Bagging

→ One of the technique called → RF

- Ensemble method
  - (1) → Bagging (RF) # Base model Decision Tree
  - (2) → Boosting (AB, GB, XB, CB) → {Additive}
    - ↑ Ada, ↑ Gradient, ↑ Cat Boosting
  - (3) → Stacking — Stacking (One top of other)
  - (4) → Cascading (Both (Bag, Bag))

- **Bagging** ⇒ In a parallel way we are creating a model
  - ↳ (Homogenous model → (similar model we are going to use)
- **Boosting** ⇒ In a sequential way we are create a model
  - ↳ (Homogenous model) → (" Above reasoning) ↑

Parameter :- n_estimator ⇒ The number of trees. (or) Number of DT models

- Random forest Regressor ⇒ criterion = "squared error" (default), "Absolute error", "Poisson", "friedman_mse"
- Random forest classifier ⇒ criterion = "gini"

Task # Dataset :- U.S. Healthcare Data (Kaggle)

(1) collect data
(2) Store somewhere database.
  - ↳ Cacendra (or)
  - ↳ mongoDB (or)
  - ↳ MySQL
(3) fetch data
(4) EDA
(5) model building

Bagging ⇒ (Parallel)

↳ we are using Homogeneous model (similar)

(1) LR, (2) LR (3) SVM, (4) KNN, (5) DT ← RF

Data ⟶ Sample of Data { Bootstrap (with Replacement) ↳ then Aggregation (voting)
→ Pasting (without Replacement)

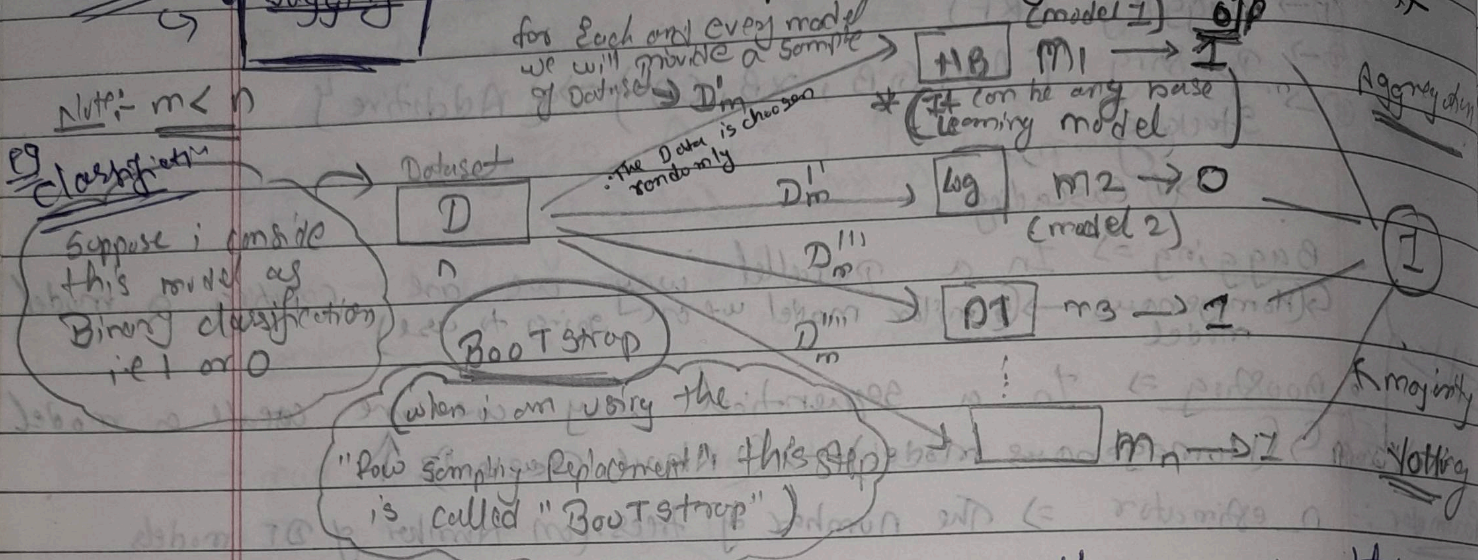
sample feature — sample row

Bagging => * In Regression => Avg (o/p of the model ...)
(value is in continuous) mean

~~Random Forest~~

You Tube

• **Bagging** is also called → Boot Strap Aggregation
Row Sampling with Replacement



→ Every model we are providing our data will be using ~~this~~ ~~PE~~ "Row sampling with Replacement" this techniques.

• There is No Relationship between M1, M2, M3 ... M3 in Bagging

Random Forest

⇒ Basic Principals of Random forest

→ It develops lots of Decision Tree based on random selection of data and random selection of Variables (is called as Bootstrap)

{ → As the trees are based on random selection of data as well as variables, these are random tree
→ Many such random trees leads to a random forest }

* Bootstrap | Pasting | oob ← out of Box sampling

obise/not obise

(weight) x_1	(Height) x_2	o/p (0/No) y
45	170	0
45.5	170.2	0
46	180	NO
47	185	0
47.2	140	0
48	140.2	NO
48.5	150	0
48.9	160.5	0
50	160.8	No
60	150	No
65	155	0
69	156	0
67	162	0
71	189	No
75	190	0

(1) ... (15) — D ← Dataset

extending Dataset $\boxed{x_3 \; x_4 \; x_5}$

Bootstrap ← It is just Data selection

→ Row sampling and colum sampling (Randomly we are selecting row + column)

→ with Replacement (Repeatition) is allowed

Creating 3-model RF (DT) ← Base model

• This is called Bootstrap

*** Pasting** ⇒ we are not going to do repetation data (without repetation)

D_1	D_2	D_3
(1 to 5)	(6 to 10)	(11 to 15)

← without repetation

This is my 1st DT

sample →

(1) D'	(2) D''	(3) D'''
(RS + CS) (row) (clm)	(RS + CS) (row) (clm)	(RS + CS) (row) (clm)
1 $x_1 x_2 x_4$	1 $x_2 x_5 x_3$	8 $x_2 x_4 x_1$
2	2	9
3	5	15
4	6	11
9	7	12
10	10	14
13	14	7
↓ DT1	↓ DT2	↓ DT3

This data is Oob

*** Oob** ⇒ The rows that we are not selecting in D' that is called Oob data

{5, 6, 7, 8, 11, 12, 14, 15}

* Aggregation
↳ (together), (voting), (majority class)

Considering 3 model

M1 $\xrightarrow{\text{predict}}$ Obese
M2 ⟶ Obese
M3 ⟶ No obese

which class i will select ?
⟶ obese
voting, majority, together

finding pro ⟶

Hard classifier = 2 obese $\xrightarrow{\text{pro}}$ $pr = \frac{2}{3}$
Soft classifier = 1 No obese $\xrightarrow{\text{pro}}$ $pr = \frac{1}{3}$

* Extra Tree classifier

↳ More Randomization

- we are not going to check Each and Every Probability in a Sequence w.r.t target variable.

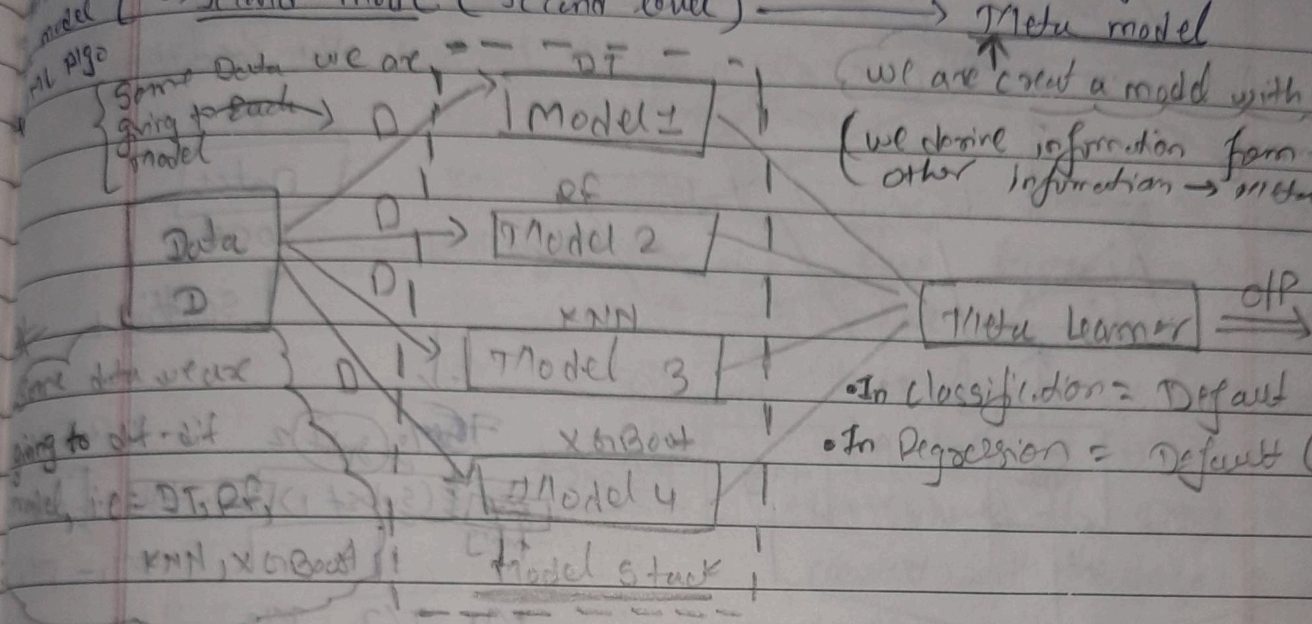
• Here, Randomly we are taking a Variable and checking Gini-Impurity and based on that we are going to define root Variable or child variable.

* ~~B~~ **Stacking** ~~#~~ we are using **Heterogeneous** model inside [Stacking
↑ different - different model (we are going to use)

→ Stacking of the model
(It is working in 2 mode
we can use diff mode → SVM, KNN, NN, DT, LoR, LinR

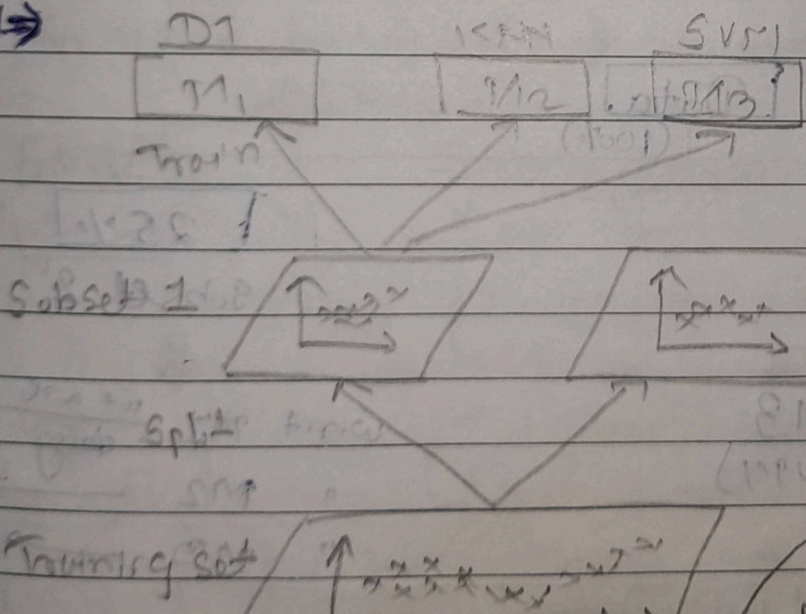
→ first mode (first level) $\xrightarrow{\text{This is called}}$ Stack of model
→ Second mode (second level) ⟶ Meta model

we are creat a model with metadata
(we derive information from the other information → metadata)



• In classification = Default (LogisticR)
• In Regression = Default (Linear R)

first level ⇒



Split 100%
75% → (Subset 1)
25% → (Subset 2)

(Subset 1) 3 model we are going to consider and train
m1 m2 m3

for Now consider subset 1

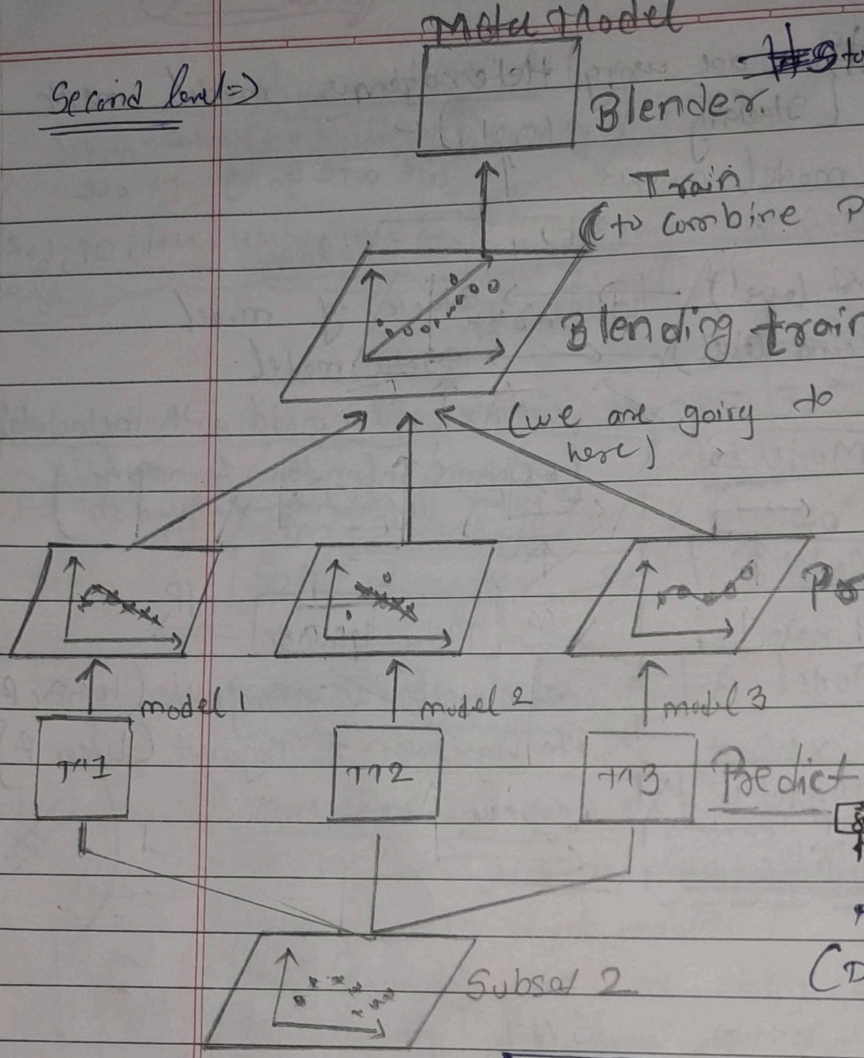
↑ This is called level "0"
(or) we are going to do stack of model

23rd April Live class - Imp

(from Data itself Prediction)

Stacking means diff - diff model

Second level =>



Data
100%
Splitting 1

75% (split 2)
80% training (subset 1) 20% test
model
M1 M2 M3
(DT) (SVM) (KNN)

25%
subset 2
M1 - prediction
M2 - prediction
M3 - prediction

①

Data
(100%)
(Split)

75%
Subset 1

M1 (DT)
M2 (KNN)
M3 (SVM)

25%
Subset 2 (we are doing Prediction)

w.r.t M1 —we are doing→ Prediction
" M2 → Prediction
" M3 →

Second level ②

100% Data

75% ← (Now this is my 100% data)
25%

80% — Train → M1, → M2, → M3
20% (prediction w.r.t M1, M2, M3)

Note • Stacking is also called → Blender

→ Blending it comes inside Stacking

Boosting

{ Additive learning (continues learning) (Sequential) learning
{ ~~my next model will be learning from previous~~ model

- AdaBoostClassifier (Adaptive Boosting)
- GradientBoosting (GBDT) (Additive learning)
- XGBoost (Extreme gradient boosting)

* AdaBoost ((YouTube Krish Sir)

wrong records

f_1 f_2 f_3 (features)	o/p	$w = \frac{1}{7}$ Sample weight
	yes	
	NO	1/7
	Yes	1/7
	NO	1/7
		1/7
		1/7
		1/7
		1/7

$w = \frac{1}{7}$

remaining records

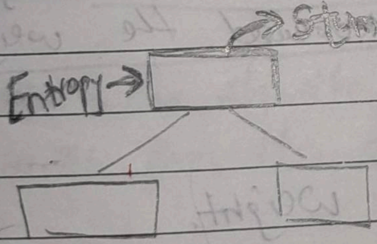
Step 1 { In Step 1 all records will get some sample weights

Step 2 :- we create our first base learner, which we need to create sequentially, so our 1st base learner, we will create with the help of DT in adaBoost.

- Over here the DT is just created with the help of only one depth.

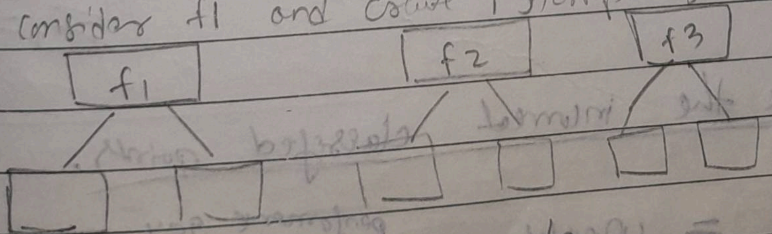
continued

This DT is basically called as Stumps [we are not creating a full depth i.e. max. depth



{ DT is just having one depth with one leaf Node (or) may having many no. of leaf nodes these are basically called as Stumps.

we consider f1 and create 1 stump similarly f2, f3.



Suppose 4 Correctly classified 4 records

1 Incorrectly classified 1 record

{ based on o/p

{ for this Incorrect classification we need to find out total error

{ Entropy and Gini concept for selecting. Compare the Entropy of Stump 1, 2, 3. whichever will be having the lesser selecting that DT as my base learning model.

$$\boxed{\text{Total Error} = \frac{1}{7}}$$

• **Step 3:-** we try to find out the performance of the stump i.e how the stump is basically classified.

$$\boxed{\therefore \text{Performance of Stump} = \frac{1}{2} \log_e\left(\frac{1-TE}{TE}\right)}$$

$$= \frac{1}{2} \log_e\left[\frac{1-1/7}{1/7}\right]$$

$$= \frac{1}{2} \log_e[6]$$

$$= 0.896$$

Note:- The reason for calculating Total error and Performance Stump is because, we need to update this weights.

→ In Boosting technique, only the wrong records from this particular DT one or the stump one will be passed to the next DT or next stump.

→ for that, I have to ↑ increase the weights of the wrong classified records and ↓ decreased the weights for the correctly classified records

• **Step 4:-** we need to update the weight.

→ Based on the performance of stump we are going to update the weights.

→ 1st we update the incorrect classified points.

$$\text{New sample weight} = \text{weight} \times e^{\text{performance say}}$$

$$= \frac{1}{7} \times e^{0.895}$$

$$= 0.349$$ (Now my weight is 0.349 which is basically increased ↑)

→ 2nd ~~for~~ updating the correctly classified points

New sample weight = weight $\times e^{-\text{performance say}}$

$$= \frac{1}{7} \times e^{-0.895}$$

$$= 0.05$$

Based Normalized weight we will try to divide them into Bucket

f1 f2 f3	O/P	Sample weight	Updated weight	Normalized weight	Create Bucket
	Yes	1/7 ↓	0.05	0.05/0.68 ⇒ 0.07	0 to 0.07
	No	1/7 ↓	0.05	0.05/0.68 ⇒ 0.07	0.07 to 0.07 (add) = 0.14
incorrect classified	Yes	1/7 ↑	0.349	0.349/0.68 ⇒ 0.513	0.14 + 0.51 (add) = 0.65
	No	1/7 ↓	0.05	0.07	.
	Yes	1/7 ↓	0.05	0.07	:
	Yes	1/7 ↓	0.05	0.07	:
	No	1/7 ↓	0.05	0.07	
			$\Sigma = 0.68$	$\Sigma = 1$	

⇒ Now our algorithm will run 7 itteration to select diff-diff records from this particular hidden ~~dat~~ data set.

↳ Installation of XGBoost

Gradient Boosting (Additive Learning)

Data →

Height	Gender	weight
160	M	75
170	M	60
180	F	65
190	F	78
195	M	80

$\frac{75+60+65+78+80}{5}$

$= 70$

$\boxed{Res2 = y - (Avg - Res1)}$

$Res3 = y - (Avg + Res1 + Res2)$

Avg ← Base Model

Both can be solved using Gradient Boosting
- Regression :- target column is Numerical
- Classification :- target colum is categorical

- **Last Result of Gradient Boosting :**

↳ first we have Base model then learning Rate into Decision Tree then...

Base model + η × DT1 + η × DT2 + η × DT3 + ... N

These entire DT we are going to train over Residual value

→ So, we are going to arrange all the, ~~weak learner~~ week learner and we are going ~~make a~~ → strong learner, this is is called a Additive learning

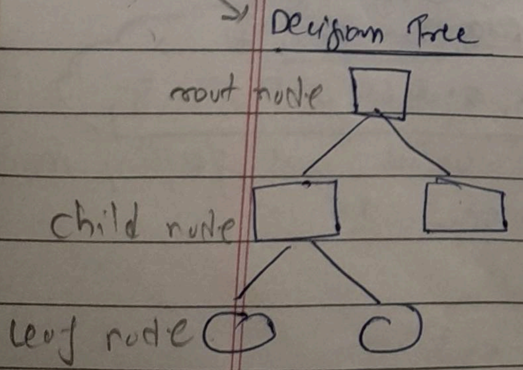
1st we have to create a base model

- **Base Model :-**

↳ Initilize our data with some constant ~~static~~ Value

- This is full depth DT

Decision Tree



→ • At initial point we are creating only 1 leaf node

leaf → ◯ ← having some value

→ • Initially we are trying to take Average value of our Dataset

→ 1st we initialize Average Value
→ Then we will find Residual w.r.t Avg value

YouTube =>

Exp	Degree	(y) Salary	$\hat{y}$	$(y-\hat{y})$ R_1	R_2	....R_n
2	BE	50K	75	-25	-23	(very very less R value than I shall stop)
3	Masters	70K	75	-5	-3	
5	Masters	80K	75	5	3	
6	PHD	100K	75	25	23	

R → Residual

→ Steps in DT

Step I:- Compute the base model, which will give me I o/p i.e Average of Salary.

$$\frac{50+70+80+100}{4} \approx 75 \text{ (This is my 1st base model)}$$

Step 2:- Compute Residuals, i.e (Errors, Pseudo Residual)
• Log function $(y-\hat{y})$

Step3:- After base model, i will add one DT sequentially, Construct a DT.
⇒ I/p → Exp and Degree (x_i)
⇒ O/p → R1 (Here o/p will not be target variable it will R_1)

$\{\overset{I/P}{x_i}, \overset{o/p}{R_1}\}$ $\boxed{\alpha = 0 \text{ to } 1}$

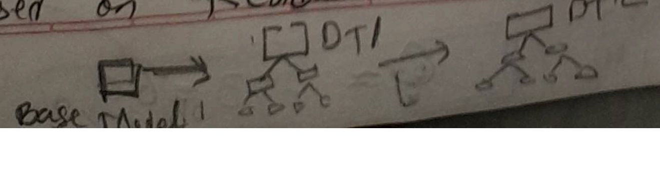
Learning Rate

$$f(x) = \underbrace{h_0(x)}_{\text{Base model}} + \underbrace{\alpha_1 h_1(x)}_{DT1} + \underbrace{\alpha_2 h_2(x)}_{DT2} + \dots + \underbrace{\alpha_n (h_n x)}_{DTN}$$

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

Note:-

→ {Here we are adding after one base model, (creating sequential DT based on Residuals that we are getting.)

⇒ 

Gradient Boosting Maths

Exp	Degree	Salary (y)	$\hat{y}$	r_{im}
2	BE	50K	60	-10
3	PHD	70K	60	10
4	Masters	60K	60	0

- Initial requirement for this ML algorithm
- → 1st we need to provide I/P
  $\{x_i, y_i\}$ — $\underbrace{x_i}_{\text{Independent}}$ → Exp, Degree and $\underbrace{y_i}_{\text{Dependent}}$ → Salary
- → 2nd thing I have to provide in I/P is my loss function in con different for both i.e Regression and classification.

  Actual value; should be differentiable

  $L(y, F(x))$ — predicted value

- → 3rd → No. of Tree

\# wikipedia

Pseudo Algo ⇒ (1) Initialize model with Constant Value

$$f_0(x) = \underset{\gamma\,(\text{Gamma})}{\arg\min} \left(\sum_{i=1}^{n} \underset{(\text{Loss})}{L}(y, \gamma) \right)$$

Gamma (It is predicted value)

arg min of this Gama value

$(y-\hat{y})$ ← Residual

This me

$$\text{Loss} = \sum_{i=1}^{n} \frac{1}{2}(y-\hat{y})^2 \quad \downarrow \text{ minimize the loss}$$

$$= \frac{1}{2}(50-\hat{y})^2 + \frac{1}{2}(70-\hat{y})^2 + \frac{1}{2}(60-\hat{y})^2$$

⇓ To minimize the loss, doing 1st order Derivatives ⇒ $x^n = nx^{n-1}$

$$= \frac{2}{2}(50-\hat{y})(-1) + \frac{2}{2}(70-\hat{y})(-1) + \frac{2}{2}(60-\hat{y})(-1)$$

$$= -50 + \hat{y} - 70 + \hat{y} - 60 + \hat{y}$$

$$= -180 + 3\hat{y}$$

$$= 3\hat{y} = 180 \qquad \boxed{\therefore \hat{y} = 180/3 = 60}$$

→ $\hat{y}$ for the base model

B

Co

+

② Iterate

$m = 1$ to M

$\left\{ \begin{array}{l} m = \text{no. of D.trees} \\ M = 2,3,4,5,6,7,8 \end{array} \right.$

(i) → Compute pseudo residual

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]$$

Actual (y), predicted ($\hat{y}$)

$y - \hat{y}$

$\left\{ \begin{array}{l} r_{11} = 50 - 60 = -10 \\ r_{12} = 70 - 60 = 10 \\ r_{13} = 60 - 60 = 0 \end{array} \right.$

for $i = 1, \dots, n$

$f(x) = f_{m-1}(x)$

(ii) → fit the Base Learner $h_m(x)$

i/p = $\{(x_i, r_{im})\}$ Now ⇒ Independent feature = Exp, Degree

Dependent feature = r_{im}

(3) → $$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + \gamma)$$

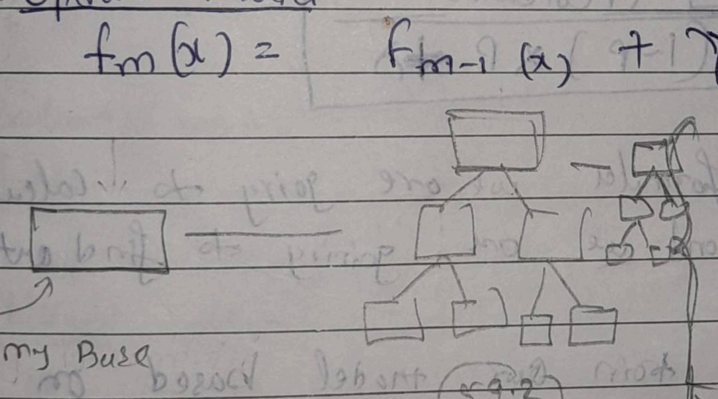
Previous model o/p $\approx \hat{y}$

$$\sum_{i=1}^{n} \frac{1}{2} (y_i - (60 - \hat{y}))^2 \Leftarrow$$ { Again i have to minimize this, Similar to Step 1 }

(4) → Update Model

learning rate (0 to 1)

$$f_m(x) = f_{m-1}(x) + \cancel{\gamma_m h_m(x)}\ \alpha(h(x))$$



This is Initially my Base model

when i give my 1st record to my Base model and 1st Additive DT getting → 59.2

Based on my Base model residuals and the I/P value i create a DT. If we pass the 1st record to the base model the o/p will be 60.

r o/p

① $= 60 + (-10) \leftarrow \alpha = 0.1$

DT $= 60 - 0.1$

$= 59.2$

α

{ the Difference is 9 (50 − 59)

now, we will try to add one more model i.e (DT) additive

② $60 + (0.1)(10)$

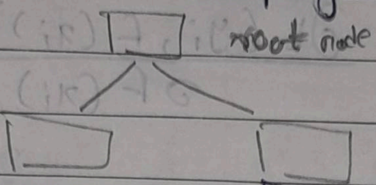
DT $= 61$

* XGBoosting ← Doing Randomization w.r.t Data

In XGBoost there will be 1st Base model, then DT₁, then DT2....

$$\text{Base model} + \eta DT_1 + \eta DT_2 + \ldots\ldots + \eta DT_n$$

- for creating DT we have Specific Method



→ entrop, IG

⇒ Here, we are going to calculate $\boxed{\text{Similarity Score}}$

Base on this we are going to calculate Information Gain

- Regression = $\boxed{\dfrac{\sum (\text{Residual})^2}{\text{No. of total residual} + \lambda}}$

- classification =

$$\boxed{\dfrac{\sum (\text{Residual})^2}{\sum (1-P) \times P + \lambda}}$$

P ← Probability

→ Based on these formula we are going to calculate Similarity Score, and we are going to find out Gain

→ we are going to train our model based on similarity Score and Gain

↓ Based on this

"we are going to do Pruning" of this tree"