




Ensemble Learning

담당자 박무성



목차

1. 왜 모델에 대해 공부해야하는가?



2. Ensemble 학습이란?

3. Ensemble 학습의 종류

4. 모델에 대한 직관적 이해

5. 모델에 대한 수학적 이해

6. 코드 구현 및 예시

목차

1. 왜 모델에 대해 공부해야하는가?
2. Ensemble 학습이란?
3. Ensemble 학습의 종류
4. 모델에 대한 직관적 이해
5. 모델에 대한 수학적 이해
6. 코드 구현 및 예시

1. 왜 모델에 대해 공부해야하는가?

항상 따라다니는 의문

“Auto ML의 성능이 좋고, Hyper Parameter는 Grid Search가 찾아주는데, 뭐하러 모델을 공부하는거죠?”

“ 모델의 장단점, Python 라이브러리 구현, 옵션의 의미만 알면되는거 아니가요? ”

1. 왜 모델에 대해 공부해야하는가?

“ 높은 Accuracy, 낮은 RMSE 외에 다른 요소를 고려하여 모델을 제작하는 경우가 분명 생긴다.”

“ 스스로 모델을 커스터마이징 할 수 없다면 아무런 차별점이 없음 ”

100억을 빌려주는 대출 심사팀

- 이자 수익은 기업의 신용도에 따라 4~7억
- 특정 기업에 대출을 '내어준다 / 내어주지 않는다' 로 분류하는 Binary Classification

1. 왜 모델에 대해 공부해야하는가?

100억을 빌려주는 대출 심사팀

- 이자 수익은 기업의 신용도에 따라 4~7억

- 특정 기업에 대출을 '내어준다 / 내어주지 않는다' 로
분류하는 Binary Classification

이 경우에 최우선 고려점은 Accuracy인가?

최대한 보수적으로 Generalization에 집중

1. 왜 모델에 대해 공부해야하는가?

100억을 빌려주는 대출 심사팀

- 이자 수익은 기업의 신용도에 따라 4~7억

- 특정 기업에 대출을 '내어준다 / 내어주지 않는다' 로 분류하는 Binary Classification

이 경우에 최우선 고려점은 Accuracy인가?

99% -> but 1% 내어주지 않아야 할 대출 승인

90% -> 약간이라도 이상하면 전부 대출 거절

최대한 보수적으로 Generalization에 집중

1. 왜 모델에 대해 공부해야하는가?

정확도보다 변수 해석이 필요한 경우라면?

데이터 특성에 범주형 자료가 많다면?

반드시 결과의 이유에 대해 설명해야한다면?

1. 왜 모델에 대해 공부해야하는가?

“Auto ML의 성능이 좋고, Hyper Parameter는 Grid Search가 찾아주는데, 뭐하러 모델을 공부하는거죠?”

상황에 맞는 대처를 하기 위해

“ 모델의 장단점, Python 라이브러리 구현, 옵션의 의미만 알면되는거 아닌가요? ”

옵션의 의미를 이해하기 위해서

1. 왜 모델에 대해 공부해야하는가?

옵션의 의미를 이해하기 위해서라도,
직관적 이해와 더불어 **수리적 이해**가 필요하다.

“ 선생님, 이해안하고 공식만 외우면 되는거 아니가요?”

- 미분의 예시

1. 왜 모델에 대해 공부해야하는가?

“Auto ML의 성능이 좋고, Hyper Parameter는 Grid Search가 찾아주는데, 뭐하러 모델을 공부하는거죠?”

“ 모델의 장단점, Python 라이브러리 구현, 옵션의 의미만 알면되는거 아닌가요? ”

상황에 맞는 대처를 하기 위해

직관적 이해

옵션의 의미를 이해하기 위해서

수리적 이해

목차

1. 왜 모델에 대해 공부해야하는가?
2. Ensemble 학습이란?
3. Ensemble 학습의 종류
4. 모델에 대한 직관적 이해
- ~~5. 모델에 대한 수학적 이해~~
- ~~6. 코드 구현 및 예시~~

목차

1. 왜 모델에 대해 공부해야하는가?
2. Ensemble 학습이란?
3. Ensemble 학습의 종류
4. 모델에 대한 직관적 이해

2. Ensemble 학습이란?

모델 여러 개를 사용하는 학습 방식

3. Ensemble Learning의 종류

< Voting >

Hard Voting

Soft Voting

Weighted Voting

< Stacking >

Meta level learning

< Bagging >

< Boosting >

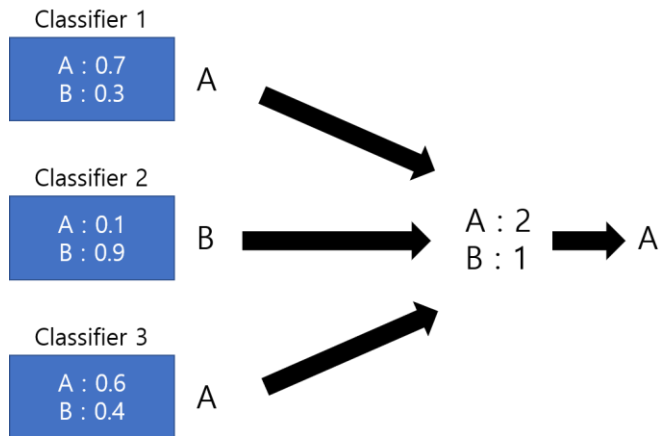
3. Ensemble Learning의 종류

< Voting >

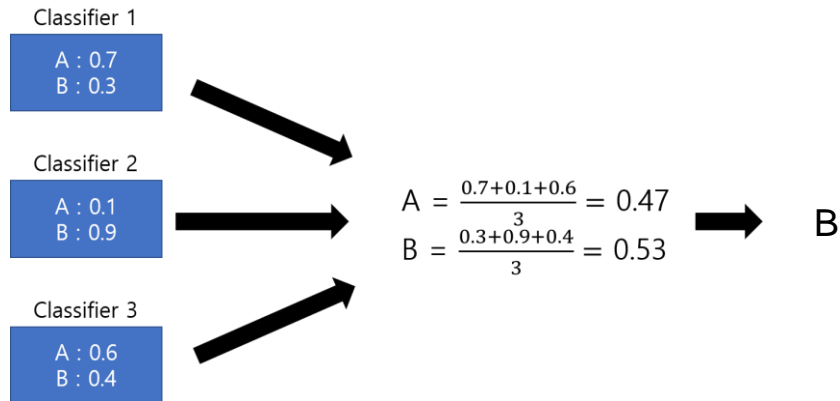
3. Ensemble Learning의 종류

< Voting >

Hard Voting



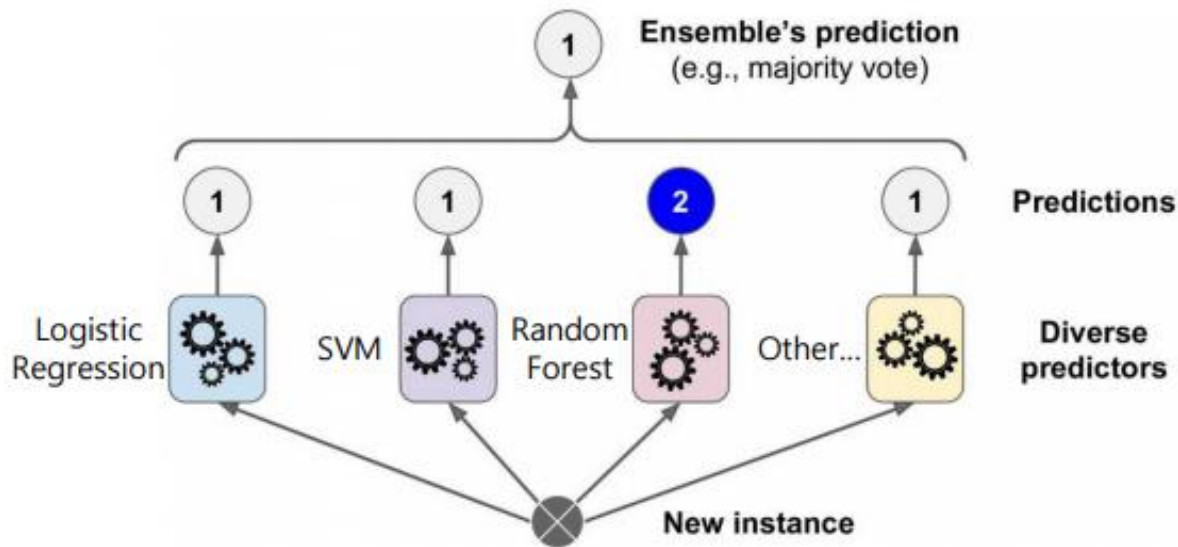
Soft Voting



+ Weighted Voting

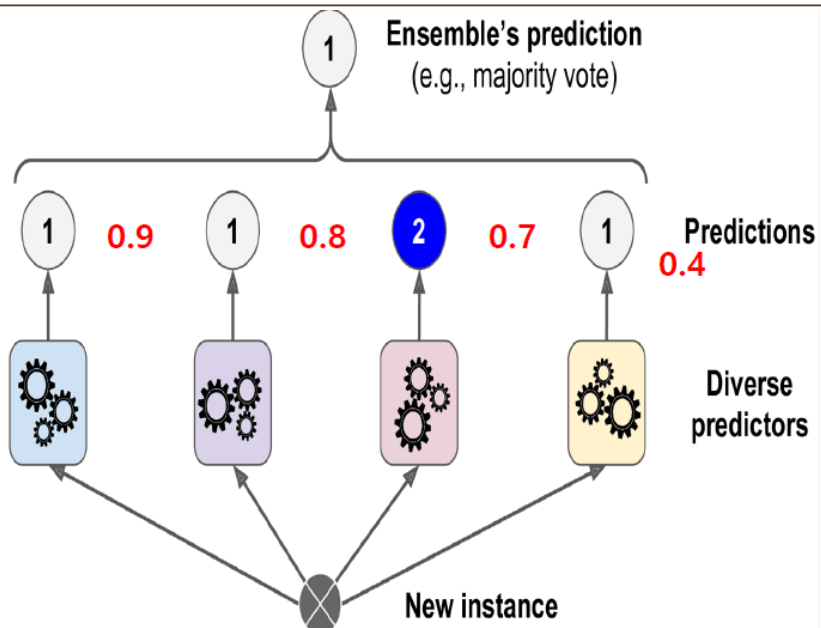
3. Ensemble Learning의 종류

< Hard Voting >



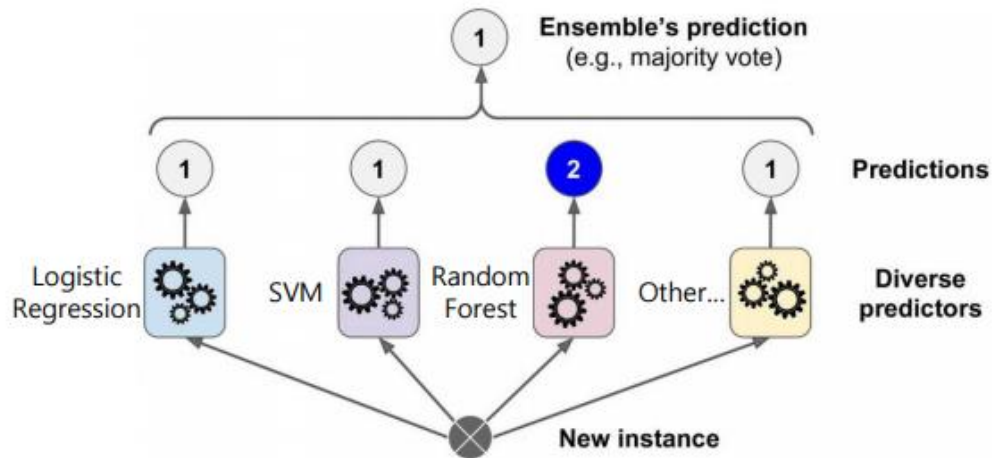
3. Ensemble Learning의 종류

< Soft Voting >



3. Ensemble Learning의 종류

< Weighted Voting >



3. Ensemble Learning의 종류

< Voting >

Hard Voting

Soft Voting

Weighted Voting

< Stacking >

Meta level learning

< Bagging >

< Boosting >

3. Ensemble Learning의 종류

< Stacking > - Meta level learning

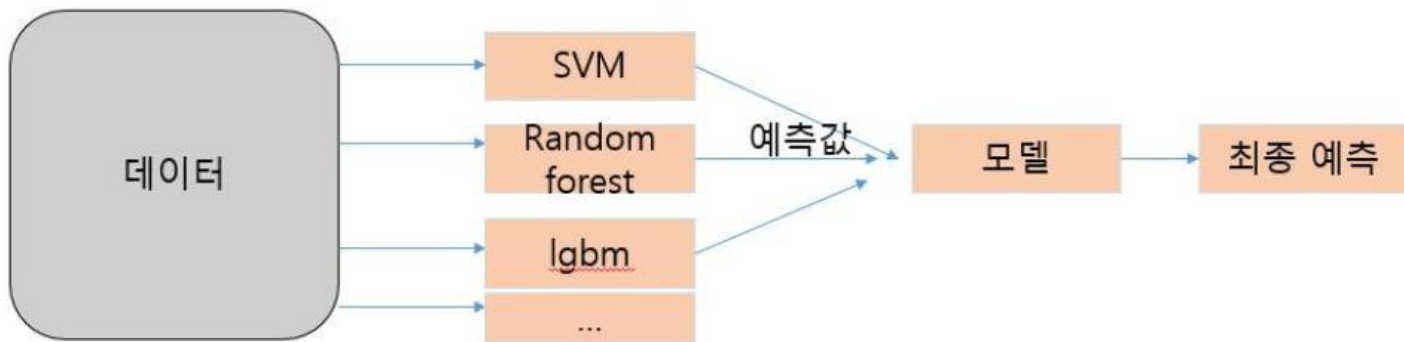
Idea : 예측값을 다시 input으로 넣어서 더 정확한 예측을 하자

직관적으로 가장 이해가 어려운 방법

3. Ensemble Learning의 종류

< Stacking > - Meta level learning

개별 모델이 예측한 결과를 다시 input data로 사용



3. Ensemble Learning의 종류

< Stacking > - Meta level learning

원본 데이터를 train / test set으로 분할

원본 training data를 각각 개별 모델로 학습

각 개별 모델들의 predict 결과를 new_training 데이터로 사용

최종 모델에 new_trainin data로 학습

3. Ensemble Learning의 종류

< Voting >

Hard Voting

Soft Voting

Weighted Voting

< Stacking >

Meta level learning

< Bagging >

< Boosting >

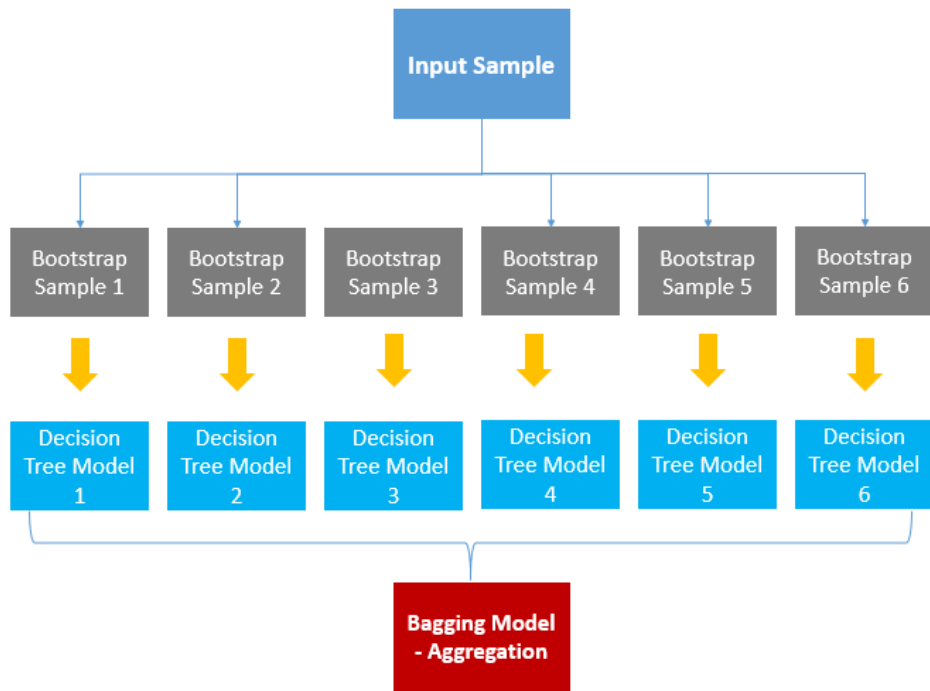
3. Ensemble Learning의 종류

< Bagging >

핵심은 복원 추출

3. Ensemble Learning의 종류

< Bagging >



원본 data set에서 데이터 랜덤으로 추출해서
모델 여러 개 돌려보자

그리고 합쳐서 결과내자

핵심은 복원 추출

목차

1. 왜 모델에 대해 공부해야하는가?
2. Ensemble 학습이란?
3. Ensemble 학습의 종류
4. 모델에 대한 직관적 이해

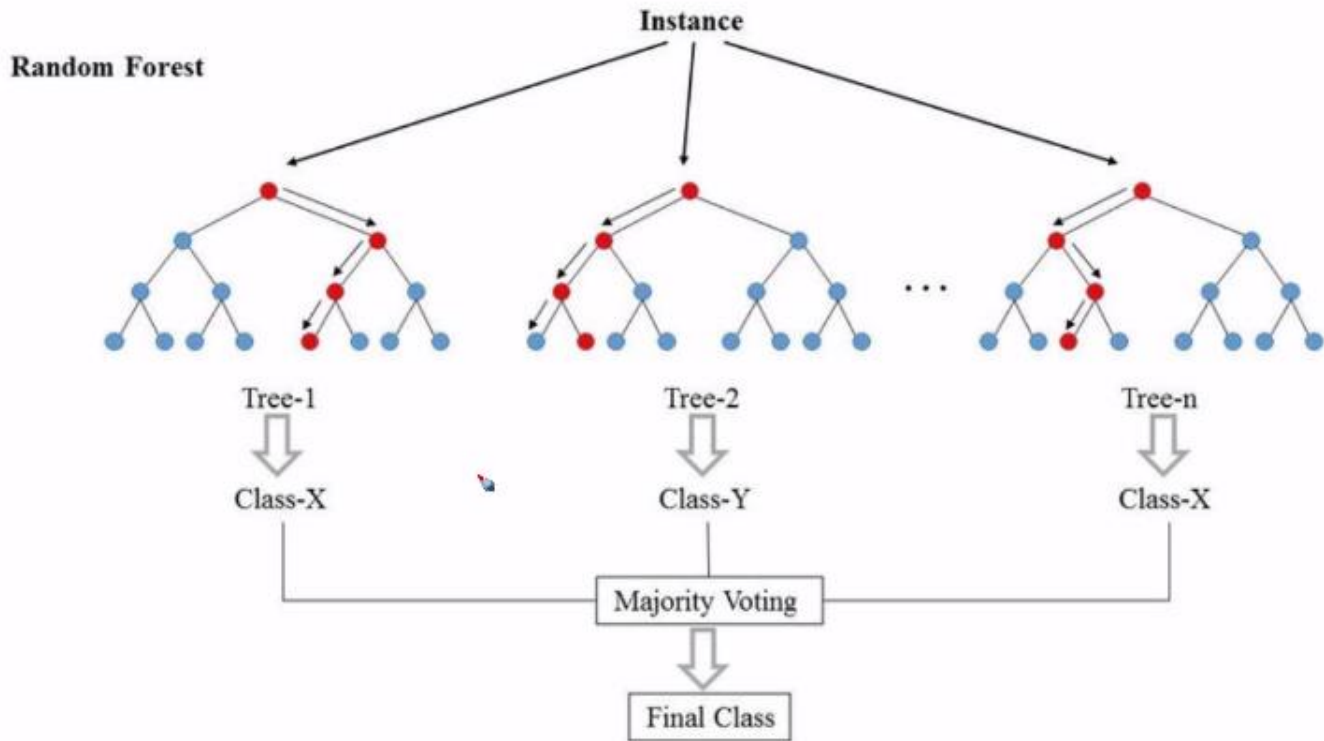
〈 Random Forest의 직관적 이해〉

Bagging의 방식을 사용한 Ensemble Model

: Random Forest

〈 Random Forest의 직관적 이해〉

< Random Forest >



〈 Random Forest의 직관적 이해〉

장점

: 어디에 가져다놓아도 중간 이상가는 알고리즘

병렬 학습 방식으로 학습 속도가 빠른편

입력 변수가 많아도 잘작동함

단점

: 이론적으로 트리 깊이를 엄청 깊게까지 내리면 100% accuracy를 구현할 수 있음

-> 트리 개수, 깊이같은 hyperparameter를 잘못 설정할 경우 **Overfitting**의 가능성이 매우 높음

목차

1. 왜 모델에 대해 공부해야하는가?
2. Ensemble 학습이란?
3. Ensemble 학습의 종류 : Boosting
4. 모델에 대한 직관적 이해

3. Ensemble Learning의 종류

< Voting >

Hard Voting

Soft Voting

< Stacking >

Meta level learning

< Bagging >

< Boosting >

3. Ensemble Learning의 종류

< Boosting >

Boosting은 오답 노트다

틀린거에 대해서 집중하는 알고리즘

ADA Boost

Gradient Boosting Machine

Extreme Gradient Boosting Machine

Light Gradient Boosting Machine

Cat Boost

3. Ensemble Learning의 종류

< Normal >

< Bagging >

< Boosting >

학생 A

: 10개년 6,9,수능 문제 1회독

학생 B

: 10개년 6,9,수능 전체 문제에서
80% 랜덤 복원 추출

-> 10번 반복

학생 C

: 10개년 6,9,수능 전체 문제에서
80% 랜덤 복원 추출

이때, 틀린 문제는 반드시
포함해서 추출

-> 10번 반복

학생 D

: 10개년 6,9,수능 전체 문제 1회독
-> 틀린 문제만 뽑아서 다시 1회독
-> 다시 틀린 문제만 뽑아서 1회독
-> 다시 틀린 문제만 뽑아서 1회독

·
·
·

안틀릴때까지 반복

Ensemble Learning 中 Boosting 목차

1. Adaptive Boosting Ada Boost
2. Gradient Boosting Machine GBM
3. Extreme Gradient Boosting Machine XG Boost
4. Light Gradient Boosting Machine Light GBM
5. Categorical Boosting Machine CAT Boost

4. Ensemble 모델의 직관적 이해

< 이번주 학습 내용 >

Ada boost

Gradient Boosting Machine

Extreme Gradient Boosting Machine

< 다음주 학습 내용 >

Light Gradient Boosting Machine

Cat Boost

〈 Ada boost의 직관적 이해 〉

Adaptive Boosting (Adaboost)

❖ AdaBoost

- 각 단계에서 새로운 base learner를 학습하여 이전 단계의 base learner의 단점을 보완
- Training error가 큰 관측치의 선택 확률(가중치)을 높이고, training error가 작은 관측치의 선택 확률을 낮춤
 - ✓ 오분류한 관측치에 보다 집중!
- 앞 단계에서 조정된 확률(가중치)을 기반으로 다음 단계에서 사용될 training dataset를 구성
- 다시 첫 단계로 감
- 최종 결과물은 각 모델의 성능지표를 가중치로 하여 결합 (앙상블)

- Weak Model 을 여러번
- 순차적 학습
- 오답 노트

〈 Ada boost의 직관적 이해 〉

Bagging의 아이디어와 비슷

But, 복원 추출할때 오답 data에 더 가중치를 줘서 선택 될 확률을 높임

Point. “ 틀린 문제는 다음 추출때 반드시 포함시켜서 학습시키겠다.”

“순차적으로 문제 풀면서 오답노트 업데이트 하겠다.”

학생 C

: 10개년 6,9,수능 전체 문제에서
80% 랜덤 복원 추출

이때, 틀린 문제는 반드시
포함해서 추출

-> 10번 반복

〈 Ada boost의 직관적 이해 〉

❖ AdaBoost algorithm

1. Set $W_i = \frac{1}{n}, i = 1, 2, \dots, n$ (impose equal weight initially)

2. for $j = 1$ to m (m : number of classifiers)

Step 1: Find $h_j(x)$ that minimizes L_j (weighted loss function)

$$L_j = \frac{\sum_{i=1}^n W_i I(y_i \neq h_j(x))}{\sum_{i=1}^n W_i}$$

Step 2: Define the weight of a classifier: $\alpha_j = \log\left(\frac{1-L_j}{L_j}\right)$

Step 3: Update weight: $W_i \leftarrow W_i e^{\alpha_j I(y_i \neq h_j(x))}, i = 1, 2, \dots, n$

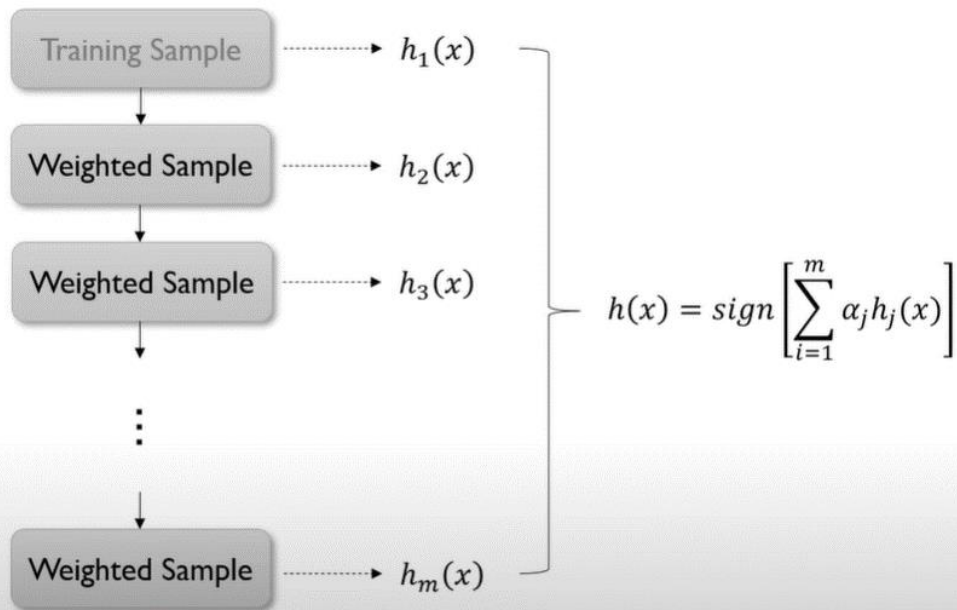
endfor

3. Final boosted model: $h(x) = \text{sign}\left[\sum_{i=1}^m \alpha_j h_j(x)\right]$

- 순차적 학습

- 오답 노트

〈 Ada boost의 직관적 이해 〉

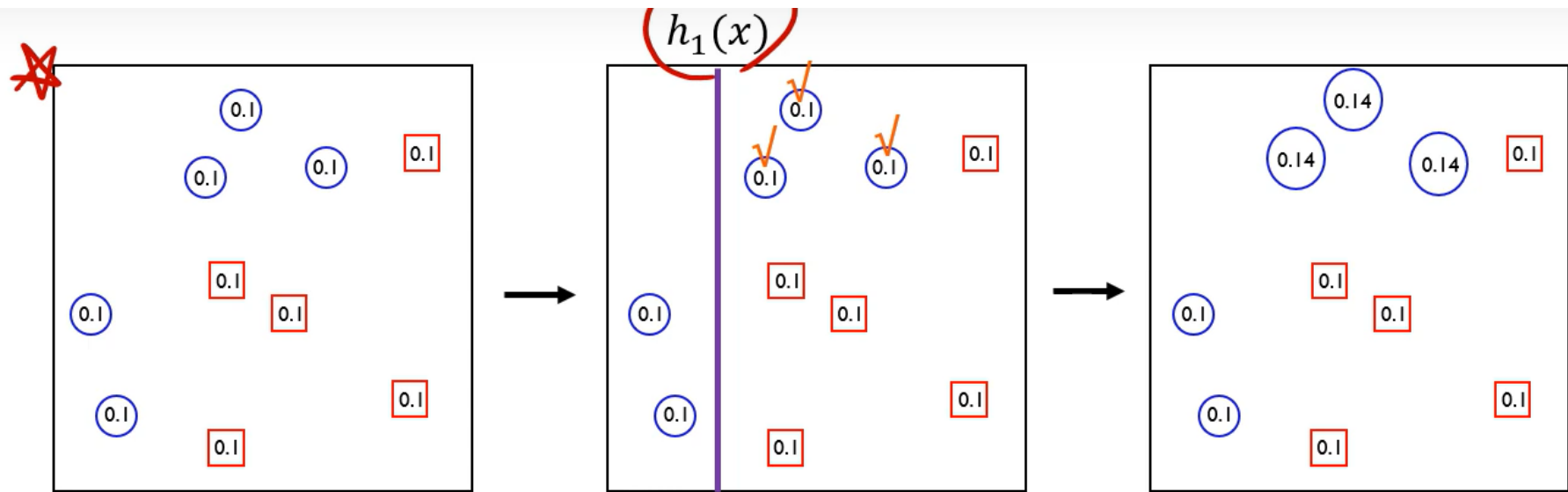


- 순차적 학습

- 오답 노트

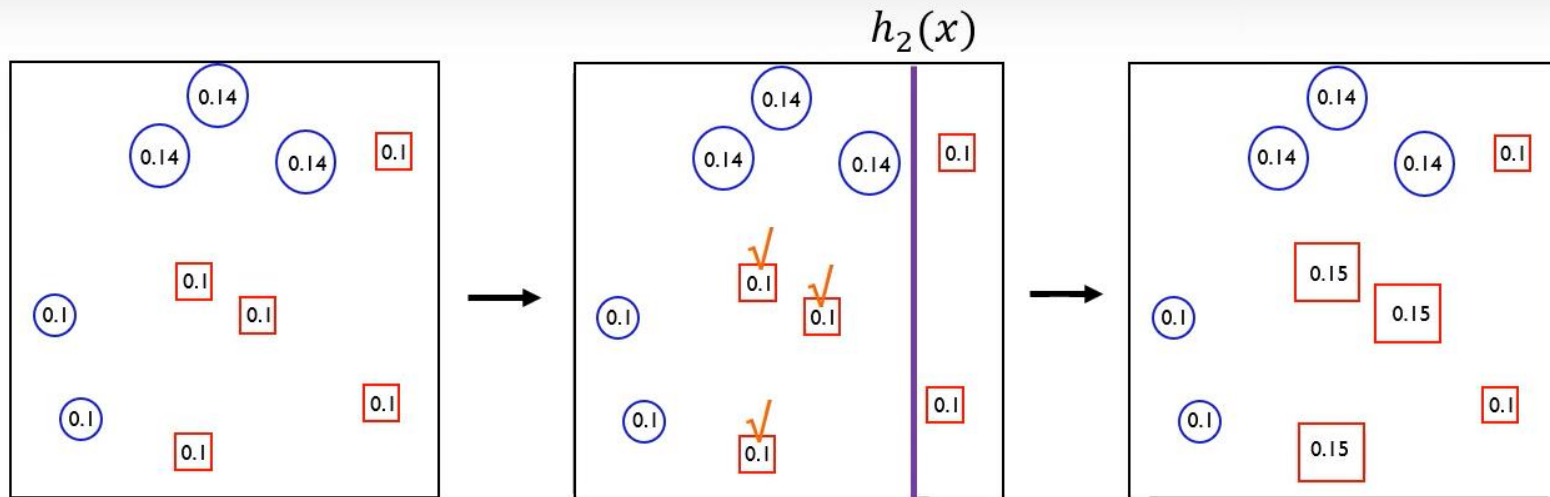
$h_1(x)$ 를 만들고, 이를 바탕으로 $h_2(x)$ 를 만들고, 이를 바탕으로 $h_3(x), \dots$

〈 Ada boost의 직관적 이해 〉



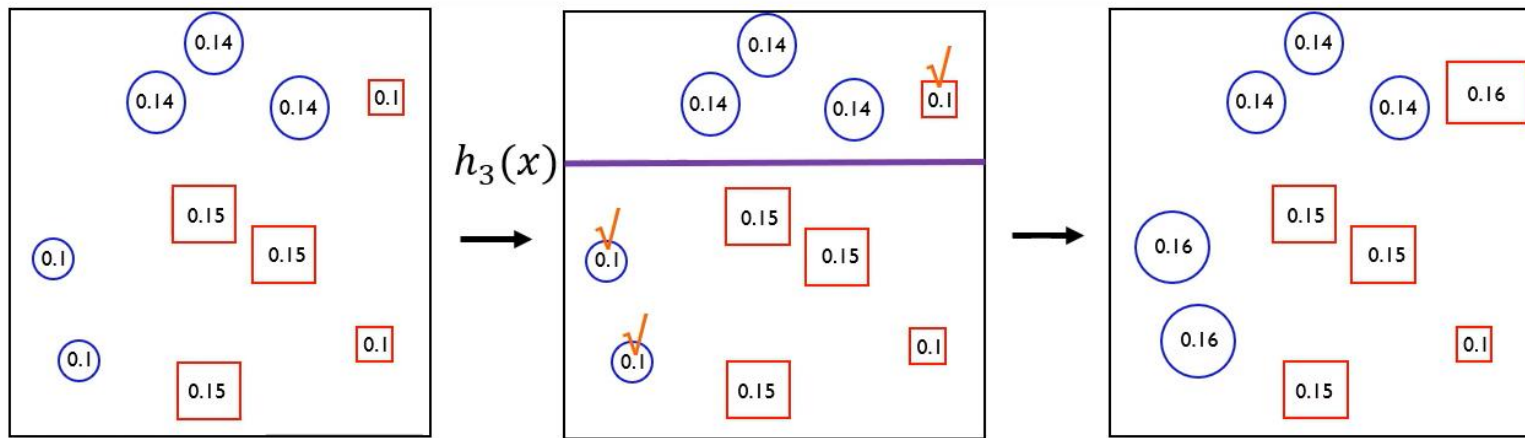
도형 안 숫자는 다음 단계의 학습 데이터셋에 선택될 확률을 의미

〈 Ada boost의 직관적 이해 〉



도형 안 숫자는 다음 단계의 학습 데이터셋에 선택될 확률을 의미

◁ Ada boost의 직관적 이해 ▷



$$L_3 = \frac{\sum_{i=1}^n W_i I(y_i \neq h_2(x))}{\sum_{i=1}^n W_i} = \frac{0.1 \times 3}{0.1 \times 4 + 0.14 \times 3 + 0.15 \times 3} = 0.24 \quad \text{10개중 3개 오분류}$$

$$\alpha_3 = \log\left(\frac{1 - 0.24}{0.24}\right) \approx 0.5$$

〈 Ada boost의 직관적 이해 〉

$$h(x) = \text{sign} \left[\sum_{i=1}^{m=3} \alpha_i h_i(x) \right]$$

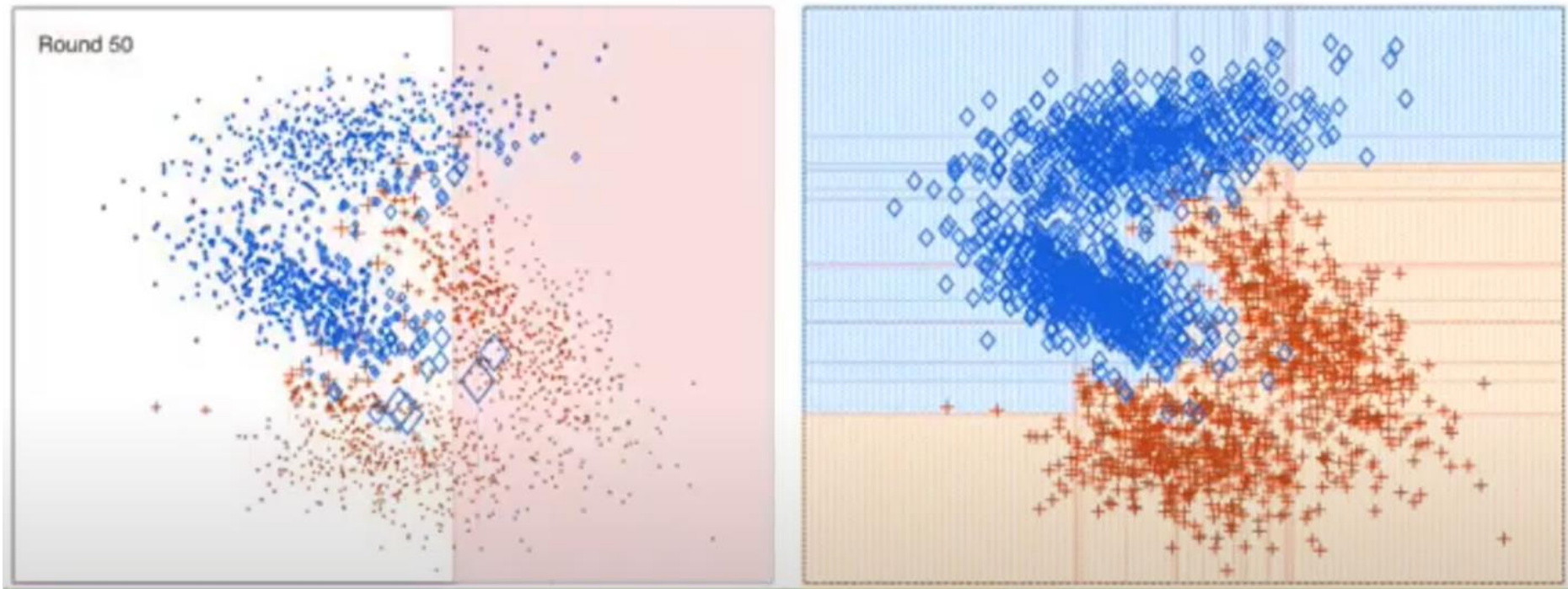
$$h(x) = \text{sign} \left[0.37 \begin{array}{|c|c|} \hline \begin{array}{c} h_1(x) \\ \text{Diagram 1} \end{array} & \begin{array}{c} h_2(x) \\ \text{Diagram 2} \end{array} \\ \hline \end{array} + 0.43 \begin{array}{|c|c|} \hline \begin{array}{c} h_2(x) \\ \text{Diagram 2} \end{array} & \begin{array}{c} h_3(x) \\ \text{Diagram 3} \end{array} \\ \hline \end{array} + 0.5 \begin{array}{|c|c|} \hline \begin{array}{c} h_3(x) \\ \text{Diagram 3} \end{array} & \begin{array}{c} h_3(x) \\ \text{Diagram 3} \end{array} \\ \hline \end{array} \right]$$

$$= \begin{array}{|c|c|c|} \hline & \text{Blue Circle} & \text{Red Square} \\ \hline \text{Blue Circle} & \text{Red Square} & \text{Red Square} \\ \hline \end{array}$$

- 순차적 학습

- 오답 노트

〈 Adaptive Boosting 성능 시각화 〉



〈 Ada boost의 직관적 이해 〉

〈복습〉

〈 Ada boost의 직관적 이해 〉

Adaptive Boosting (Adaboost)

❖ AdaBoost

- 각 단계에서 새로운 base learner를 학습하여 이전 단계의 base learner의 단점을 보완
- Training error가 큰 관측치의 선택 확률(가중치)을 높이고, training error가 작은 관측치의 선택 확률을 낮춤
 - ✓ 오분류한 관측치에 보다 집중!
- 앞 단계에서 조정된 확률(가중치)을 기반으로 다음 단계에서 사용될 training dataset를 구성
- 다시 첫 단계로 감
- 최종 결과물은 각 모델의 성능지표를 가중치로 하여 결합 (앙상블)

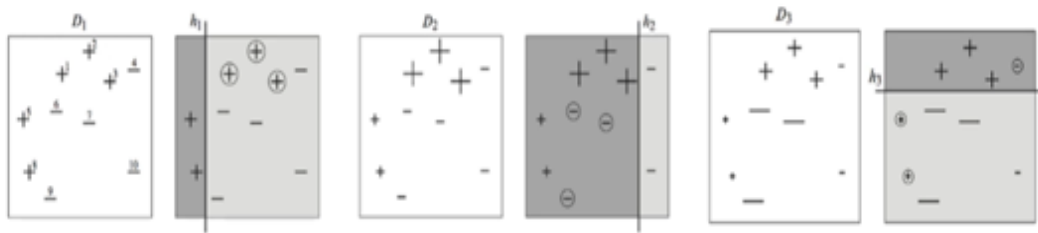
- Weak Model 을 여러번
- 순차적 학습
- 오답 노트

Ensemble Learning 中 Boosting 목차

1. Adaptive Boosting Ada Boost
2. Gradient Boosting Machine GBM
3. Extreme Gradient Boosting Machine XG Boost
4. Light Gradient Boosting Machine Light GBM
5. Categorical Boosting Machine CAT Boost

〈 Gradient Boosting Machine의 직관적 이해 〉

- Adaboost



틀린 부분에 대해 가중치를 부여해서, 더 높은 확률로 다음 학습때 포함시켜 사용하겠다.
-> 결론적으로 틀린걸 더 많이 반복하여 학습해서 Strong Model로 만들겠다.

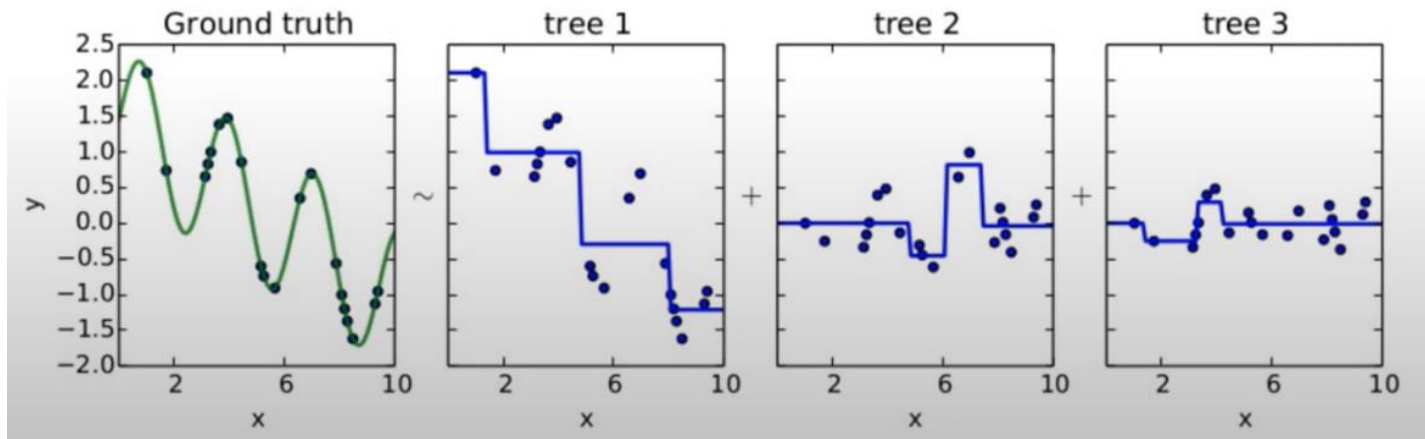
오류값에 대한 새로운 학습을 residual을 사용해서 진행하겠다.

〈 Gradient Boosting Machine의 직관적 이해 〉

오류값에 대한 새로운 학습을 residual을 사용해서 진행하겠다.

〈 Gradient Boosting Machine의 직관적 이해 〉

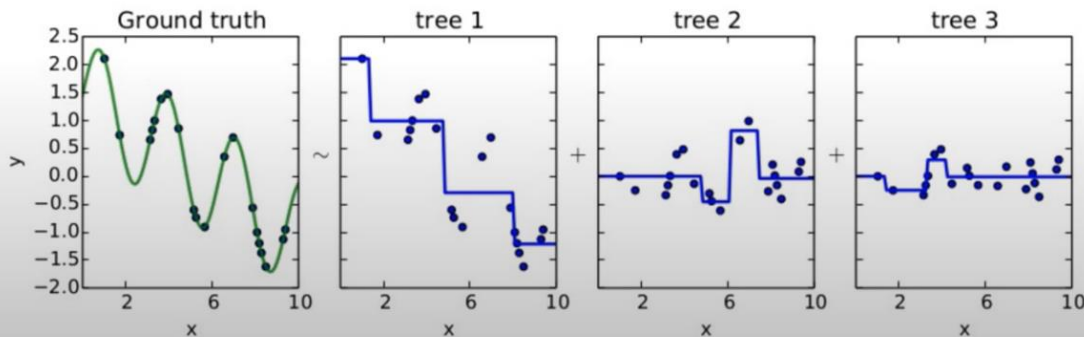
Gradient Boosting = Gradient Descent + Boosting



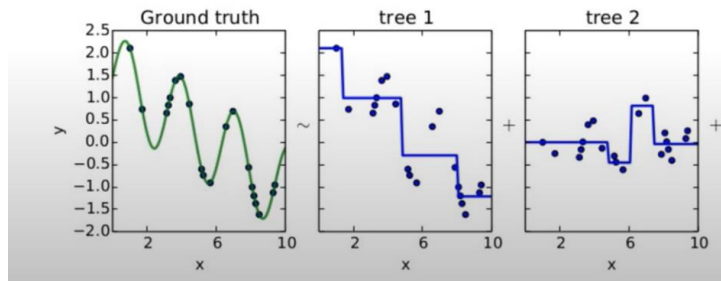
오류값에 대한 새로운 학습을 Residual을 사용해서 진행하겠다.

〈 Gradient Boosting Machine의 직관적 이해 〉

- Gradient boosting = Boosting with gradient decent
- 첫번째 단계의 모델 tree1을 통해 Y를 예측하고, Residual을 다시 두번째 단계 모델 tree2를 통해 예측하고, 여기서 발생한 Residual을 모델 tree3로 예측
- 점차 residual 작아 짐
- Gradient boosted model = tree1 + tree2 + tree3



- Gradient boosting = Boosting with gradient decent
- 첫번째 단계의 모델 tree1을 통해 Y를 예측하고, Residual을 다시 두번째 단계 모델 tree2를 통해 예측하고, 여기서 발생한 Residual을 모델 tree3로 예측
- 점차 residual 작아 짐
- Gradient boosted model = tree1 + tree2 + tree3



• Main idea

Original Dataset

x^1	y^1
x^2	y^2
x^3	y^3
x^4	y^4
x^5	y^5
x^6	y^6
x^7	y^7
x^8	y^8
x^9	y^9
x^{10}	y^{10}

Modified Dataset 1

x^1	$y^1 - f_1(x^1)$
x^2	$y^2 - f_1(x^2)$
x^3	$y^3 - f_1(x^3)$
x^4	$y^4 - f_1(x^4)$
x^5	$y^5 - f_1(x^5)$
x^6	$y^6 - f_1(x^6)$
x^7	$y^7 - f_1(x^7)$
x^8	$y^8 - f_1(x^8)$
x^9	$y^9 - f_1(x^9)$
x^{10}	$y^{10} - f_1(x^{10})$

Modified Dataset 2

x^1	$y^1 - f_1(x^1) - f_2(x^1)$
x^2	$y^2 - f_1(x^2) - f_2(x^2)$
x^3	$y^3 - f_1(x^3) - f_2(x^3)$
x^4	$y^4 - f_1(x^4) - f_2(x^4)$
x^5	$y^5 - f_1(x^5) - f_2(x^5)$
x^6	$y^6 - f_1(x^6) - f_2(x^6)$
x^7	$y^7 - f_1(x^7) - f_2(x^7)$
x^8	$y^8 - f_1(x^8) - f_2(x^8)$
x^9	$y^9 - f_1(x^9) - f_2(x^9)$
x^{10}	$y^{10} - f_1(x^{10}) - f_2(x^{10})$

...

$$\hat{y} = f_1(x) \quad y - f_1(x) = f_2(x) \quad y - f_1(x) - f_2(x) = f_3(x)$$

〈 Gradient Boosting Machine의 직관적 이해 〉

〈 Gradient Boosting Machine의 직관적 이해 〉

- How is this idea related to the gradient?

✓ Loss function of the ordinary least square (OLS)

$$\min L = \frac{1}{2} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$$

✓ Gradient of the Loss function

$$\frac{\partial L}{\partial f(\mathbf{x}_i)} = f(\mathbf{x}_i) - y_i$$

✓ Residuals are the negative gradient of the loss function

$$y_i - f(\mathbf{x}_i) = -\frac{\partial L}{\partial f(\mathbf{x}_i)}$$

Gradient Boosting = Gradient Descent + Boosting

$$F(x) = h_1(x) + h_2(x) + h_3(x) \dots$$

“ Residual 사용해서 모델 쌓는거면서 왜 Gradient Boosting 이라고 하나?”

Gradient 가 결국 Residual

오류값에 대한 새로운 학습을 residual을 사용해서 진행하겠다.

오류값에 대한 새로운 학습을 gradient를 사용해서 진행하겠다.

〈 Gradient Boosting Machine의 직관적 이해 〉

- Gradient Boosting Algorithm

1. Initialize $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$.

2. For $m = 1$ to M :

- 2.1 For $i = 1, \dots, N$ compute

$$g_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x_i)=f_{m-1}(x_i)}$$

- 2.2 Fit a regression tree to the targets g_{im} giving terminal regions

$$R_{jm}, j = 1, \dots, J_m.$$

- 2.3 For $j = 1, \dots, J_m$ compute

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

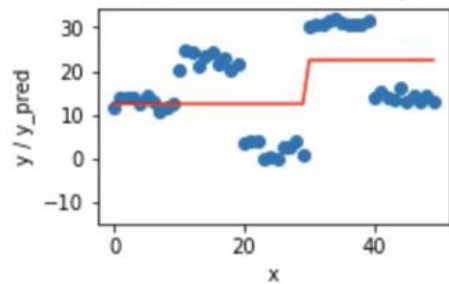
- 2.4 Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

3. Output $\hat{f}(x) = f_M(x)$.

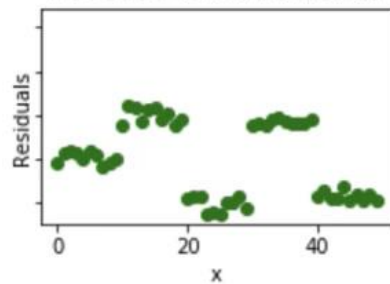
Regression, classification은

loss 함수를 뭘 쓰냐에 따라 달라진다.

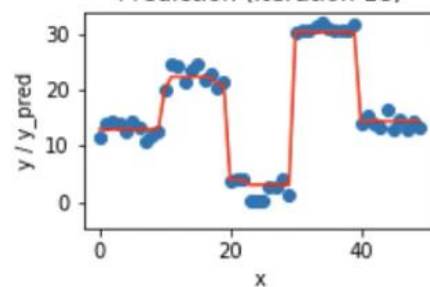
Prediction (Iteration 1)



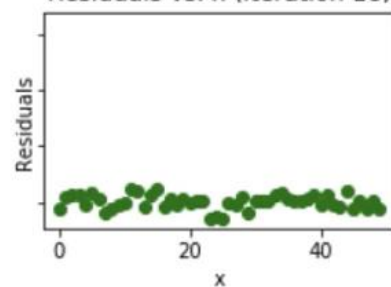
Residuals vs. x (Iteration 1)



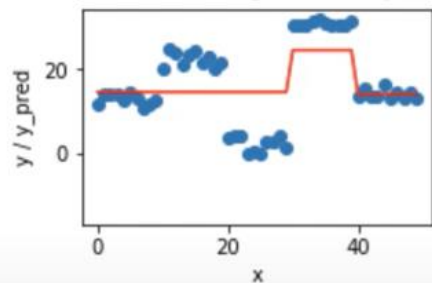
Prediction (Iteration 18)



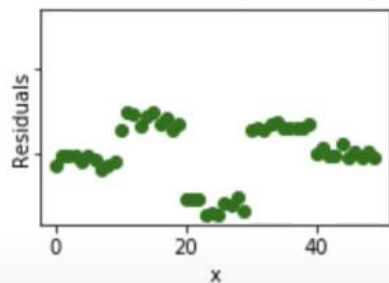
Residuals vs. x (Iteration 18)



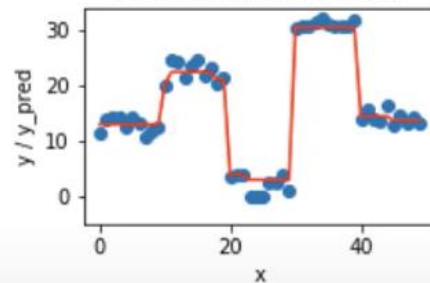
Prediction (Iteration 2)



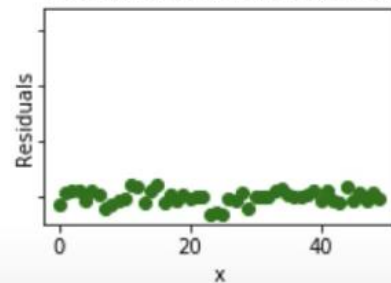
Residuals vs. x (Iteration 2)



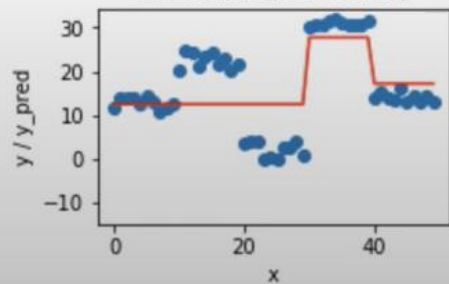
Prediction (Iteration 19)



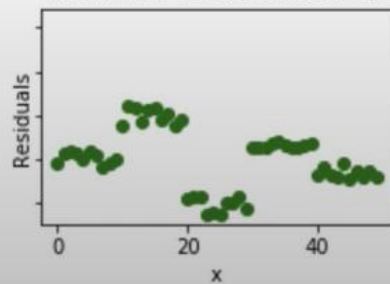
Residuals vs. x (Iteration 19)



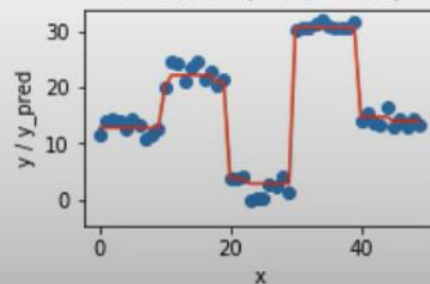
Prediction (Iteration 3)



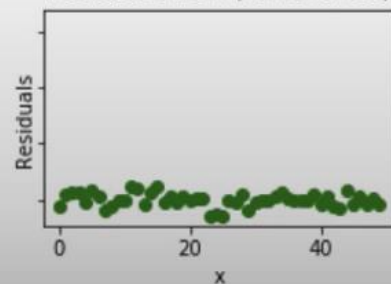
Residuals vs. x (Iteration 3)



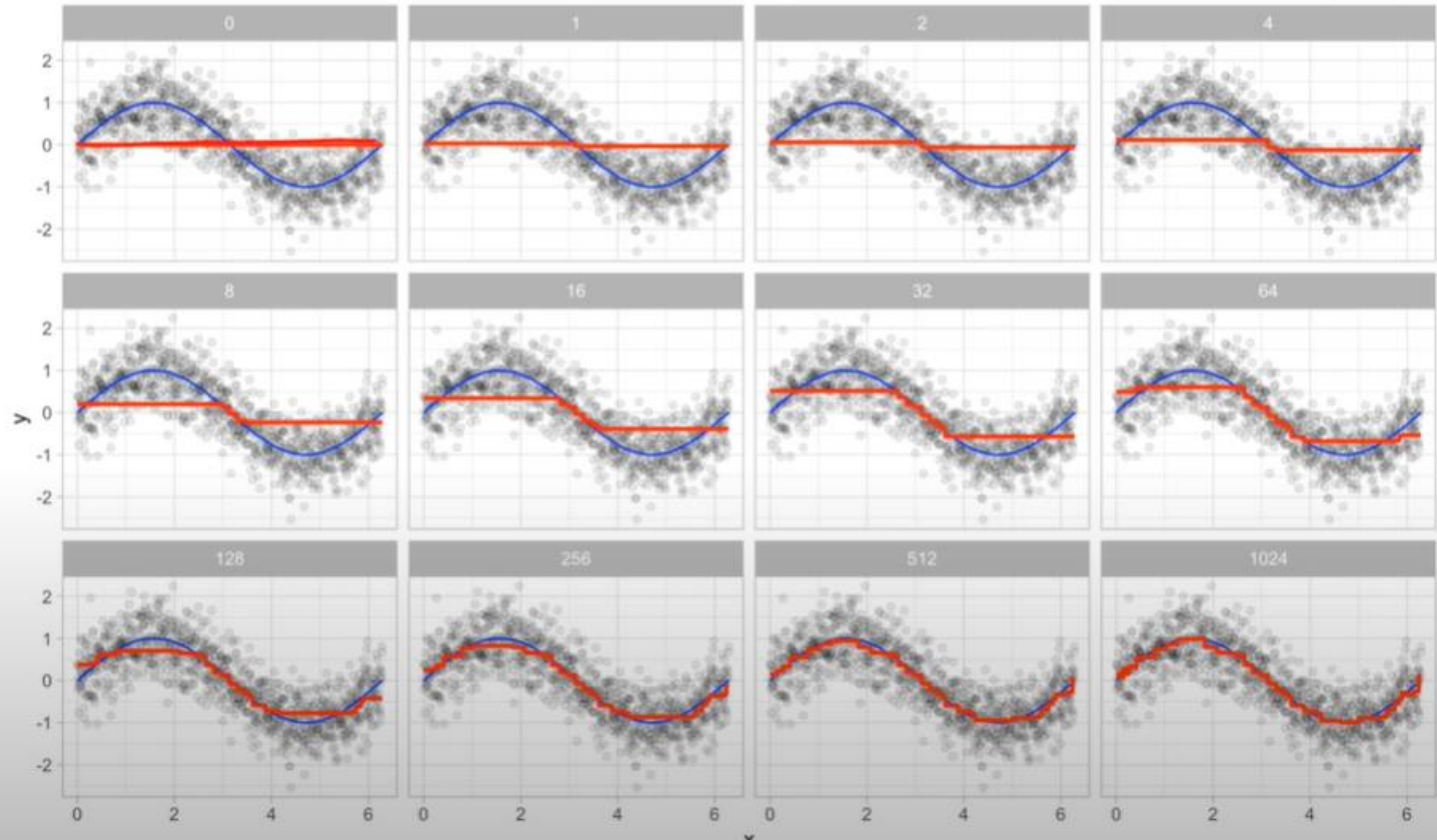
Prediction (Iteration 20)



Residuals vs. x (Iteration 20)



- GBM Regression Example 3



〈 Gradient Boosting Machine의 직관적 이해 〉

GBM 특징

장점

: 최근 유명한 모델들 모두 GBM 기반 (성능 좋음)

Variable Importance 계산 가능

단점

: Over Fitting 가능성

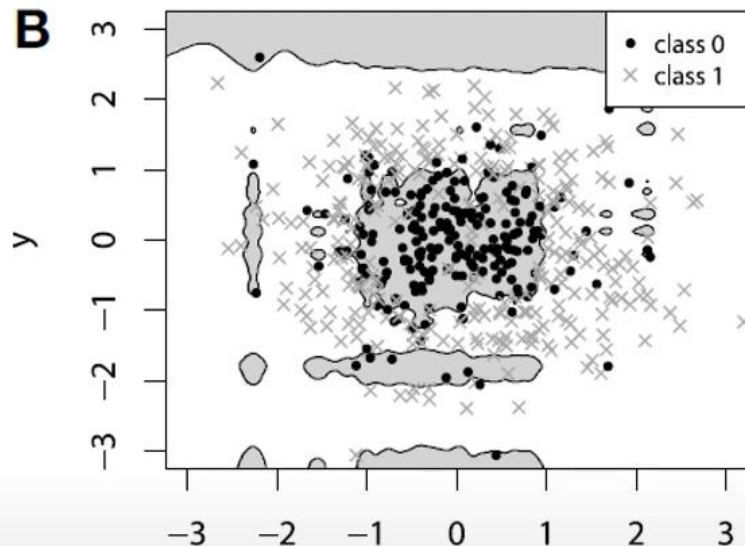
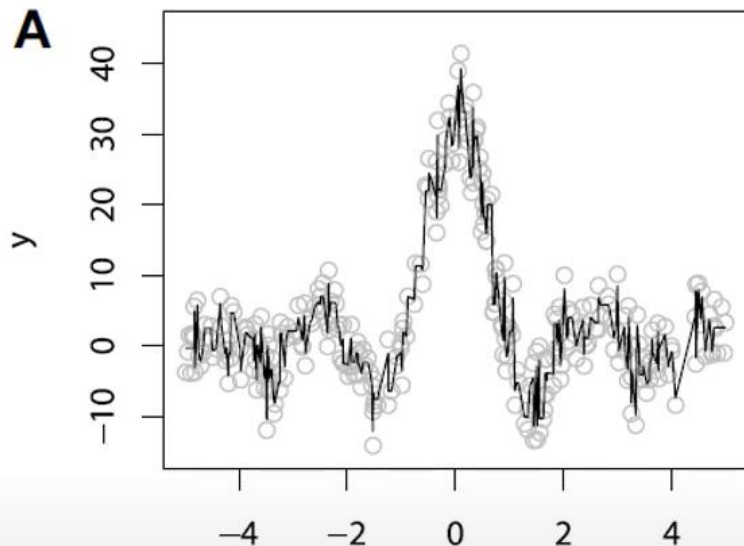
연산 속도가 오래 걸림

✓ Variable importance of Gradient boosting

$$Influence_j = \frac{1}{M} \sum_{k=1}^M Influence_j(T_k)$$

Gradient Boosting Machine의 단점

Overfitting problem in GBM



Gradient Boosting Machine의 단점

Overfitting problem in GBM

- Regularization

- ✓ Subsampling

- At each learning iteration, only a random part of the training data is used to fit a consecutive base-learner.
 - The training data is typically sampled without replacement, but bagging can be also acceptable.

Gradient Boosting Machine의 단점

Overfitting problem in GBM

✓ Shrinkage

- Used for reducing/shrinking the impact of each additional fitted base-leaners.
- Better to improve a model by taking many small steps than by taking fewer large steps.

Gradient Boosting Machine의 단점

Overfitting problem in GBM

✓ Early Stopping

- Use the validation error

〈 Gradient Boosting Machine의 직관적 이해 〉

GBM 특징

장점

: 최근 유명한 모델들 모두 GBM 기반 (성능 좋음)

Variable Importance 계산 가능

단점

: Over Fitting 가능성

연산 속도가 오래 걸림

✓ Variable importance of Gradient boosting

$$Influence_j = \frac{1}{M} \sum_{k=1}^M Influence_j(T_k)$$

Gradient Boosting Machine의 단점

연산 속도의 문제

〈 Gradient Boosting Machine의 직관적 이해 〉

장점

: 최근 유명한 모델들 모두 GBM 기반 (성능 좋음)

Variable Importance 계산 가능

단점

: Over Fitting 가능성

연산 속도가 오래 걸림

“장점을 살리면서 GBM의 단점을
줄이는 알고리즘은 없을까?”



2014년 Extreme Gradient Boost (XG Boost)

2017년 Light Gradient Boost (LGBM)

2017년 Categorical Gradient Boost (Cat Boost)

Gradient Boosing Machine 의 발전

