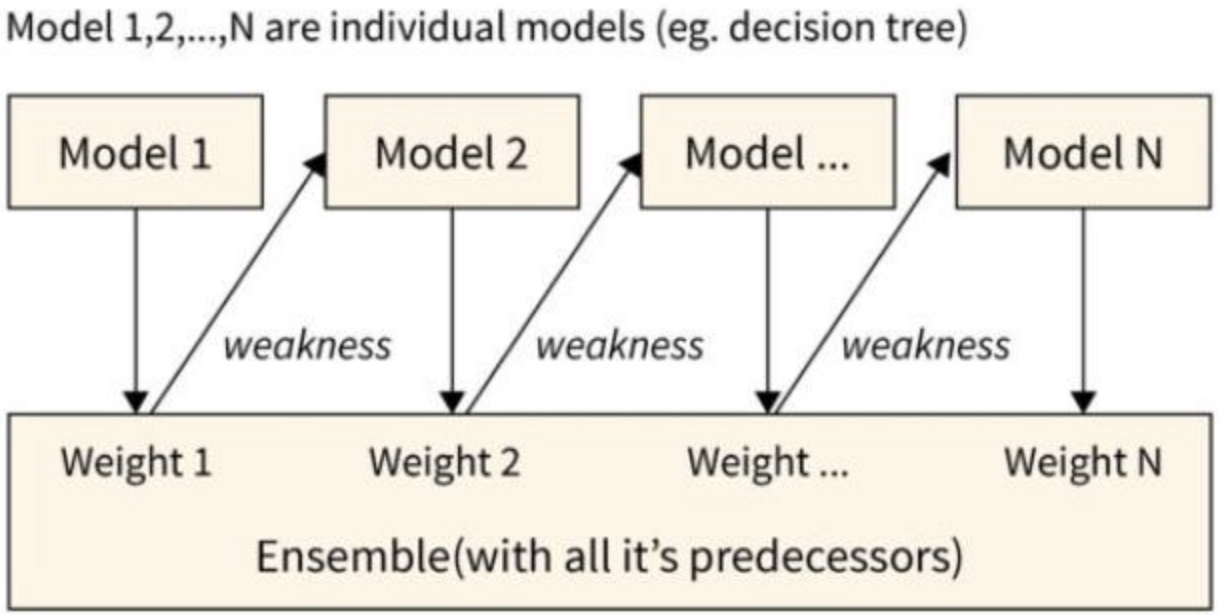
Ada boost - Regression

AdaBoost, short for Adaptive Boosting, works by sequentially combining multiple "weak" learning models (like decision trees) to create a single "strong" model.

Ada boost Regression diagram



- **Start**: Give all data points equal importance (weights).
- **Train Weak Models**: Build a simple model (e.g., a small decision tree) to make predictions.
- **Adjust Focus**: Increase the weight of data points the model got wrong, so the next model pays more attention to them.
- **Combine Models**: Add the new model to the ensemble, giving it a weight based on how accurate it was.
- **Repeat**: Keep training new models and adjusting weights until a set number of models is reached or performance stops improving.
- **Final Prediction**: Combine all models' predictions, with better models having more influence, to get the final result.

This formula is used in boosting (like AdaBoost) to decide how much weight to give to a weak learner based on its error.

$$\alpha_t = \frac{1}{2} ln \frac{(1 - TotalError)}{TotalError}$$

- α_t is the weight for the learner.
- $\mathrm{TotalError}$ is the fraction of mistakes it makes.
- The $\ln$ part compares correct predictions $(1 - \mathrm{TotalError})$ to mistakes $(\mathrm{TotalError})$.
- The $\frac{1}{2}$ adjusts the weight.

If the learner is good (low error), it gets a higher weight (α_t). If it's bad (high error), it gets a lower weight. This helps the algorithm focus on fixing mistakes in the next round.

The final prediction formula for the AdaBoost algorithm, when used for regression its weighted sum of the predictions from all weak learners.

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

- $F(x)$: The final prediction for input x.
- T: The total number of weak learners (iterations).
- $h_t(x)$: The prediction of the t-th weak learner for input x.
- α_t: The weight assigned to the t-th weak learner, calculated as $\alpha_t = \frac{1}{2} \ln \frac{(1-\text{TotalError})}{\text{TotalError}}$, reflecting its accuracy.

In practice, the weak learners' predictions are combined, with better-performing learners (lower error) contributing more to the final output through higher α_t values.