

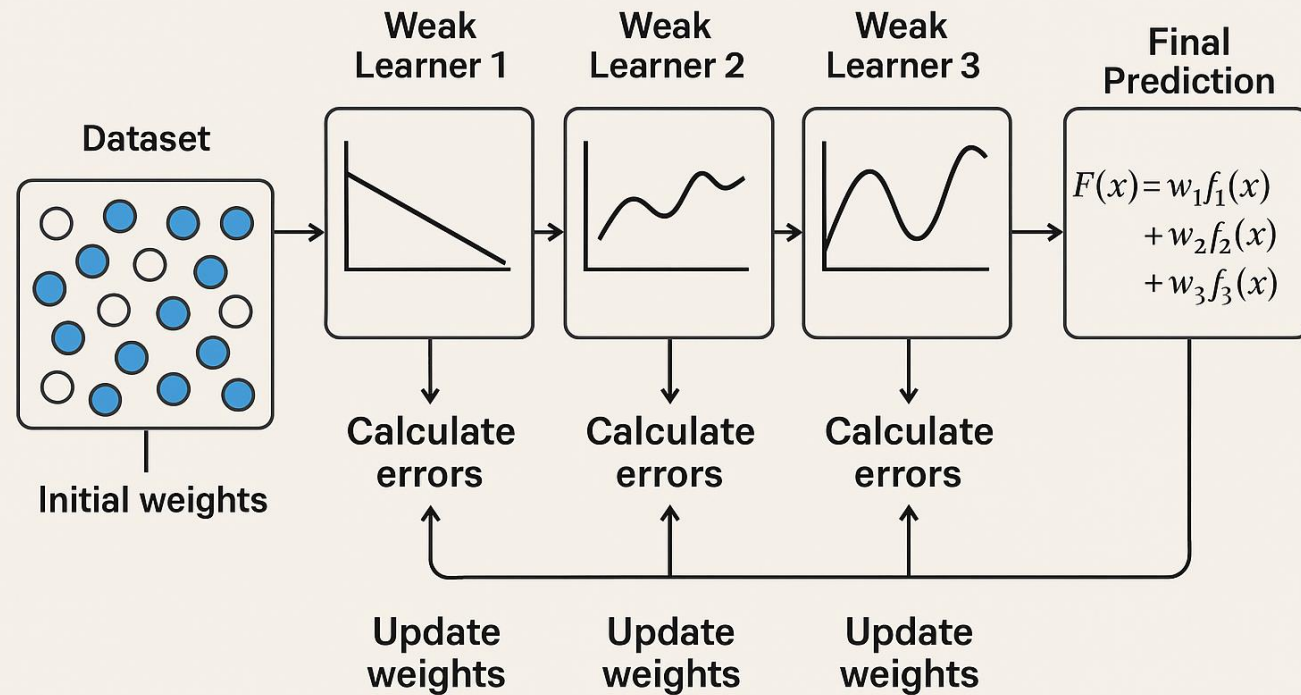
# BOOST ALGORITHM-REGRESSOR

1. ADA BOOST REGRESSOR
2. XG BOOST REGRESSOR
3. 3. LG BOOST REGRESSOR

# ADA BOOST REGRESSOR

- ADA BOOST(Adaptive Boosting) is an ensemble learning technique that improves the performance of weak learners by combining multiple models iteratively. In regression, AdaBoost assigns weights to data points and adjusts them based on prediction errors, focusing more on difficult-to-predict samples.
- **How AdaBoost Regressor Works**
  1. **Initialize Weights:** Each data point starts with equal weight.
  2. **Train Weak Learners:** A weak model (often a decision tree) is trained on the weighted dataset.
  3. **Calculate Errors:** The model's performance is evaluated, and misclassified points get higher weights.
  4. **Update Weights:** The next weak model focuses more on difficult cases.
  5. **Combine Models:** The final prediction is a weighted sum of all weak learners.

# AdaBoost Regressor



- In the dataset visualization, the blue marker rounds typically represent the **weights assigned to data points**. Initially, all data points have equal weights, but as AdaBoost iterates, the weights of harder-to-predict points increase. This forces subsequent weak learners to focus more on those difficult cases, improving overall accuracy.
- Am using insurance\_prediction dataset:
- The number of weak learners is controlled by the parameter `n_estimators`. The `learning_rate` parameter controls the contribution of the weak learners in the final combination. By default, weak learners are decision stumps. Different weak learners can be specified through the `estimator` parameter. The main parameters to tune to obtain good results are `n_estimators`

# Evaluation Metrics : R value

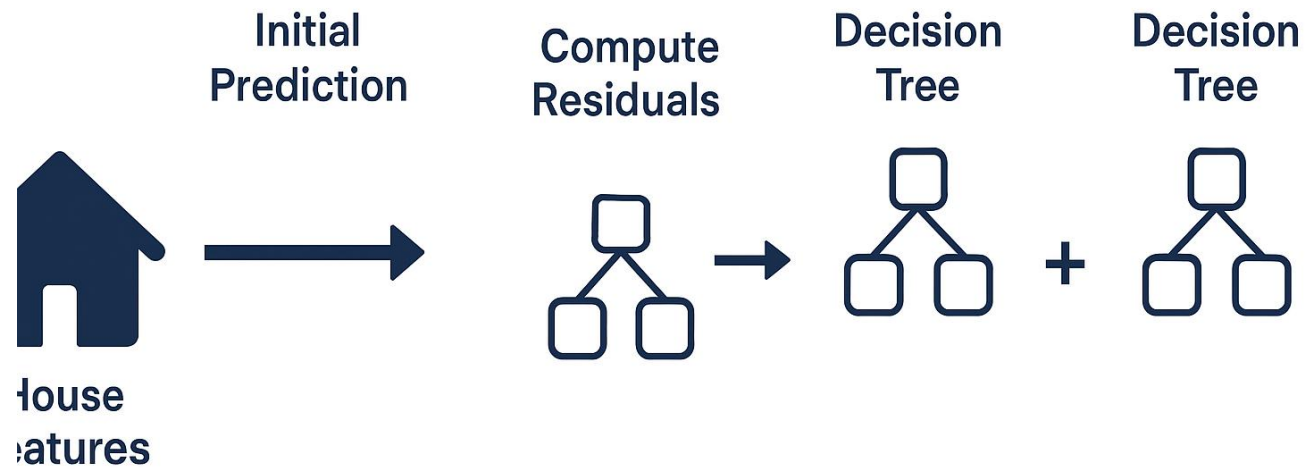
SNO	N_estimators	Random state	Learning rate	R value
1	150	0	1.0	0.844747
2	200	2	1.5	0.8224836
3	100	0	1.0	0.844747
4	100	50	1.0	0.848572
5	200	90	1.0	0.862539

Best model in ADABOOST  
REGRESSOR : 0.862539

# XG BOOST REGRESSOR

- XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm used for **regression** and **classification** tasks. It's an optimized version of gradient boosting that is fast and efficient, making it popular in real-world applications.
- **How XGBoost Works (Simplified)**
- XGBoost is built on **ensemble learning**, meaning it combines multiple weak models (decision trees) to create a strong predictive model. Here's the core idea:
  - 1. Starts with a simple decision tree**
    1. The first tree makes initial predictions, often not very accurate.
  - 2. Calculates errors (residuals)**
    1. It looks at how far the predictions are from the actual values.
  - 3. Builds new trees to correct previous mistakes**
    1. Each new tree **focuses on reducing the errors** of the previous one.
    2. Trees are added sequentially, each improving the model.
  - 4. Combines all trees for final predictions**
    1. The trees work together to produce a **strong, accurate model**.

# XGBoost Regressor



Final Prediction = Initial Prediction  
+ Trees' Contributions

# Evaluation Metrics: R Value

SNO	N_estimators	Learning_rate	Max_depth	R value
1	100	0.1	3	0.8883087
2	200	0.8	5	0.77798
3	100	0.3	3	0.8847196

BEST MODEL OF XG REGRESSOR : 0.8883087



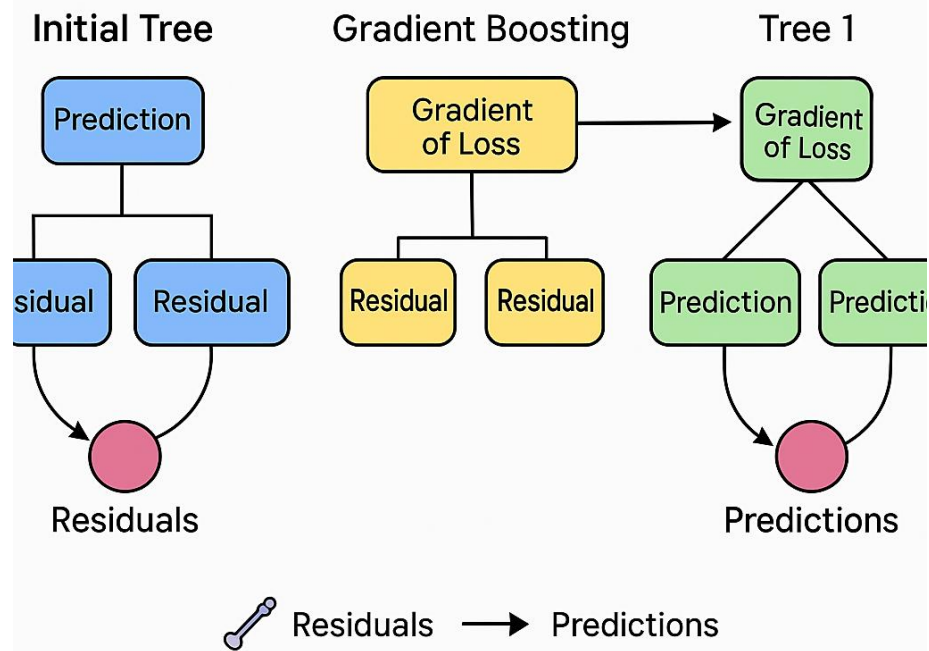
# LG BOOST REGRESSOR

- LightGBM (Light Gradient Boosting Machine) is a powerful and efficient **gradient boosting algorithm** that is designed for **speed and scalability**. It is particularly useful when working with large datasets because it optimizes how trees grow.
- **How LightGBM Works (Step-by-Step)**
  - 1. Starts with an initial weak model**
    1. The first tree makes basic predictions.
    2. Errors (residuals) are calculated.
  - 2. Focuses on reducing errors**
    1. New trees are added iteratively.
    2. Each tree corrects the previous errors using **gradient boosting**.
  - 3. Uses leaf-wise growth strategy**
    1. Instead of growing trees **depth-wise**, LightGBM grows them **leaf-wise** (optimizing splits).
    2. This **reduces loss faster** than traditional boosting.
  - 4. Efficient feature selection**
    1. LightGBM automatically selects relevant features.
    2. Uses histogram-based computation for speed.

# Difference between LG Boost and XG Boost

Feature	LightGBM	XGBoost
<b>Tree Growth</b>	Leaf-wise	Depth-wise
<b>Training Speed</b>	Faster (uses histograms)	Slower than LightGBM
<b>Handles Large Data</b>	Excellent	Good
<b>Works with Missing Values</b>	Yes	Yes

# LightGBM



# Evaluation Metrics: R value

SNO	N_estimators	Learning_rate	Max_depth	Num_leaves	reg_alpha	R value
1	100	0.1	2	20	Le-3	0.88685
2	100	0.1	12	20	Le-3	0.84473
3	100	0.2	3	20	Le-3	0.84473
4	90	0.3	3	20	Le-3	0.883085

BEST MODEL OF LG BOOST REGRESSOR : 0.88685