Boosting Techniques:

Boosting Algorithms

- ADABoost
- Gradient Boost
- Xgboost

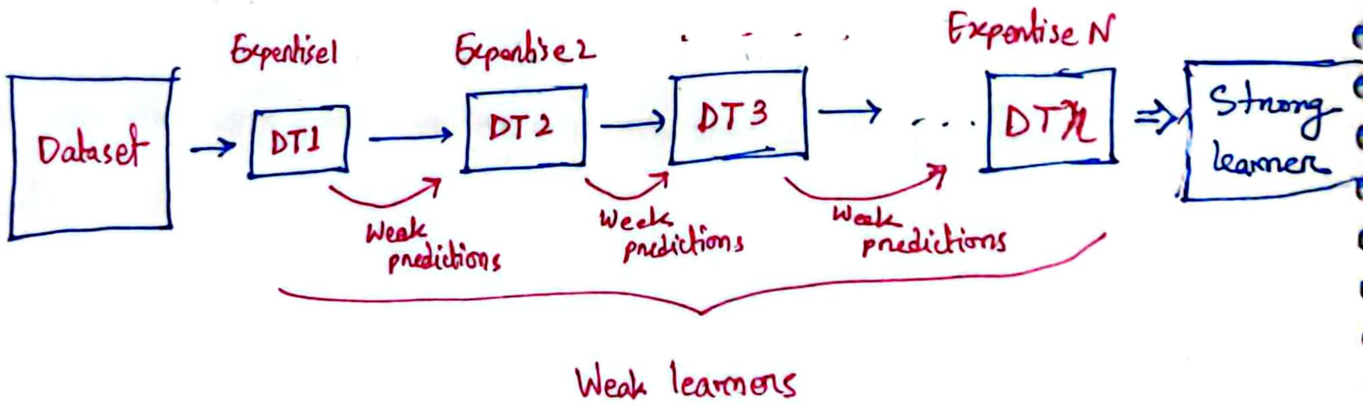
We already discussed about the overfitting issue of Decision Tree in ~~every~~ previous Bagging technique.

We saw that →

Overfitting { Low Bias, High Variance } —Random Forest→ { Low Bias, Low Variance }

In Bagging technique, all of our decision trees were trained parallely with sample rows and sample features.

But in "Boosting technique", decision trees are connected sequentially



In Boosting technique, training dataset goes to first decision tree model. It gets train. ~~B~~ Then it sends all the wrong prediction it does to the next Decision tree model (DT2). It is the weakest learner. Then DT2 get trained and it also does wrong predictions besides doing write predictions. Then it sends it wrong predicted data to the next decision tree to get trained on. This goes on and on till the nth decistion tree (DTn). Then combining all the models we get a final model which is called "Strong learner".

Weak learners : Haven't learned much from the training dataset.

In boosting technique, every model will provide a confidence value of how well it can predict, which is called "weight."

Boosting

$$f = \alpha_1(M_1) + \alpha_2(M_2) + \cdots + \alpha_n(M_n)$$

→ Classification

→ Regression

$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4 \ldots \alpha_n\}$ → weights (confindence)

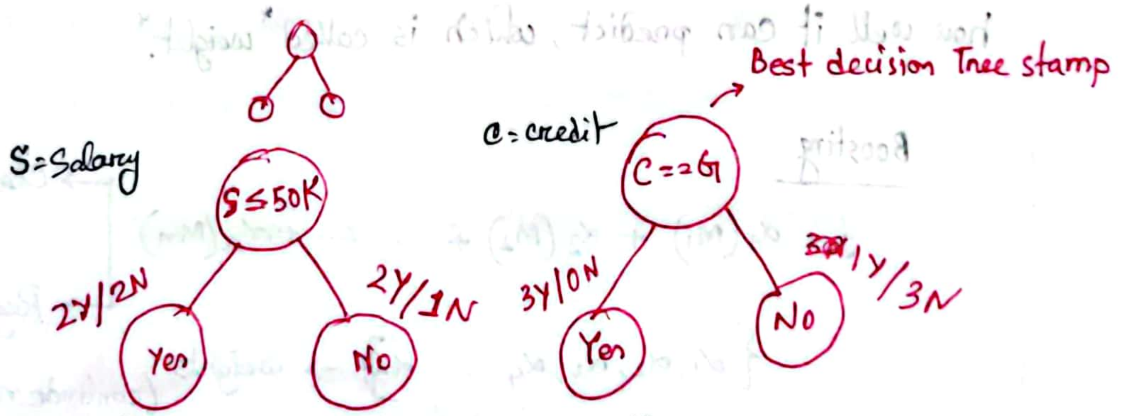
AdaBoost Classifiers:

Dataset

Salary	Credit	Approval → target
<=50K	B	No
<= 50K	G	Yes
<= 50K	G	Yes
> ~~<=~~ 50K	B	No
> ~~<=~~ 50K	G	Yes
> ~~<=~~ 50K	N	Yes
≤ ~~<=~~ 50K	N	No

Procedure: We create decision tree stamp and we select the best stamp.

Decision Tree stamp → 1 level of depth in the decision Tree



Select the best dt using Entropy or Gini Impurity and Info gain.

2nd step: We will assign each record a sample weight.

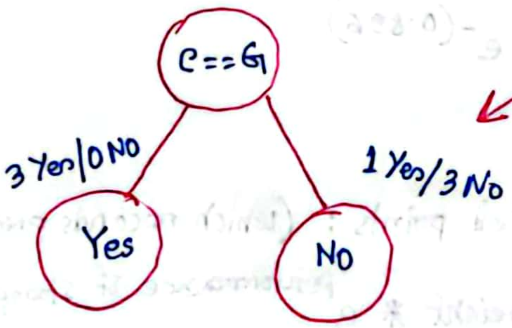
Salary	Credit	Approval	Sample Weight
<= 50	B	No	1/7
<= 50	G	Yes	1/7
<= 50	G	Yes	1/7
> 50	B	No	1/7
> 50	G	Yes	1/7
> 50	N	Yes	1/7
<= 50	N	No	1/7

(Because we have 7 records)

row 6 above predicting "Yes" which is wrong

3rd Step: Sum of total errors and performance of stamp.

Best DT stamp



As, here 1 record is false, sum of total error = $1 \times 1/7$ (TE)

For, n it would be = $n \times 1/7$

Performance of stamp:

$$\frac{1}{2} \ln\left[\frac{1-TE}{TE}\right]$$

For above case

$$= \frac{1}{2} \ln\left[\frac{1-1/7}{1/7}\right] = \frac{1}{2}\ln[6] \approx 0.896$$

Our Adaboost formula was →

$$f = \alpha_1(M_1) + \alpha_2(M_2) + \ldots\ldots + \alpha_n(M_n)$$

M_1 = This would be our Best DT stamp

α_1 = performance of stamp = weights = 0.896

4th step: Update the weights for correctly or incorrectly classified points.

For correctly classified points:

$$\text{sample weight} * e^{-\text{Performance of stamp}}$$

$$= 1/7 * e^{-(0.896)}$$

$$= 0.058$$

For incorrectly classified points: (which records provided wrong prediction)

$$\text{sample weight} * e^{\text{performance of stamp}}$$

$$= 1/7 * e^{0.896}$$

$$= 0.349$$

Salary	Credit	Approved	Sample weight	Updated weight
<=50K	B	NO	1/7 ↓	0.058
<=50K	G	Yes	1/7 ↓	0.058
<=50K	G	Yes	1/7 ↓	0.058
>50K	B	NO	1/7 ↓	0.058
>50K	G	Yes	1/7 ↓	0.058
>50K	N	Yes	1/7 ↑↑	0.349
<=50K	N	NO	1/7 ↓	0.058

5th step: Normalize weights computation and assigning bins

$\boxed{\text{Updated weights} / \sum \text{Updated Weights}}$

Salary	Credit	Approval	Sample ~~Updated~~ Weights	~~Not~~ Updated Weights	Normalized Weight	Bins
<= 50K	B	No	1/7 ↓	0.058	0.083	0 - 0.083
<= 50K	G	Yes	1/7 ↓	0.058	0.083	0.083 - 0.166
<= 50K	G	Yes	1/7 ↓	0.058	0.083	0.166 - 0.24
> 50K	B	No	1/7 ↓	0.058	0.083	0.24 - 0.32
> 50K	G	Yes	1/7 ↓	0.058	0.083	0.32 - 0.40
> 50K	N	Yes	1/7 ↑↑	0.349	0.500	0.40 - 0.90
<= 50K	N	No	1/7 ↓	0.058	0.083	0.90 - 0.98
				$\Sigma = 0.697$	$\Sigma = 1$	

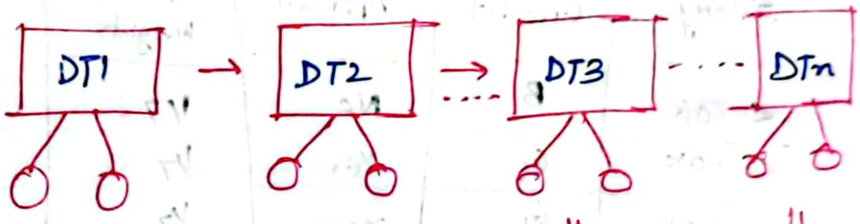
Here, we can see that, the records which provided wrong prediction, their bin sizes' are larger in range than other records. So, our model will loop through the bin column and pick random records to pass on to the next decision Tree model. The probability of picking larger bin size will be in ~~prigrow~~ priority. So, incorrect data are getting picked to train the next DT model.

So, the ~~x~~ next M2 model will follow all the steps again and pass the wrong data to M3 model and so on. It continues till the n number of DT model chosen by us.

Final prediction:

For any testdata having $(\le 50K, G)$ (Salary ↓, Credit ↓)

f = Test data →



DT1	DT2	DT3	DTn
⇓ Yes	⇓ No	⇓ Yes	⇓ No
$\alpha_1 = 0.896$	$\alpha_2 = 0.650$	$\alpha_3 = 0.24$	$\alpha_n = -0.30$

$$f = 0.896(\text{Yes}) + 0.650(\text{No}) + 0.24(\text{Yes}) \cancel{+0.0} - 0.30(\text{No})$$

$$= 1.136(\text{Yes}) + 0.350(\text{No})$$

Performance of say (Yes) = 1.136

Performance of say (No) = 0.350

Yes > No

So, final prediction will be Yes.