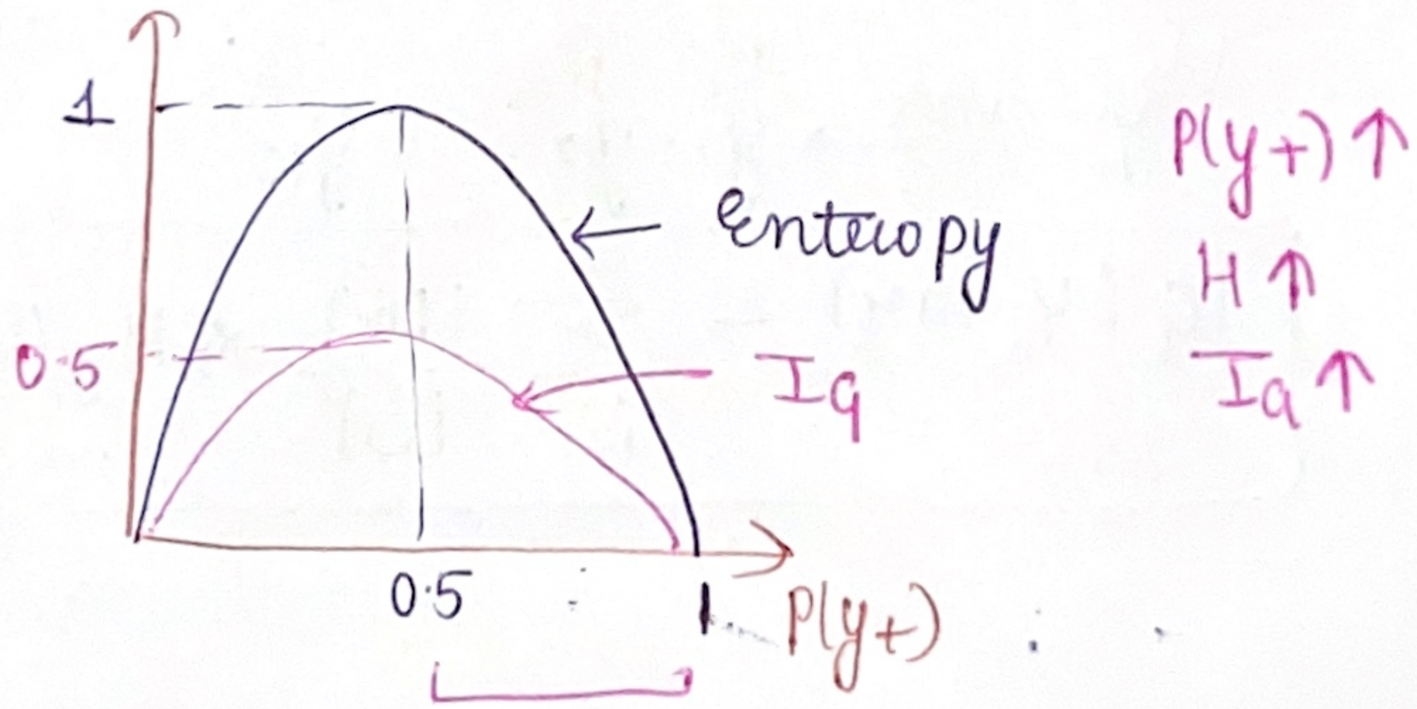
Case 2 $P(y+) = 1$

$P(y_0) = 0$

$I_G(Y) = 1 - (1+0) = 0$

$H(Y) = 0$

2 - Category Case $y+, y-$ $\quad P(y+) = 1 - P(y-)$



$Ig(4)$

$1-(P(y+)^2+P(y-)^2$

↓

no log

more computationally easy and efficient to calculate

$H(4)$

$-P(y+)\log_2 P(y+) -$

$P(y-)\log_2(Py-)$

↓

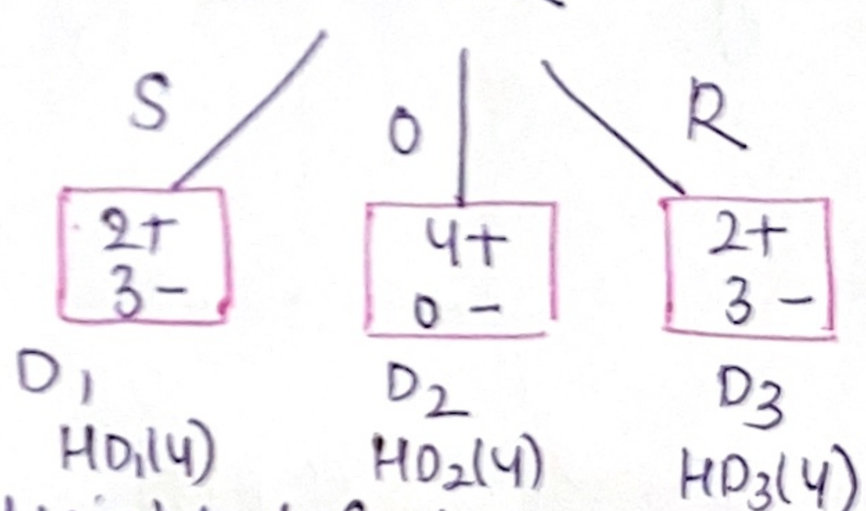
log

Construct a DT: H, IG, Ig

① $D \rightarrow 9+, 5-$

$H_D(Y) = 0.94$

② $\xrightarrow{D=0.94}$ Outlook $\quad 0.94-0.69$

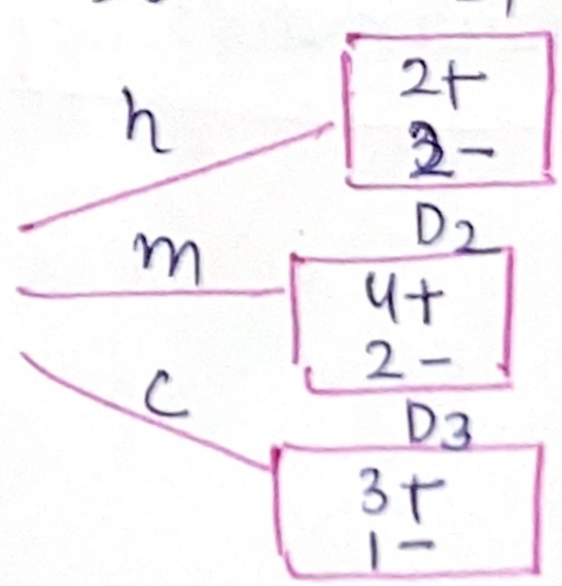


Weighted Entropy = 0.69

$0.94 - 0.69 = 0.25$

$IG(Y, Outlook) = 0.25$

③ Temperature



$IG(4, Temp) =$

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

$D \longrightarrow$

- $IG(Y, \text{outlook}) = 0.85$
- $IG(Y, \text{temp}) =$
- $IG(Y, \text{Humidity}) =$
- $IG(Y, \text{windy}) =$

$IG(Y, f)$ = entropy @ parent level − weighted entropy @ child level

1. pure node – stop growing the node
2. can't grow the tree anymore because of lack of pts.
3. if we are too deep.

depth of tree ↑ ⇒ overfit ↑ (few pts)

depth is small ⇒ underfit

DT: hyperparameter: depth

↓

CV

Splitting numerical features

Construct a DT: splitting a node → IG

IG :— Entropy
\
gini impurity → computationally efficient

Discrete RV or categorical features we have seen till now

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 cat. val was easy
↓
f_2: 3 Categories

① Sort the numerical feature in ascending order

②
$f_1 < 2.2$
$f_1 < 2.6$
$f_1 < 3.5$
$f_1 < 4.6$
$f_1 < 5.3$

Numerical features

$f_1 < t_1$ → D_1 / D_2

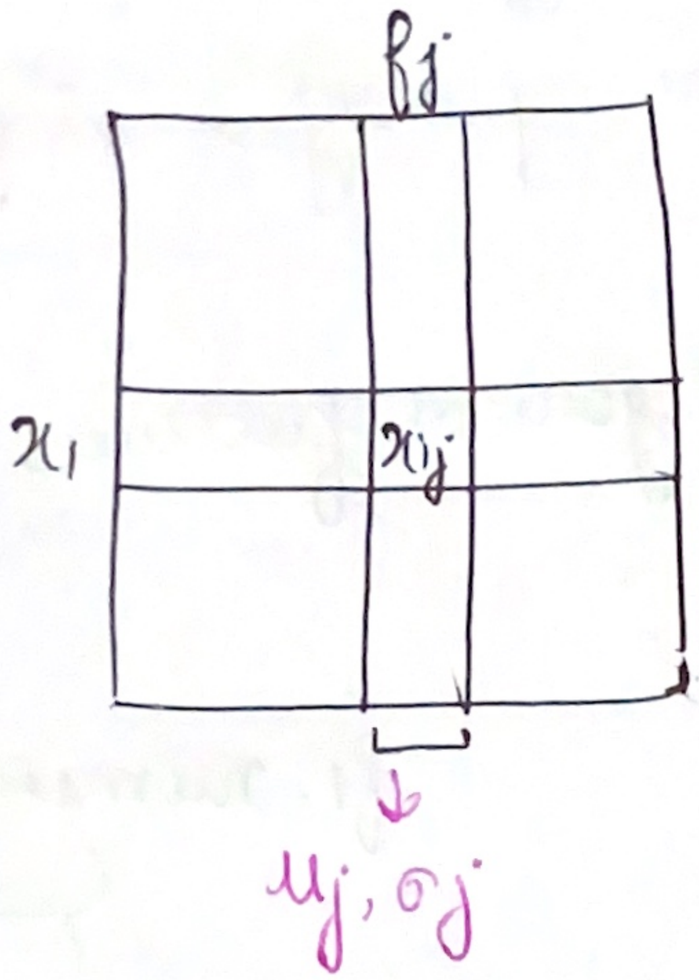
$f_2 < t_2$ → D_1 / D_2

$f_3 < t_3$ → D_1 / D_2

Feature Standardization

- logistic reg :
- SVM
- KNN

$\rightarrow$ feature stdn

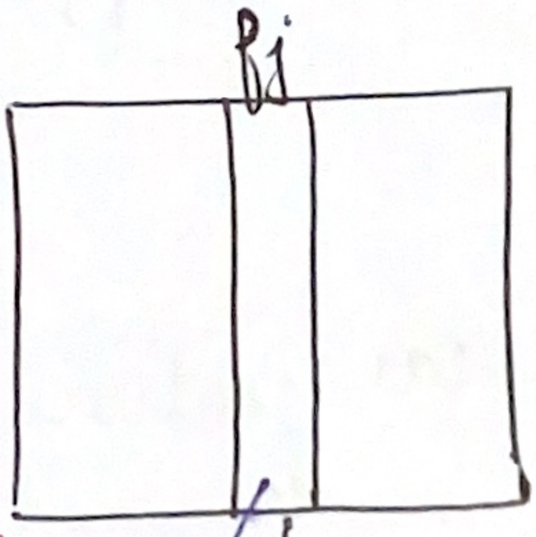


u_j, σ_j

$$x_{ij}' = \frac{x_{ij} - u_j}{\sigma_j}$$

Decision Trees

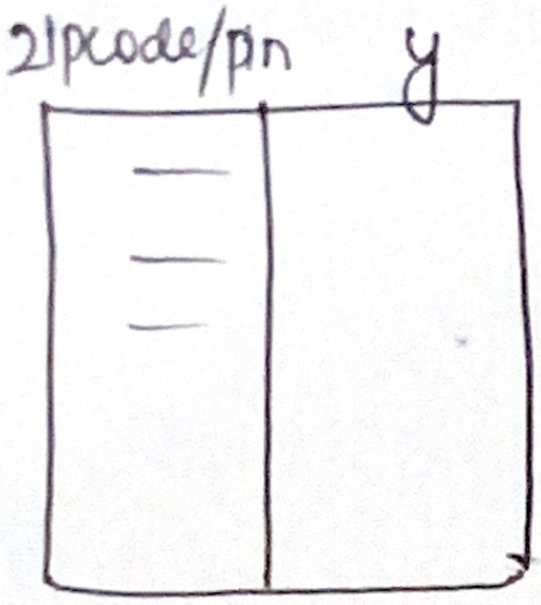
not a distance based method

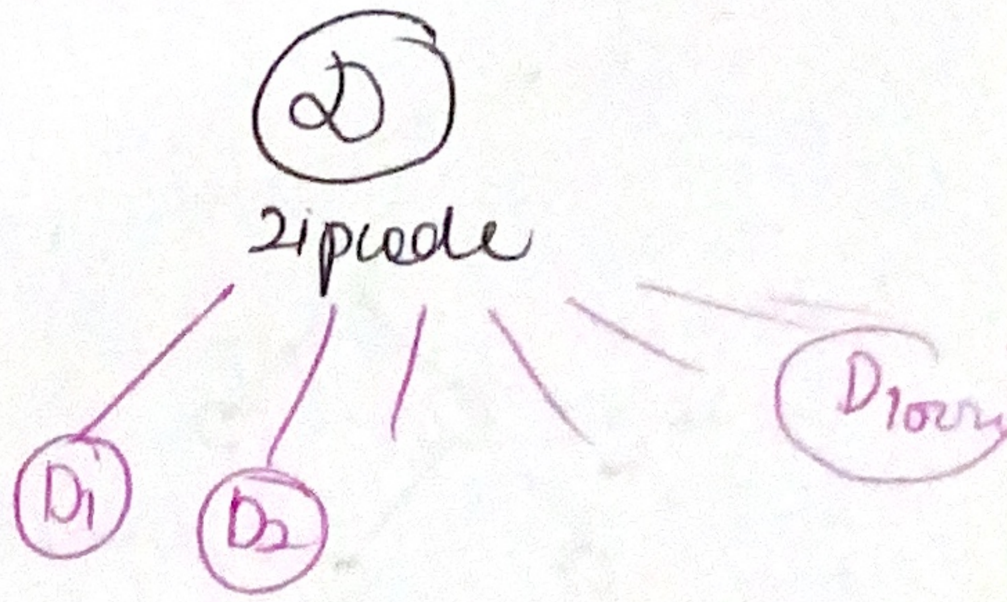


- do not need to perform feature Standardization
- Sort this column
- only order matters

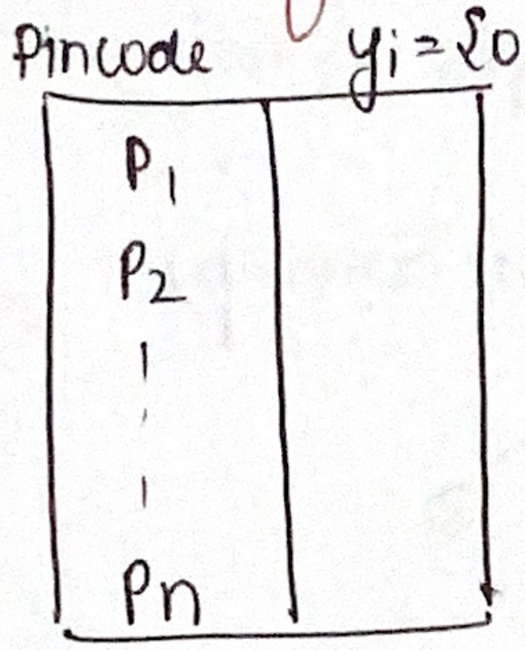
Categorical features with many categories
↳ levels.

pincode / zipcode
↳ 1000's





Hack (feature engg)



20 times Pj
19 times $y_i = 1$

pincode / zipcode ← categ-
↓
convert it to
numerical feature

Pj = □

$P(y_i = 1 \mid P_j) =$

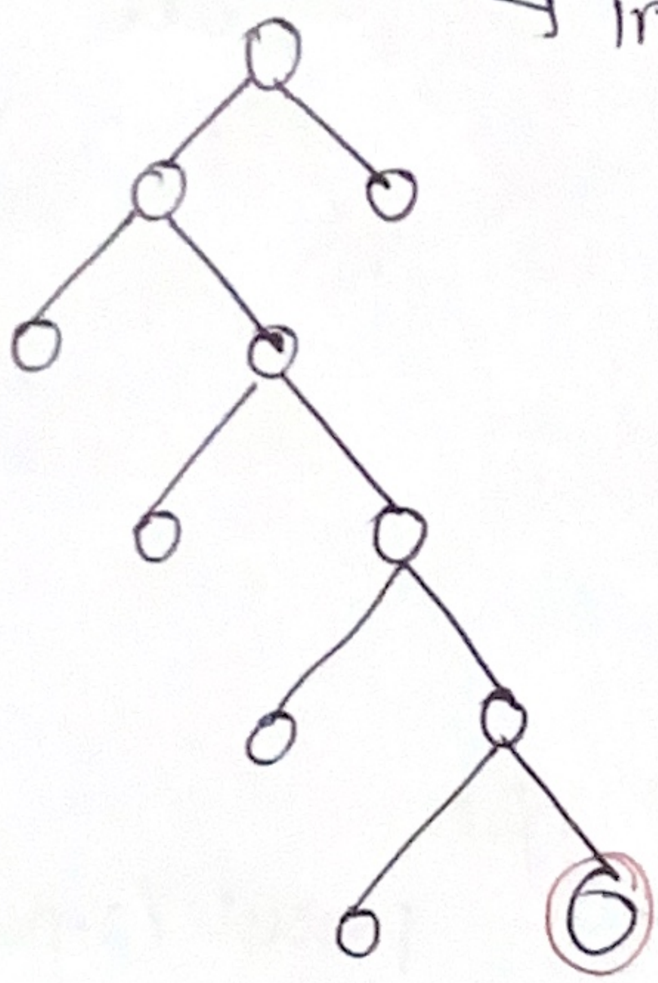
$$= \frac{\#\, y_i = 1 \;\&\; P_j}{\#\, P_j}$$

$P_j \rightarrow$ numerical feature $= P(y_i = 1 \mid P_j)$

D → same pincodes feature
↓
numerical features

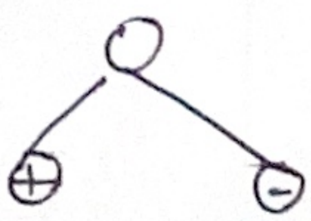
Overfitting & Underfitting

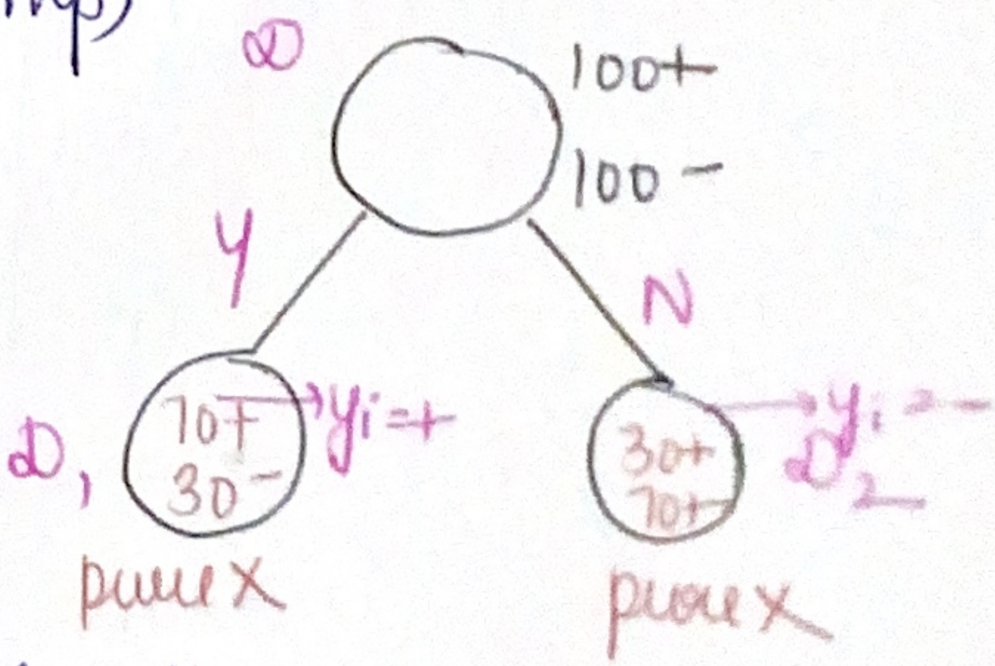
depth ↑ : possibility of having very few pts at leaf node ↑ → Interpretability of model ↓



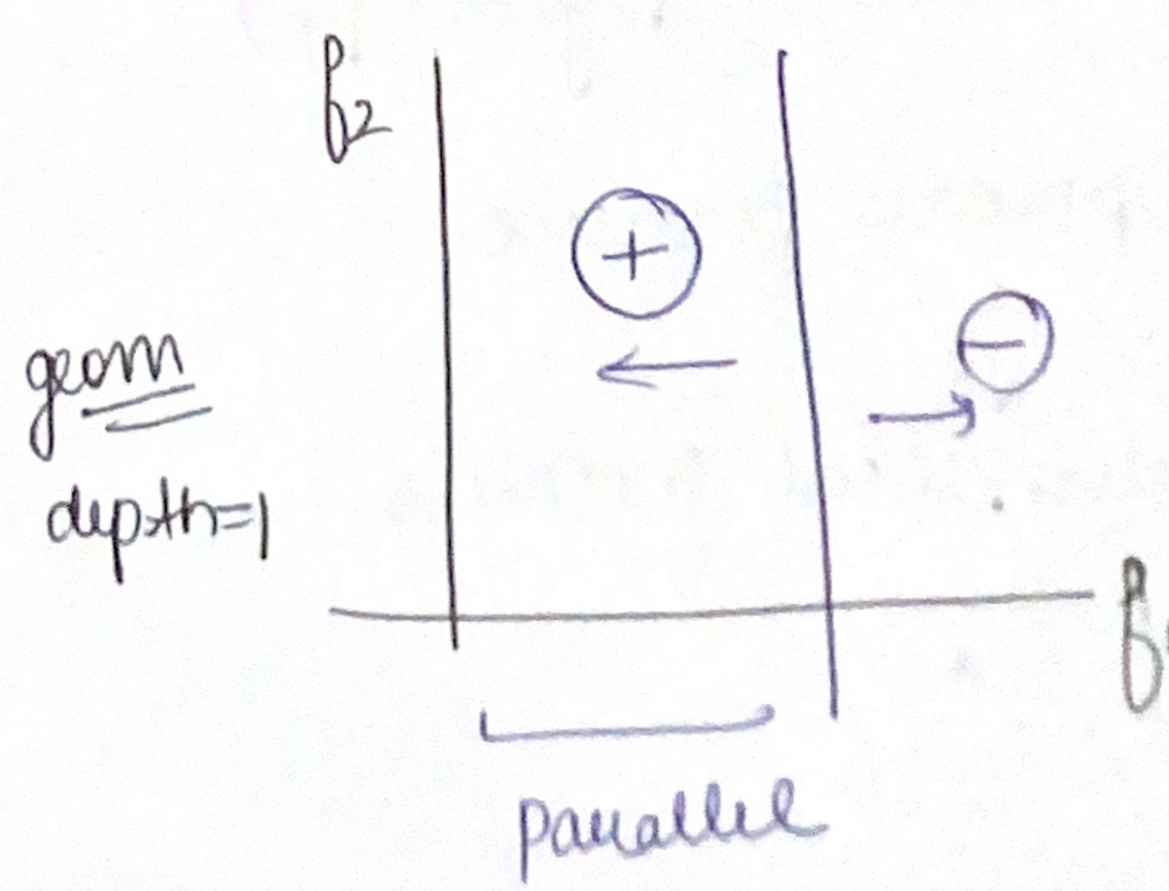
2 pts → noisy → overfitty my model to noise

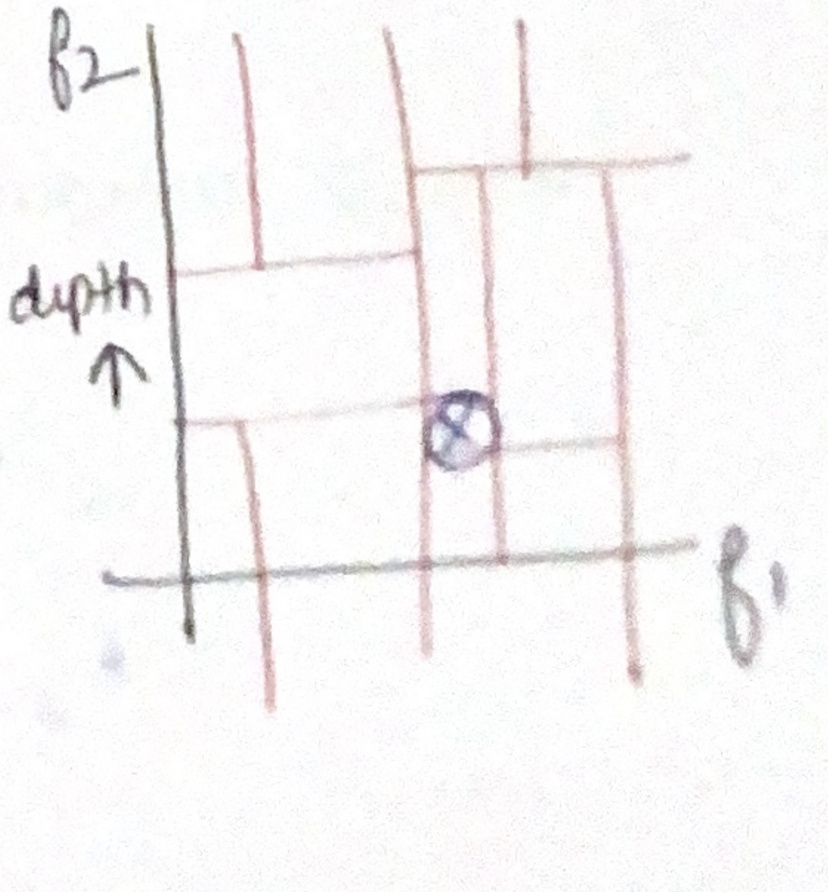
depth ↓ : =1 = (decision Stump)





depth determined by cross Validation





Train & Run Time Complx

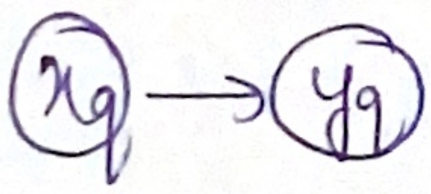
Train: $\sim O(n \lg n) d)$ n = #pts in D_{train}
d = dim

numerical features: (threshold)

↳ algorithmic methods

After Training:

at Runtime Space



Store my Dtree

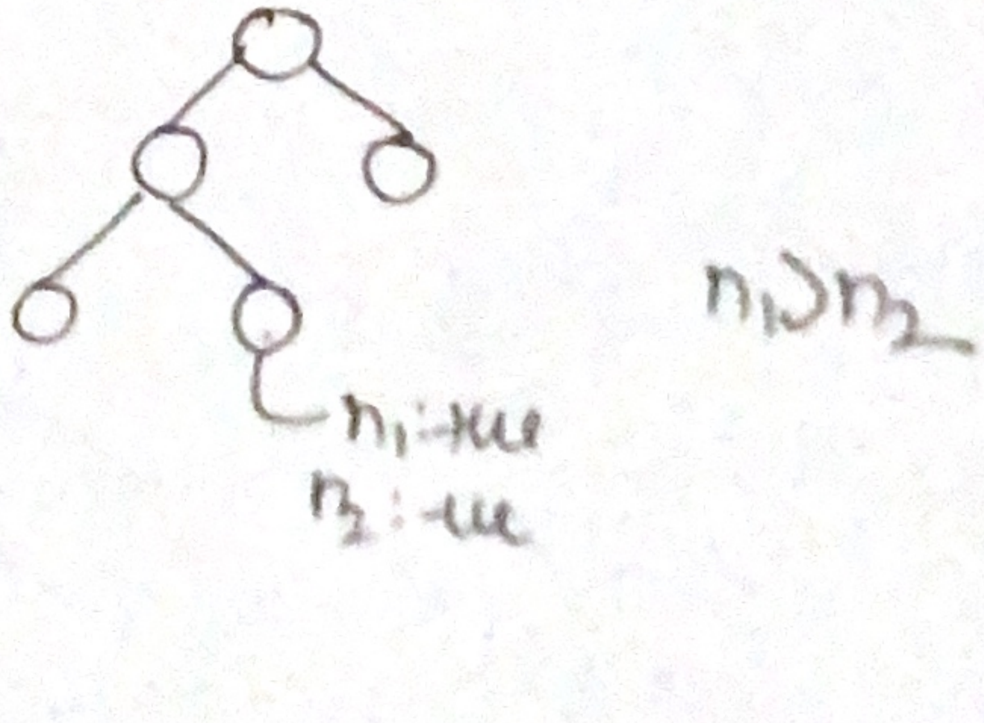
↳ if else } nested if else

[# internal nodes
+
leaf nodes

DT: large data, dimis small
low latency → O(depth)

Regression Using Decision Trees:

DT → clasfn



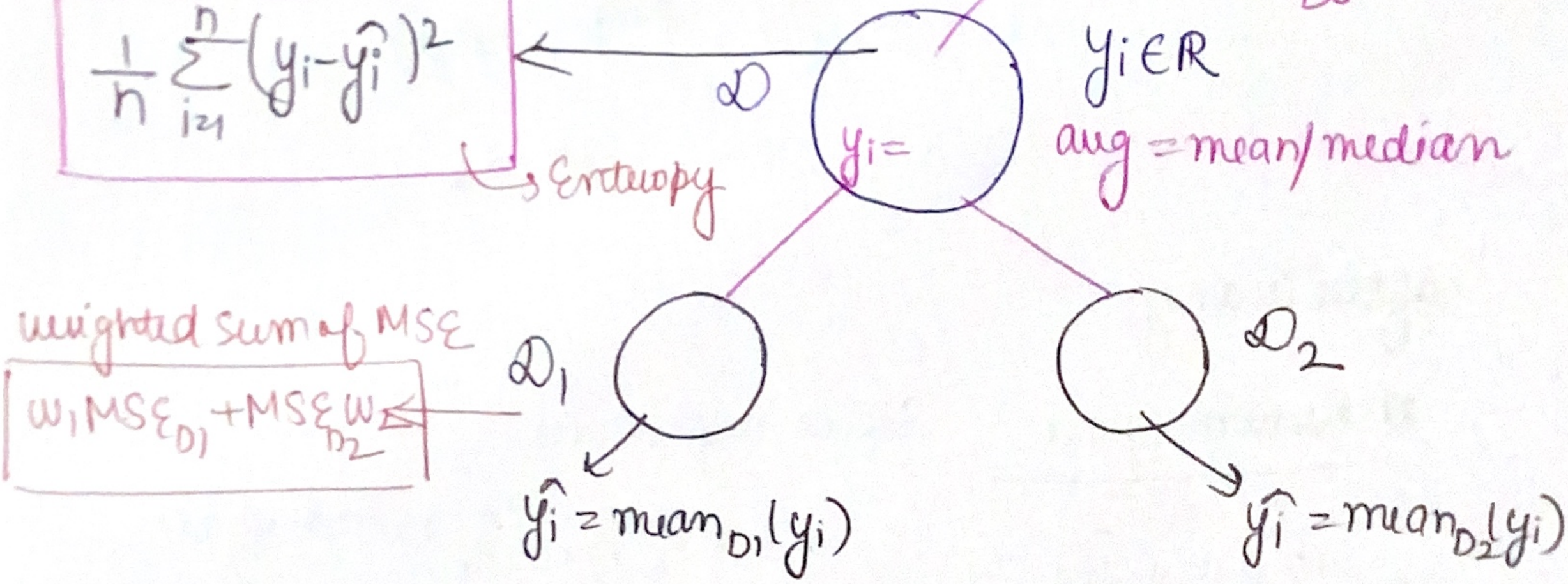
IG → classification only , Regression X

MSE or MAE → Regression

Mean Squared Error (MSE)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y_i})^2$$

↳ Entropy



classification → β_i :- reducing entropy : "0"

regression → β_i : reduce MSE → "0"

$$MSE_D(y_i) = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y_i})^2$$

(MAE) = MAD → "0"

Median $(|y_i-\hat{y_i}|)$