1. DTR.
2. Pre Pruning | Post Pruning
3. hyperParameter
4. Practical

DTC => 1. Entropy, 2. Gini-impurity → (Information gain)

Target => (Classes)

I/P: {Class/Num} → target: (Classes) Binary, multiclass

2 Scenario

Target => Numerical

Decision tree Regressor

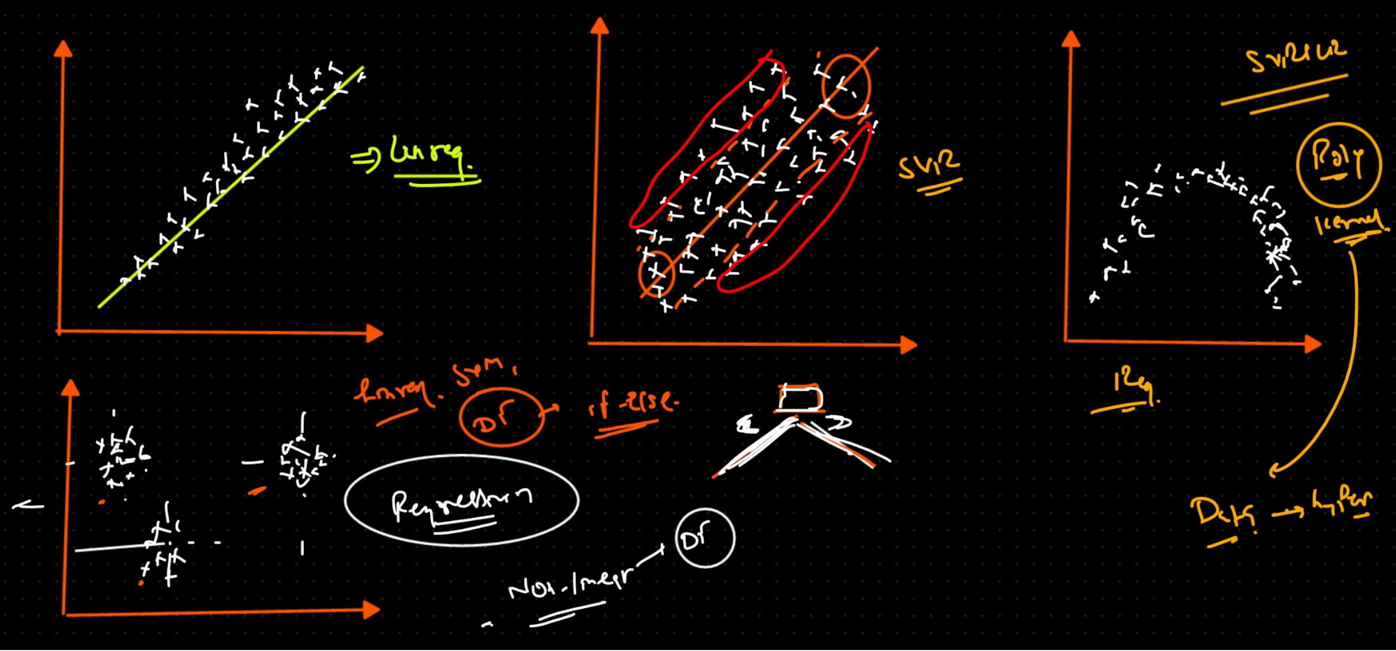
Num. (Height) → target (DTR) (Weight)

Height	Weight
160	50
165	65
180	50
170	85
175	70

DTR: Class → Num

Gender	Weight
M	60
F	50
M	50
F	65
M	70

DTR/DTC => 4



complex ⇒ DT ⇒ ensemble

Height	Weight
165	65
160	50
180	90
170	85
175	70

{ Classification ⇒ Gini impurity, Entropy } → info gain → Root ⇒ IG highest

{ Regression ⇒ Variance → Reduction in Var ⇒ Root ⇒ RV lowest

1. Sort the value in ascending order
2. average of adjacent value in index col.
3. variance → Reduction in variance

Height	Weight
160	50
165	65
170	85
175	70
180	90

162.5, 167.5, 172.5, 177.5

[50, 65, 85, 70, 90]

Height (best split) — Var = 206

≤ 162.5 → [50], Var = 0

> 162.5 → [65, 85, 70, 90], Var = 106.25 {Deviation from the mean}

1 Var:
- Var (Root)
- Var (Left)
- Var (Right)

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

(y_i: D.P., $\hat{y}$: Mean)

$$\left\{ \frac{\sum_{i=1}^{N}(y_i - \hat{y})^2}{N} \right\}$$

Var Root = [50, 65, 85, 70, 90] ⇒ Avg = $\frac{50+65+85+70+90}{5}$ = 72

$$= \frac{(72-50)^2 + (72-65)^2 + (72-85)^2 + (72-70)^2 + (72-90)^2}{5} = 206$$

Var(L) = 50 ⇒ 0

Var(R) = [65, 85, 70, 90]

$= \frac{65+85+70+90}{4} = 77.5$

$= \frac{(77.5-65)^2 + (77.5-85)^2 + (77.5-70)^2 + (77.5-50)^2}{4}$

$= 106.25$

Minimum RSV = that is going to my Root Node

Reduction in var = $Var(Root) - \sum_{i=1}^{n} w_i \times var(Childnode)$

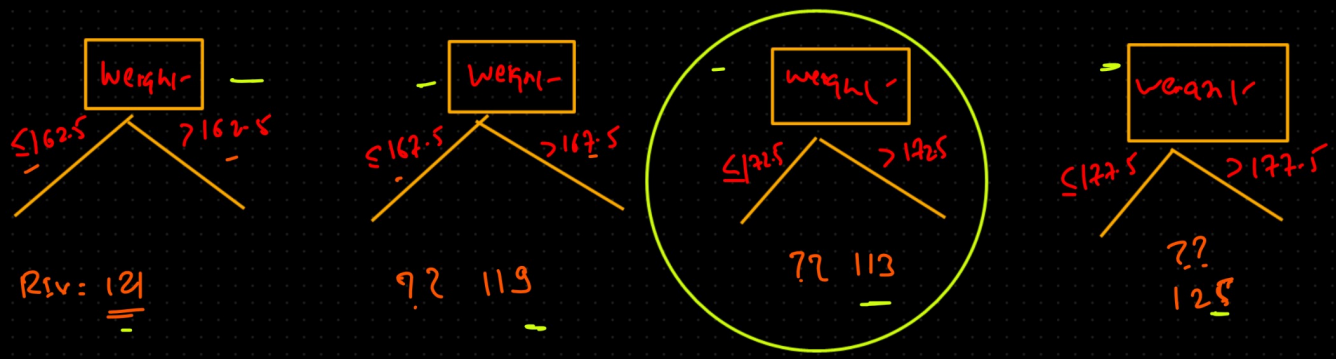
$= 206 - \left[\frac{1}{5} \times 0 + \frac{4}{5} \times 106.25\right]$

$= 206 - [85]$

$= 121 \Rightarrow$ RSV

162.5

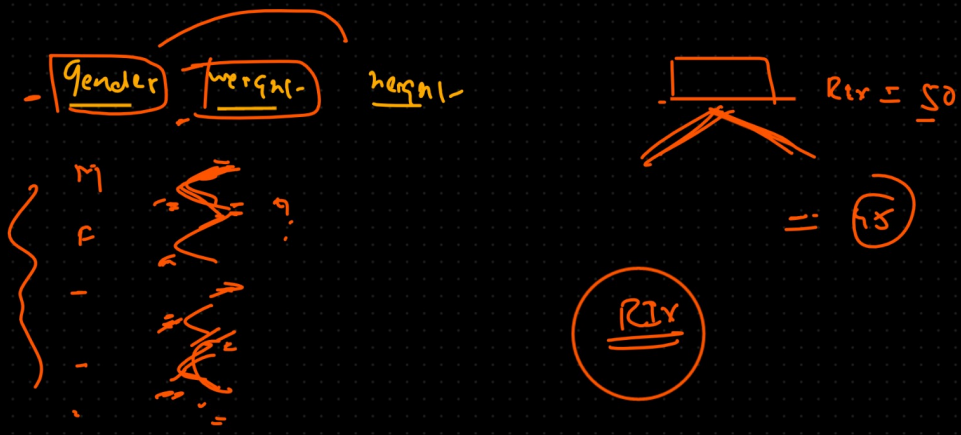
Gender Margin



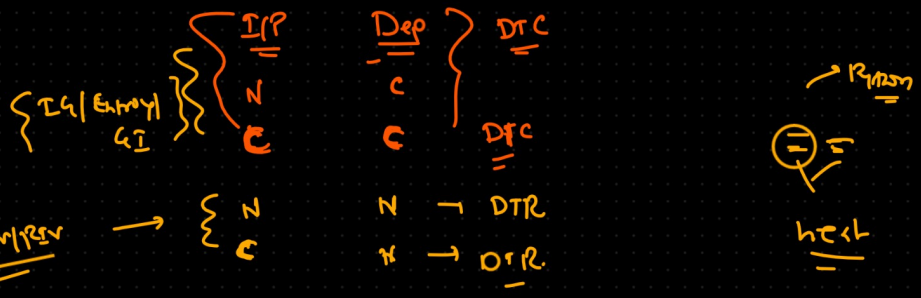
- Can you calculate it?

How to select Root node in DTR

Assume ⇒ Minimum Rev



Rev ⇒ DTC (DTR) Var ⇒ DTR. entropy ⇒ DTC



Pruning

1. Pre Pruning
2. Post Pruning

Pre Pruning ⇒ While i'm building the DT at that time only we decide the **structure** of the tree.

1. Max_depth
2. max_sample_split
3. min_sample_leaf
4. max_feature

Structure / Depth

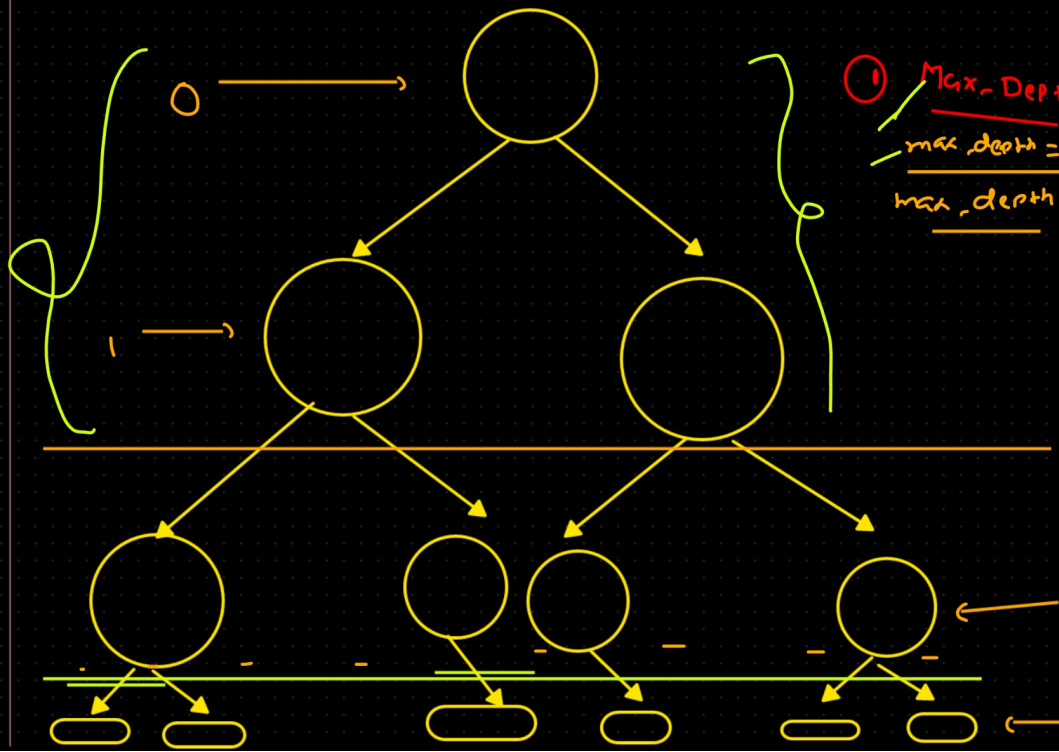
= {PrePruning} / PostPruning { Overfitting } reduce the

error will low

⇒ Bias will be low → training acc will be high
Var will be high → test acc will be low
error will higher

Underfitting = {training acc low → error ↑ / testing acc low → error ↑} { High bias, High var }

Balance tree ⇒ ~~Underfitting~~ / ~~overfitting~~



① Max_Depth = 1
max_depth = 2
max_depth = {till that level my DT is going to build?

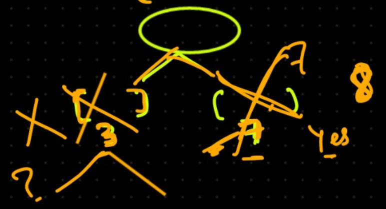
Max_depth = 5

→ 0
→ 1
→ 2
→ 3
→ 4
→ 5

Data
DT

full DT

10
[50, 60, 70 ... 80, 90]



(2) min_sample_split ⇒ Split ⇒ sample min

= (5)

min_s_sample

split

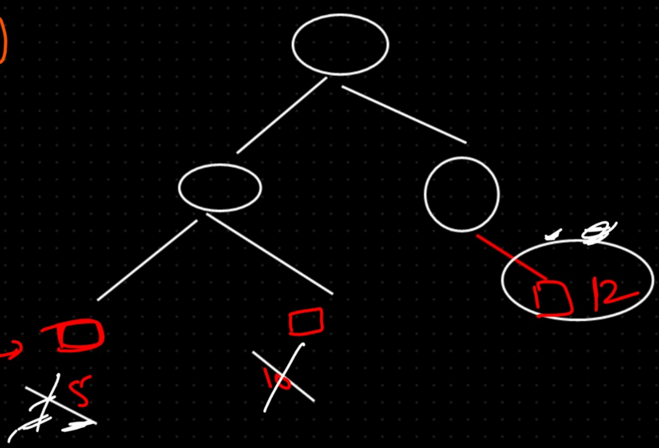
further

< 5

min_sample = [ ]

(3) Min_Sample_Leaf

leaf

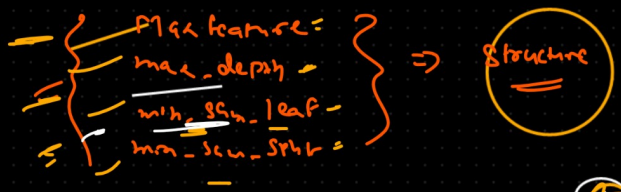


(4) max_feature = {1, 2, 3, 4, ... n}

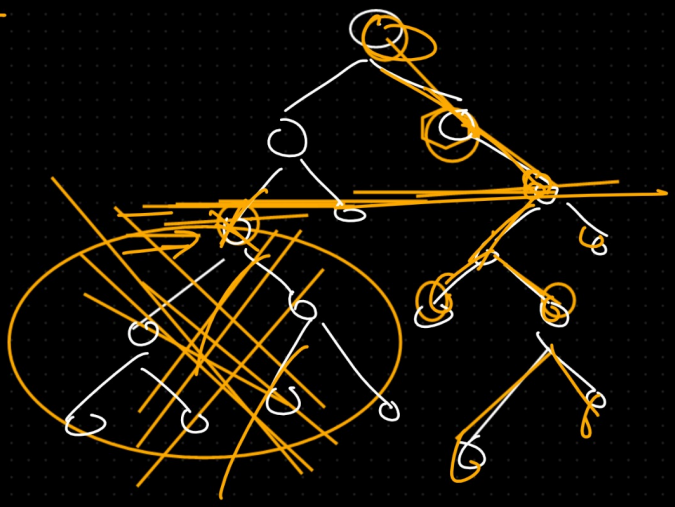
DT

10

PrePruning



full complete

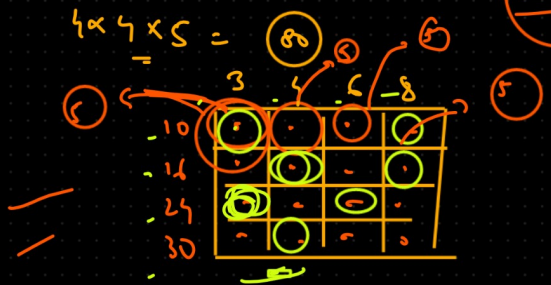


HyperParameter
1. Grid search
2. Random Search

Max_Depth = [3, 4, 6, 8]

min sample_split = [10, 14, 24, 30]

G.S.CV 5 => Combination => multiply

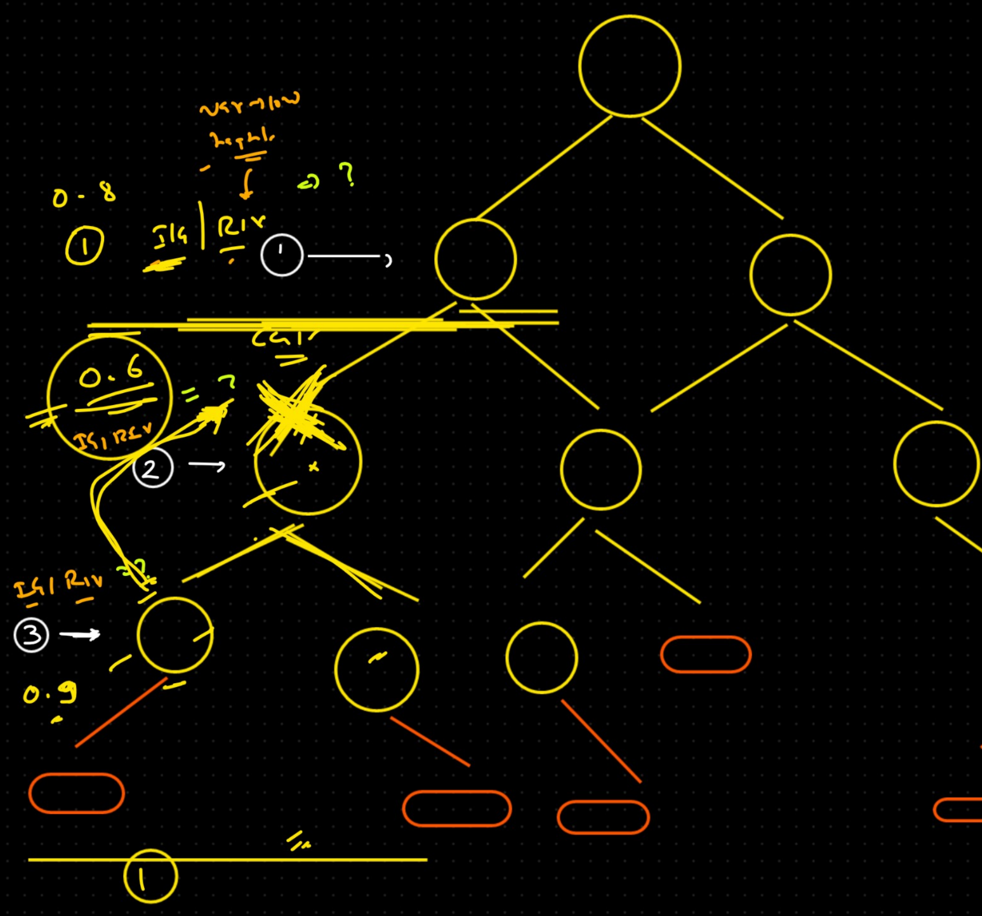


80 => Post Pruning

Post Pruning =>

- Grow Complete DT
- Build complete DT

Prune it or you need to cut it?



Complete / full DT

Prunerir

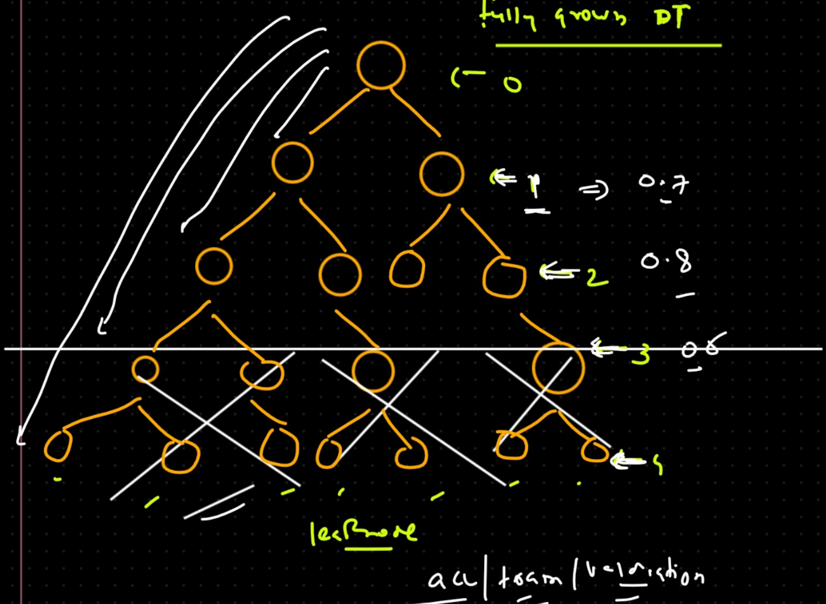
Threshold = 0.7

Recursive

Pre Pruning = Deciding the Depth of tree | Structure of the tree while building it.

Post Pruning => first Create | Build Complete DT then cut the branches Based on defined threshold

↳ ccp-alpha



ccp-alpha = 0.7 => minimum

0.7 = 0.7 ✓

0.8 > 0.7 ✓

0.5 < 0.7

Entropy = [0-1]

(0-∞)

ccp_range = [...]

h.p.T

Intell

PrePruning

Adv (1) PrePruning can be computationally less expensive

huge amount of data

(2) Reduce the overfitting

=

D.S ⇒ Parameter ⇒ tuning ⇒ always it might not lead to the good result

bad

less

Post Pruning ⇒ (1) more (more) accuracy ← (growing a (complete tree) full tree (decide)

(2) remove unnecessary branches based on a given threshold.

D.S ⇒ Computationally expensive

huge data ✗