Reg. Classification

Ensembles

- → multimodels
- → Custom Bagging
- → Bagging → Random forest
- → Boosting :- ① ADA Boost
  - ② Gradient Boosting
  - ③ XG Boost.
- → KNN.
- → PCA.
- → USL
- → Anomaly detection & time series.

* Boosting

Ensembles → Bagging, Boosting

Baselearner Bagging

→ Random Forest.

Boosting

- → AdaBoost
- → Gradient
- → XG Boost.

⇒ both regression & classification

Bagging : Baselearner (Decision Trees)

⇓

Bootstrap aggregation

Boosting :- weak learners (Decision trees)

* Boosting (Boost करना)

India - Pakistan

Bumrah

Data → DT_1 → DT_2 → DT_3 → - - DT_n

Weak learner, wrong predict, Weak, wrong predict, Weak, Weak

⇒ Strong learner.

* weak learner → Model is very simple and not properly from data.

Analogy
→ Football
→ one stick

* Principle of Boosting :→ Build a first model on training data and then build a second model to rectify the errors present in the first model.

Bagging

→ classification : voting
→ Regression : Average

Boosting

$M_1 \rightarrow M_2 \rightarrow M_3 \ldots M_n$

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

$\alpha_1, \alpha_2 \ldots \alpha_n$ are weights

* AdaBoost (Adaptive Boosting)
* work both for classification & Regression

Yoav Freund and Robert Schapire in 1995.

* Boosting →

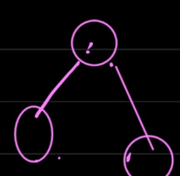
Weak learner   Weak learner

* Bagging → reduces variance
* Boosting → reduces bias.

* Ada-Boost → concept of boosting

* Decision stumps

A decision tree with only one level of depth.



→ 3

Simple
↳ Underfitting (High bias → high training error.
↳ train acc ↓

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

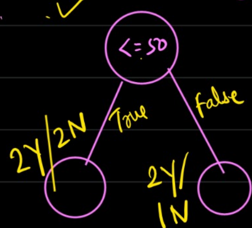
→ AdaBoost solves the problem of high bias

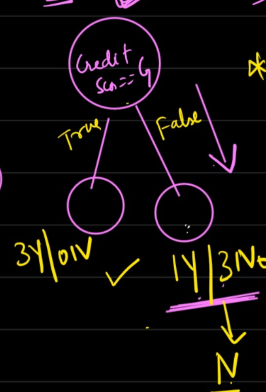
* Mathematical Explanation of AdaBoost

B-bad
G-Good
N-Not Avail

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

Step-1 Create decision tree stump and Predict

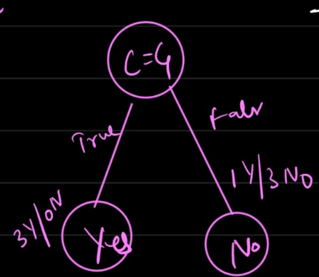




* split based on feature with highest IG.

* if CS != G
↳ N

Salary	Credit Score	Approval	Wi
<=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-2 Assign equal weights to all the dp

Step-3 - Total Error & Performance of Stump.

Total Error = 1/7
(Sum of wts of misclassified dp's)

* Performance of Stump:-

* boosting Prediction fn

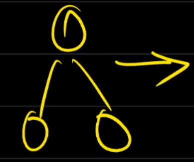
$$f = \alpha_1 M_1 + \alpha_2 M_2 + \dots \alpha_n M_n$$

$\uparrow$ 0.896

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

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

$$= \frac{1}{2}\ln[6] \approx 0.896$$



TE / performance

next step is to give more priority to wrong prediction

* Performance of individual stump (model) will be considered as weight for that specific model.

Step-4 → Give more wts to incorrect classified dp & lesser wt to correct datapoint

* For correct classified dps

$$Wt_{new} = Wt_{old} * e^{-\text{performance of stump}}$$

* for incorrect classified dp

$$Wt_{new} = Wt_{old} * e^{+\text{performance of stump}}$$

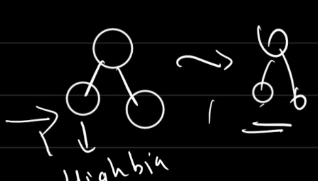
$e = 2.73$

Correct → $\frac{1}{7} \times e^{-0.896} = 0.058$

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

Salary	Credit score	Approval	Wt	updated wt	Normalised wt	Bin assignment
<= 50k	B	No	1/7	0.058	0.083	0 – 0.083
<= 50k	G	Yes	1/7	0.058	0.083	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	0.5007	0.40 – 0.90
≤ 50k	N	No	1/7	0.058	0.08	0.90 – 0.98
			1	total updated = 0.697	≈ 1	

$$\text{normalizing wt} = \frac{\text{update wt}}{\text{total updated wt}}$$



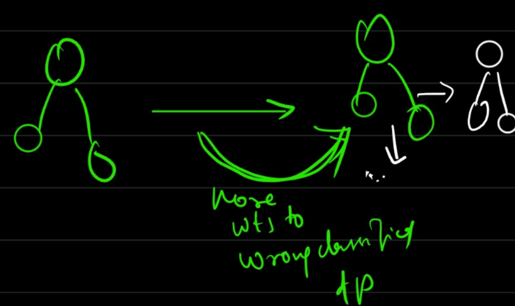
0 — 0.08 0.16

+0.08

Salary	Credit score	Approval	Wt	Updated wt	Normalised wt	Bin assignment
<= 50K	B	No	1/7	0.058	0.083	0 – 0.083
<= 50K	G	Yes	1/7	0.058	0.083	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	0.5007	$\boxed{0.40 - 0.9}$
< 50K	N	No	1/7	0.058	0.08	0.90 – 0.98
			⇓ 1	⇓ total = 0.697	⇓ ≈ 1	

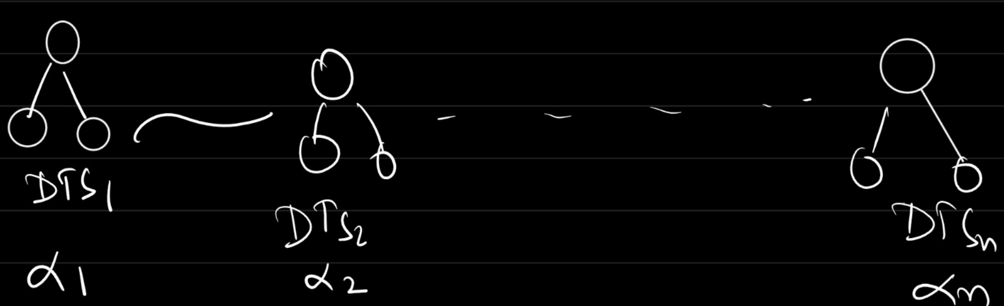
Random nos (0 to 0.98)

⇓
0.3
0.4
0.4
0.5
0.4
0.1
0.9
0.2



Note

The row with higher bin range due to misclassification will be repeating
⇓
more priority on misclassified ip's



* Final Prediction

Perfo = 0.896 (performance of stumps)

$\alpha_2 = 0.55$

$\alpha_3 = 0.2$

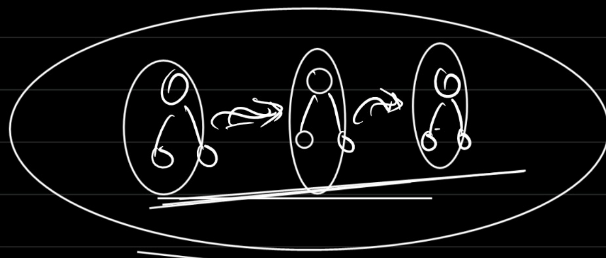
$\alpha_n = -0.30$

$$f = \alpha_1(M_1) + \alpha_2(M_2) + \alpha_3(M_3) + \cdots \alpha_n(M_n)$$

$$= 0.896(\text{Yes}) + 0.55(\text{No}) + 0.25(\text{Yes}) + \cdots -0.3(\text{No})$$

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

Final prediction $\Rightarrow$ Yes



High bias
→ High

* Adaboost Regressor

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

→ during prediction.

$$f = \alpha_1 m_1 + \alpha_2 m_2 + \cdots \alpha_n m_n$$