Ensemble

→ group of musician

↓

In m/c learning

↓

group of models

Ensemble

→ Bagging

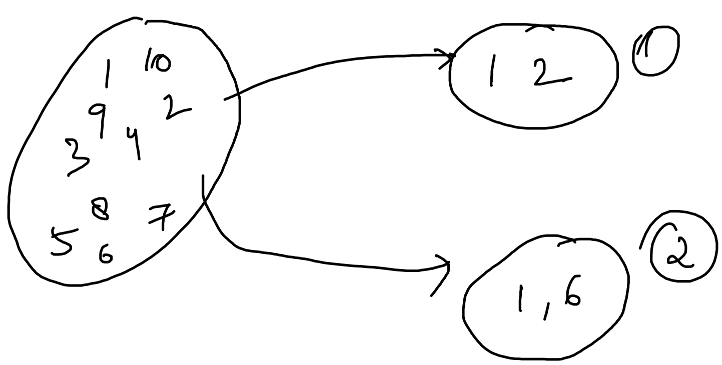
→ Boosting

BAGGING

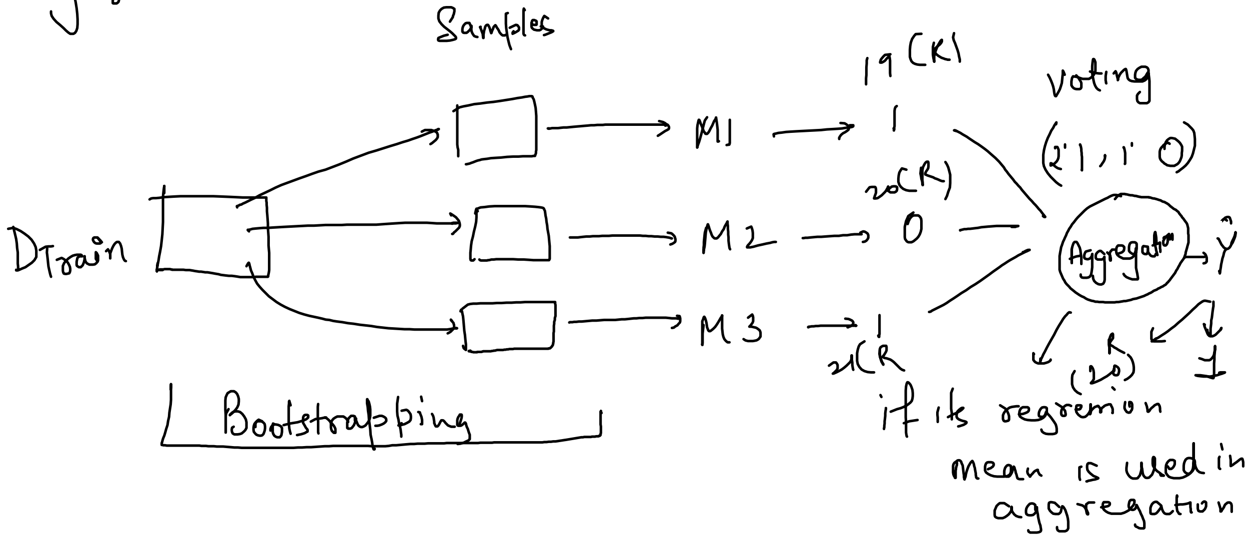
→ Bootstrapping

→ (sampling with replacement)

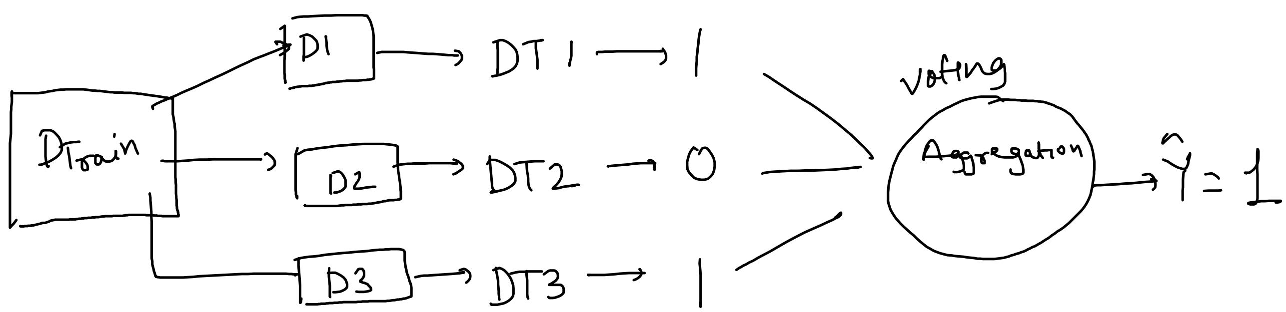
→ Aggregation



Bagging Algo:



Random forest:
→ collection of Decision Trees
→ DT should be of good depth (overfitting)



Models should be different from each other

	CGPA	IQ	EXTRA	SOCIAL	PLACED
(1)	7	110	10	9	1
(11)	8	112	9	8	0

II	8	112	9	8	0
III	9	118	8	7	0
IV	10	125	7	6	1

lets choose I & III : (50% of columns)

$\boxed{D_1} \rightarrow DT1$ $\qquad\qquad$ $\boxed{D_2} \rightarrow DT2$

CGPA	Extra	Placed
7	10	1
9	8	0

IQ	SOCIAL	PLACED
110	9	1
118	7	0

Different models ⟶ different samples ⟶ column sampling & row sampling

RF ⇒ $\underbrace{\text{low bias \& high variance}}_{\text{Overfitting}}$ DTs + Row sampling + column sampling

Output ← Aggregation

d'(#dimensions)

no. of d (dimensions) D_{Train} n (# rows)

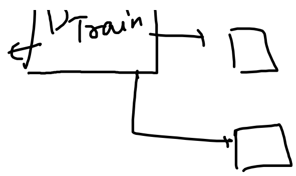
m (no. of r

$d > d'$

$m > n$

$m > n$

m(no. of rows)



Hyper parameters:

1) No. of models ↑ (n_estimators) = overfitting ↑ [100-2000]

2) max_features → ["auto", "sqrt", "lg", 0.7, 0.3]

3) row_sampling ⇒ $\frac{n}{m}$

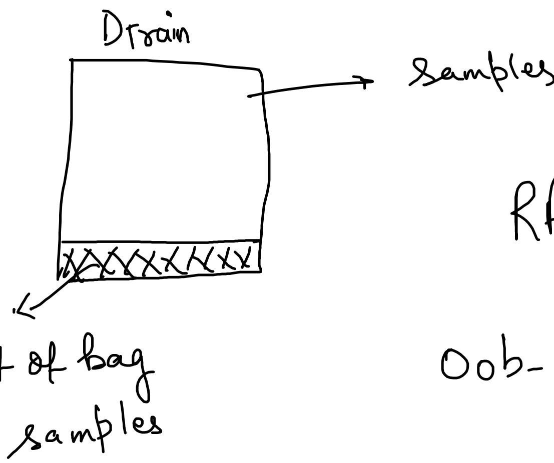
4) n_jobs ⇒ -1 ⇒ Capability of your CPU

5) max_depth

OOB Score

↓

out of bag



RF(oob_score = True)

oob_score ∝ accuracy

Advantages:

Disadvantages:

Advantages:

→ Feature Importances

Disadvantages:

→ Black box

Hyperparameters Tuning ↙ GSCV ↘ RSCV

Boosting

(I) Bagging → overfitting DTs

(II) Boosting → underfitting DTs → (high bias & low variance)

Flow Chart:

D_{Train} → M_0 [x_i, y_i] ($h_0(x)$) → predictions [$\hat{y}_i$] → errors [e_i] → M_1 [x_i, e_i] [$h_1(x)$] → pred → errors ↓ repeat till errors are min

($\hat{y}_i$)

0) $D_{Train} = [x_i, y_i]_{i=1}^{n} \Rightarrow M_0 \rightarrow$ predictions $(\hat{y}_i)$ $\rightarrow$ errors

$M_0 \xrightarrow{[h_0(x)]} [x_i, y_i]_{i=1}^{n}$

$e_1 = y_i - \hat{y}_i$

$e_i = y_i - h_0(x)$

2) $M1 \xrightarrow{h_1(x)} [x_i, e_i]_{i=1}^{n}$ $\quad e_i = y_i - h_0(x)$

Model at end of stage 1:

$$\underbrace{F_1(x)}_{\text{New predictions}} = \underbrace{\alpha_0 h_0(x)}_{\substack{\text{old prediction} \\ \text{(at stage 0)}}} + \alpha_1 h_1(x)$$

$e_i = y_i - F_1(x)$

2) $M_2 \xrightarrow{h_2(x)} \{x_i, e_i\}_{i=1}^{n}$ $\quad e_i = y_i - F_1(x)$

Model at end of stage 2,

$$\underbrace{F_2(x)}_{\text{new} \ldots} = \underbrace{\alpha_0 h_0(x) + \alpha_1 h_1(x)}_{} + \alpha_2 h_2(x)$$

prediction

$f_2(x) = f_1(x) + \alpha_2 h_2(x) \Rightarrow$ additive combination

new prediction — $f_2(x)$

old prediction — $f_1(x)$

current model — $h_2(x)$

$n\rangle$ $f_n(x) = \alpha_0 h_0(x) + \alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_n h_n(x)$

$$\boxed{f(x) = f_{n-1}(x) + \alpha_n h_n(x)} \rightarrow \text{final model}$$

or

$$\boxed{f_n(x) = \sum_{i=1}^{m} \alpha_i h_i(x)}$$

$n \Rightarrow$ # no of models $\Rightarrow$ hyperparameters

**** Residual & loss f^n:

$$L(y, f_n(x)) = [y_i - f_n(x)]^2$$

$$\frac{\partial L}{\partial f_n(x)} = \frac{\partial [y_i - f_n(x)]^2}{\partial f_n(x)} = -2\,[y_i - f_n(x)]$$

error

$\partial f_n(x)$ $\partial f_n(x)$

$$-\frac{\partial L}{\partial f_n(x)} = \overbrace{[y_i - f_n(x)]}^{\text{error}}$$

negative gradient
or
pseudo-residual

Gradient Boosting

I/P ⇒ $\langle x_i, y_i \rangle_{i=1}^{n}$ + differentiable loss f^n

0) $F_0 = \underset{\gamma}{\operatorname{argmin}} \left(\sum_{i=0}^{n} L(y_i, \gamma) \right) \longrightarrow \gamma = \bar{y}_i$

1) for $m=1$ to M ⇒ m ⇒ # models

$$r_m = - \left[\frac{\partial L(y_i, F_{m-1}(x))}{\partial f_{m-1}(x)} \right]$$

$$L = (y_i - f_1(x))^2$$

for $m=2$,

$$r_m = -\left[\frac{\partial L(y_i, f_1(x))}{\partial f_1(x)}\right]$$

1

2> $h_m(x)$ that can fit of r_m

↳ train h_m on $[x_i, r_{im}]_{i=1}^{n}$

$$F_2 = F_1(x) + \alpha_2 h_2(x)$$

3> $$\gamma_m = \underset{\gamma}{\text{argmin}} \left[L(y_i, F_{m-1}(x_i) + \gamma_m h_m(x)) \right]$$

4> $$\underbrace{F_m(x)}_{\text{New pred}} = \underbrace{F_{m-1}(x)}_{\text{old pred}} + \underbrace{\gamma_m h_m(x)}_{\text{Current model}}$$

Hyperparameters ⇒ M ⇒ # models↑ → overfitting↑

<u>shrinkage</u>: $F_m = \underbrace{F_{m-1} + \nu \, \gamma_m h_m}$

↳ learning rate

↳ $0 < \nu < 1$

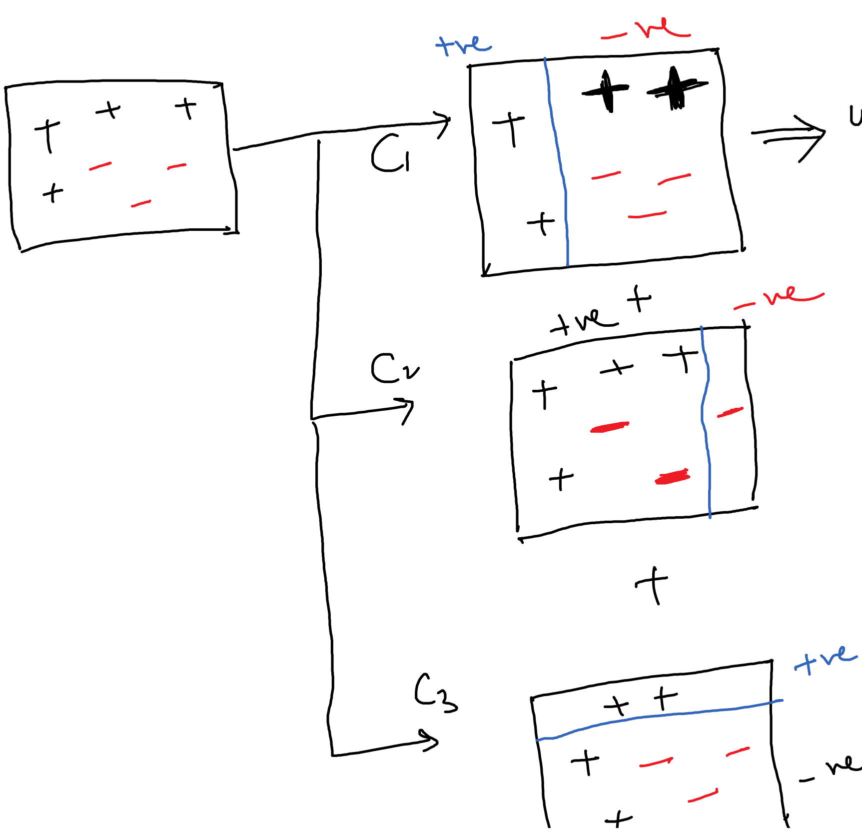
$\nu \rightarrow$ reduces $\gamma_m h_m \rightarrow$ to reduce overfitting

GBDT → models are DTs

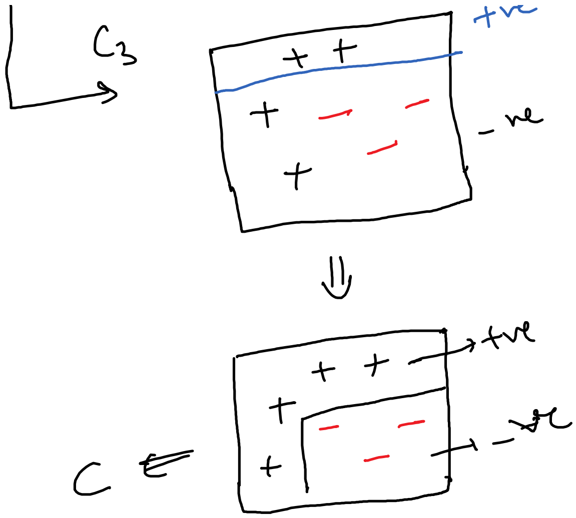
→ very slow → optimized → Taylor Series → XgBoost

!pip install XgBoost

Adaboost → Adaptive, Boosting



⇒ upsample misclassified points



$$C = r_1 C_1 + r_2 C_2 + r_3 C_3 + - - - -$$

	X_1	X_2	Y	$\hat{Y}$	weight $= 1/n$, $n = \#$rows
1)	3	9	1	1	$0.2 = 1/5$
2)	2	4	0	1 ✗	0.2 ✓
3)	1	5	1	0 ✗	0.2 ✓
4)	4	6	0	0	0.2
5)	5	7	0	0	0.2

α = error rate

error = algebric sum of weights of misclassified rows/points

error $= 0.2 + 0.2 = 0.4$

$$\alpha = \frac{1}{2} \ln\left(\frac{1 - \text{error}}{\text{error}}\right)$$

$$= \frac{1}{2} \ln\left(\frac{1 - 0.4}{0.4}\right) = 0.2$$

new weight for misclassified pts $= e^{+\alpha} \times$ old weight

$$= e^{0.2} \times 0.2 = 0.24$$

new weight of correctly classified points $= e^{-\alpha} \times$ old weight

$= e^{-0.2} \times 0.2$

$= 0.16$

	X_1	X_2	Y	$\hat{Y}$	Weights	New Weights	Normalized Weights	Range
1	3	9	1	1	0.2	0.16	$0.16/0.96 = \frac{1}{6} = 0.167$	0 — 0.167
2	2	4	0	1 ✗	0.2	0.24	$0.24/0.96 = 0.25$	0.167 – 0.417
3	1	5	1	0 ✗	0.2	0.24	$0.24/0.96 = 0.25$	0.417 – 0.667
4	4	6	0	0	0.2	0.16	$0.16/0.96 = 1/6 = 0.167$	0.667 – 0.834
5	5	7	0	0	0.2	0.16	$0.16/0.96 = 1/6 = 0.167$	0.834 – 1
					1	0.96	1	

0.167 — 0.500 — 0.667

Randomly any no b/w 0 & 1

0.1 0.4 0.5 0.6 0.7

1 2 3 3 4

New Dataset

1
2
3
3
4

upsampling ← [2, 3, 3] → misclassified rows in previous model