

DATA  SCIENCE
D R E A M J O B

Boosting Trees

One tree is not enough...

Table of Contents

1. Introduction

- a. What is Boosting?
- b. Strengths/Weaknesses of Boosting
- c. Important Concepts / Terminologies
- d. AdaBoost vs Gradient Boosting

2. AdaBoost

- a. Explanation & Example

3. Gradient Boosting

- a. Explanation & Example
- b. Algorithm Deep Dive
- c. Gradient Descent == Gradient Boosting?

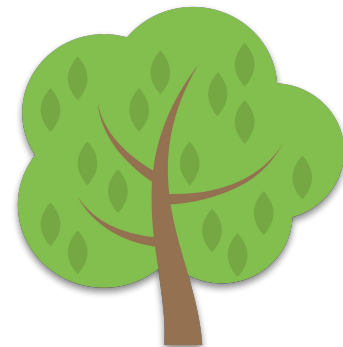
4. Notebook Code Demo

- a. XGBoost Implementation
- b. Hyperparameters & Learning Curves
- c. Implement GridSearchCV
- d. Understand Early Stopping

5. Activity

6. Questions

Introduction to Boosting



Goal: Convert weak learners → Strong Learner

- ❑ Combines simple models into one composite model
- ❑ **Composite Model = Simple Model (1) + Simple Model (2) + ...**
- ❑ Each simple model **focuses on the errors** of the previous one
- ❑ Each model is dependent upon the previous one
- ❑ Each model that is added tries to improve the overall performance of the ensemble

Gradient Boosting Strength/Weakness

Strengths:

- ❑ Robust + Powerful
- ❑ Directly optimizes cost function
- ❑ Works for both regression/classification
- ❑ Can capture non-linear relationships
- ❑ Can capture various interactions in data
- ❑ Implicit variable selection (feature importance)

Weaknesses:

- ❑ Requires careful tuning
- ❑ Prone to overfitting
- ❑ Several hyperparameters

Important Concepts / Terminologies

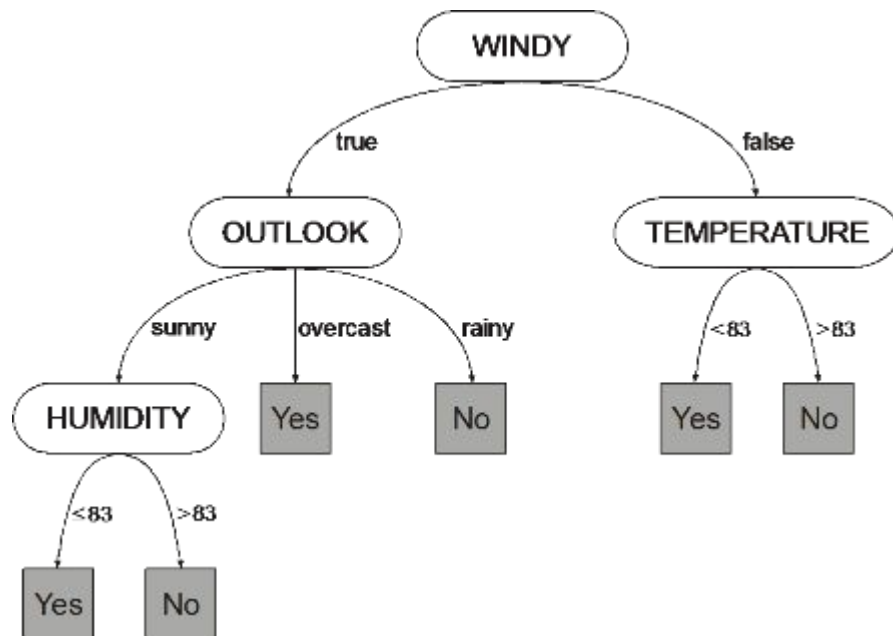
1. What is a Decision Tree?
2. What is an Ensemble?
3. What is an Additive Model?
4. What is a Residual/Error?
5. What is Gradient Descent? What is a Loss Function?

What is a Decision Tree?

Decision Trees look for the best split in your data and partitions it into sub-groups.

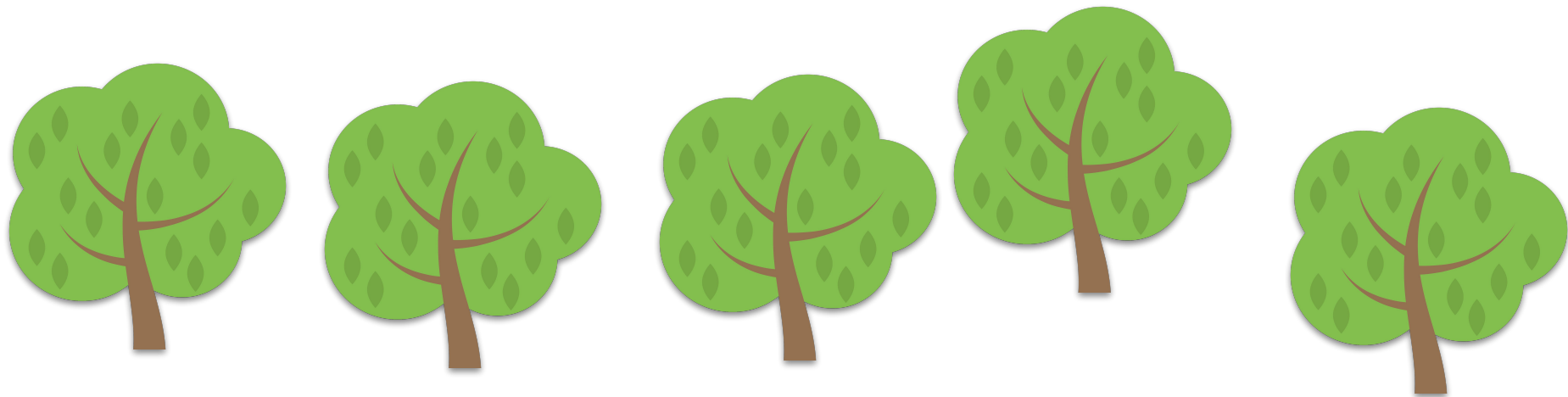
The importance of a feature is defined on top of the tree and is lowered as it goes down

1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
4. **Leaf/Terminal Node:** Nodes do not split is called Leaf or Terminal node.

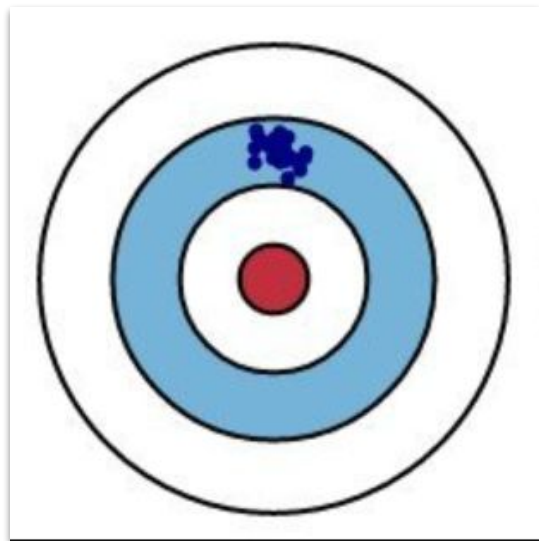


What is an Ensemble?

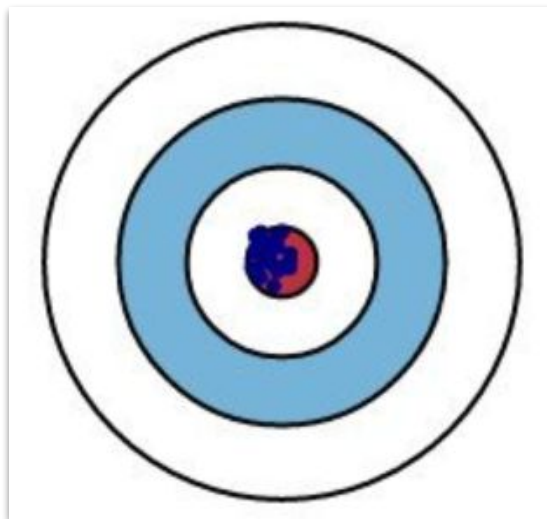
- ❑ It's a **collection** of predictors that comes together to make a final prediction
- ❑ Ensembles learn many weak classifiers that are good at different parts of the input space.
- ❑ Helps reduce bias and variance error
- ❑ **Two Types:** Boosting & Bagging



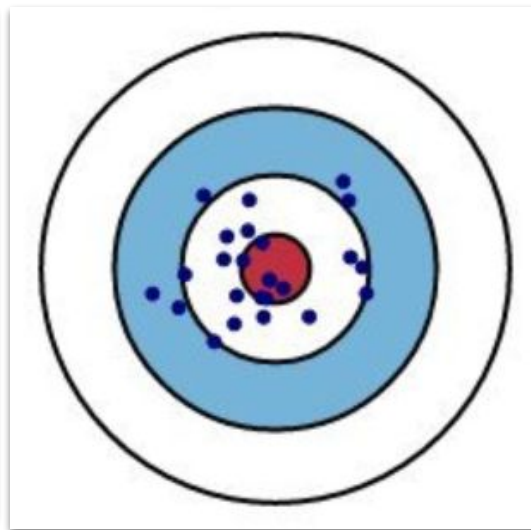
Bias Variance Tradeoff



High Bias



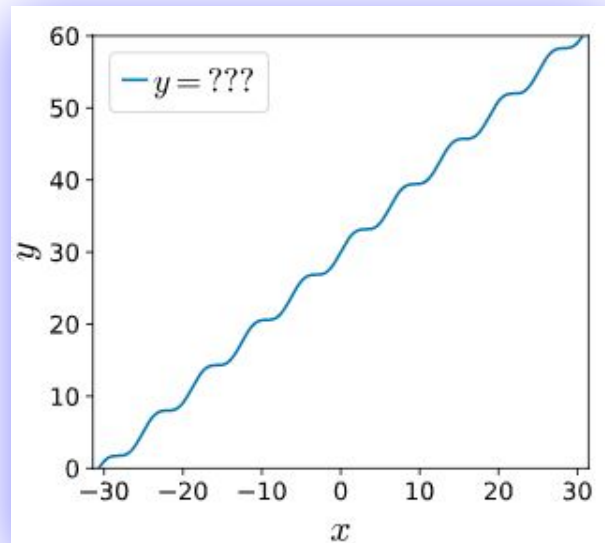
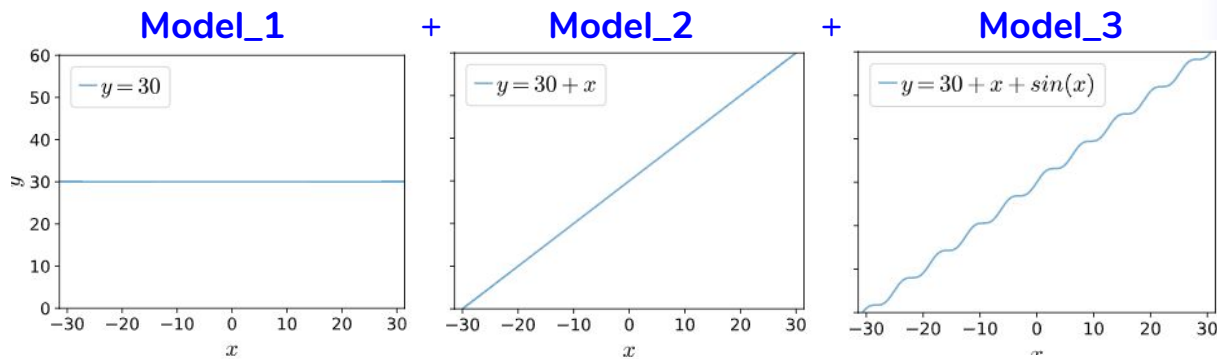
Low Bias & Low Variance



High Variance

What is an Additive Model?

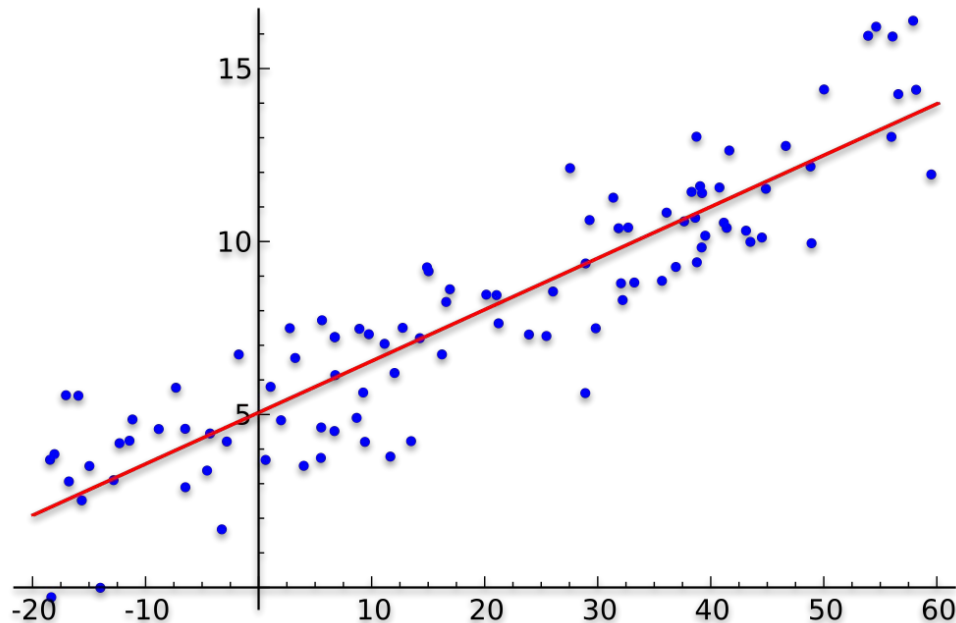
- ❑ **Big Idea:** Add a bunch of simple functions (models) to create a more complex function
- ❑ Boosting is an Additive Model
- ❑ Foundation of boosting algorithm
- ❑ $F(x) = f_1(x) + f_2(x) + f_3(x)$



What function is this??

What is a Residual?

- ❑ Residual = Actual - Predicted
- ❑ Also known as the **errors**
- ❑ Tells you how far off you are from the actual value



Source: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3a/Linear_regression.svg/1000px-Linear_regression.svg.png

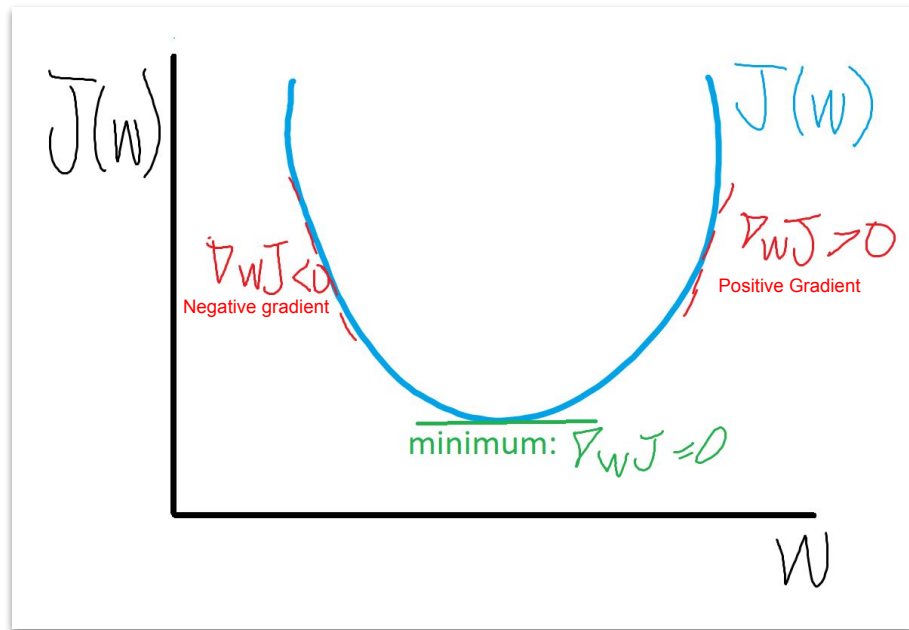
What is Gradient Descent?

What is a Loss Function?

Goal of Gradient Descent:
Find optimal weight that
minimizes the loss function

$$Loss = MSE = \sum (y_i - y_i^p)^2$$

where, y_i = ith target value, y_i^p = ith prediction, $L(y_i, y_i^p)$ is Loss function



Types of Boosting Methods

1. AdaBoost 
2. Gradient Boosting 
3. XGBoost 
4. LightGBM 
5. CatBoost 

AdaBoost vs Gradient Boosting

They differ on **how they create the weak learners** during the additive stages.

- ❑ In each stage, the learner will try to **fix** the “**weaknesses**” of the previous learner.
- ❑ Gradient Boosting uses **different** training samples at each stage
- ❑ AdaBoost uses the **same** training samples at each stage

Gradient Boosting

“Weaknesses” = “Residuals”

AdaBoost

“Weaknesses” = “High-weight data points”

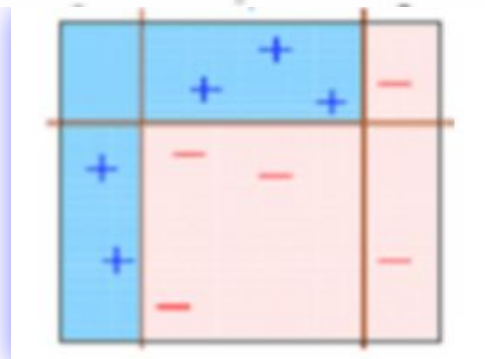
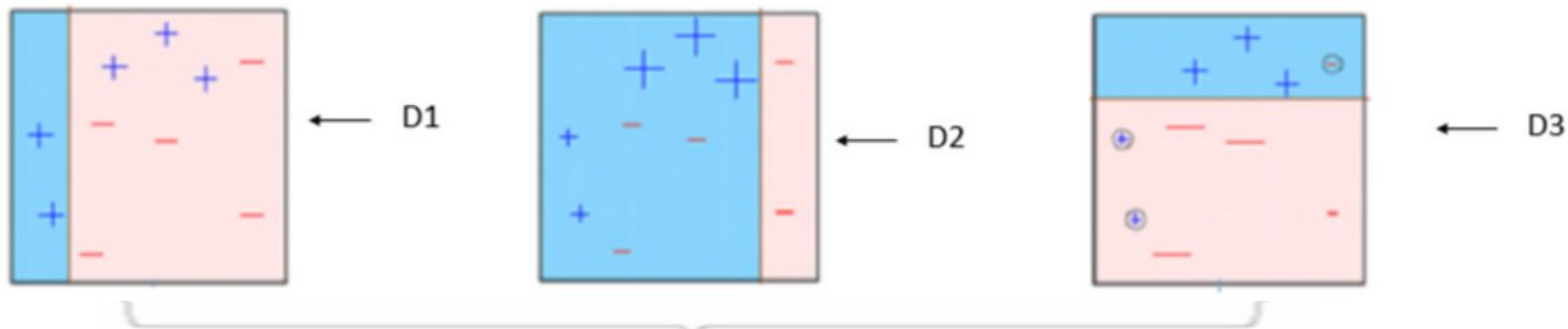
- ❑ These “**weaknesses**” tell us how to improve our model

AdaBoost

General idea is to have each learner concentrate on instances that are difficult to correctly classify

- ❑ A model is fitted in a **forward stage-wise** fashion
- ❑ **Increases** the weights of the **wrongly** predicted instances
- ❑ **Decreases** the weights of the **correctly** predicted instances
- ❑ Weak learner focuses more on the difficult instances (errors) from the previous learner

AdaBoost Example



D1 + D2 + D3
(Ensemble)

Gradient Boosting

- ❑ Intuitively, Gradient Boosting is a **residual fitting method**
- ❑ A model is fitted in a **forward stage-wise** fashion
- ❑ Focuses on learning the remaining errors (aka **error fitting**)
- ❑ Can utilize other loss functions

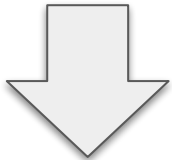
3 Parts of Gradient Boosting

1. A **loss function** to be optimized
2. A **weak learner** to make predictions
3. An **additive model** to add weak learners to minimize loss function

- ❑ **Loss function** must be differentiable (e.g. MSE, MAE, LogLoss)
- ❑ **Weak learners** are usually decision trees
- ❑ An **additive model** is a model of many other learners

3 Steps of Gradient Boosting

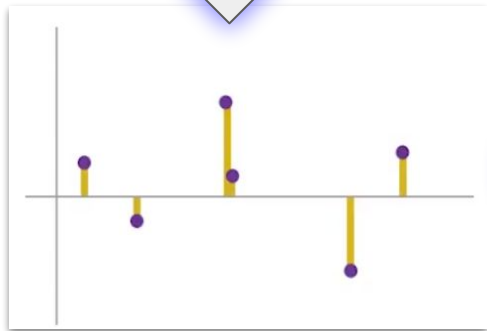
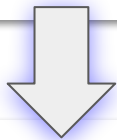
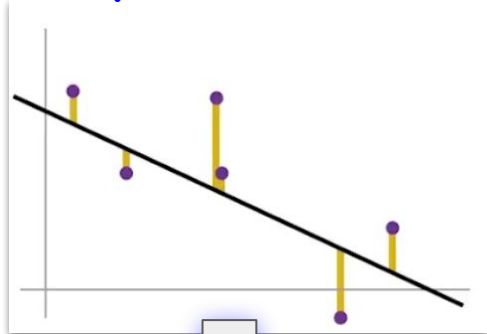
1. You fit a model
2. You compute the errors or residuals
3. You fit another model to the residuals



Combine models together

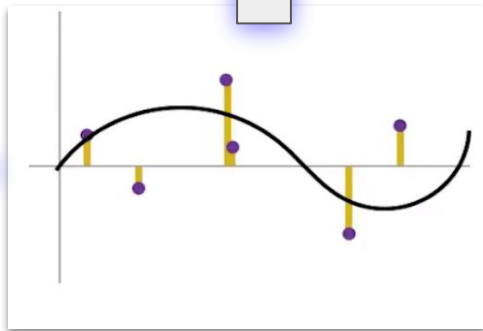
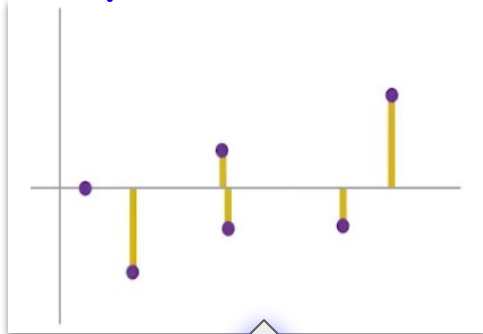
Gradient Boosting Example

Step 1: Fit base model (1)



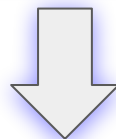
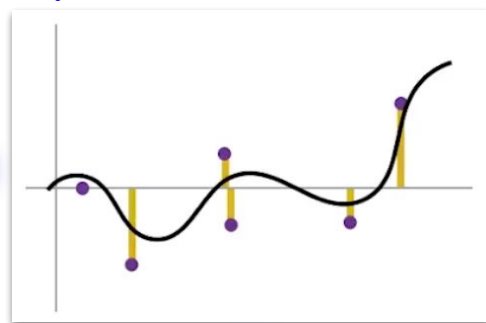
Step 2: Compute residuals

Step 4: Compute residuals



Step 3: Fit model (2) on residuals

Step 5: Fit model (3) on residuals



Step 6: Combine models

$$M_1(x) + \gamma_2 m_2(x) + \gamma_3 m_3(x)$$

model(1)

model(2)

model(3)

Gradient Boosting Algorithm

From Wikipedia:

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, number of iterations M .

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma).$$

2. For $m = 1$ to M :

1. Compute so-called *pseudo-residuals*:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n.$$

2. Fit a base learner (e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.

3. Compute multiplier γ_m by solving the following **one-dimensional optimization** problem:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)).$$

4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Reference:

https://en.wikipedia.org/wiki/Gradient_boosting#Algorithm

Gradient Boosting Algorithm Simplified

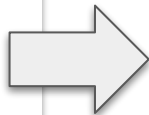
Step 1: Initialize a base model with constant value $M_0(X) = c$

Step 2: Define boosting rounds **For** $t=1$ **to** M :

a. Calculate residual $R = M_{t-1}(X) - Y$

b. Fit model to new target value M_t to (X, R)

Step 3: Update final ensemble $M_t(X) = M_{t-1}(X) + m_t(X)$



Final Ensemble

$$M_3(X) = M_2(X) + M_3(X)$$

$$= M_1(X) + M_2(X) + M_3(X)$$

$$= M_0(X) + M_1(X) + M_2(X) + M_3(X)$$

GRADIENT BOOSTING

$$\hat{y}[i] = \hat{y}[i] + \text{alpha } \underline{f[i]}$$

GRADIENT DESCENT

$$\theta = \theta - \eta * \underline{\nabla_{\theta} J(\theta)}$$

The updates to
reduce error

Cost Function
(MSE)

$$J(.) = \sum (y[i] - \hat{y}[i])^2$$

Gradient of Cost
Function

$$\nabla J(y, \hat{y}) = (y[i] - \hat{y}[i])$$

“Residuals”

★ By taking the **derivative of our cost function** $J(.)$ w.r.t. \hat{y} , then we simply get the **residual**

How is Gradient Boosting Related to Gradient Descent?

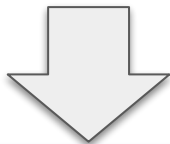
- ❑ **Gradient Boosting**: Learning from the **residuals** nudges us to the right direction towards the **true y**
- ❑ **Gradient Descent**: Following the **negative gradient** of the loss function nudges us to the right direction towards the **true y**
- ❑ So by following the **residuals/errors** in Gradient Boosting we are indirectly following the **negative gradient** of the loss function.

3 Steps of Gradient Boosting (AGAIN)

Residual \Leftrightarrow **Negative Gradient**

1. You fit a model
2. You compute errors on **residuals** \Leftrightarrow You compute errors on **negative gradient**
3. Fit model to **residuals** \Leftrightarrow Fit model to **negative gradients**

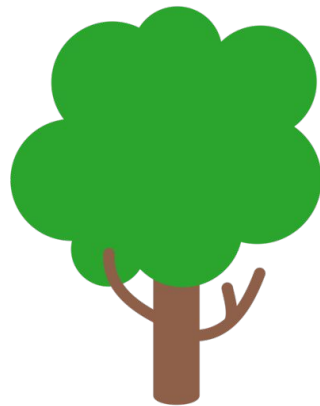
Update model based on **residual** \Leftrightarrow Update model based on **negative gradient**



Hence, we are actually **updating** our model using **gradient descent**!

Combine models together

Summary



Ensemble methods:

- ❑ Combine multiple models to make a “better” one
- ❑ Each model is going to learn something the others can’t
- ❑ Wisdom of the crowd

Boosting method:

- ❑ Combine “weak” learners → “strong” learner
- ❑ Each additive model focuses on the “weaknesses” of the previous one
- ❑ The “weaknesses” tells us how to improve our model
- ❑ Sum of predictions makes it more accurate and complex
- ❑ It learns sequentially

Sources & Reference

Decision Tree Diagram:

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwisSkn6zfAhUIFnwKHXcEA5kQjRx6BAqBEAU&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FFile%3ADecision_tree_for_playing_outside.png&psig=AQvVaw37_R5dcmt9hVaNU1hj31mG&ust=1545320944074870

Bias/Variance Diagram: <https://qph.fs.quoracdn.net/main-qimg-a55358a5a12b02c3f71010c965a2c4dc>

AdaBoost Diagram: <https://slideplayer.com/slide/9092209/27/images/19/Algorithm+Adaboost+-+Example.jpg>