

# Ada Boost

---

1. Ada boost starts with building a very small tree(level=1) a stump from the training data.
2. The amount of say that the stump has on the final output is based on how well it compensates for those previous errors.
3. Then Ada Boost builds the next stump based on the errors that the previous stump made.
4. It keeps on building the stump on the errors that the previous stump made until it has made the number of stumps you asked for , or it has a perfect fit.

## Gradient Boost for Regression

---

1. Loss function

$$Loss = \frac{1}{2}(observed - predicted)^2$$

1. We start with a leaf that is the average value of the variable we want to predict.
2. Then we add a tree(larger than the stump in practice leaf could be between 8 and 32) based on the residuals(difference between the Observed values and the Predicted values.)
3. Calculate the output values of the leaves of the fitted tree.The output value is the average of the value of the records present in that leaf.
4. We scale the tree's contribution to the final Prediction with a Learning Rate. Unlike in ada boost where all trees are scales by the same amount.
5. Then we add another tree based on the new residuals and we keep adding new trees based on the errors made by the previous tree.

## Gradient Boost for Classification

---

1. We start with a leaf that represents the initial prediction for every individual record using log(odds). For eg. In binary classification with 60 Yes and 40 No.

$$\log(odds) = 60/40 = 1.5$$

2. We convert log(odds) to probability using Logistic function to use it for classification.

$$P(\text{Yes}) = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

3. For each record,calculate the pseudo residual ,the difference between the Observed and the predicted.
4. Then we add a tree(larger than the stump in practice leaf could be between 8 and 32) based on the residuals(difference between the Observed values and the Predicted values.)
5. For each leaf node in the tree,calculate the output values of the leaf of the fitted tree. As the output of the leaf is represented as probability , we need to do a transformation to make it log(odds) comparable with the previous predictions for residual calculation.

$$T = \frac{\sum Residuals_i}{\sum [Previous Probability_i * (1 - Previous Probability_i)]} \text{ where } i \text{ is the records in the leaf node}$$

3. For each record, we scale the tree's contribution to the final log(odds) with a Learning Rate.
4. For each record, convert the log(odds) prediction into predicted probability.

$$Probability = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

5. For each record, calculate the new residuals, the difference between the Observed and the predicted probability.
6. Then we add another tree based on the new residuals and we keep adding new trees based on the errors made by the previous tree.

## Reference

---

[Gradient-Boosting](#)