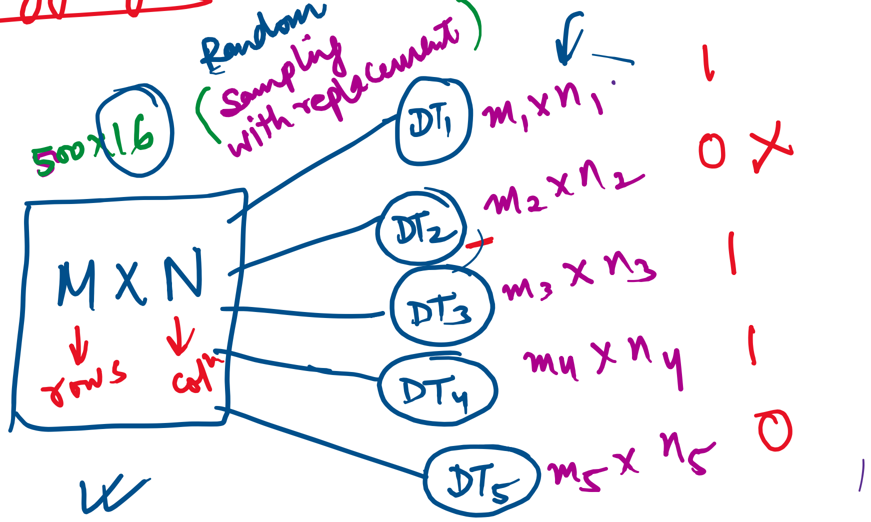
Ensemble Model.

D.T. (Collection/Aggregation)

- Bagging (bootstrap aggregation). (Parallel)
- Boosting (Sequentially).

16 →

① Bagging. (Random Forest).



$m_i \leq M$

$n_i \leq N$

$m_i \Rightarrow \left(\frac{2}{3}\right) M$

$n_i \Rightarrow \sqrt{N}$

Random forest.

→ forest is made of Trees & R.F. is made of D.T.

→ Ensemble model, Trained using Bagging technique.

→ <u>Adv.</u> – Combination of learning models.

DT: Searches for most imp. feature while splitting a node.

RF: Searches for most imp. feature among a random subset of feature.

⇓

(High diversity)

Most imp. features of DT is not included (by chance) in DT_3, then also the split will happen. (based on best available feature).

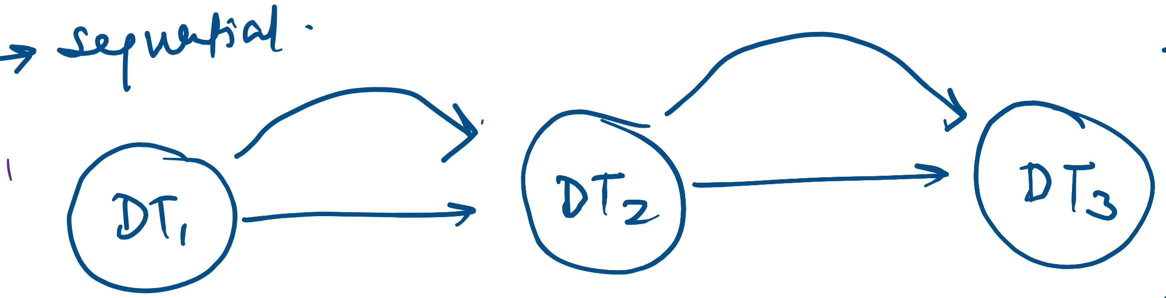
② Boosting.

→ sequential.

1000

→ 9

→ 7



weak learners :

Strong learners :

			y	y_pr
			0	0
			1	0
			1	1
			0	1
			0	0
			1	1

Step 1 : Initialize equal weights.

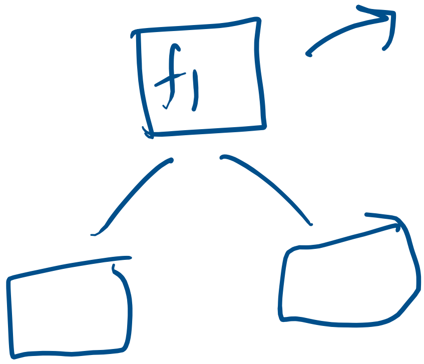
	x_1	x_2	x_3	x_4	x_5	y	o/p	weights
1							yes	$\frac{1}{10}$
2							no	$\frac{1}{10}$
3							yes	
4							no	
5							yes	
6							yes	
7							no	
8							yes	
9								
10								$\frac{1}{10}$

$\rightarrow \frac{1}{10}e^{-x}$

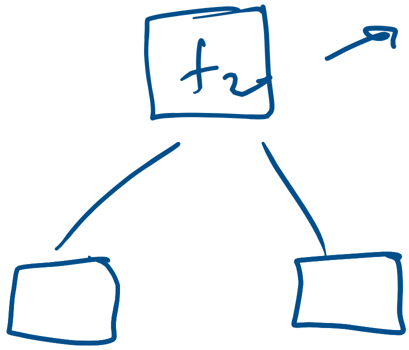
$\frac{1}{10}e^{x}$

$$\frac{1}{n} = \frac{1}{10}$$

Step 2 : create Base learners in sequential fashion.



→ Stump



→ Stump.

Lesser Entropy : High info gain.

9 1

e_t = sum of all the sample weights of that error row $= \frac{1}{10}$

Step 3: Performance of stump.

$$= \frac{1}{2} \log_e \left(\frac{1 - e_t}{e_t} \right)$$

$$= x$$

Step 4:

$\uparrow$ weights of incorrect classification (weak learners)

$\downarrow$ weight of correct classified record (strong learners)

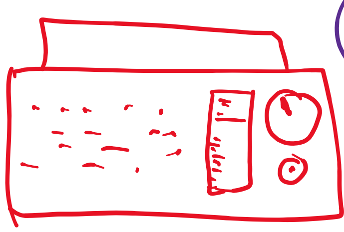
New sample weight (of incorrect classified row) $=$ old weight $\times\, e^{\text{Perf. stump}}$

$$= \frac{1}{10} \times e^{x}$$

New sample wt. (strong lear) $=$ old wt. $\times\, e^{-\text{Perf. stum}}$

$$= \frac{1}{10} \times e^{-x}$$

(Hyperparameter Tuning) in RF.



max_depth

4

2
3
4
5

(*) Increasing the Predictive Power.

1. n_estimators

   → More Trees

   ✓ ↑ Perf.

   ✓ ↓ Computational Power

   ✓ ↓ speed.

2. max_features :
3. min_sample_leaf :
4. Max_depth.

2. Model's speed.

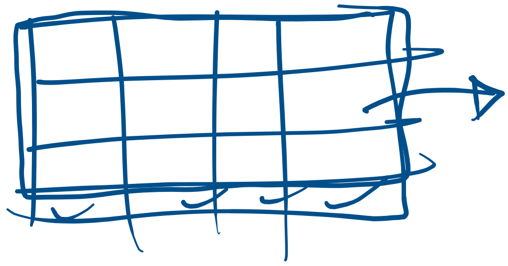
1. n_jobs :

$\boxed{1}$ → one Processor

$\boxed{-1}$ → no limit.

② oob_score

out of bag :-



60%

70%

30

20 Tr

10 Te

oob_error_rate.

10 — 2 → e_k

Grid Search CV :-

→ member of model_selection.

Max_depth = 3, 4, 5, 8, 10

= 10, 12, 15, 18

= 3, 8, 9

→ Fits to the hyperparameters & fit the estimator on Traing set.

→ you can select the best Parameters from the listed hyperparameters.

500 → G.S.CV →

W.K. — 50

1B — 300

1Bowler — 100

I A.R. — SD → (A,B,C,D)

Why Needed?

→ Manual selection of Parameters (Hyper) are Complex & Very time Consuming

→ hard to keep track of hyperparameters that we've tried & still have to try.

Micro and Macro Average for Multiclass Classification:

$$P_A = \frac{TP}{(TP+FP)}$$

Cls A : 1 TP 1 FP

Cls B : 10 TP 90 FP

Cls C : 1 TP 1 FP

Cls D : 1 TP 1 FP

$$P_A = \frac{1}{1+1} = 0.5$$

$$\frac{10}{100} = 0.1$$

0.5

0.5

$$\text{Micro Avg} = \frac{1+10+1+1}{2+100+2+2} = \square$$

$$\text{Macro Avg.} = \frac{0.5+0.1+0.5+0.5}{4} =$$

4

Capstone:

Mandatory
EDA.

Dataset
+ Problem statement

PPT
Ans

2-3

Try to
explain
to
[illegible]

EOF >>