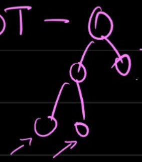
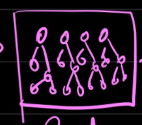
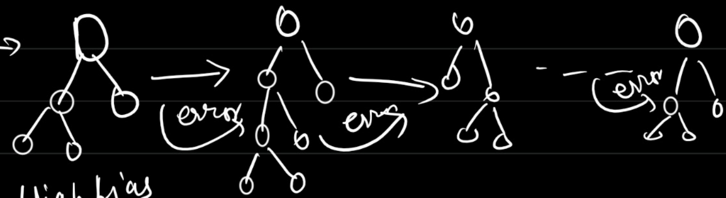
AdaBoost (Adaptive Boosting)

by Yoav Freund and Robert Scaphire in 1995.

* Works both for classification & regression problem.

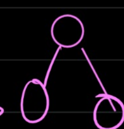
DT →  → High variance → Bagging → 

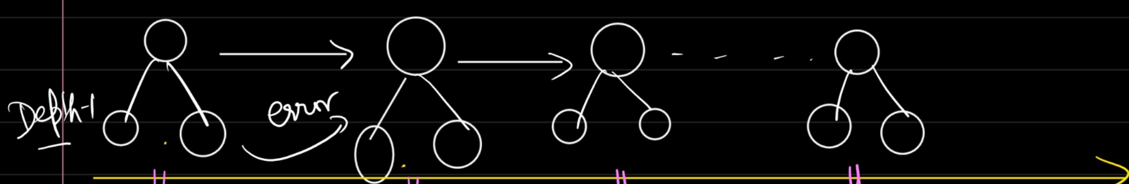
Random Forest → It reduces the high variance of of individual decision trees.

* Boosting → 

* AdaBoost follows the concept of boosting.

Decision Stumps :- A decision tree with only one level of depth

→ 



Weak learner ⇓ Simple (only 1 depth)

Weak learner ⇓ Simple

WL ⇓ Simple

WL ⇓ Simple

↳ Underfitting model (High bias — high training Error)

↳ train acc ↓ → High bias

* Each individual DT is a high bias model but error is transferred to next model leading to low bias at an overall level.

→ AdaBoost solves the problem of high bias.

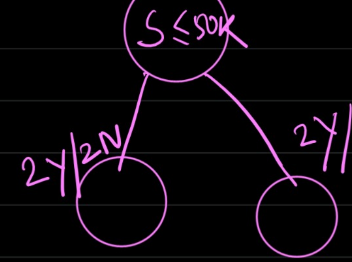
→ AdaBoost is low bias model.

→ we always want low bias & low variance model.

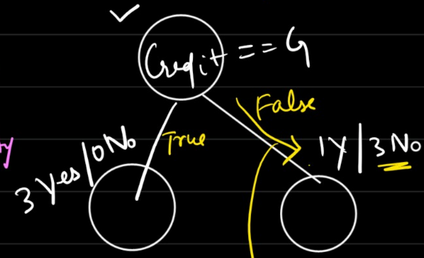
* Mathematical explanation of AdaBoost

Step-1 → Create decision tree stump.

Salary	Credit Score	Approval.
<=50k	B	No
<=50k	G	Yes
<=50k	G	Yes
>50k	B	No
>50k	G	Yes
>50k	N	Yes
≤ 50k	N	No



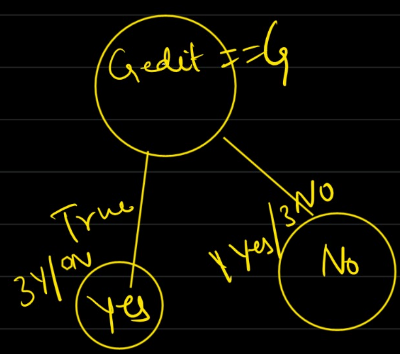
Entropy or Gini impurity



In this majority is No, that's Credit score != G, will be predicted as No but 2nd last row is predicted Yes, this is wrong prediction.

Step-2 ⟶ Assign equal weights to all of the datapoints and predict using the higher IG Decision Stump

Salary	Credit Score	Approval.	Sample weights
<=50k	B	No	1/7
<=50k	G	Yes	1/7
<=50k	G	Yes	1/7
>50k	B	No	1/7
>50k	G	Yes	1/7
>50k	N	Yes	1/7
≤ 50k	N	No	1/7



Step-3 → Total Error and performance of stump.

Total Error = 1/7

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

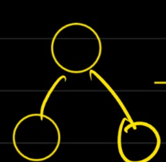
$$= \frac{1}{2}\ln\left[\frac{1-1/7}{1/7}\right]$$

$$= \frac{1}{2}\ln(6) \approx \underline{0.896}$$

↳ base to e

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

↳ $\boxed{\alpha_1 = 0.896}$ → weight of first decision stump.



(TE, performance) ⟶ next step is to give more priority to wrong prediction.

Salary	Credit Score	Approval.	Sample weights
<=50K	B	No	1/7 ↓
<=50K	G	Yes	1/7 ↓
<=50K	G	Yes	1/7 ↓
>50K	B	No	1/7 ↓
>50K	G	Yes	1/7 ↓
>50K	N	Yes	1/7 ↑
≤ 50K	N	No	1/7 ↓

Step-4 → give more weights to incorrect classified datapoint and lesser weight to correct datapoint

* For correct classified dp's

$$= \text{weight} * e^{-\text{Performance of Stump}}$$

$$= 1/7 \times e^{-0.896} = 0.058 \qquad e = 2.73$$

* Incorrect classified dp -

$$= \text{weight} \times e^{+\text{Performance of Stump}}$$

$$= \frac{1}{7} * e^{0.896} = 0.349$$

$\ominus 5$ (correct ones)

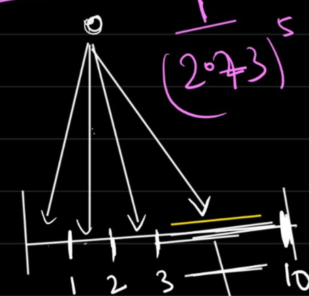
$(2.73)^5 \quad (2.73)^{-5} = \frac{1}{(2.73)^5}$

Step-5 Normalized wt & Bin assignment.

Salary	Credit Score	Approval.	initial SW	Updated wt	Normalized wt	Bin assignment
<=50K	B	No	1/7 ↓	0.058	0.08	0–0.08
<=50K	G	Yes	1/7 ↓	0.058	0.08	0.08–0.16
<=50K	G	Yes	1/7 ↓	0.058	0.08	0.16–0.24
>50K	B	No	1/7 ↓	0.058	0.08	0.24–0.32
>50K	G	Yes	1/7 ↓	0.058	0.08	0.32–0.40
>50K	N	Yes	1/7 ↑	0.349	0.50	0.40 – 0.90
≤ 50K	N	No	1/7 ↓	0.058	0.08 ≈ 1	0.90 – 0.98

total weight = 0.697

Here total weight is not 1, then we have to normalize



Higher prob of stone falling in this region

$\alpha_1 = 0.896$ Preparing the dt

Step-6 Generate a random no between 0 to 1

0.3
0.4
0.5
0.4
0.6
0.2
0.7
0.8
0.9

Salary	Credit Score	Approval.	SW	Updated wt	Normalized wt	Bin assignment
<=50K	B	No	1/7 ↓	0.058	0.08	0–0.08
<=50K	G	Yes	1/7 ↓	0.058	0.08	0.08–0.16
<=50K	G	Yes	1/7 ↓	0.058	0.08	0.16–0.24
>50K	B	No	1/7 ↓	0.058	0.08	0.24–0.32
>50K	G	Yes	1/7 ↓	0.058	0.08	0.32–0.40
>50K	N	Yes	1/7 ↑	0.349	0.50	$\boxed{0.40 \rightarrow 0.90}$
≤ 50K	N	No	1/7 ↓	0.058	0.08 ≈ 1	0.90 – 0.98

Salary	Credit Score	Approval	Bin assignment
<=50K	B	No	0-0.08
<=50K	G	Yes	0.08-0.16
<=50K	G	Yes	0.16-0.24
>50K	B	No	.24-0.32
>50K	G	Yes	.32-0.40
>50K	N	Yes	0.40 → 0.90
≤ 50K	N	No	0.90-0.98

1st, 2nd, 3rd, 4th

rn b/w 0 to 1

0.4
0.5
0.4
0.6
0.2
0.7
0.8
0.9

* These records will be sent to next Decision Stump

>50K	N	Yes
>50K	N	Yes
<50K	G	Yes
≤50	G	Yes
>50K	N	Yes

* Note: → The row with higher bin range due to misclassification can be repeat ⇒ more priority to misclassified dp.

→ TE
→ performance of stump
→ weight, bin assignment

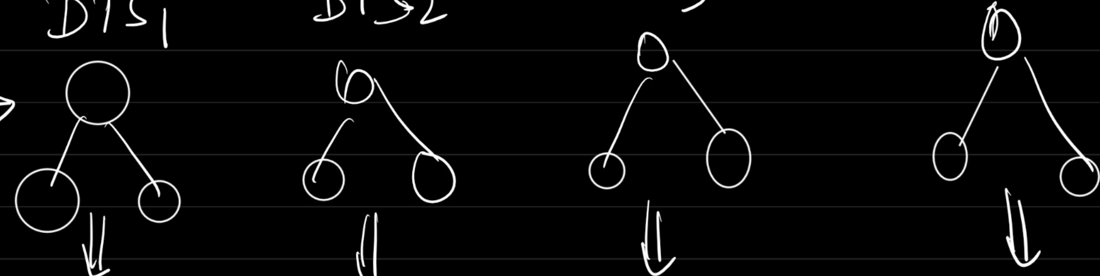
DTS1 α_1 DTS2 α_2 α_n

* Final Prediction

Test data (≤ 50K, G)

DTS1 DTS2 DTS3 - - - DTSN

test data →



DTS1	DTS2	DTS3	DTSN
⇓	⇓	⇓	⇓
Yes	NO	Yes	No
$\alpha_1 = 0.896$	$\alpha_2 = 0.55$	$\alpha_3 = 0.25$	$\alpha_n = -0.30$

Performance of Stump

$$f = \alpha_1(M_1) + \alpha_2(M_2) + \alpha_3(M_3) \; \ldots \; \alpha_4(M_4)$$

$$= 0.896(\text{Yes}) + 0.55(\text{No}) + 0.25(\text{Yes}) \; \ldots \; 0.30(\text{No})$$

$$= 1.15(\text{Yes}) + 0.25(\text{No})$$

→ Performance of NO = 0.25

⇓

Performance of Yes = 1.15

final Output ⇒ Yes.

* multiclass → Same method → there will three Aggregated alphas.

* AdaBoost Regressor

→ All the steps will be same except instead of Information gain, you will use variance reduction to select the Decision tree stump.

→ during prediction

$$f = \alpha_1 m_1 + \alpha_2 M_2 + \ldots \; \alpha_n M_n$$

$m_1, m_2 \ldots m_n$ ⇒ will be a Continuous Value.