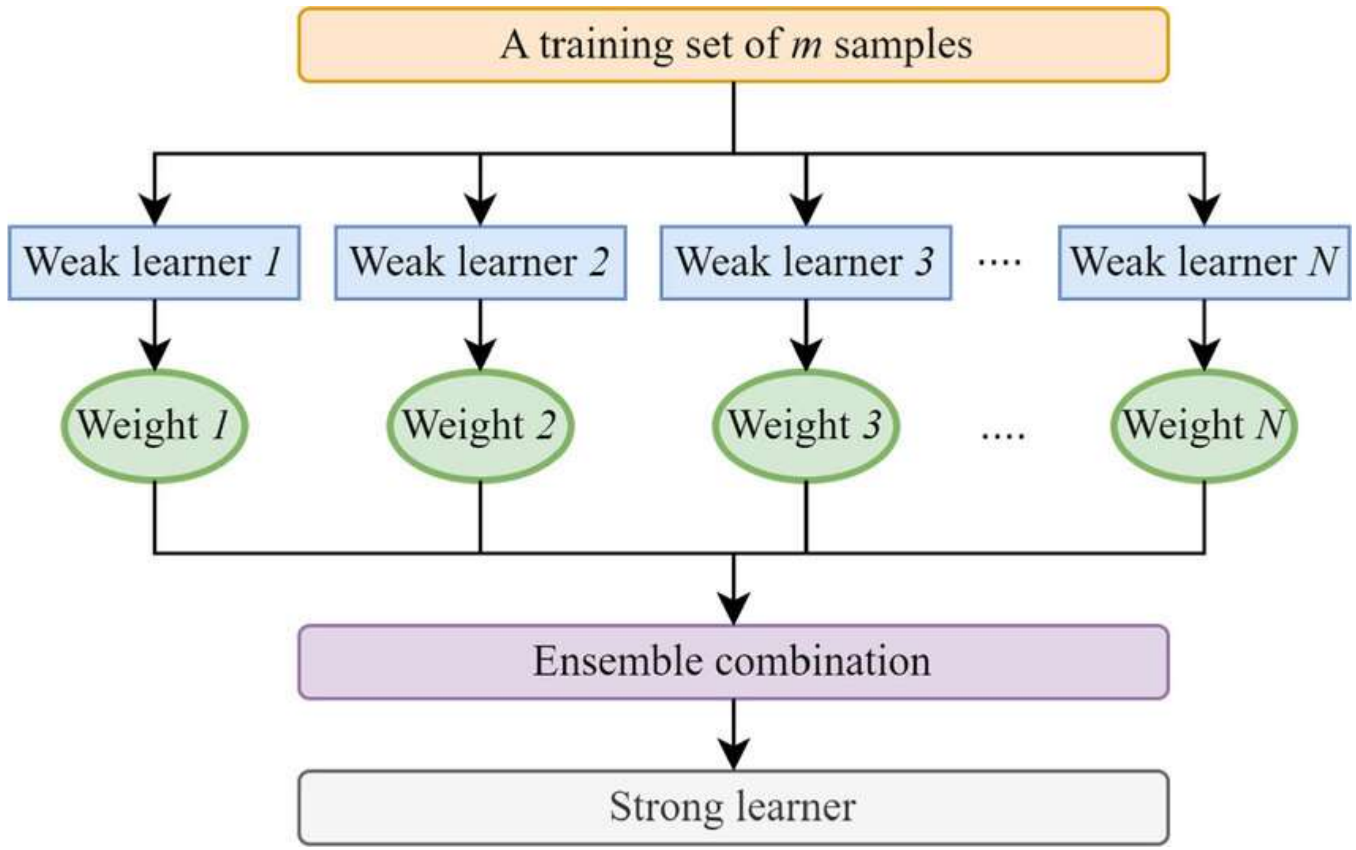
BOOSTING ALGORITHMS

JEYARAKKINI J

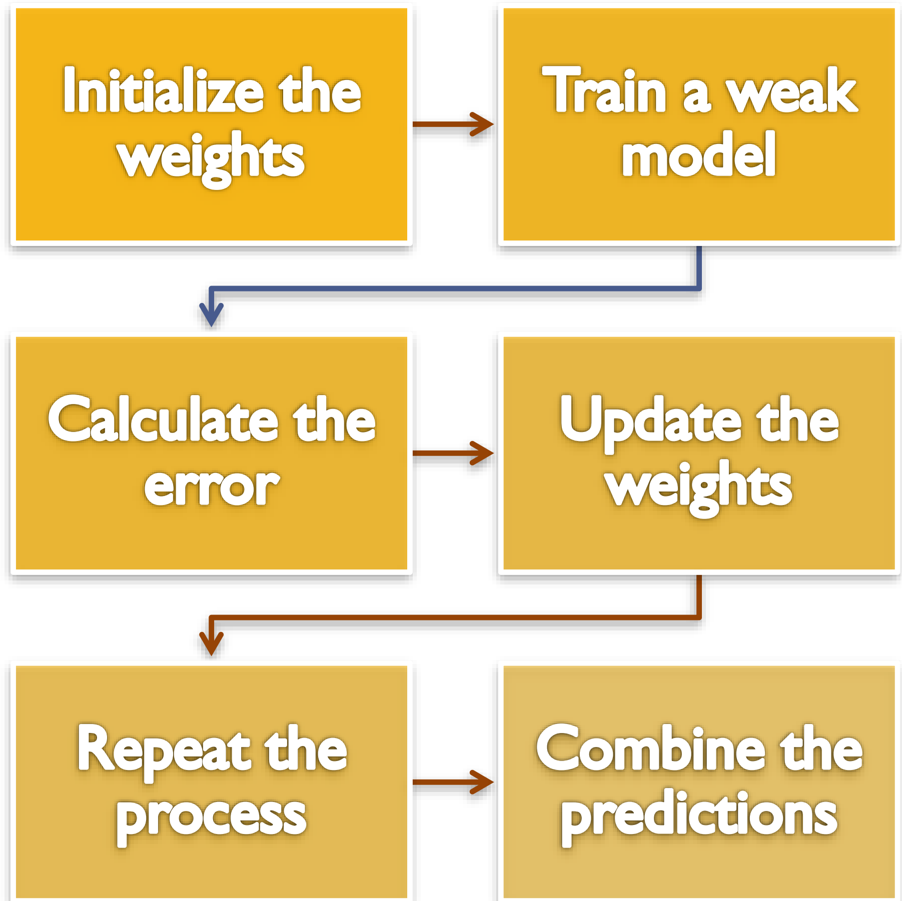
ADA BOOST FOR REGRESSION

- A machine learning algorithm for predicting continuous outcomes.
- Ada Boost algorithm falls under ensemble boosting technique.
- The base model can be a linear model or a decision tree.

ADA Boosting algorithm combines multiple weak models to create a strong predictive model.



Process


Initialize the weights
Train a weak model
Calculate the error
Update the weights
Repeat the process
Combine the predictions

$$y = \sum(\alpha_m * h_m(x)) + c$$

y : predicted output value

α_m : weight assigned to the mth weak model

$h_m(x))$: prediction made by mth weak model

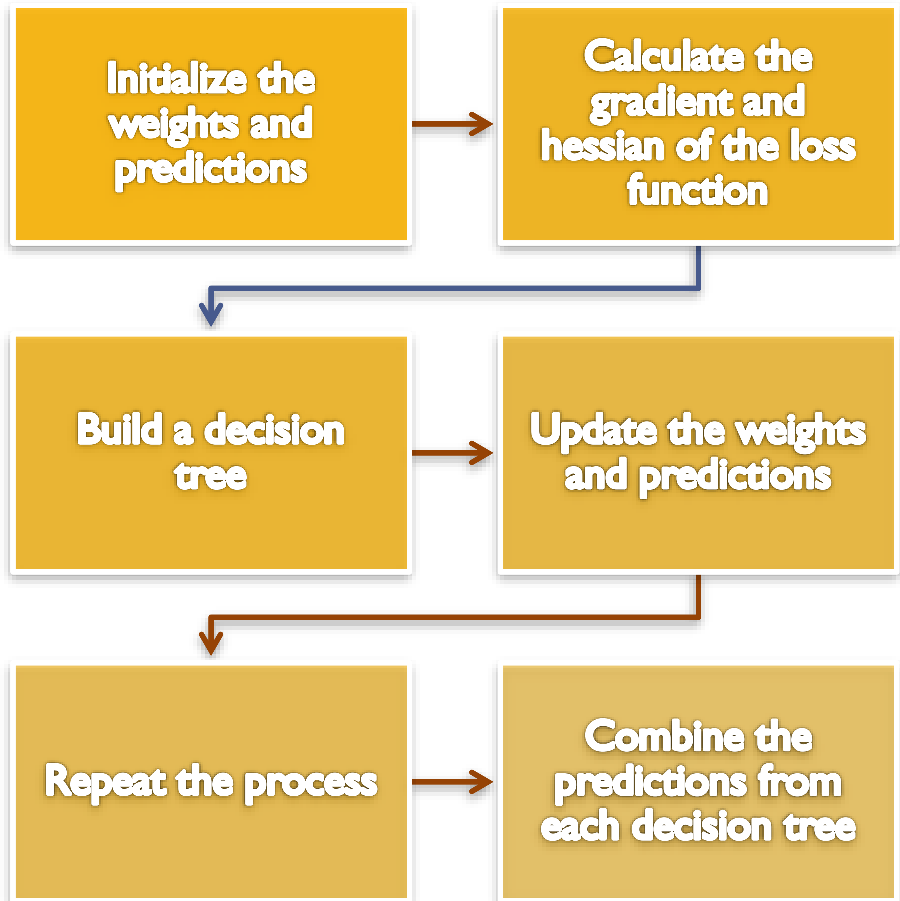
c : bias term

$\sum$: indicates combination of predictions from multiple weak models.

XG BOOST REGRESSION

- Popular gradient boosting algorithm, widely used for regression tasks
- Known for its speed and accuracy
- Uses a decision tree as a base model

Process


Initialize the weights and predictions
Calculate the gradient and hessian of the loss function
Build a decision tree
Update the weights and predictions
Repeat the process
Combine the predictions from each decision tree

$$\text{Objective} = \text{Loss} + \text{Regularization Term}$$

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \text{Loss}(y_i, \hat{y}_i) + \sum_{k=1}^{K} \left(\gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \right)$$

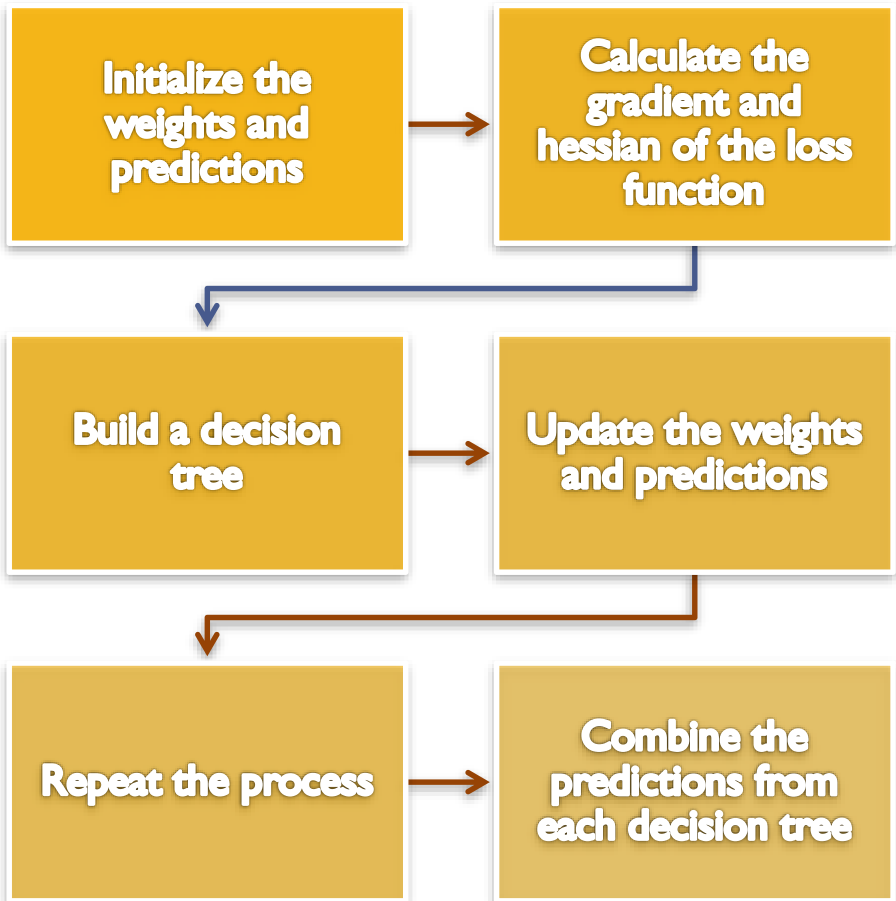
Loss - measures the error.

Regularization - prevents the model from being too complex and over fitting the data.

LG BOOSTING IN REGRESSION

- LightGBM (Light Gradient Boosting Machine) is a fast and efficient implementation of gradient boosting.
- It is designed to handle large datasets, and it is known for its speed and scalability.

Process


Initialize the weights and predictions
Calculate the gradient and hessian of the loss function
Build a decision tree
Update the weights and predictions
Repeat the process
Combine the predictions from each decision tree

Initialize the model by assigning weight:

$$\hat{y}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Calculate the error r_i (residuals):

$$r_i = y_i - \hat{y}_{m-1}(x_i)$$

Fit the weak model to the residual:

$$h_m(x_i) = \arg\min_h \sum_{i=1}^{n} [y_i - \hat{y}_{m-1}(x_i) - h(x_i)]^2$$

Update the model:

updated prediction = previous prediction + (learning rate * error))

$$\hat{y}_m(x) = \hat{y}_{m-1}(x) + \eta \cdot h_m(x)$$

П – learning rate

Final model after M prediction:

Adding all the updated predictions.

$$\hat{y}_M(x) = \hat{y}_0(x) + \sum_{m=1}^{M} \eta \cdot h_m(x)$$

Key differences:

Feature	AdaBoost	LG Boosting	XD Boost
Base learner	Decision tree (usually stumps)	Decision tree	Decision tree (can be deeper)
Loss function	Exponential loss	Logistic loss	Customizable loss with regularization
Handling of errors	Focuses on misclassified points	Focuses on residuals	Focuses on residuals with regularization
Speed and Efficiency	Moderate, as it doesn't parallelize much	Moderate. Can be slow for large datasets	Very fast, highly optimized with parallelization
Overfitting Control	Weak overfitting control, sensitive to noisy data	Moderate overfitting control	Strong overfitting control via regularization

Conclusion:

- Boosting improves accuracy of the model.
- Most commonly, XD boosting algorithm is used as it provides high accuracy,