1

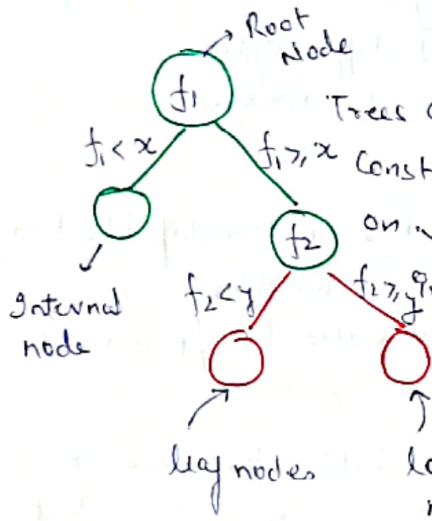
Ensemble Techniques (Classification + Regression)

(we ensemble or club various models together to make a prediction)

Bagging
↓
Random forest
(Decision Tree)

Boosting
- Ada boost
- gradient boost
- XG boost

Decision Tree



Trees are being constructed based on various features in your data

Lets data

f_1	f_2	f_3	O/P (binary)
			Y
			N

Decision Tree
- classification trees - Qualitative data | Categorical data
- Regression trees → Quantitative data | Continuous data

✓ There are very popular methods owing to interoperability of this technique across various divisions.

Simple decision trees are preferred over complex decision trees to prevent overfitting.

* Multiple way decision tree can be better sometimes than a binary decision tree because it increases the gain in information manifolds.
* As the number of splits increase in decision trees their complexity rises - Pruning.

In order to prevent overfitting in decision trees we adopt a method called pruning which actually curtails the branches of the tree that fit data too specifically.

Pruning
- Pre-Pruning - stop growing DT branches when information become unreliable
- Post Pruning: first you grow a full-flegged DT and then start pruning irrelevant branches.

Decision Tree Algorithms

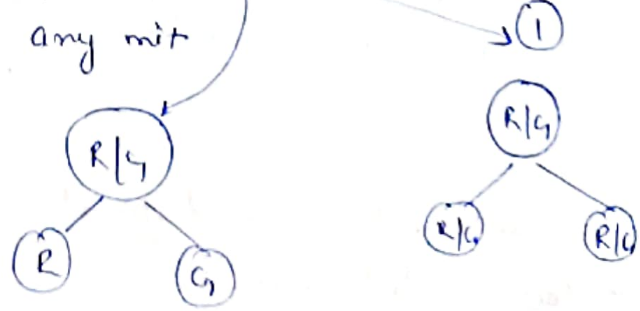
CHAID
(Chi-squared Automatic Interaction Detection)

- Regression:- f-test → target means/nodes/leafs.
  - ↳ if significant diff:- split
  - ↳ if not : merge
- Classification:- Chi square test → relationship b/w two variables
  - most relatable variable to o/p - node of split
  - Merge those variables that are not very significantly related to the output

Short comings:-
1. Very horizontal DTs
2. does not handle missing values handled as a different class.
3. no pruning option available

CART
(binary classification/ Regression Trees)

- classification:- Gini Impurity - purity of split
  - (0-1)
  - 0 → split does not have any mix
  - 1 → the split has mix of both classes in each node



- Regression
  - capture more information ← Least Square Deviation (LSD)
  - Variance reduction, minimizes Sum of sq distances, deviations
  - Predicted - observed, sq residual ↓

CART produces a sequence of DTs, each of which is a candidate for "optimal tree". This optimal tree is identified by evaluating the performance of every tree through testing or performing cross-validation ③

does not use any internal performance measure

ID3

(used for classification tasks, not very effective with regressions)

Information Gain : decrease in level of randomness in a set of data

favouritism towards attributes with more outcome

how much information, a feature gives us about a class. attribute with highest information gain will split first.

Entropy : A concept to which information gain is directly related to.

(0 - 1)

0 → perfectly predictable

1 → perfect randomness unpredictable

$$\text{Information gain} \propto \frac{1}{\text{Entropy}}$$

C4.5 → Improvement over ID3

(Regression + classification Trees)

Gain Ratio : attribute with maximum gain ratio is selected as the splitting attribute

reduce bias in DT with huge amount of branches by taking into considerations number & size of branches while choosing an attribute

Pruning + technique $\boxed{\text{Windowing}}$ → DT is trained first on a batch of data from training data and rest of the cases are used as a test measure for this DT. If it fits well process stops. Save memory & computations.

Limitations of Decision Trees

1. High Variance - small change in data can result in various sets of splits.
2. High Bias :- If some classes dominate over others. (Problem with Unbalanced datasets)
3. Greedy :- locally optimal rather than globally optimal
4. Regression :- boundaries.

All of these problems of decision trees can be resolved by ensemble techniques
- Bagging
- Boosting

Mathematics behind DTs

Chi-square test: Hypothesis test when one wants to determine the relationship b/w two categorical variables.

Gender	Preffered NP	frequency of TV	Highest education
1. M	1. Washington P.	1. daily	1. without grad
2. F	2. Hindu	2. several times/week	2. college
	3. USA today	3. more rarely	3. bechalor's

Is there a relationship b/w gender & prefered NP. | freqn & highest education
or a correlation
↳ chi-square test

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

chi square value

$O_i \rightarrow$ observed value

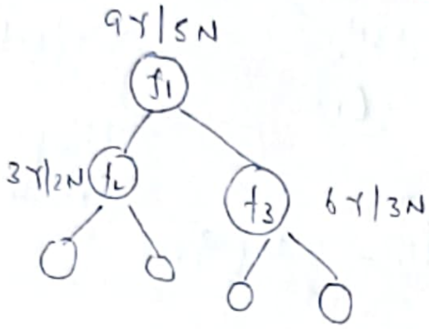
$E_i \rightarrow$ expected value

from table

if chi square value < Critical chi square value
null hypothesis is not rejected

Hence there is no relationship b/w these variables.

Decision Tree Entropy : Measures the purity of the split



$$H(S) = -P_{(+)} \log_2 (P_+) - P_{(-)} \log_2 (P_{(-)})$$

$P_+ | P_-$ = % of +ve class / % of -ve class

$$H(S) = -3/5 \log_2 3/5 - 2/5 \log_2 (2/5)$$

$$= 0.78 \text{ bits.}$$

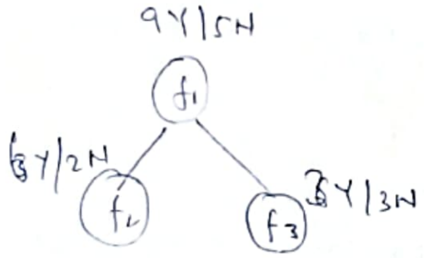
Entropy 0-1   1 - worst split   0 → best split

Decision Tree Information Gain

$$\text{Gain}(S, A) = H(S) - \sum_{v \in val} \frac{|S_v|}{|S|} H(S_v)$$

$|S_v|$ → subset after splitting

$H(S_v)$ → subset entropy



$$H(S) = H(f_1) = -\frac{9}{14} \log_2 9/14 - 5/14 \log_2 (5/14)$$

$$= 0.94$$

$$\boxed{H(f_2) = 0.81 \quad H(f_3) = 1} \; H(S_v)$$

$$\text{Gain}(S, f_1) = H(S) - \frac{8}{14} H(f_2) - \frac{6}{14} H(f_3)$$

$$= 0.049.$$

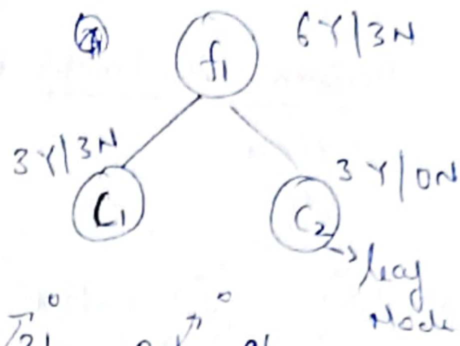
Gini Impurity

$$GI = 1 - \sum_{i=1}^{n} (P)^2$$

$$= 1 - [(P_+)^2 + (P_-)^2]$$

Entropy $H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$ ⑥

f_1	f_2	f_3	o/p
c_1	d_1		Y
c_2	d_2		N



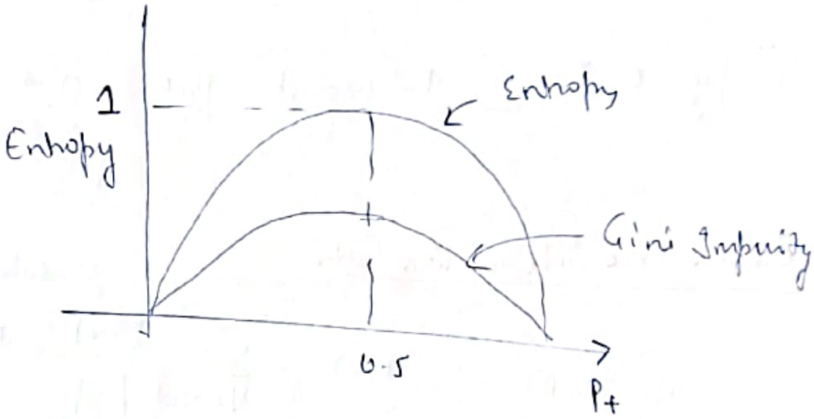
$H(C_2) = -\frac{3}{3} \log_2 3/3 - \frac{0}{3} \log_2 0/3$

$= 0$ (Pure split)

$H(C_1) = -3/6 \log_2 3/6 - 3/6 \log_2 3/6$

$= 1$

~~$H(C_2) = \frac{3}{3} \log$~~



$GI(C_1) = 1 - [(1/2)^2 + (1/2)^2]$

$= 1/2$

Gini impurity is taken as a parameter by random forest & XG boost because it is computationally efficient.

How decision tree split Numerical Variable

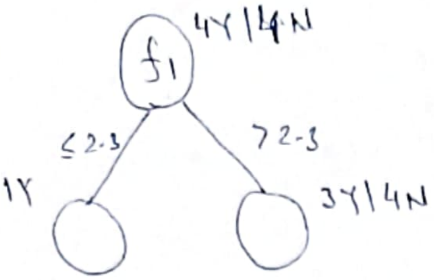
f_1	o/p
2.3	Y
3.6	N
4	'
5.2	/
6.7	/
8.9	/
10.5	

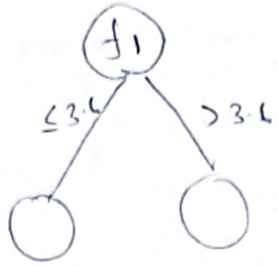
① Sort all the values

② will consider some threshold values lets say 2.3

$x_i \le 2.3$ ↳ branch 1

$x_i > 2.3$ ↳ branch 2





like this we go with each and every feature

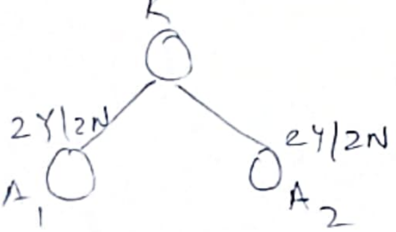
We will calculate the best entropy and Information gains.

* Disadvantage is time complexity for millions of rows in performing this operations

Gain Ratio

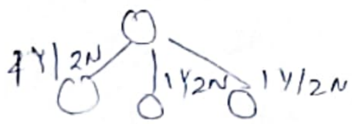
$$\text{Split Entropy}(R, A) = -\sum_{i=1}^{p} \frac{|R_i|}{|R|} \log\left(\frac{|R_i|}{|R|}\right)$$

Split entropy increases with the number of divisions increases



$$= -\frac{1}{2}\log \frac{1}{2} - \frac{1}{2}\log \frac{1}{2}$$

$$= 1$$



$$= -\frac{1}{3}\log \frac{1}{3} - \frac{1}{3}\log \frac{1}{3} - \frac{1}{3}\log_2 \frac{1}{3}$$

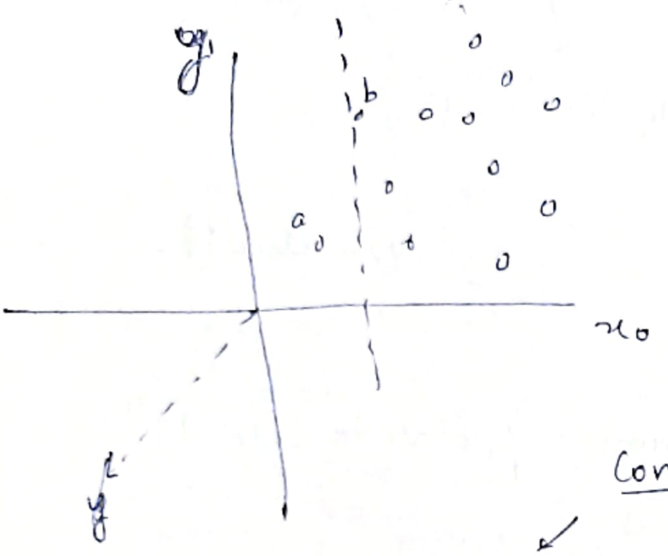
$$= \log_2 3 = 1.6.$$

$$\text{gain ratio} = \frac{\text{Gain}(R, A)}{\text{Split}(R, A)}$$

→ Prevent the number of Branches.

Its same as MNP with loss f^n as MAE | MSE.

Decision Tree Regression | Random forest Regression



$x_0 \leq 1$

a, b → $x_0 \leq -7$

rest → $x_0 \leq -12$

Condition

If this condition is true we move left and if this condition is false we move right

So like this we keep splitting the data until we achieve leaf nodes. It is a sort of clustering if one sees.

* Now we have understood the intuition behind decision tree algorithm. Most important bit is to find the condition for splitting at the node

Condition 1 → Split 1, Split 2

Condition 2 → Split 1, Split 2

Which of these conditions is better? Condition 1 or Condition 2

Ans- Variance Reduction

In case of regression variance reduction works best like in case of classification we have gini coefficient

$$Var = \frac{1}{n} \sum (y_i - \hat{y})^2$$

Higher value of variance will mean a poor classification.

$$(Var)_{root} > w_1 (Var)_{split1} + w_2 (Var)_{split2}$$

$$(Var)_{red^n} = (Var)_{parent} - \sum w_i \, Var(child\,i).$$

$$w_1 = \frac{n_{split1}}{\text{Total points}} \text{ (number of points in split 1)}$$

Various split algorithms

- ID3 - C4.5 - CHAID - MARS

Random forest Regressor (Bagging Technique)

Bagging ??

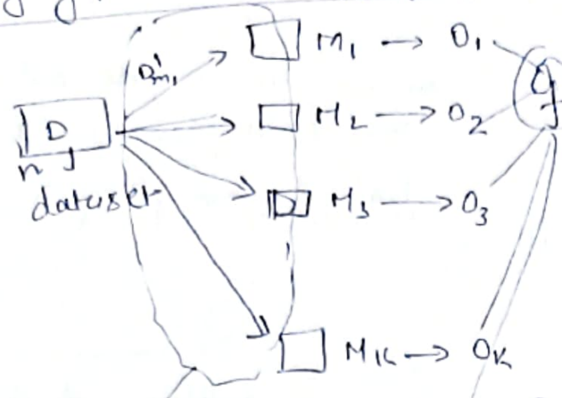
Ensemble Techniques (combining multiple models)

Bagging (Bootstopf Aggregation)

Boosting

Random Forest

Bagging | Bootstrap Aggregation



We will create multiple models from a big dataset D and to each model we will provide just a sample of datatset D i.e D'

Row Sampling with replacement] We will again sample the data and provide it to the another model.

* $O_1, O_2, O_3, \ldots O_K$ are outputs by different model.
* We will combine all the outputs and provide the result for test data.

m_1 will get trained on D'_{m_1}

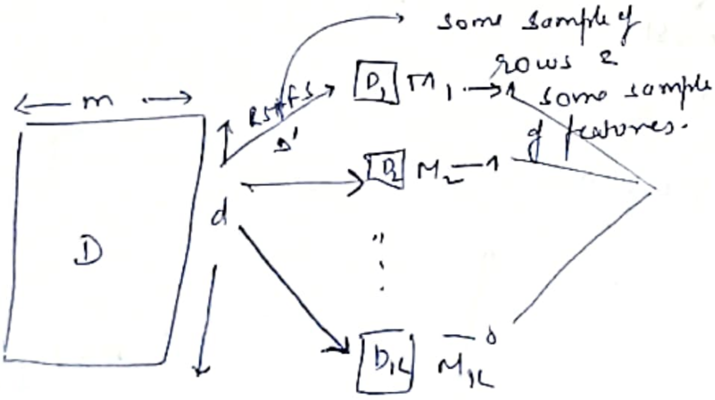
m_2 will get trained on D'_{m_2}

⋮

m_K " " " " D'_{m_K}

Bootstrap aggregation

Random Forest



$D' < D$

There are many decision trees that one creates sampling rows and columns randomly from a big dataset. This sampled data is then passed and used for training different decision trees.

The output of these trees is taken and aggregated to form a common output on a random input.

Base learner in random forest is a decision tree.

Decision Tree

Low bias and high variance] when we create a decision tree to its complete depth it leads to overfitting

✓

Decision tree in complete depth will have a high variance

Random forest overcome this limitation of over fitting of decision tree. As we provide random rows and random features to decision tree

→ low variance

It is owing to this fact that Random forest works very well in most of the cases that we use.

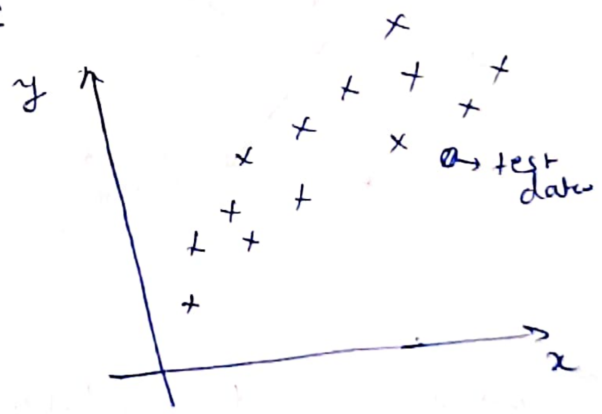
In regression → we take mean or median of the decision tree output that we have

*Hyperparameter → how many decision tree we have to use using Random forest

KNN Regression

K Nearest Neighbour

- Classification
- Regression



K=5 , hyper parameter

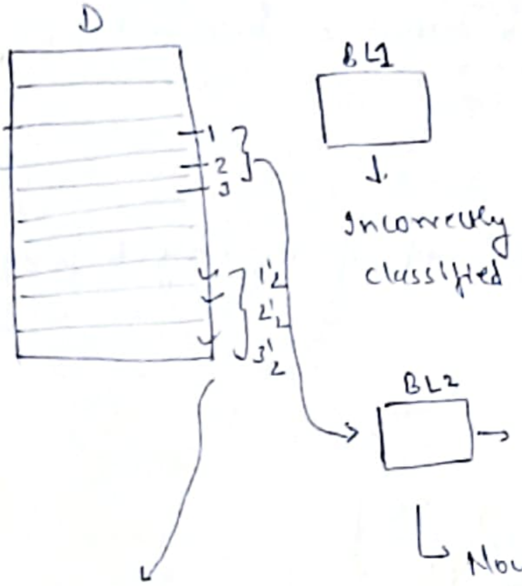
i.e we will look for 5 nearest points to the test point

average of all the values of 5 data points

average of nearest datapoints.

Value = average of nearest datapoints

Boosting



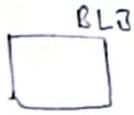
① We pass entire data to the base learner and then we see that which of the data has been incorrectly classified.

② Let's say record 1, 2, 3 are incorrectly classified.

→ error records from base learner 1 will pass to base learner 2

└ Now lets say base learner 2 give wrong values for other three records

These records will now pass to Base learner 3



This cycle will keep going on in the same manner. This's a boosting technique

Adaboost → base learner - Decision Tree

Sample dataset

f_1	f_2	f_3	o/p	Sample weight
-	-	-	-	$1/7$
-	-	-	-	$1/7$
-	-	-	-	$1/7$
-	-	-	-	$1/7$
-	-	-	-	$1/7$
-	-	-	-	$1/7$

→ sample total error = $1/7$

Cycle 1 formula for Sample weight

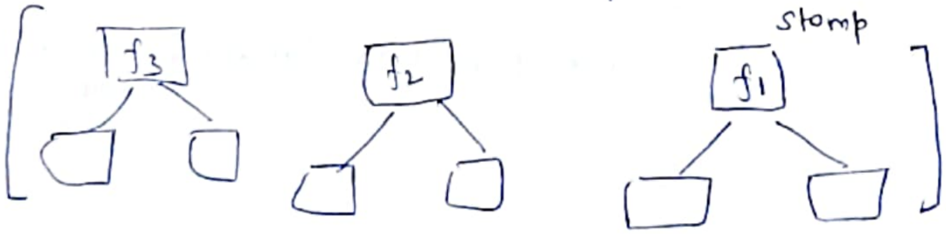
$w = 1/n$

$= 1/7$

all records are basically assign the same weight

* Decision Tree are base learners and are created with just one depth. These are called as stomps

We create stomp for each feature with a depth 1



(12) We will select base learner as a decision tree which has least entropy
Now lets say it correctly classifies 3 record & incorrectly classifies 4 records.

Step 2 Calculate the total error - just sum up the weights for the error rows.

Step 3

$$\text{Performance of stump} = \frac{1}{2} \log_e \left[\frac{1-TE}{TE}\right]$$

TE → Total error $= \frac{1}{2} \log_e [6] = 0.896.$

Step 4 update the weights → Increase the weight for wrongly classified records and decrease the weights for correctly classified ones

Update weight

Incorrectly classified record New weight = Old weight × $e^{\text{Performance of stump}}$

$= \frac{1}{7} \times e^{0.859} = 0.359$

For correctly classified point New weight = $e^{-(\text{Performance of stump})}$

$= e^{-0.895} = 0.05$

Now we will normalize the weights by dividing all these newly obtained weights by sum of these weights.

Create a New dataset for next Base learner

(13) Normalized weight

New dataset

Normalized weight	Category
0.07	0 - 0.07
0.51	0.07 - 0.55
0.07	0.55 - 0.65
0.07	0.65 - 0.72
0.07	
0.07	

* Now it will select a random weight from one of these buckets.

Lets say random weight is

0.34 →
* which bucket it lies?
(3)
* Selects one or two records from this bucket.
* Make a data set.

Now owing to updated weights there is a more probability of a wrong weight getting selected.

This cycle will continue until it pass through sequential decision tree. We will obtain the sequence of stomps and for a test data we will obtain value from all stomps and will either take mean or median

* We combine multiple weak learners to obtain a strong learner

Gradient Boost

Exp	Degree	y Salary	$\hat{y}$
2	BE	50K	75
3	Masters	70K	75
5	masters	80K	75
6	PHD	100K	75

Step 1 Base model - 1 (average)

$$\frac{50+70+80+100}{4} \simeq 75K$$

what ever the input is, I will give the output as 75K.

Step 2 Compute pseudo residuals / residuals / error

$y-\hat{y}$

Exp	Degree	Salary	$\hat{y}$	R_1	R_2	R_3
2	BE	50K	75	-25	-23	1
3	MCA	70K	75	-5	-3	1
3	MCA	80K	75	5	3	1
10	PHD	100K	75	25	20	

Step 3 : Construct the decision tree

Output is residual and rest all the feature

$\{x_i, R\}$

Base learner average $\cancel{5}$ 75 $\rightarrow$ Decision Tree 1 $\rightarrow$ DT_2 (x_i, R_2) $\rightarrow$ DT_3 (x_i, R_3)

How to compute salary

$75 + R_1 + \cancel{R_2 +}$ ——

like $75 - R_3 \simeq 52$ which is very near to 50K

This is a problem as my model is overfitting

Now we need a decision tree with low variance & low bias.

Hence we will introduce a learning rate

$75 + \alpha(R)$
$\hookrightarrow (0-1)$

$$F(x) = h_0(x) + \alpha_1 h_1(x) + \alpha_2 h_2(x) + \cdots + \alpha_n h_n(x)$$

$$= \sum_{i=1}^{n} \alpha_i h_i(x) + h_0(x)$$

This residual value will be decrease and one with more residual will be decreasing slowly but other near values residual will eventually come to zero.

pseudo algorithm (gradient boost)

Exp	Degree	y Salary	$\hat{y}$	r_1	r_2
2	BE	50K	60	-10	
3	PHD	70K	60	10	
4	MAST	60K	60	0	

I/P

① $\{x_i, y_i\}$: Independent and dependent variables

② $L(y, f(x))$: loss function (differential)

Regression :- Mean squared error

Classification :- Hinge loss | log loss

③ Number of Trees

Psudo Algorithm

① Initialise the model with constant value

$$F_0(x) = \underset{r}{\text{argmin}} \left(\sum_{i=1}^{n} L(y, r) \right)$$

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

Find the r value for which loss f^n is minimal

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

↓ 1st order derivative

$$\frac{\partial L}{\partial \hat{y}} = -(50 - \hat{y}) - (70 - \hat{y}) - (60 - \hat{y}) = 3\hat{y} - 180 = 0$$

$\boxed{\hat{y} = 60}$ – average

② Iterate the steps from 1 to (M) ~ number of DT_s

Compute psudo residuals.

$$r_{im} = \left[\frac{\partial L(y, F(x_i))}{\partial (F(x_i))} \right] \quad i = 1 \text{ to } m$$

16

$Loss = \frac{1}{2}(y-\hat{y})^2$

$$\frac{\partial L}{\partial \hat{y}} = (y-\hat{y})(-1)$$

$$\boxed{-\frac{\partial L}{\partial \hat{y}}} = y-\hat{y}$$

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

Now this residual error is nothing but the difference of actual & predicted value

Once we get the residual we create a Decision Tree where my dependent feature will be the residue and independent feature will be experience and degree

- again fit a base learner. $h_m(x)$

$i/p \{x_i, r_{im}\}$

Step 3

prev model output

$$\gamma_m = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} \left(L(y_i, f_{m-1}(x_i)) + \gamma\right)$$

$$L(y_i, F_{m-1}(x_i)) = \sum_{i=1}^{n} \frac{1}{2}(y_i - (60+\hat{y}))^2$$ - minimize this.

Step 4

$$F_m(x) = F_{m-1}(x) + \alpha(h(x))$$

$\hookrightarrow$ learning rate

XG Boost Regressor

Average salary = 51K

Base model = 51K

(14)

Exp	Gap	Salary	Residual 1	o/p	Res2 → Another tree
2	Yes	40K	-11K	46	
2.5	Yes	42K	~~10K~~ -9	46	
3	No	52K	1K	53.5	
4	No	60K	9K	62	
4.5	No	62K	11K	63	

DT (Exp, Gap, Res1) only binary trees

[-11, -9, 1, 9, 11]

$\boxed{\text{Exp}}$ $SW = \frac{1}{6} = 0.16$

<= 2 / > 2

-11 | -9, 1, 9, 11

$SW = \frac{121}{1+1} = 65.5$

$SW = \frac{\cancel{81+1+81+121}}{\cancel{4+1}} \frac{(-\cancel{9}+1+\cancel{9}+11)^2}{4+1} = \frac{144}{5} = 28.5$

Step 2 : Similarity weights

$$= \frac{\Sigma (\text{Residual})^2}{\text{No. of Residuals} + \lambda}$$

→ hyper parameter

$\lambda \simeq 1$

Gain / Information gain = (LSW + RSW) − Root SW

= 65.5 + 28.5 − 0.16 = 93.84

Now similarly we will make other split and calculate the gain

≤ 2 93.84 | we decide the split

≤ 2.5

We can create any number of decision trees. and select 1 DT and Calculate output avg = 51

[-11, -9, 1, 9, 11]

Exp

≤ 2.5   > 2.5

[-11, -9]   [1, 9, 11]

o/p - 10

Y   N

[9, 11]   [1, 9]

o/p = 11   o/p = 5

Now the value for exp < 2.5

$= 51 - \alpha(10)$

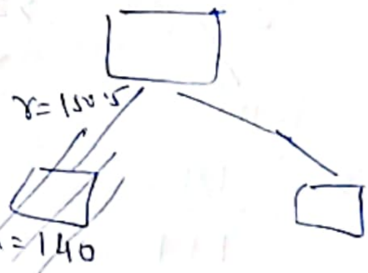
Let $\alpha = 0.5$

$= 46.$

$o/p = \text{Base model} + \alpha_1(T_1) + \alpha_2 T_2 + \alpha_3 T_3 \ldots + \alpha_n T_n$

There is one more hyper parameter γ

Let's say $\gamma = 150.5$



a gain $- \gamma < 0$

prune this tree means

↳ Cut it → prevent the overfit

if positive don't prune this.