Relationship B/w Variance & Bias. (1)

Both Variance & Bias are reciprocal of each other.

$$B \propto \frac{1}{V}$$

There is always trade of b/w "V" & "B".

For prediction:

$$MSE = Bias^2 + Variance + Irreducible\ error$$

$$E[(y-\hat{y})^2] = E[(y - E(\hat{y}) + E(\hat{y}) - \hat{y})^2]$$

$$= E[(y-E(\hat{y}))^2] + E[(E(\hat{y}) - \hat{y})^2] + 2E[(\hat{y} - E(\hat{y}))(E(\hat{y}) - \hat{y})]$$

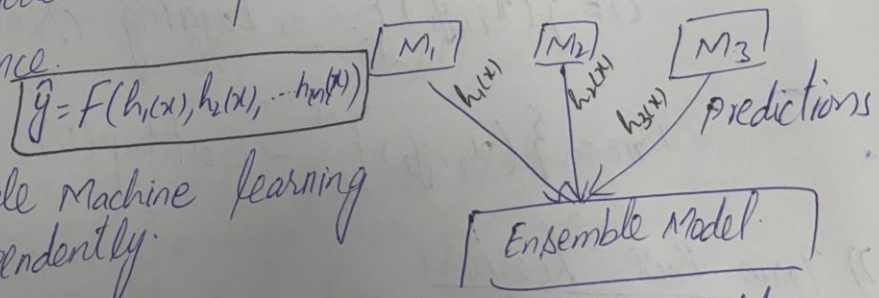
Thus

$$MSE = Bias^2 + Variance + Irreducible\ Error.$$

Ensemble Technique:

A technique that combine Multiple machine learning models to improve overall predictive performance.

$$\hat{y} = F(h_1(x), h_2(x), \cdots h_n(x))$$

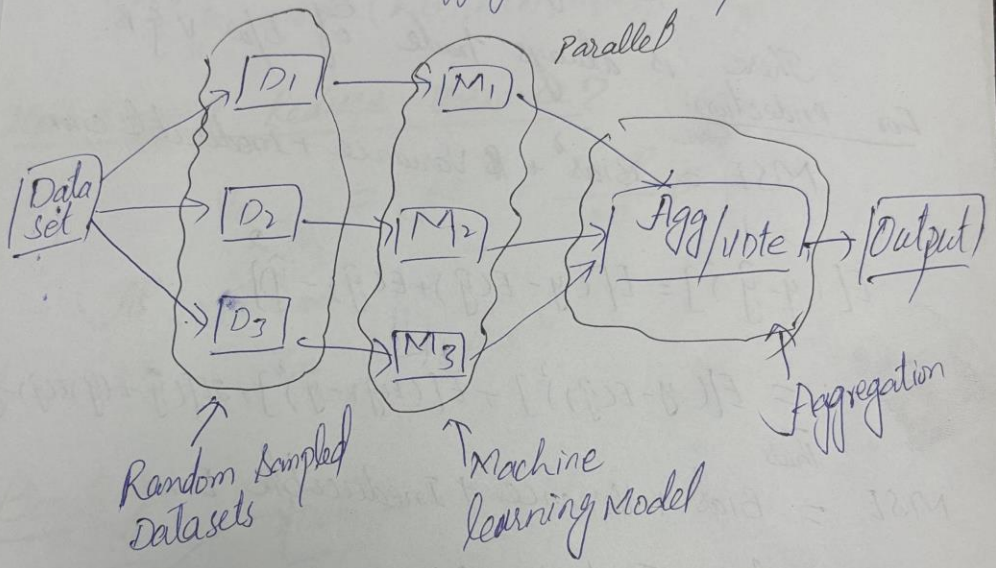


Step 1
→ Multiple Machine learning trained independently.

Step 2:
Done prediction as voting, average or weights.

3) Bagging

Bagging (bootstrap aggregation) is an ensemble method that involve training models independently on random datasets & aggregate their predictions.



Mathematics

1) Bootstrap:

$D = \{(x_i, y_i)\}_{i=1}^{N}$, bagging creates M datasets.

$D_m = \{(x_i, y_i)\}_{i=1}^{N} \sim \text{Bootstrap}(D), \quad m = 1, 2, \ldots, M$

2) Train Base Learner:

$h_m = \text{Train}(D_m)$

3) Combine learners:

For Regression $\hat{y} = \frac{1}{M} \sum_{m=2}^{M} h_m(x)$

For classification

$\hat{y} = \text{mode}(h_1(x), h_2(x), h_3(x), \ldots h_M(x))$

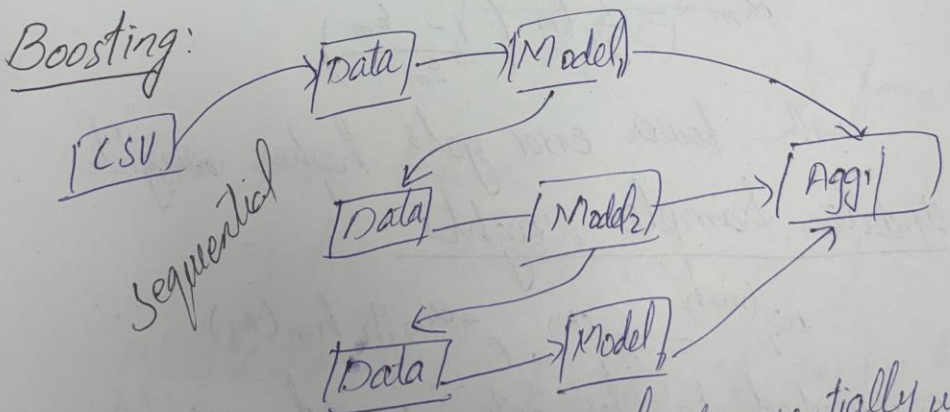
How Bagging Reduce Variance?

$$Var_{bag} = \rho\sigma^2 + \frac{1-\rho}{M}\sigma^2$$

As $M \to \infty$

$$Var_{bag} \to \rho\sigma^2$$

Boosting:



Boosting models are trained sequentially with each model learning from the errors of previous one. Boosting assign weights based on accuracy. Boosting reduce bias.

Mathematics of Boosting: (Adaboost, Gradient Boosting)

1) Initialize sample weights

$$w_i^{(1)} = \frac{1}{N}$$

2) Train weak learner

At round m.

$h_m(x)$

computed weight error

$$\varepsilon_m = \sum_{i=1}^{N} w_i^{(m)} \cdot 1(h_m(x_i) \neq y_i)$$

3) Computed Model weight

$$\alpha_m = \frac{1}{2} \ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$$

with lower error gets higher weight.

4) Update Sample weights.

$$w_i^{(m+1)} = w_i^{(m)} \cdot e^{-\alpha_m y_i h_m(x_i)}$$

Misclassified samples gets increased weights.

Normalization

$$w_i^{(m+1)} = w_i^{(m+1)} \Big/ \sum_{i=2}^{N} w_j^{(m+1)}$$

5) Final Output

$$H(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m h_m(x)\right)$$

Gradient Boost:

Negative gradient of loss function

Given $L(y, f(x))$

→ Start initial model $f_0(x)$

→ At each step fit pseudo-residuals

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f = f_{m-1}}$$

Train weak learner

$$h_{(m)} = \text{Train}\left(x_i, r_i^{(m)}\right)$$

update Model $f_m(x) = f_{m-1}(x) + \eta \cdot h_m(x)$

η = learning rate

Stacking

use prediction of Multiple Models as input

Math

Given $z_m(x) = h_m(x)$

New Dataset

$$Z = \{(z_1(x_1), z_2(x_2), \dots z_m(x_i), y_i)\}$$

Train meta learner

$$g = \text{Train}(Z)$$

Final Prediction

$$\hat{y} = g(h_1(x), h_2(x), \dots h_M(x))$$