19-NOV-2022

Boosting

→ Sequential weak learners.
→ To remove underfitting problem.

Adaboost

stump
1 depth

{ Underfitting
Training data Accuracy → low
Test data Accuracy → High }

Boosting { Low Bias ⟸ High Bias
Low Variance ⟸ B.Low Variance

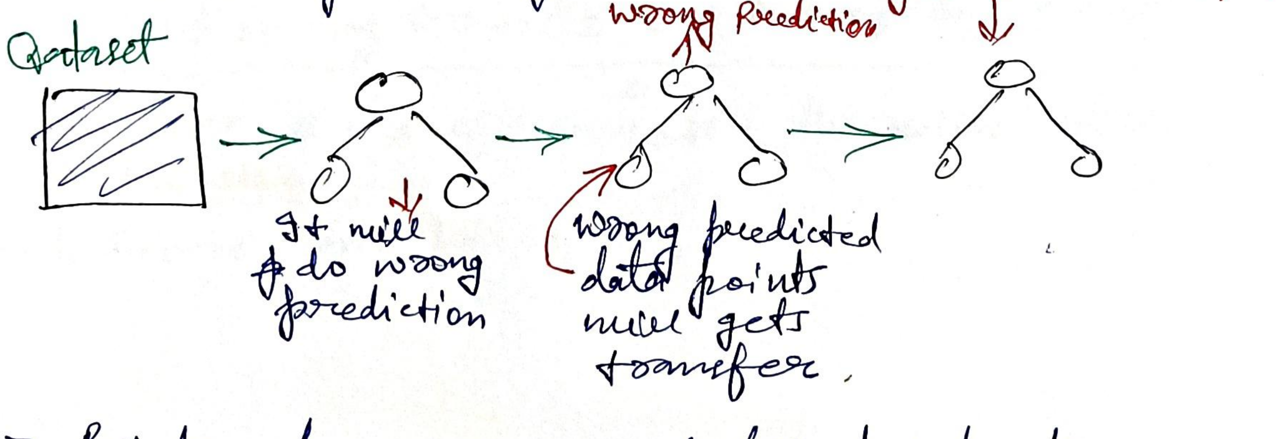
→ combination of weak learners become strong learners.

Random Forest

→ final output will be majority of vote. in classification problem.
→ final output will be average of vote in regression problem.

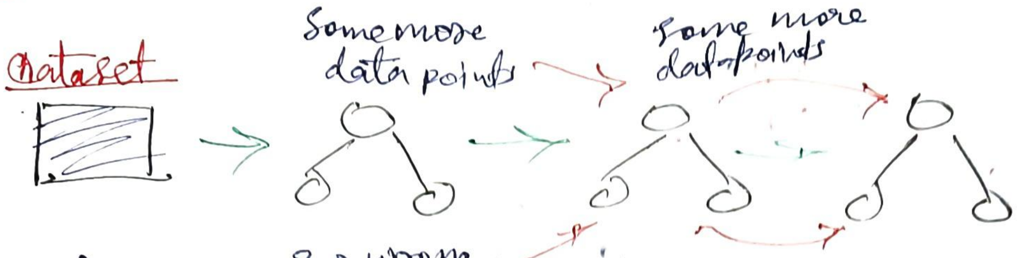
Adaboost

→ Add the output of the weak learners with some weight assigned to it. wrong data points transfer


wrong Prediction
Dataset
It will do wrong prediction
wrong predicted data points will gets transfer.

→ By transfering wrong data points to each weak learners, we will get strong learners.

Eg



out of 100 data points →

20 wrong predictions, 80 correct predictions

10 wrong predictions, 90 correct predictions

Adding weights

$$\boxed{f = \alpha_1 M1 + \alpha_2 M2 + \alpha_3 M3 + \dots + \alpha_n Mn}$$

where,

α = weights

if α = +ve, importance of model will get increase

if α = -ve, model is less importance.

M1, M2, M3, M4, ..., Mn → weak learners → Depth=1

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$ ..., α_n → weights

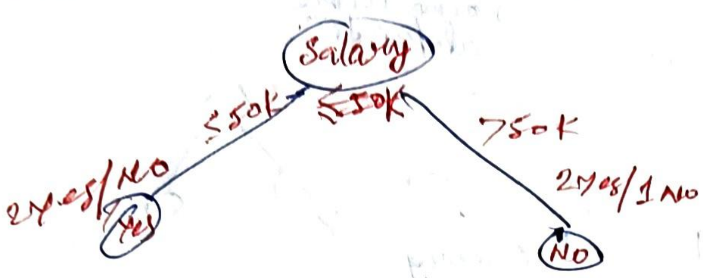
→ If depth = 1, then weak learners.

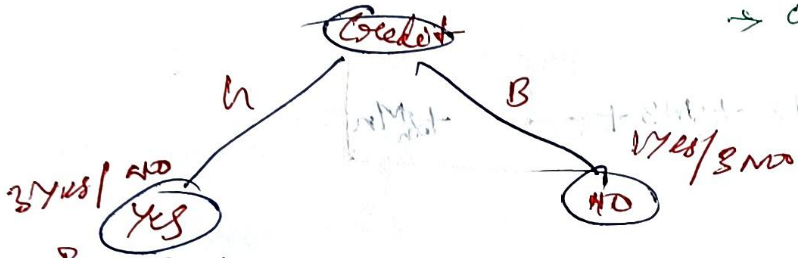
Eg

Salary	Credit	Approach	Weights
<=30K	B	NO	1/7
<=50K	G	Yes	1/7
<=50K	G	Yes	1/7
>50K	B	NO	1/7
>50K	~~H~~ G	Yes	1/7
>50K	N	Yes	1/7
<=50K	N	NO	1/7

→ wrongly Predicted

Step1 : Create Decision Tree Stump(1 depth DT)





Pure Split

→ choose this as weak learner.

How to determine which is weak learnr?

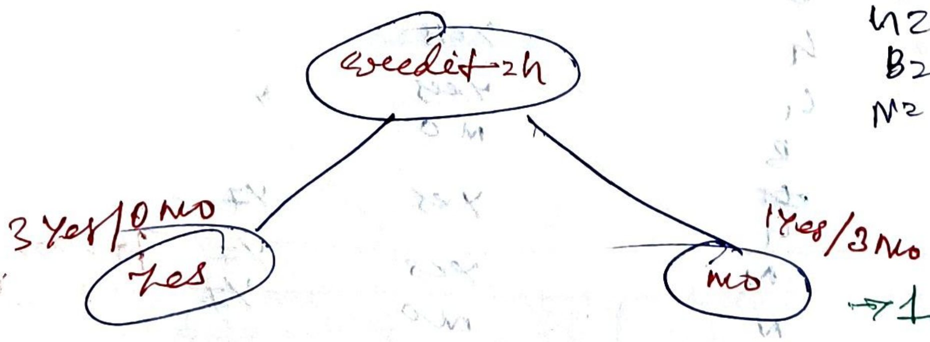
~~Ans~~ using entropy or Gini, we will decide better entropy.

$$\boxed{H(S) = -P_y \log_2(P_y) - P_N \log_2(P_N)}$$

Step2 : Assing weights (1/7)

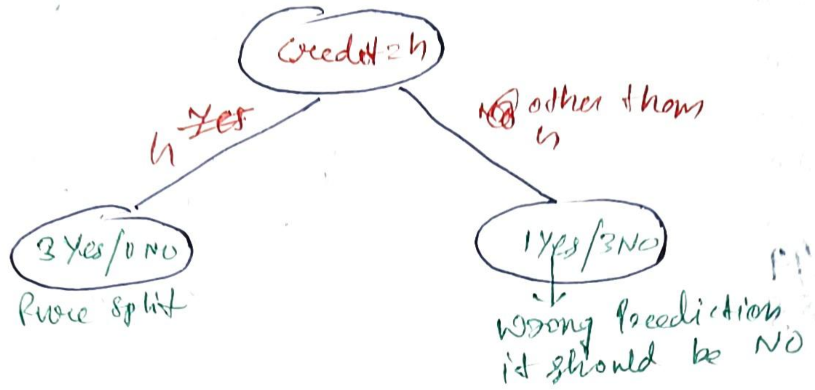
→ weights treats all the data points equally.

Letls take "credit" as weak learner



G=Good
B=Bad
N=Normal

→1 wrong Prediction



step 3: calculate total error

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

{Because 1 data point ~~was~~ is wrongly predicted. If more data ~~will~~ would there then add all the data points}.

step 4: ~~St~~ Performance of stump

$$\boxed{\text{Performance of stump} = \frac{1}{2}\ln\left\{\frac{1-TE}{TE}\right\}}$$

Here

$$\text{Performance of stump} = \frac{1}{2}\ln\left\{\frac{1-1/7}{1/7}\right\} = 0.896$$

$$f = \alpha_1(M1) + \alpha_2(M2) + \alpha_3(M3) + \dots + \alpha_n(M_n)$$

$\alpha_1 = 0.896$ (weight)

M1 → credit model prediction

step 5: Update the weight for correctly and incorrectly data points

→ 1 was wrongly predicted.

Update Weights

0.058

0.058

0.058

0.058

0.058

$\boxed{0.349}$ ← Incorrected Prediction

0.088

↓ weight got increased.

~~For~~

total = 0.697

$$\left\{ \text{For correctly predicted datapoints} = \text{weight} \times e^{-\text{Performance}} \right\}$$

Here

$$= \frac{1}{7} \times e^{-(0.896)}$$

$$= 0.058$$

$$\left\{ \text{For Incorrect predicted datapoints} = \text{weight} \times e^{\text{Performance}} \right\}$$

Here

$$= \frac{1}{7} \times e^{0.896}$$

$$= 0.349$$

→ ~~As~~ Give wrong predicted data points to next model

Step 6: Normalize the weights and assign bins

		Normalized Weights
0.058	0.058/0.697	= 0.08
0.058	0.058/0.697	= 0.08
0.058		0.08
0.058		0.08
0.058		0.08
0.349	0.349/0.697	= 0.50
0.058		0.08
0.697		1

Bins Assignment

0 – 0.08

0.08 – 0.16

0.16 – 0.24

0.24 – 0.32

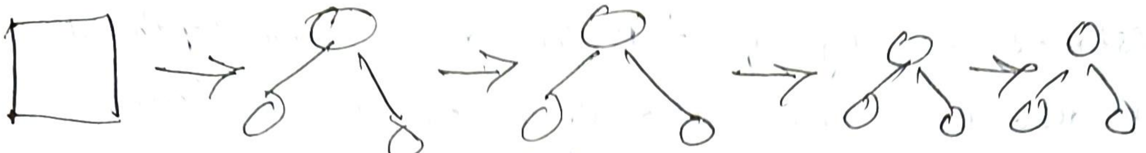
0.32 – 0.40

0.40 – 0.90

0.90 – 1

- > Random bin will get selected.
- → ~~Along~~ Along with wrong predicted data points, some ~~wro~~ correctly predicted points ~~wil~~ will also get selected.

→



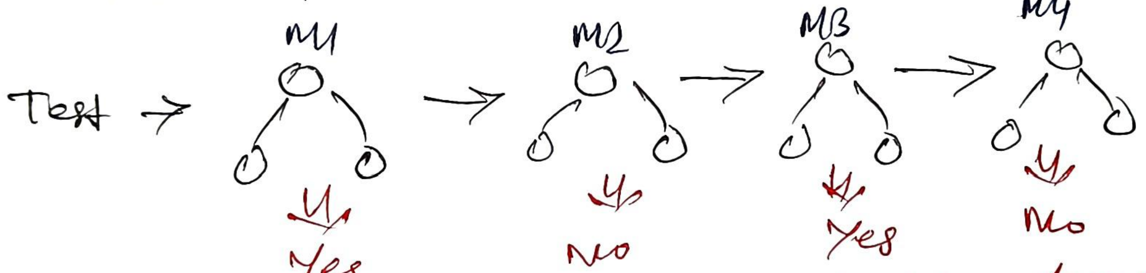
$\alpha_1(M1) + \alpha_2(M2) + \alpha_3(M3) + \dots \alpha_n(Mn)$

→ wrong predicted data points have large bin size.

Step 7: Final Prediction

↓

Test (≤50K, 4)



$\alpha_2 = 0.650$ $\alpha_3 = 0.38$ $\alpha_4 = 0.20$

Random assignments → $\alpha_1 = 0.896$

$$f = \alpha_1(M1) + \alpha_2(M2) + \alpha_3(M3) + \alpha_4(M4)$$

$$= 0.896(Yes) + 0.650(NO) + 0.38(Yes) + 0.20(NO)$$

$$= \underbrace{1.2(Yes) + 0.85(NO)}$$

⇓

Yes (Because weight of yes is more)

In Regression

→ We use MSE instead of entropy or Gini and repeat the complete process.

Adaboost Parameters

base_estimator = By default Decision Tree with depth = 1
→ We can also use other algorithm as weak learner.

n_estimators = 50 (by default) → can be changed.

learning_rate = by default 1.0

* Weights $\alpha_1 M1 + \alpha_2 M2 + \alpha_3 M3 + \dots + \alpha_n M_n$