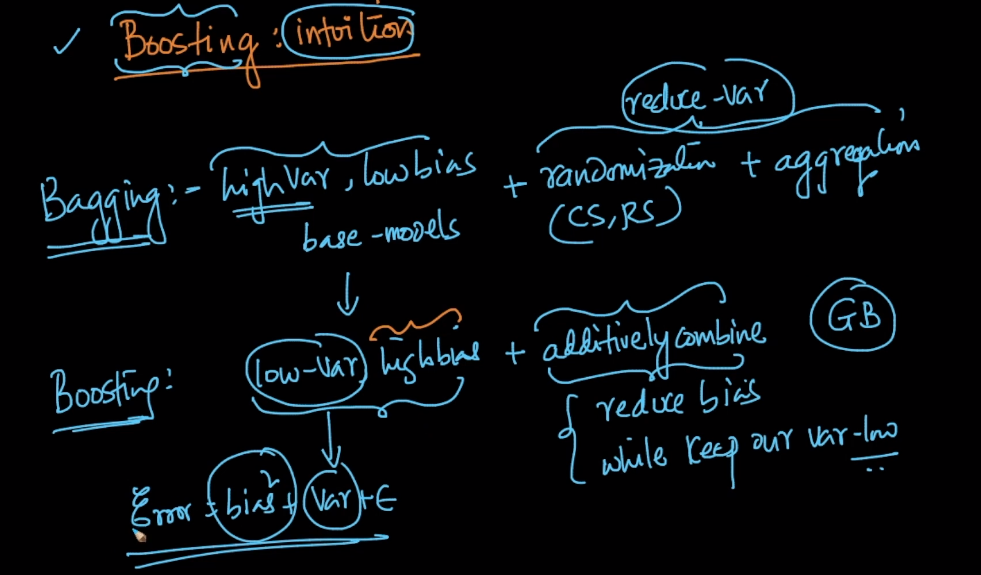
**Boosting Intuition**

In bagging we have high variance and low bias base models and then we use randomization(CS,RS) and aggregation to reduce variance

In Boosting we have low variance and high bias base models and then we use additively combine to reduce bias while keeping our variance low

Our ultimate aim is to reduce error and Error = bias2 + variance + e

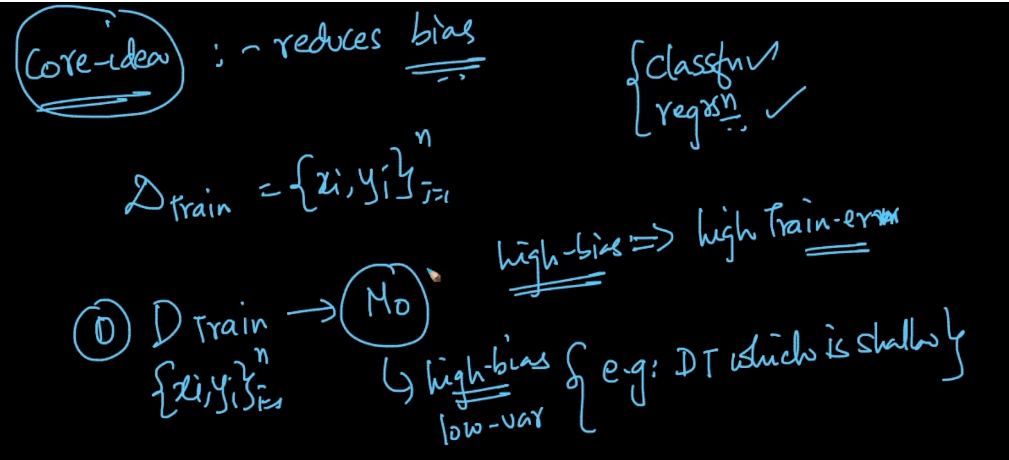
In boosting we already have low variance, we just have to reduce bias to reduce error



So the core idea of boosting is to reduce bias and we can apply it on both regression and classification.

We training data Dtrain we build a model M0 and train it using Dtrain and this model is high bias and low variance (eg. DT which is shallow i.e which have low depth)

High bias means we got high training error



Steps of boosting :

**Stage 0 :**

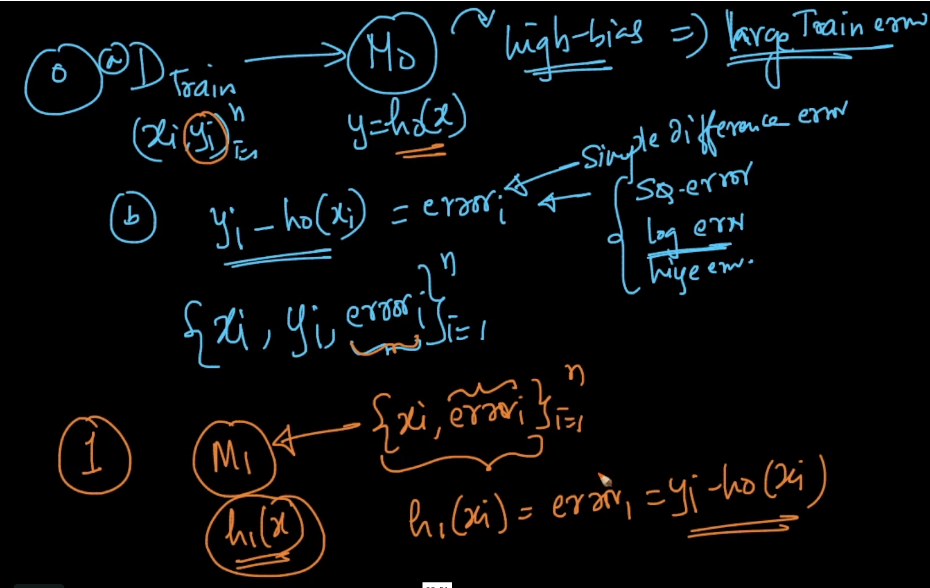
In this first we train model M0(function of M0 is y = h0(x)) by Dtrain and this model is high bias i.e large training error

Now we calculate the error by difference of yi and output generated by model h0(x)

i.e yi – h0(xi) = errori

there are many types of errors like square error, logistic error, hinge error here we use simple error this is for regression if there is classification then we can use logistic error.

Now we have data of { Xi , yi , errori }



**Stage 1 :**

In this stage we train a model M1(h1(x)) using data as xi and errori (this error we got from above model )as shown in above figure.

Here input to model will be xi and output is errori, that mean we are predicting error from input.

Here model h1(xi) gives output which is errori which is nothing but approxy

yi – h0(xi)

now at the end of the stage 1 we got F1(x) (as shown in below pic) which is nothing but weighted sum of two base models

F1(x) = alpha\_0 \* h0(x) + alpha\_1 \* h1(x)

and the value of alpha\_0 and alpha\_1 is hyperparameter we’ll discuss later

**why we are adding h0(x) and h1(x) in F1(x) :**

correct value(yi) = yi’ + error

we use this intuition that by adding predicted value with error we got correct value.

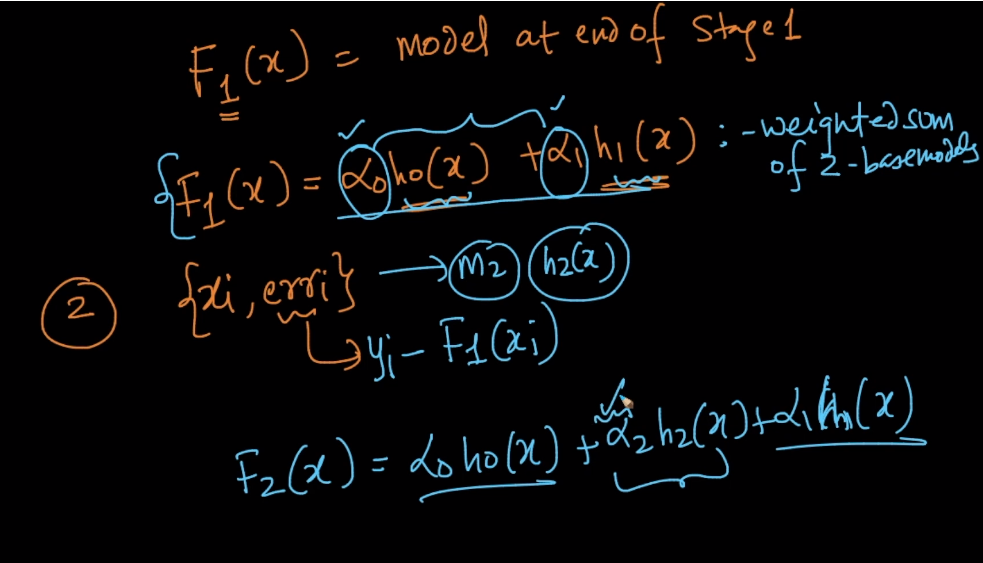
So by model 0 we got predicted value and from model 1 we got predicted error therefore we got correct value but as error got from model 1 is not much enough to sum up by predicted y to get correct y therefore we need to this similar thing for many more models and then sum up all the predicted errors from all models with predicted y got on model 0 i.e at stage 0 to get correct y

**Stage 2:** In this stage we train model M2(h2(x)) by data xi and errori (this error is

Yi – F1(Xi))

After this stage we got F2(x) which is shown in below fig.

F2(x) = alpha\_0 \* h0(x) + alpha\_1 \* h­1(x) + alpha\_2 \* h2(x)



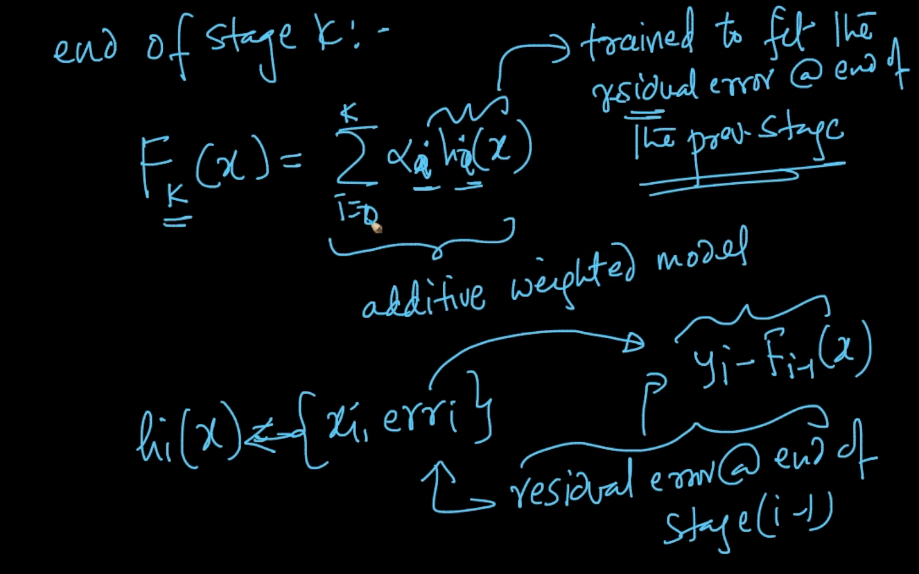
At the end of k stage we got function we got Fk­(x)

F­k(x) = sum\_i=0-to-k(alpha\_i \* h\_i(x))

And because of this it is called additive weighted model

Here hi(x) is trained to fit the residual(remaining) error at the end of the previous stage

The errori we give as a input to model hi(x) is residual error at the end of stage i-1 which is nothing but yi – Fi-1(x)

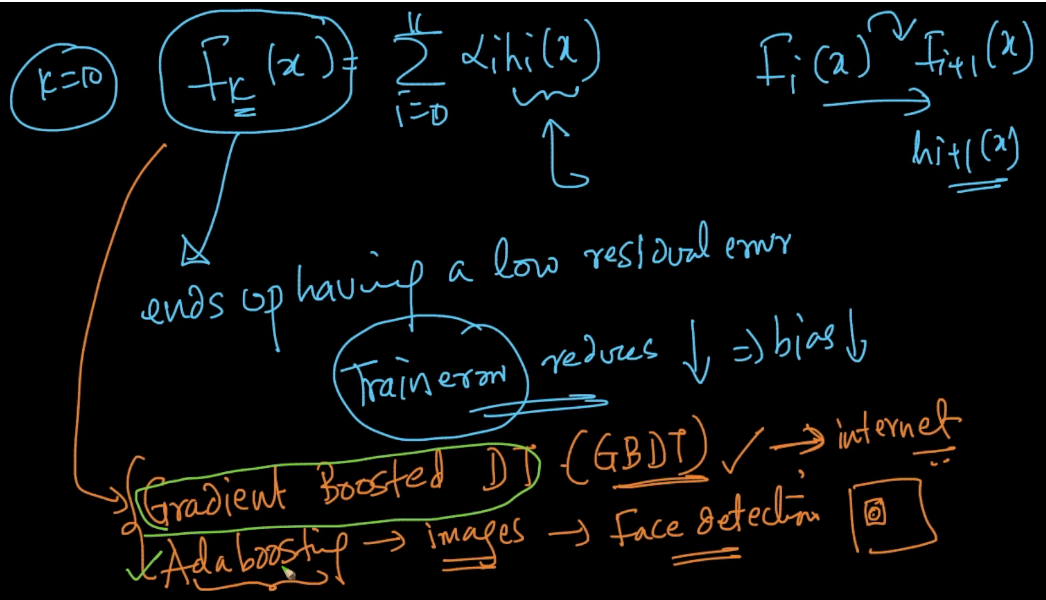


So in this way we are reducing error in each stage by passing residual error from previous stage and at the end Fk(x) ends up having a low residual error.

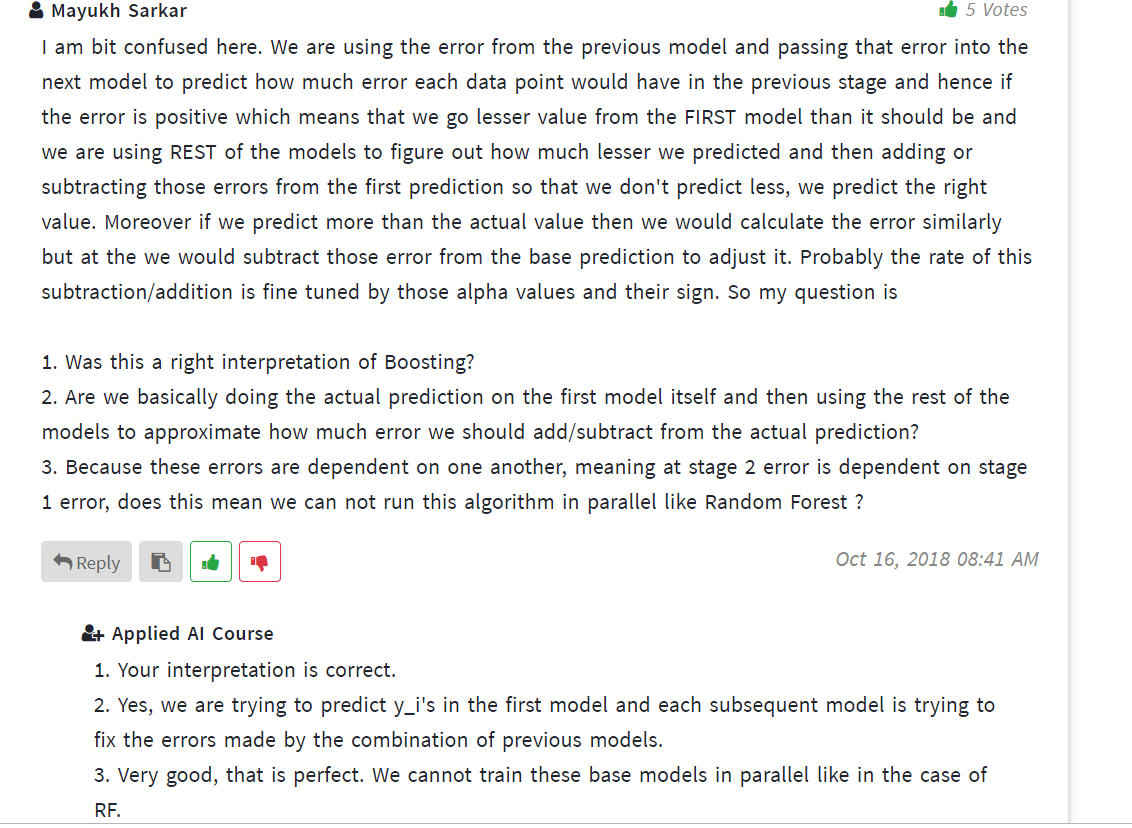
So as the train error reduces, bias also reduces.

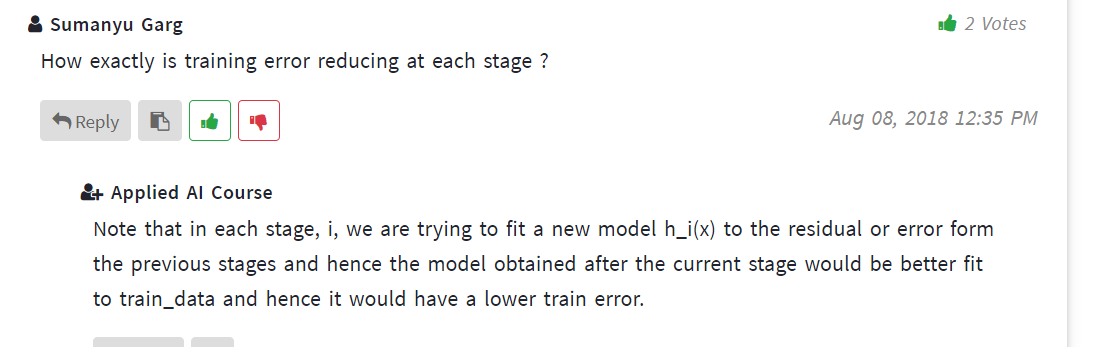
There are many boosting techniques like gradient boosted decision tree(GBDT) which uses very much in internet application

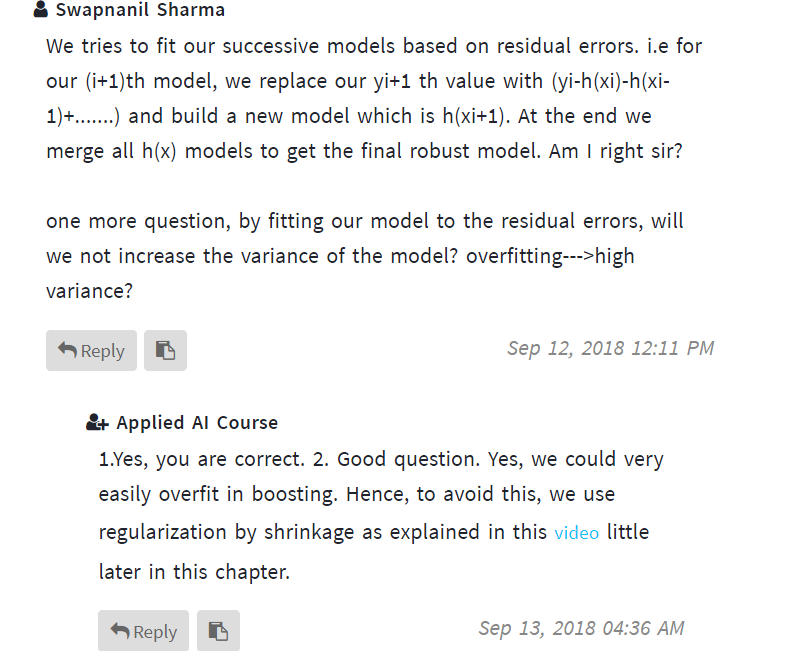
Another technique is Ada Boosting which is used for image like face detection



Comments :







Link :

For classification problem : <https://www.youtube.com/watch?v=gmok1h8wG-Q>