

# What is Boosting?

In Boosting,

- we build a bunch of simple models and
- Each model is trained using the residual of the previous model.
- use these models to build an additive weighted model

Base learners typically have low variance and high bias

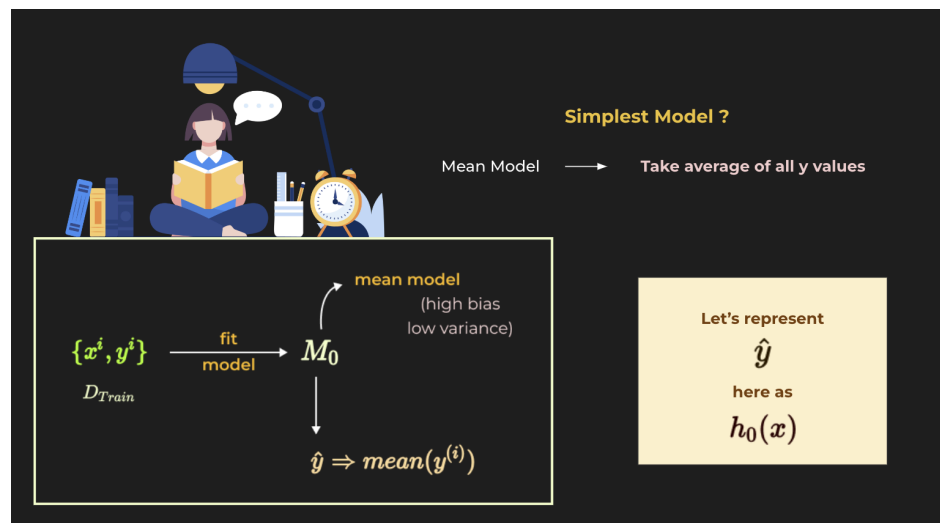
What sort of DT models have high bias?

- Shallow Trees or Decision Stump

The output of these models is combined in an Additive Manner

## Process of creating a boosting ensemble

1. A simple mean model is used and trained on training data.



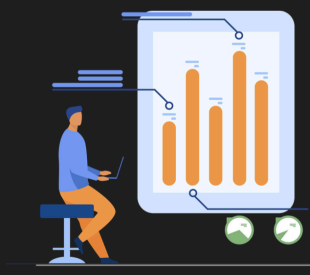
- The error becomes  $\rightarrow err_0^{(i)} = y^{(i)} - h_0(x^{(i)})$ ; where  $i \in \{1, \dots, N\}$
  - Therefore  $y^{(i)} = h_0(x^{(i)}) + err_0^{(i)}$
2. Now use a second model with the features  $x$  and target variable as the error from previous model  $err_0$ 
    - Why make the error of the previous model as a target for the new model ?

How would predicting the **error** help in **reducing** it ?

Suppose, predicted error =  $e_{pred}$

Total Prediction =  $h_0(x^i) + e_{pred}$

Closer to actual prediction  
than  $h_0(x^i)$  alone



**Conclusion :** We are reducing the error by predicting it & adding it to previous prediction

- With each addition of a new model a weight  $\gamma$  is associated

How to do that ?

→ Multiply the model pred with weight values

$$F_1(x) = h_0(x) + \gamma_1 h_1(x)$$

Prediction at stage 0      Prediction at stage 1      Weight

Lower Residual ? → Good Model → Give high influence in final prediction → **Large weights**

Higher Residual ? → Give Low influence in final prediction → **assign small weights**

3. Repeat the addition of new models for M stages where M is an hyperparameter

We keep doing this till **M stages**

$$F_m(x) = h_0(x) + \gamma_1 h_1(x) + \gamma_2 h_2(x) + \dots + \gamma_m h_m(x)$$

We can also write it as :

$$F_m(x) = h_0(x) + \sum_{i=1}^m \gamma_i h_i(x)$$

Note:  
M is a hyper parameter

## Train/Test Time of Boosting

What happens at Train & Test time ?

**At train time**

- Fits all base learners ( DT )
- Find the value of weights ( $\gamma_m$ )

**NOTE:**

- Training is bit slow.
- As it is a sequential process.

**At test time**

- We have already found hyper parameter M at train time .

Say , M = 3

For a query point

- Just pass it through DTs (base learners) & get predictions
- Multiply predictions with weight values ( $\gamma_m$ )

## Gradient Boosting/Pseudo residual - Intuition

- As models are built sequentially rather than parallel, boosting uses gradient boosting algo to minimize the loss

How Gradient boosting algo reduces loss ?

- By using **Pseudo residuals** which is the negative gradient of the loss function with respect to the predicted values,

Currently , we are using

$$\begin{aligned} err^{(i)} &= y^{(i)} - F_{j-1}(x^i) \\ err^{(i)} &= residual \end{aligned}$$

→

Instead use,

$$\begin{aligned} err^{(i)} &\approx pseudo\ residual \\ &= \frac{-\partial L}{\partial F_{j-1}}(x^i) \end{aligned}$$

[More content](#)