

데이터마이닝팀

4팀

김수빈
조건우
김보현
이지원
조성우

CONTENTS

1. 트리 기반 모델

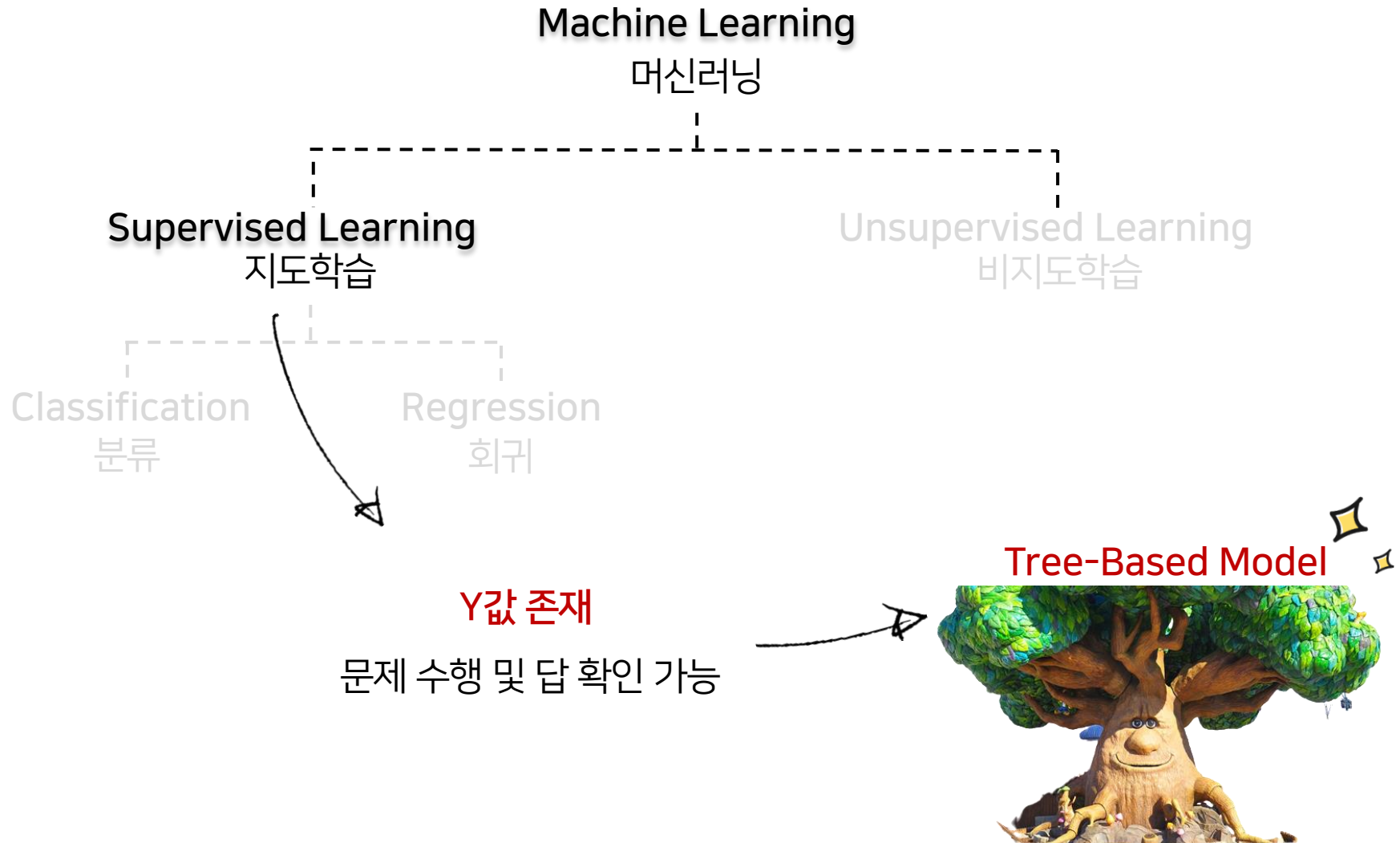
2. 비선형 모델

1

트리 기반 모델



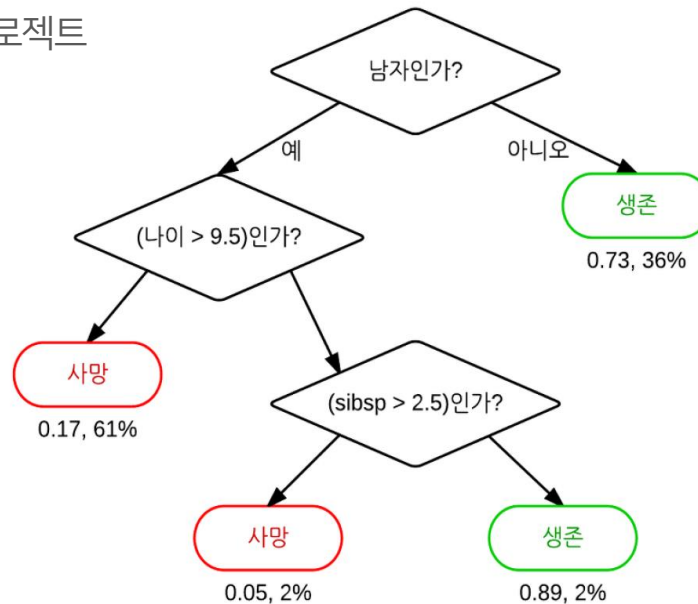
트리 기반 모델



트리 기반 모델

의사결정나무(Decision Tree)

Ex) 타이타닉 프로젝트

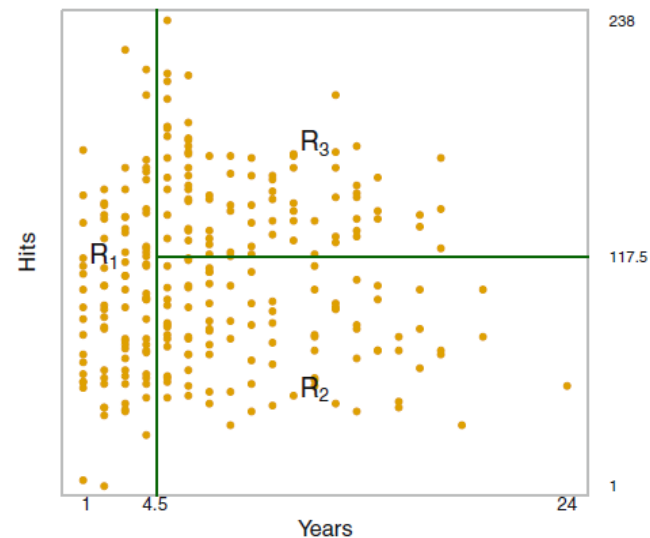
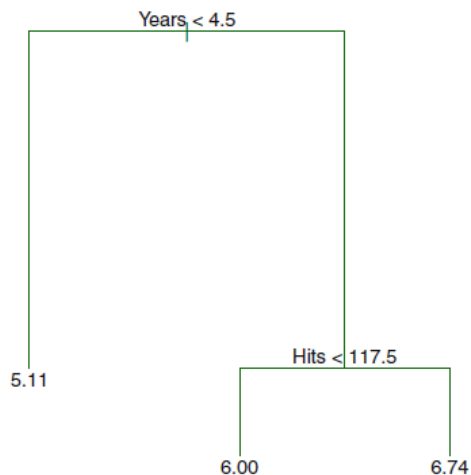


사람들의 정보에 기반하여 생존 여부를 예측하는데,
각 질문에 대한 대답으로 예측값 결정



트리 기반 모델

의사결정나무(Decision Tree)

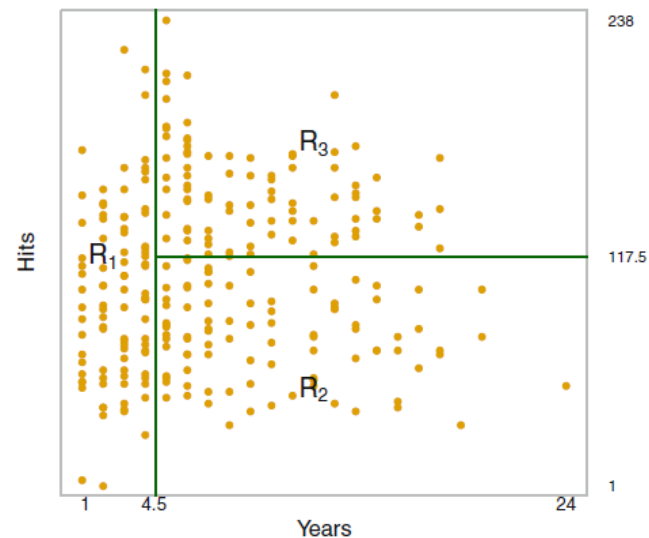
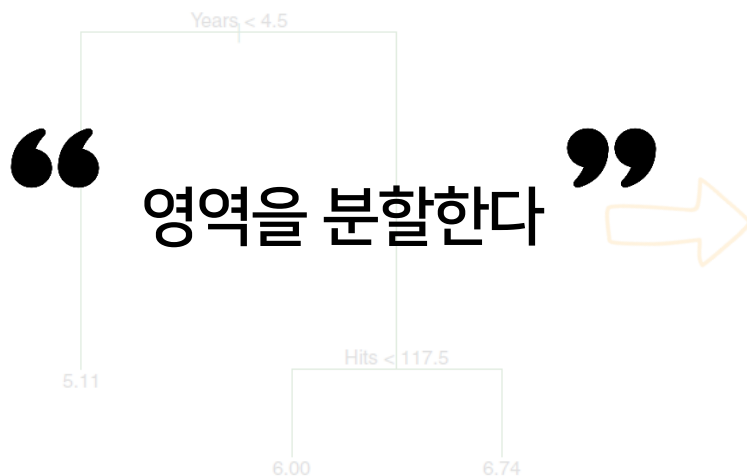


좌측의 트리 예시를 2차원 평면으로 옮기면
오른쪽과 같이 **분할된 영역**의 형태로 나타남



트리 기반 모델

의사결정나무(Decision Tree)

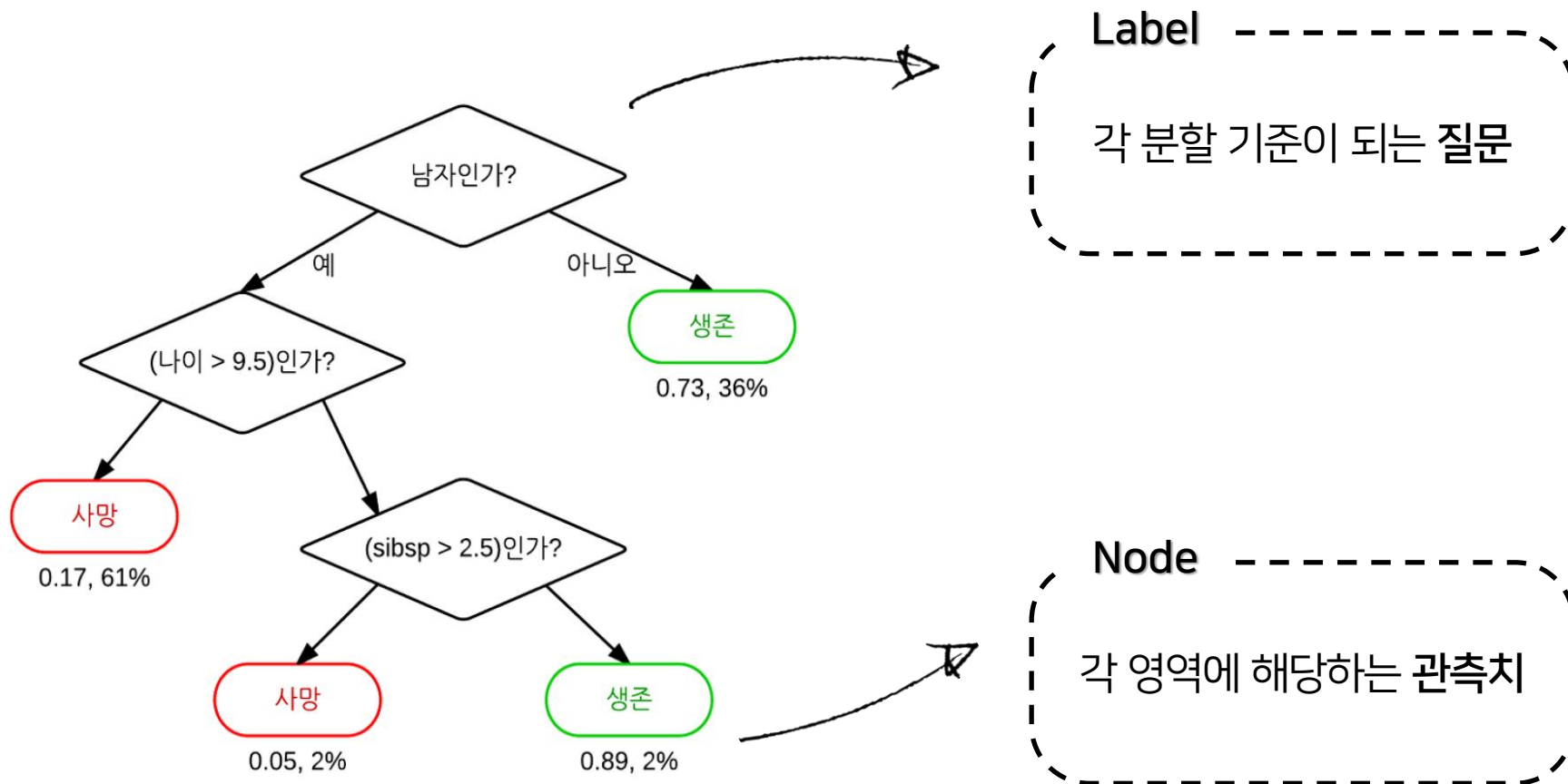


좌측의 트리 예시를 2차원 평면으로 옮기면
오른쪽과 같이 분할된 영역의 형태로 나타남



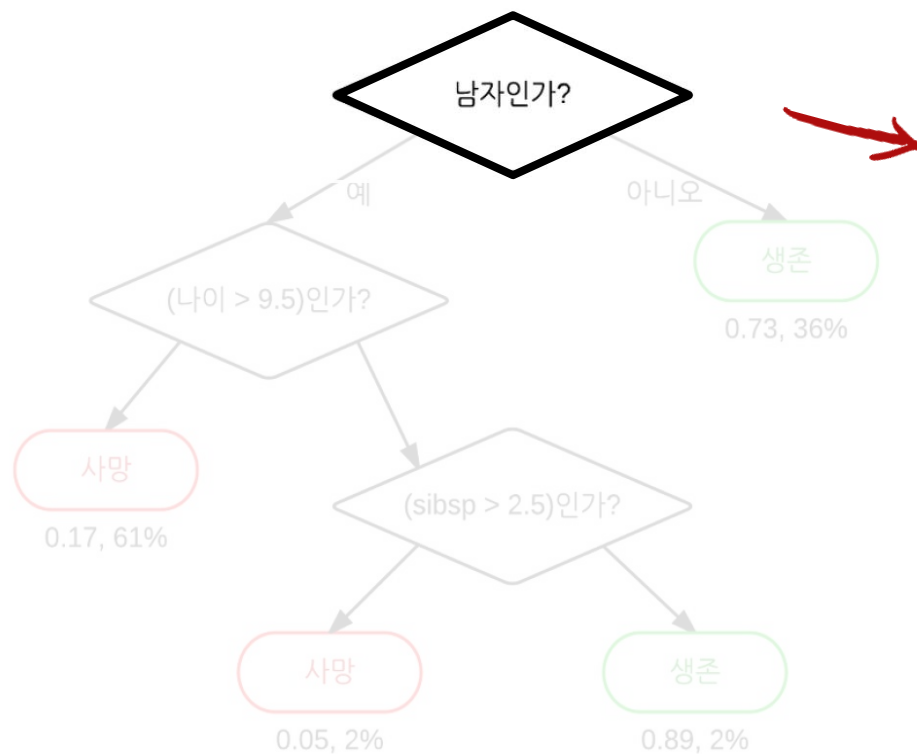
트리 기반 모델

의사결정나무의 구성요소



트리 기반 모델

의사결정나무의 구성요소



Root Node

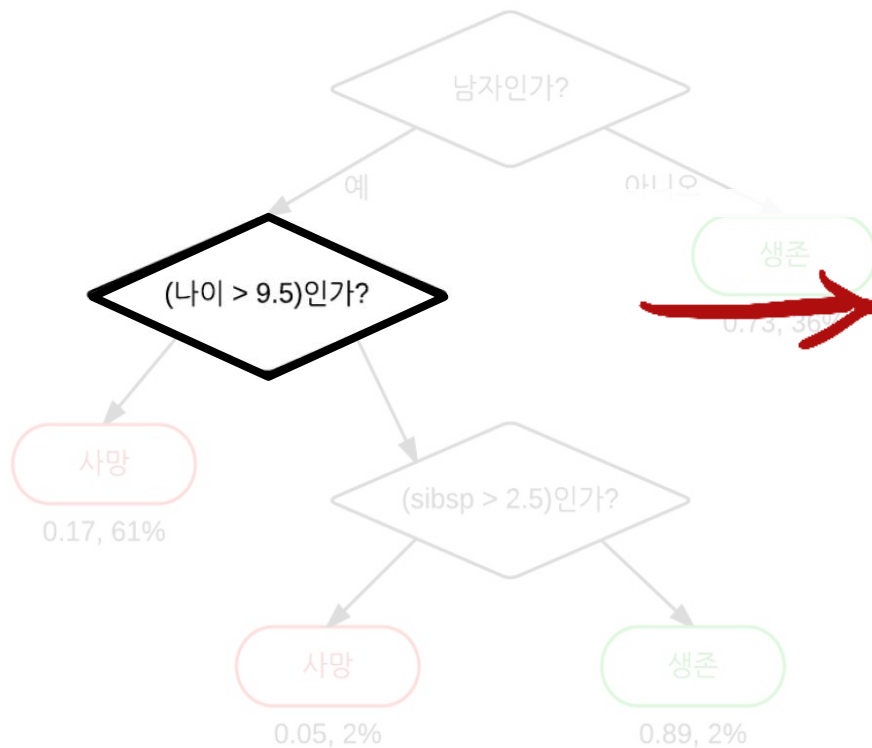
영역이 나뉘지지 않았을 때 항목

SBS대기획
보리밭은나무



트리 기반 모델

의사결정나무의 구성요소



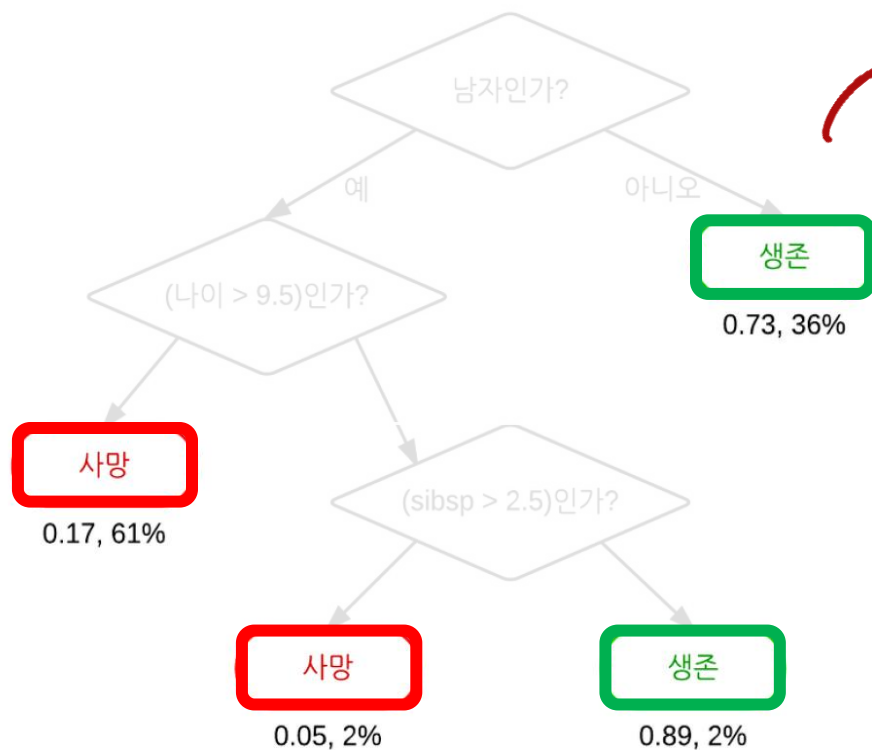
Intermediate Node

더 나뉘질 수 있는 항목



트리 기반 모델

의사결정나무의 구성요소



Terminal Node

더 이상의 하위 항목을 갖지 않는 노드
적합이 끝난 후 결과값이 들어감



트리 기반 모델

의사결정나무의 구성요소

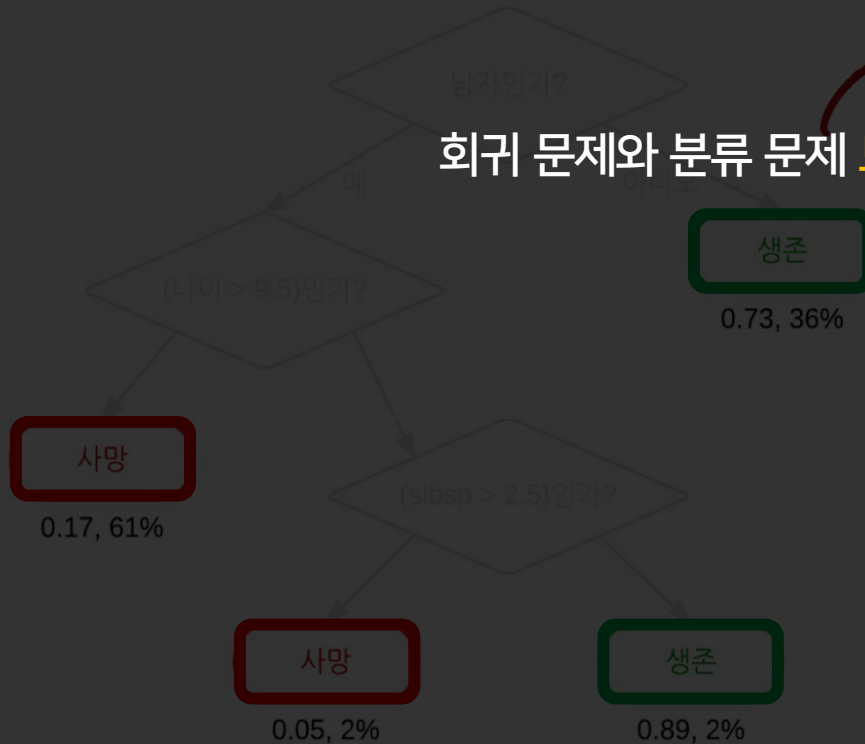
의사결정 나무의 특징

회귀 문제와 분류 문제 **모두**에서 사용 가능



Terminal Node

더 이상의 하위 항목을 갖지 않는 노드
적합이 끝난 후 결과값이 들어감





트리 기반 모델

의사결정나무의 구성요소

의사결정 나무의 특징

회귀 문제와 분류 문제 **모두**에서 사용 가능



Terminal Node

더 이상의 하위 항목을 갖지 않는 노드
적합이 끝난 후 결과값이 들어감

사망
0.17, 61%



회귀 모델

Terminal Node 안의 관측치들의 **평균**

사망
0.05, 2%



분류 모델

Terminal Node 안의 관측치들의 **최빈값**

생존
0.89, 2%

Decision Tree Regressor & Decision Tree Classifier

대표적인 트리 생성 알고리즘

CART

Classification and Regression Tree

독립변수에 상관없이 활용 가능

한 노드에 대해 두 개로만 분기

분기 기준으로 지니 계수 사용

ID3

Iterative Dichotomiser

독립변수가 범주형인 경우 활용 가능

한 노드에 대해 모든 클래스로 분기

분기 기준으로 엔트로피 사용

Decision Tree Regressor & Decision Tree Classifier

대표적인 트리 생성 알고리즘

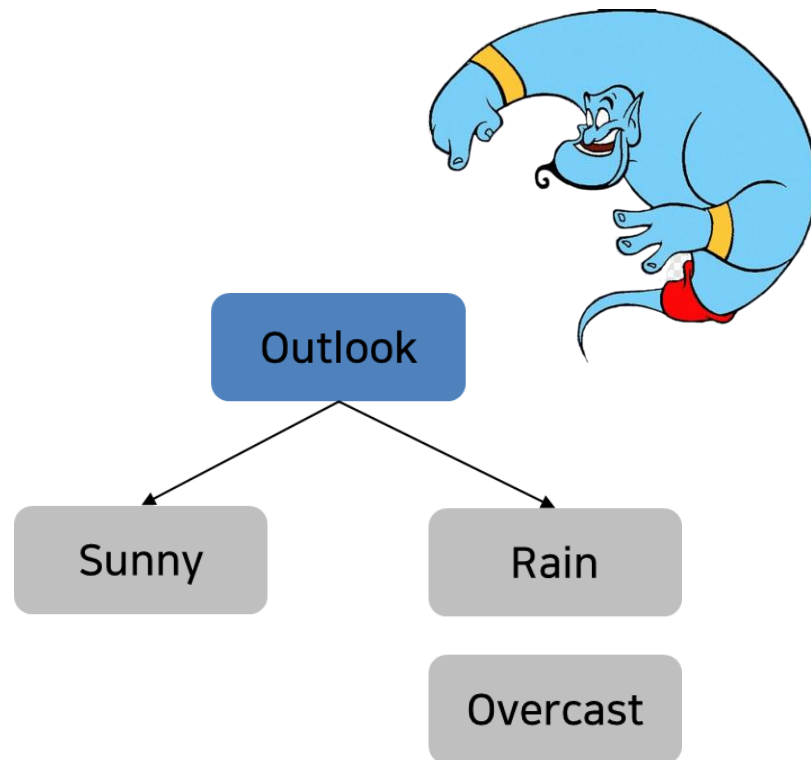
CART

Classification and Regression Tree

독립변수에 상관없이 활용 가능

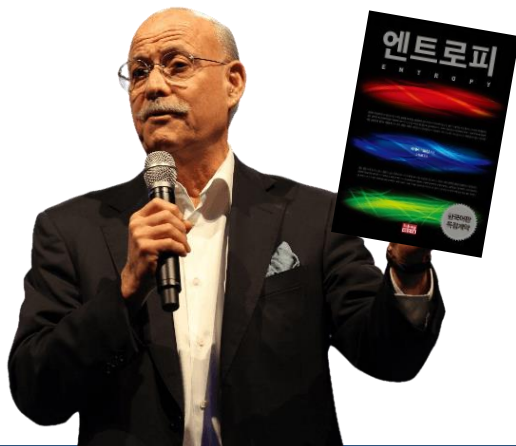
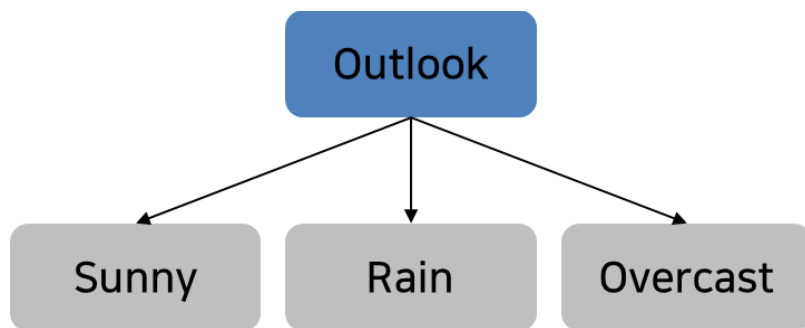
한 노드에 대해 두 개로만 분기

분기 기준으로 지니 계수 사용



Decision Tree Regressor & Decision Tree Classifier

대표적인 트리 생성 알고리즘



ID3

Iterative Dichotomiser

독립변수가 범주형인 경우 활용 가능

한 노드에 대해 모든 클래스로 분기

분기 기준으로 엔트로피 사용

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

?

우리의 목표는 실제 Y값에 가까운 모델을 찾는 것
실제 모델과 예측된 모델이 얼마나 떨어져 있는가?

MSE (Mean squared error)

$$MSE(\text{Mean square error}) = \frac{RSS(\text{Residual sum of square})}{\text{degree of freedom}}$$

Decision Tree의 모델 학습은 RSS 값을 줄이는 방향으로 작동



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Objection function(Least square criterion)

$$\min_{C_m} \sum_{i=1}^N \{y - f(x)\}^2 = \min_{C_m} \sum_{i=1}^N \left[y - \sum_{m=1}^M C_m I(X \in R_m) \right]^2$$

C_m : m번째 node의 최종 예측값(해당 노드의 관측값들의 평균)

N: 전체 관측값의 개수

M: 전체 노드의 개수



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Objection function(Least square criterion)

$$\min_{C_m} \sum_{i=1}^N \{y - f(x)\}^2 = \min_{C_m} \sum_{i=1}^N \left[y - \sum_{m=1}^M C_m I(X \in R_m) \right]^2$$

C_m : m번째 node의 최종 예측값(해당 노드의 관측값들의 평균)

N: 전체 관측값의 개수

M: 전체 노드의 개수



i번째 관측치의 실제 값에 i번째 관측치의 예측값을 뺀 오차의 제곱을 최소화

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Review Rating
10.0	10
11.0	10
12.0	10
13.0	10
...	
18.0	44
18.5	52
19.0	55
...	
30	100
31.0	100



 분기 기준
 Price ≤ 13

R1

Review Rating 평균

$$\hat{c}_1 = 10$$

R2

Review Rating 평균

$$\hat{c}_2 = 70.625$$

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Review Rating
10.0	10
11.0	10
12.0	10
13.0	10
<hr style="border-top: 1px dashed red;"/>	
...	
18.0	44
18.5	52
19.0	55
...	
30	100
31.0	100

RSS of R1

$$(10 - 10)^2 + (10 - 10)^2 + \dots \\ \cdot + (10 - 10)^2 = 0$$

RSS of R2

$$(10 - 70.625)^2 + (13 - 70.625)^2 + \dots \\ \cdot + (100 - 70.625)^2 = 13455.75$$

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Review Rating
10.0	10
11.0	10
12.0	10
13.0	10
...	
18.0	44
18.5	52
19.0	55
...	
30	100
31.0	100

Total RSS

$$\begin{aligned} &\text{RSS of R1} + \text{RSS of R2} \\ &= 13455.75 \end{aligned}$$





Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Review Rating
10.0	10
11.0	10
12.0	10
13.0	10
...	
18.0	44
18.5	52
19.0	55
...	
30	100
31.0	100

분기 조건에 따라 RSS 값이 달라짐!



Total RSS

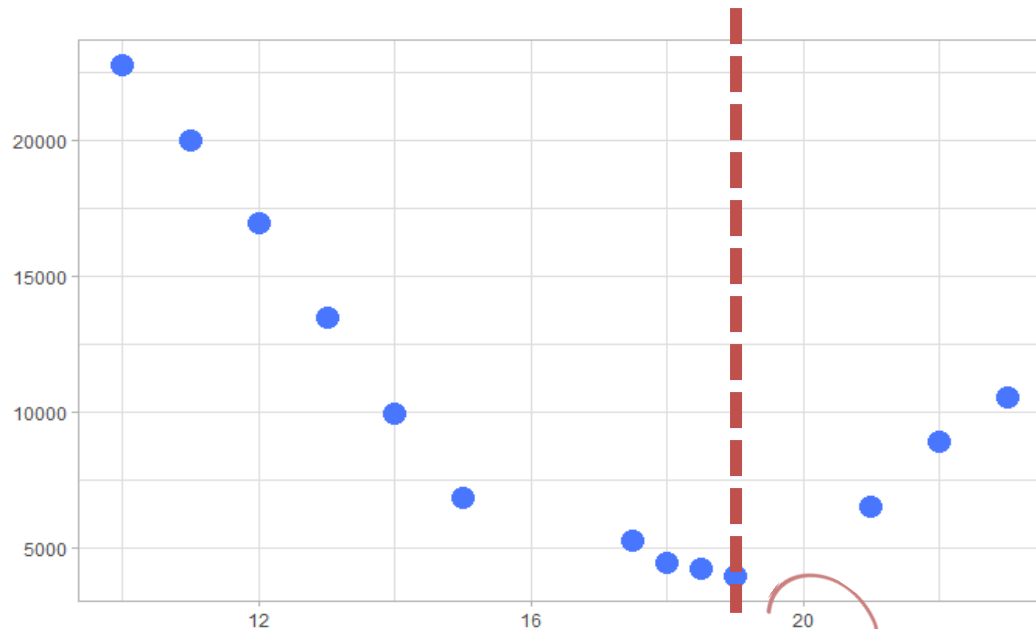
RSS값이 가장 작은 조건을

의사 결정 나무의 분기 기준으로 채택



Decision Tree Regressor & Decision Tree Classifier

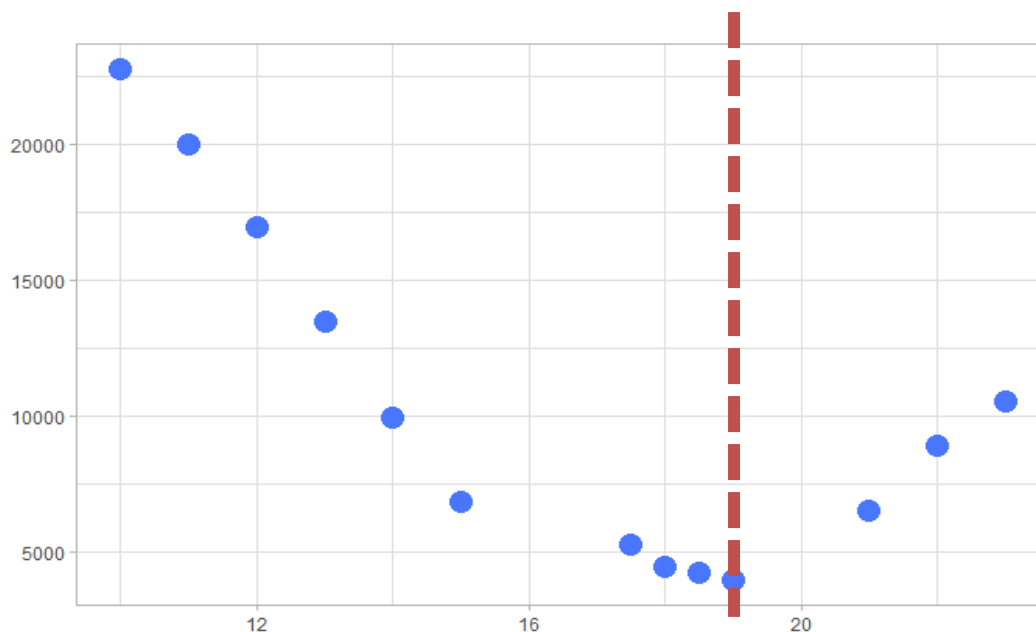
Decision Tree Regressor



Price=19 일 때 RSS값이 최소
따라서, 첫번째 분기 기준은 $\text{Price} \leq 19.0$

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor



RSS값이 가장 작은 기준을 찾아가는 알고리즘을 따름

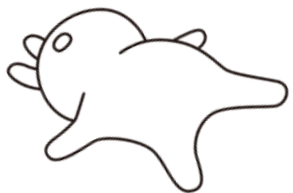
“ 탐욕적 (Greedy) 알고리즘 ”



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Parking Fee	Foreign	Review Rating
10	0.0	0	10
11.0	1.0	0	10
...
17.5	7.5	1	35
18.0	8.0	1	44
18.5	8.5	1	52
19.0	9.0	0	55
...



만약 독립변수가 2개 이상인 경우 어떻게 해야 할까?



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Parking Fee	Foreign	Review Rating
10	0.0	0	10
11.0	1.0	0	10
...
17.5	7.5	1	35
18.0	8.0	1	44
18.5	8.5	1	52
19.0	9.0	0	55
...



각 독립 변수 별로 RSS 값을 최소로 만드는 기준 찾기

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Parking Fee	Foreign	Review Rating
10	0.0	0	10
11.0	1.0	0	10
...
17.5	7.5	1	35
18.0	8.0	1	44
18.5	8.5	1	52
19.0	9.0	0	55
3961.8	64214.8	11685	...

RSS



각 기준에 따라 RSS 값을 비교하여

가장 작은 RSS 값을 도출한 독립변수를 분기 기준으로 정함

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Parking Fee	Foreign	Review Rating
10	0.0	0	10
11.0	1.0	0	10
...
17.5	7.5	1	35
18.0	8.0	1	44
18.5	8.5	1	52
19.0	9.0	0	55
3961.8	64214.8	11685	...

RSS



Price가 최소 RSS를 산출



Price ≤ 19.0를 Label로 설정

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Regressor

Price	Parking Fee	Foreign	인생view Rating
10	0.0	0	평안없는 반복 10
11.0	1.0	0	반복에 10
...	지긋지긋한 자가 ...
17.5	7.5	-1	성취한다 35



나뉜 R1, R2 안에서 다시 최소 RSS를 산출하는 독립변수 찾아 반복

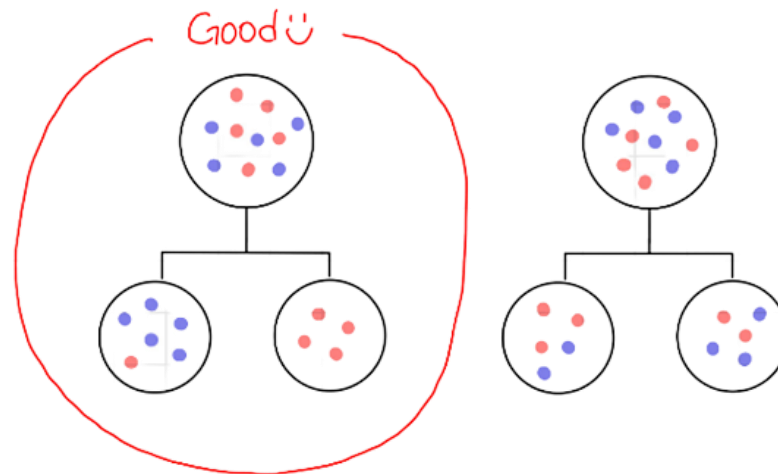


RSS 값이 더 이상 작아지지 않을 때까지 분할하면 terminal node에는 각 상황에서의 예측값의 평균이 배치

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

분류 모델인 Decision Tree Classifier의 경우
영역들의 불순도를 기준으로 분기



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

분류 모델인 Decision Tree Classifier의 경우
영역들의 불순도를 기준으로 분기

불순도(Impurity)

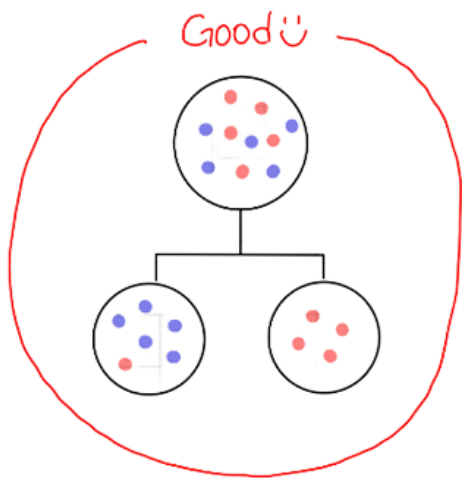
분기된 영역 내에서 각기 다른 다양한 범주들의 개체가
얼마나 포함 되어있는지에 대한 지표

아주 불순하네!

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

분류 모델인 Decision Tree Classifier의 경우
영역들의 불순도를 기준으로 분기



영역 내의 output들의 클래스가 잘 분리되게
불순도가 낮아지도록 학습을 진행

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

불순도(Impurity)의 지표

Misclassification Rate, Gini index, Entropy

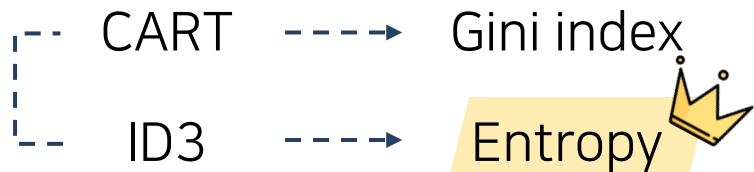
CART	----->	Gini index
ID3	----->	Entropy

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

불순도(Impurity)의 지표

Misclassification Rate, Gini index, Entropy



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier



\hat{p}_{mk} 를 한 영역 내의 **특정 클래스의 비율**로 생각하고 엔트로피의 수식을 보자!

$$Entropy = - \sum_{k=1}^K \hat{p}_{mk} \log_2(\hat{p}_{mk})$$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

\hat{p}_{mk} : k번째 사건의 발생확률
k: 사건의 종류

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier



영역 내의 특정 클래스의 비율이 비슷하면

\hat{p}_{mk} 를 한 영역 내의 특정 클래스의 비율이라고 생각하면 엔트로피의 공식을 보자!

해당 영역은 특정 Class로 분류하기 어려움

$$Entropy = - \sum_{k=1}^K \hat{p}_{mk} \log_2(\hat{p}_{mk})$$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in X_m} I(y_i = k)$$

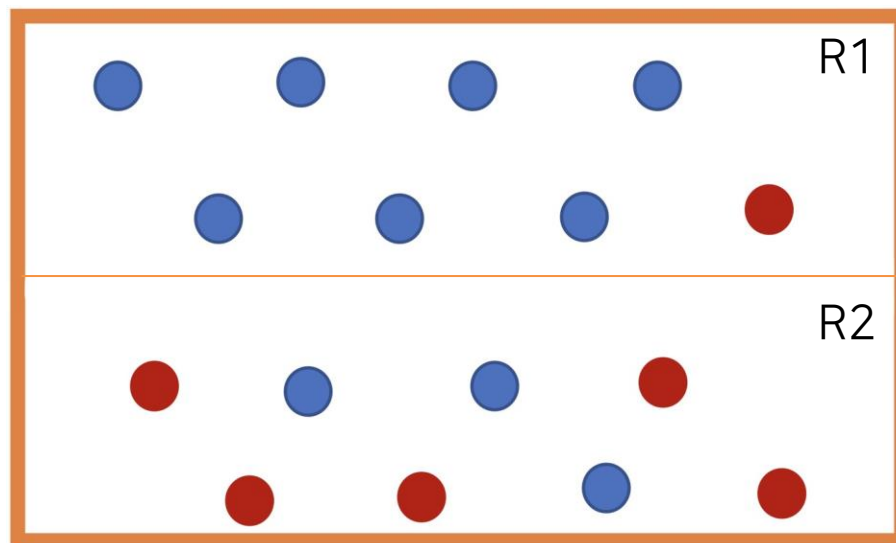
해당 영역의 불순도(Entropy)가 높다

\hat{p}_{mk} : k번째 사건의 발생확률

k: 사건의 종류

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

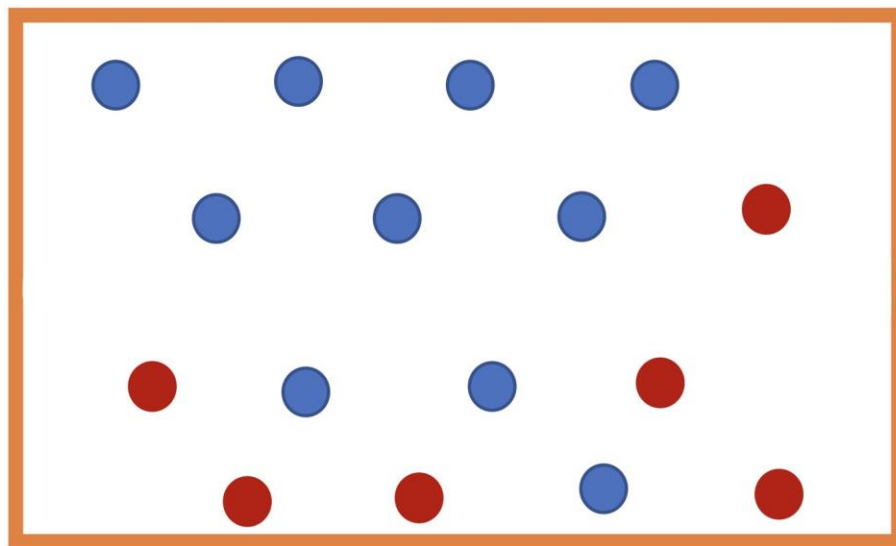


분기 이전의 영역의 엔트로피와
분기 이후 영역들의 엔트로피는 어떻게 변할까?



Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

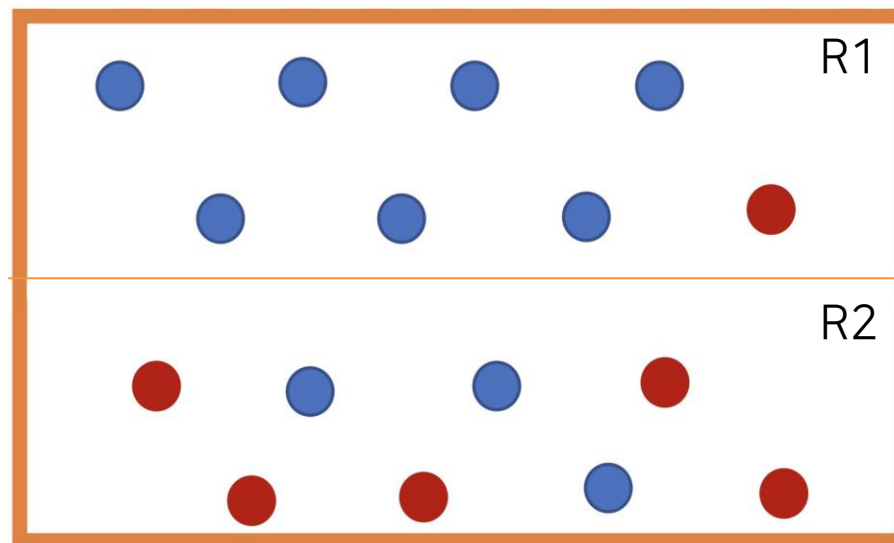


분기 전

$$Entropy(X) = -\frac{10}{16}\log_2\left(\frac{10}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) \approx 0.95$$

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier



가중평균을 이용하여 계산

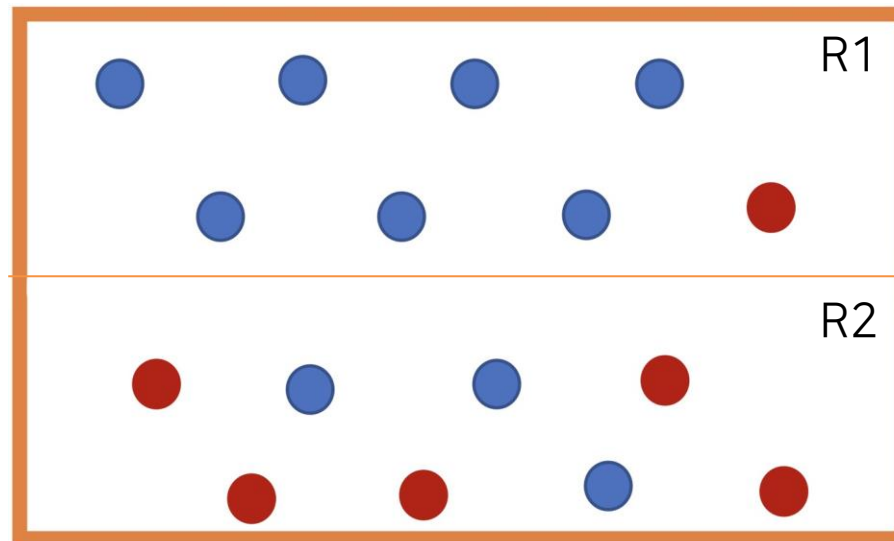
분기 후

$$E(S, P) = \sum_p \frac{|P|}{|S|} Entropy(S)$$

$|S|$: 전체 데이터 개수, $|P|$: 영역의 데이터 개수

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier




분기 후

$$E(S, P) = -\frac{8}{16} \text{Entropy}(R_1) + \frac{8}{16} \text{Entropy}(R_2) \approx 0.75$$

Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

분기 후 Entropy 감소


$$Entropy(X) = -\frac{10}{16}\log_2\left(\frac{10}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) \approx 0.95$$

$$E(S, P) = -\frac{8}{16}Entropy(R_1) + \frac{8}{16}Entropy(R_2) \approx 0.75$$

0.2 감소



전체 불순도 합을 줄이는 방향으로 분류 실행
더 이상 불순도가 줄어들지 않을 때 중지





Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

그렇다면 계속 분기하여

분기 후 Entropy 감소

완전하게 **Pure**하게 만들면 안되는 걸까?

$$Entropy(X) = -\frac{10}{16}\log_2\left(\frac{10}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) \approx 0.95$$

0.2 감소

$$E(S, P) = -\frac{8}{16}Entropy(R_1) + \frac{8}{16}Entropy(R_2) \approx 0.75$$



전체 불순도 합을 줄이는 방향으로 분류 실행

더 이상 불순도가 줄어들지 않을 때 중지





Decision Tree Regressor & Decision Tree Classifier

Decision Tree Classifier

그렇다면 계속 분기하여

분기 후 Entropy 감소

완전하게 **Pure**하게 만들면 안되는 걸까?

$$Entropy(X) = -\frac{10}{16}\log_2\left(\frac{10}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) \approx 0.95$$

0.2 감소

$$E(S, P) = -\frac{8}{16}Entropy(R_1) + \frac{8}{16}Entropy(R_2) \approx 0.75$$

필연적으로 **과적합**이 발생



전체 불순도 합을 줄이는 방향으로 분류 실행

더 이상 불순도가 줄어들지 않을 때 중지

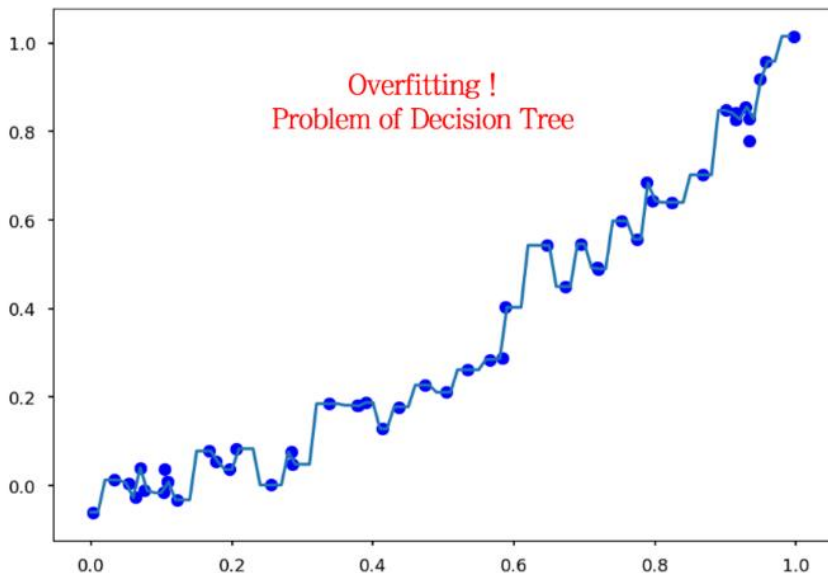


트리 기반 모델 과적합 방지법



모든 데이터셋이 불순도가 0에 근접할 정도로 분기 진행 시

Overfitting 발생



Low Bias

High Variance

Non-Robust Model



가지치기(Pruning)

가지치기(Pruning)

트리 모델에서 과적합을 방지하기 위해 적절한 수준에서 Terminal node 결합

사전 가지치기

트리 모델의
하이퍼파라미터인
Tree의 깊이를
사전에 제한두는 것

사후 가지치기

Full Tree를 만든 후
적절한 수준에서
Terminal Node를
결합하는 방법

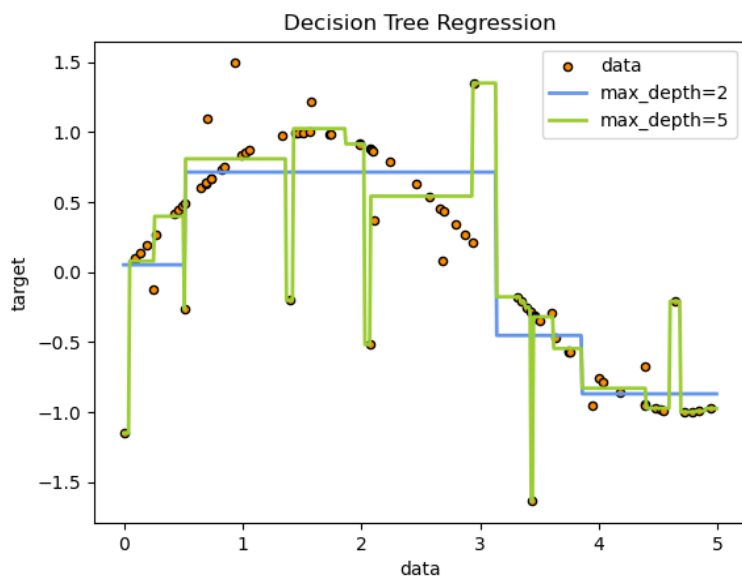


가지치기(Pruning)

사전 가지치기

“ Max_depth ”

트리가 분기하는 깊이에 제약을 걸어주는 하이퍼파라미터



Max_depth의 값이 커질수록
모델이 복잡해져 성능 저하 초래

✧ 하이퍼파라미터 튜닝으로
최적의 값을 찾아야함

가지치기(Pruning)

사후 가지치기

데이터에 대해 충분히 설명력 가지면서
Input space가 지나치게 분류되는 것을 방지



트리 모델을 끝까지 분기



밑에서부터 하나씩 자식노드의 Error와 부모노드의 Error 비교



만약 자식노드의 Error가 부모노드의 Error보다 더 크다면 자식 노드 제거

앙상블 기법 (Ensemble Method)

트리 모델의 한계



트리는 Non-Robust하고 Variance가 크기에
가지치기를 하더라도 **Overfitting**의 한계를 벗어나지 못함



어떻게 해결할 수 있을까



앙상블 기법 (Ensemble Method)

트리 모델의 한계



트리는 Non-Robust하고 Variance가 크기에
가지치기를 하더라도 **Overfitting**의 한계를 벗어나지 못함



어떻게 해결할 수 있을까



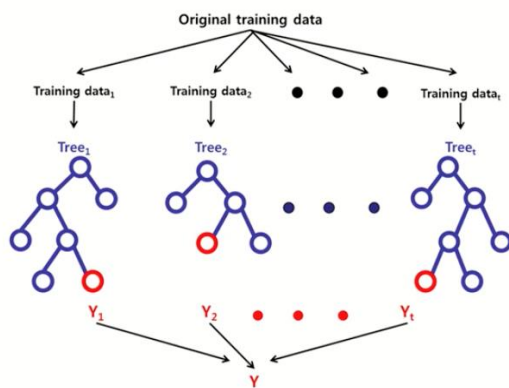
Decision Tree들을 모아 활용하는 **Ensemble 기법**

앙상블 기법 (Ensemble Method)

앙상블 기법 (Ensemble Method)

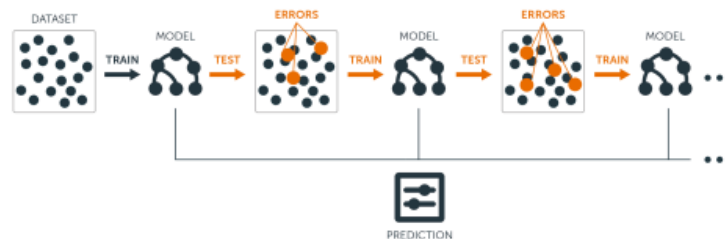
여러 개의 의사결정 나무를 합쳐서 하나의 강한 모델을 만들어내는 방식

Bagging



EX) RandomForest

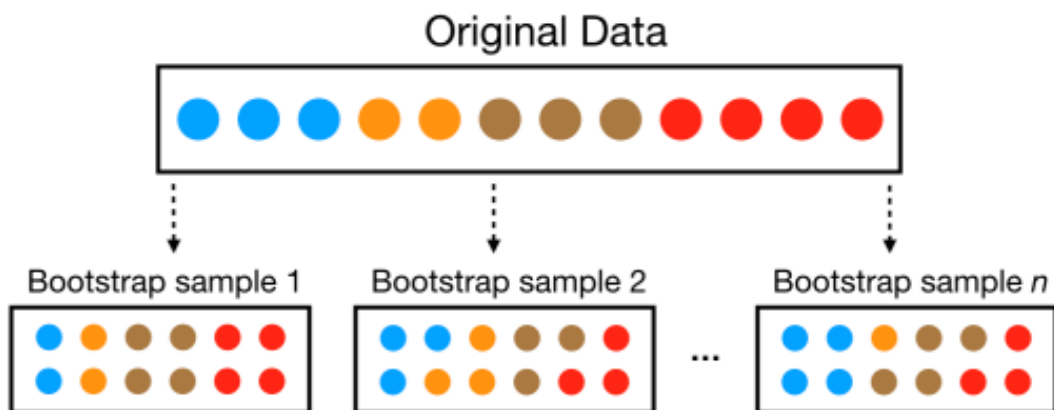
Boosting



EX) GBM

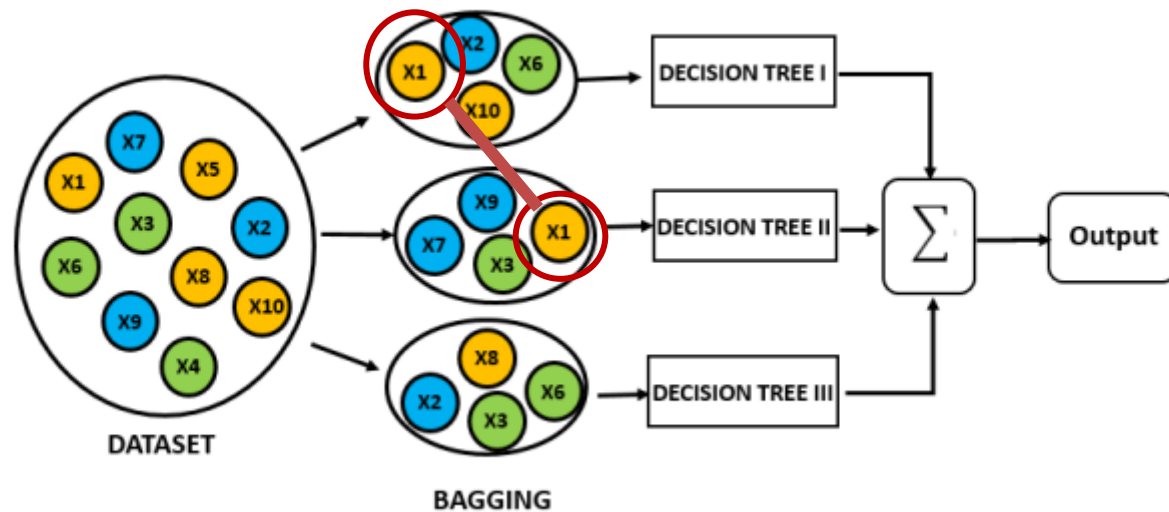
배깅 기법(Bagging Method)

부트스트랩 (Bootstrap)



샘플을 추출할 때 복원 추출이 가능해
각 샘플들마다 중복되는 관측값 존재

배깅 기법(Bagging Method)



여러 개의 의사결정 나무를 만들되, 각각의 N개의 의사결정 나무를 **부트스트랩 기법**으로 추출된 N개의 데이터 셋으로 학습을 시키는 기법

데이터들의 부분집합에 중복 값 허용!

배깅 기법(Bagging Method)

효과와 한계

효과

각각의 데이터셋에 중복값 허용
→ N개 모델들 간의 분산을 줄임

한계

각각의 데이터셋에 중복값 허용
→ 최종 모델의 분산은 크게 줄어들지 않음

배깅 기법(Bagging Method)

효과와 한계



효과

각 모델들의 분산을 줄여주기 위해서
→ 각각의 데이터셋에 중복값 허용
→ N개 모델들 간의 분산을 줄임

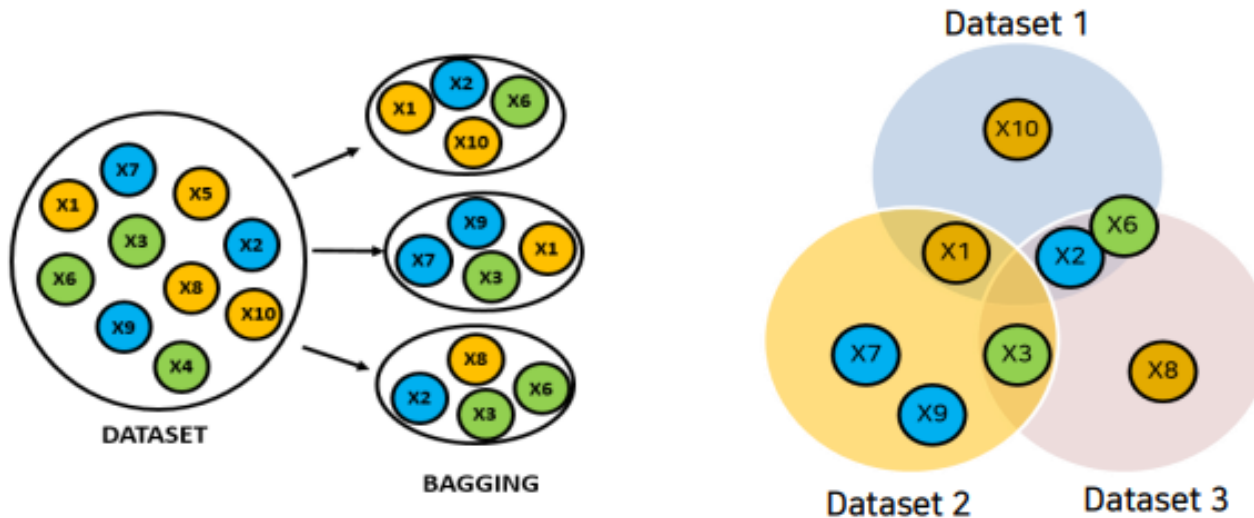
표본의 **중복값을 허용**했는데, 왜 이게 문제가 되는 걸까?

한계

각각의 데이터셋에 중복값 허용
→ 최종 모델의 분산은 크게 줄어들지 않음

배깅 기법(Bagging Method)

배깅(Bagging)의 한계



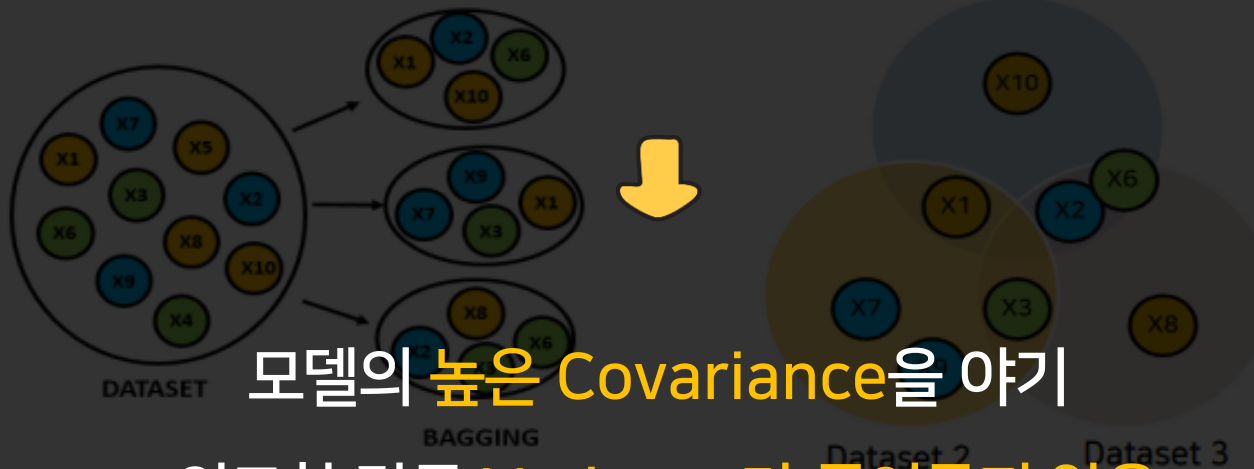
데이터의 중복값을 허용하면, 데이터들은 서로 **correlated**되어
트리가 분기하는 기준 비슷해짐

배깅 기법(Bagging Method)



배깅(Bagging)의 한계

비슷한 데이터로 형성된 N개의 모델들도 결국 유사한 형태



모델의 높은 Covariance을 야기
의도한 만큼 Variance가 줄어들지 않음

데이터의 중복값을 허용하면, 데이터들은 서로 correlated

→ 트리가 분기하는 기준 비슷해짐

배깅 기법(Bagging Method)

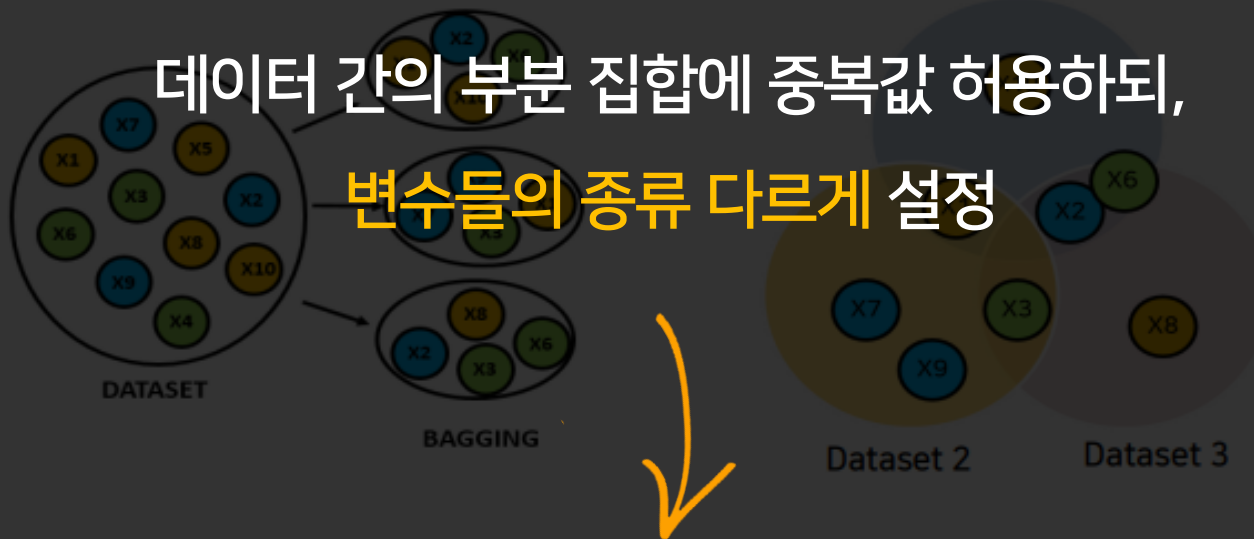
배깅(Bagging)의 한계



각각의 트리들이 Decorrelate 하게끔

데이터 간의 부분 집합에 중복값 허용하되,

변수들의 종류 다르게 설정



“

Random Forest

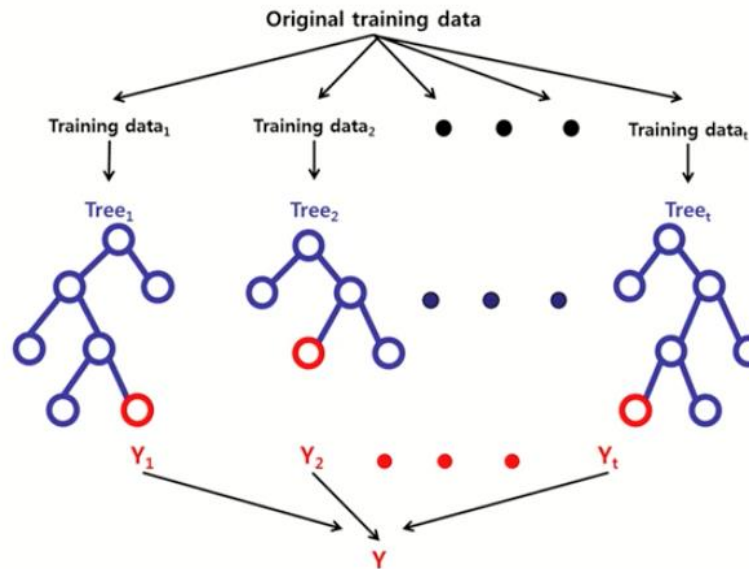
”

데이터의 중복값을 허용하진, 데이터들은 서로 correlated

→ 트리가 분기하는 기준 비슷해짐

배깅 기법(Bagging Method)

랜덤 포레스트(Random Forest)



매 모델링마다 사용되는 feature의 개수를 랜덤하게 선택하여
모델 간의 분산을 더욱 줄임

ex) 분류 - 전체 feature 개수의 제곱근 / 예측 - 전체 feature 개수의 1/3



부스팅 기법(Boosting Method)

배깅 기법 VS 부스팅 기법

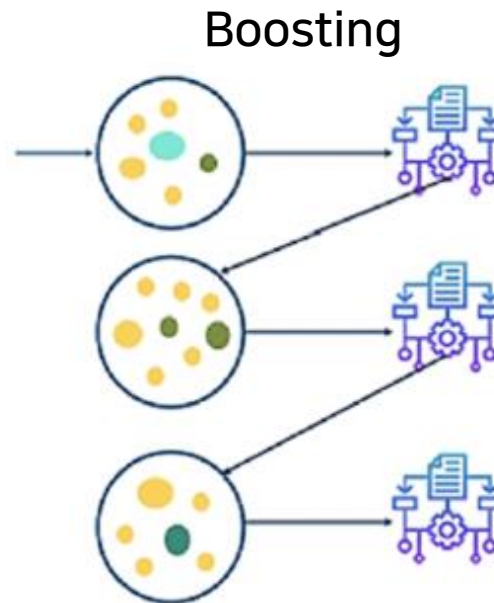


각각의 모델에 부트스트래핑으로 생성된 데이터 부분집합 적합

각각의 트리 모델 병렬적으로 생성

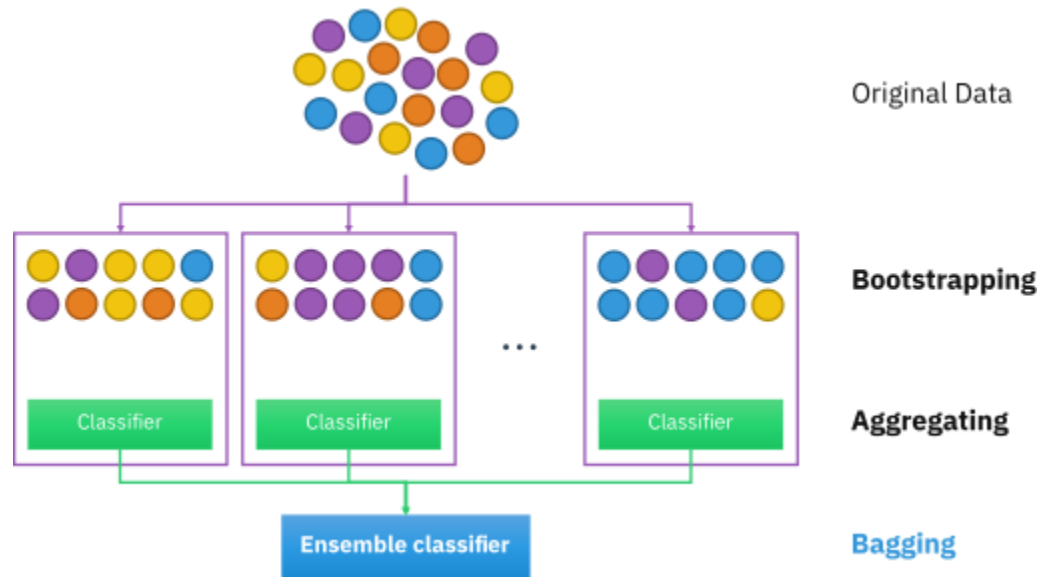
부스팅 기법(Boosting Method)

배깅 기법 VS 부스팅 기법



각각의 모델에 전체 데이터셋 적합
순차적으로 N개의 Decision Tree 생성

부스팅 기법(Boosting Method)



여러 개의 **weak model**들을 모아서
하나의 **strong model**을 만들어내는 기법

오분류율이 굉장히 높은(50% 넘는) 모델

부스팅 기법(Boosting Method)

학습 프로세스

AdaBoost

학습할 때마다 각각의 데이터에
가중치를 다르게 주는 기법

GBM

경사 하강법을 사용하여
Weak Model들을 생성

부스팅 기법(Boosting Method)

학습 프로세스

AdaBoost

학습할 때마다 각각의 데이터에
가중치를 다르게 주는 기법

GBM



경사 하강법을 사용하여
Weak Model들을 생성



XGBoost, LGBM의 기반

부스팅 기법(Boosting Method)

GBM (Gradient Boosting Method)



경사하강법을 사용해 weak model들을 생성하는 기법
손실함수를 최소화 하는 방향으로 학습을 진행

부스팅 기법(Boosting Method)

GBM (Gradient Boosting Method)



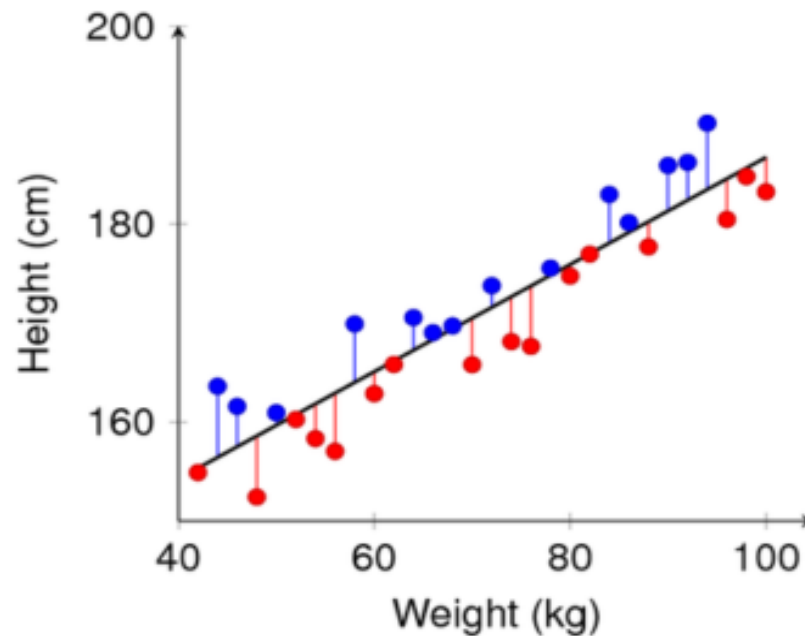
딤러닝 클린업 참고 바람!

경사하강법을 사용해 weak model들을 생성하는 기법
손실함수를 최소화 하는 방향으로 학습을 진행



GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시

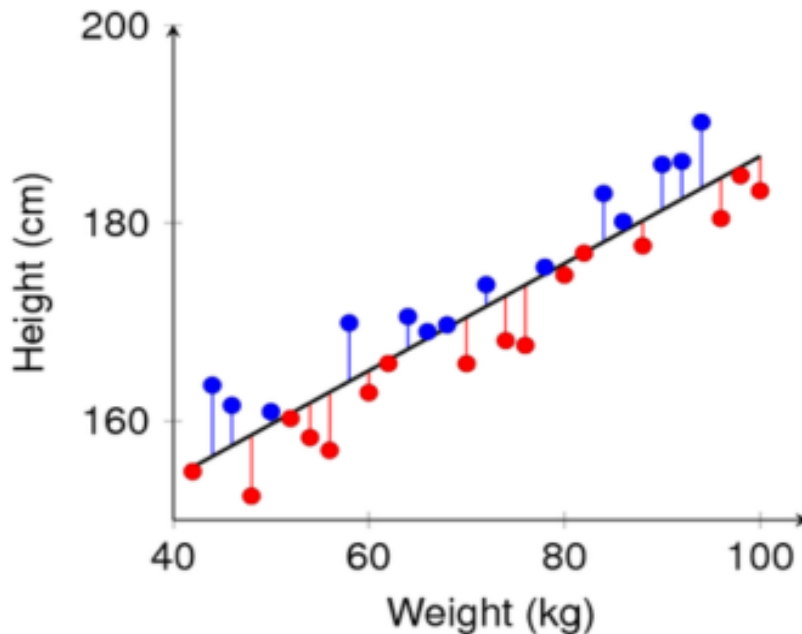


$f_1(x)$ 라는 간단한 선형회귀 모델
수직선으로 표시된 잔차



GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시



$$y = f_1(x) + \text{Residual}_1$$

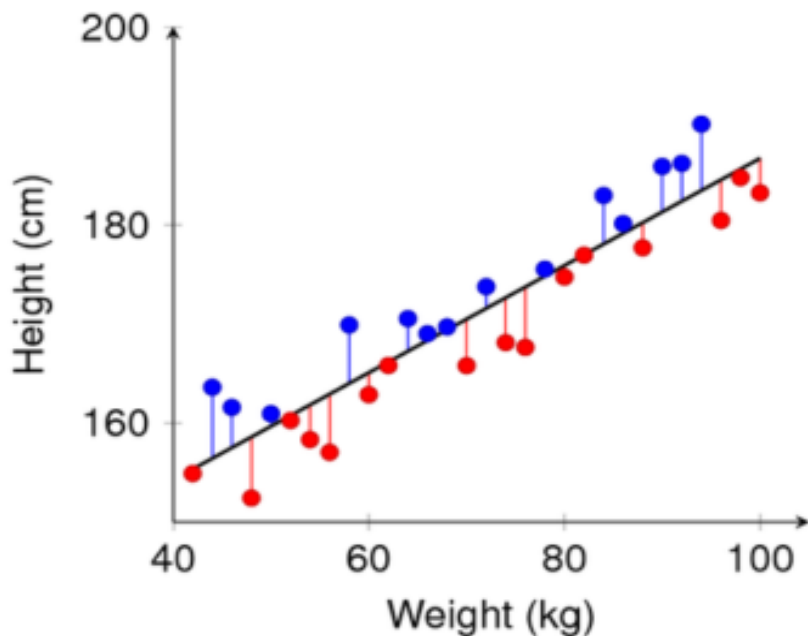
$$y - f_1(x) = \text{Residual}_1$$

$\hat{y} = f_1(x)$: 회귀 모델에 대한 예측값

Residual_1 : 모델의 잔차

GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시



$$y = f_1(x) + \text{Residual}_1$$

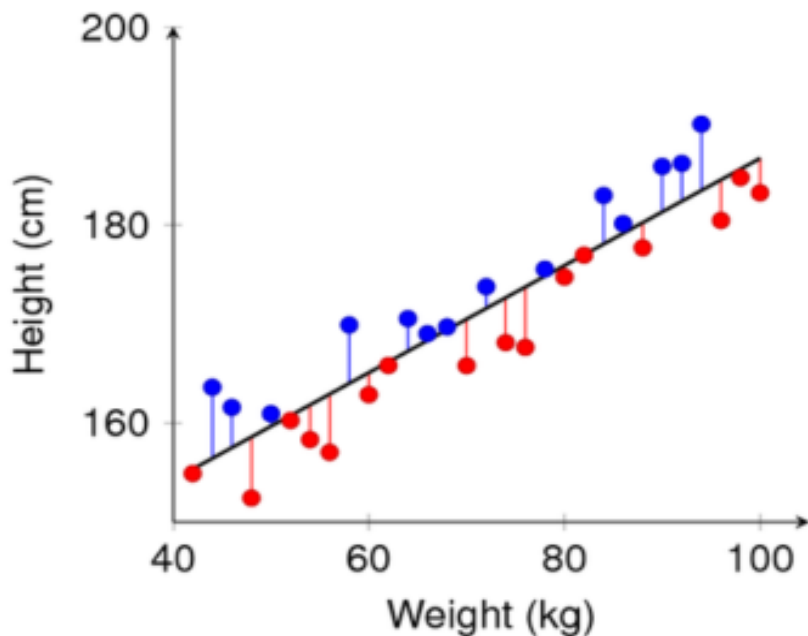
$$\text{Residual}_1 = f_2(x) + \text{Residual}_2$$

$f_1(x)$ 이 예측하지 못한 부분인

잔차 Residual_1 를 예측하는 모델 $f_2(x)$ 생성

GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시



$$y = f_1(x) + \text{Residual}_1$$

$$\text{Residual}_1 = f_2(x) + \text{Residual}_2$$



$$y = f_1(x) + f_2(x) + \text{Residual}_2$$

최종적으로 y값은 weak model $f_1(x)$, $f_2(x)$ 이 합쳐져
잔차가 더해진 꼴로 모델이 구성됨

GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시

$$y = f_1(x) + \text{Residual}_1$$

$$y = f_1(x) + f_2(x) + \text{Residual}_2$$

$$y = f_1(x) + f_2(x) + f_3(x) + \text{Residual}_3$$

...

$$y = f_1(x) + f_2(x) + f_3(x) + \cdots + f_n(x) + \text{Residual}_n$$

매우 작은 오차

잔차를 예측하는 모델을 재생성하고 이 과정을 반복
최종 모델은 weak model이 n개 합쳐진 꼴에 해당

GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시

$$y = f_1(x) + \text{Residual}_1$$

$$y = f_1(x) + f_2(x) + \text{Residual}_2$$

$$y = f_1(x) + f_2(x) + f_3(x) + \text{Residual}_3$$

...



$$y = f_1(x) + f_2(x) + f_3(x) + \cdots + f_n(x) + \text{Residual}_n$$

매우 작은 오차

BUT, 잔차를 매번 구해내는 것은 쉽지 않기에
경사하강법을 통해 구해낸 **Negative Gradient**로 대체하여 사용

GBM (Gradient Boosting Method)

단순선형회귀 모델의 예시

$$y = f_1(x) + \text{Residual}_1$$

$$y = f_1(x) + f_2(x) + \text{Residual}_2$$

잔차를 경사하강법을 통해 얻어낸

Negative Gradient와 같다고 할 수 있을까?

매우 작은 오차

$$y = f_1(x) + f_2(x) + f_3(x) + \dots + f_n(x) + \text{Residual}_n$$



BUT, 잔차를 매번 구해내는 것은 쉽지 않기에

경사하강법을 통해 구해낸 **Negative Gradient**로 대체하여 사용

GBM (Gradient Boosting Method)

Negative Gradient

$$\text{loss function} = \frac{1}{2} (y_i - f(x_i))^2$$

실제 y 값

$f(x_i)$ 통해 예측한 값

손실함수(Loss Function)로 **Squared Error**를 사용한다고 가정

GBM (Gradient Boosting Method)

Negative Gradient

$$\text{loss function} = \frac{1}{2} (y_i - f(x_i))^2$$

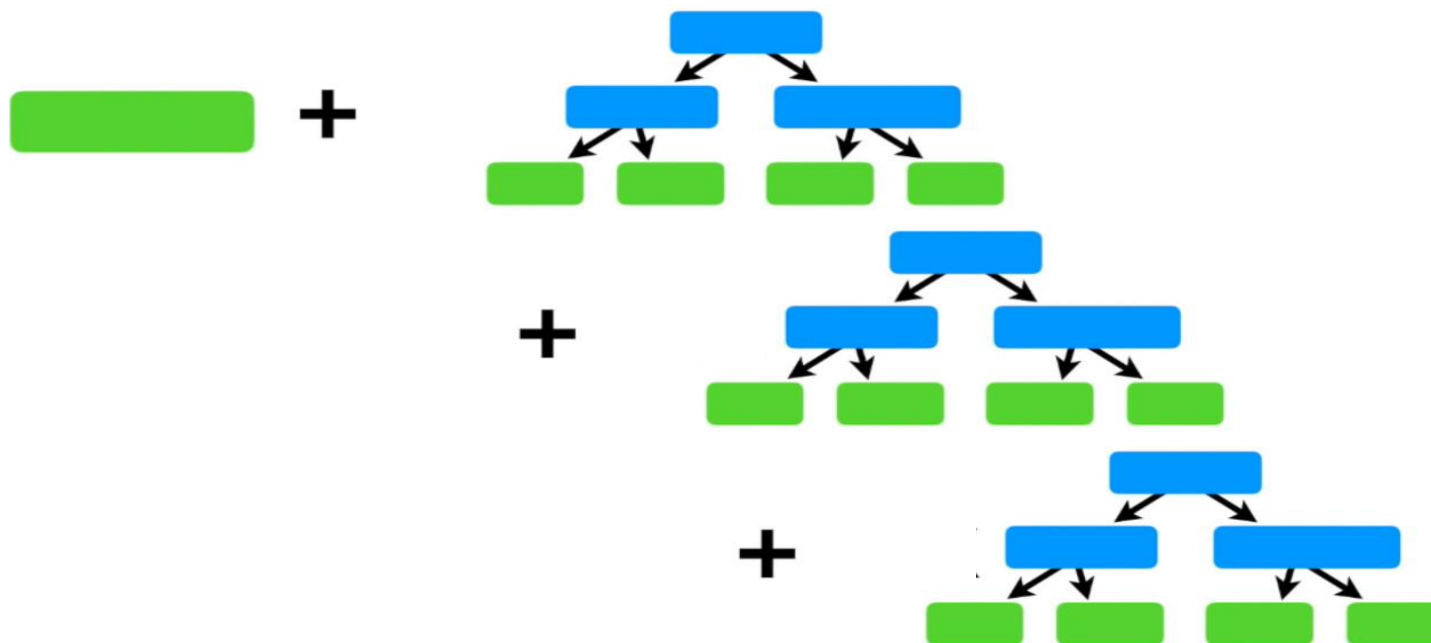
$$\text{negative gradient} = -\frac{\partial(\text{loss function})}{\partial f(x_i)} = f(x_i) - y_i = \text{residual}$$

손실함수를 편미분해 Negative Gradient를 구하면

Residual, 잔차가 나옴

GBM (Gradient Boosting Method)

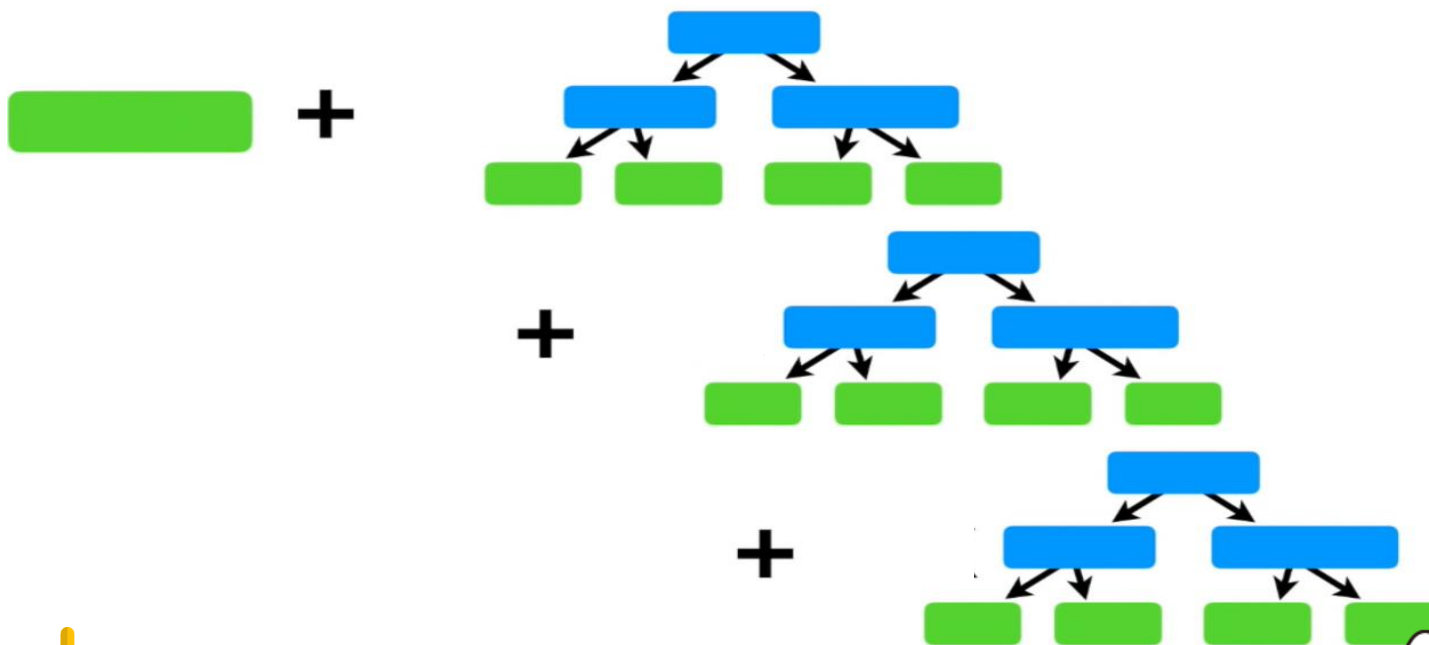
GBM의 예측방식



경사 하강법으로 생성된 여러 개의 weak tree에
전체 데이터를 순차적으로 적합시켜 예측 값을 출력

GBM (Gradient Boosting Method)

GBM의 예측방식

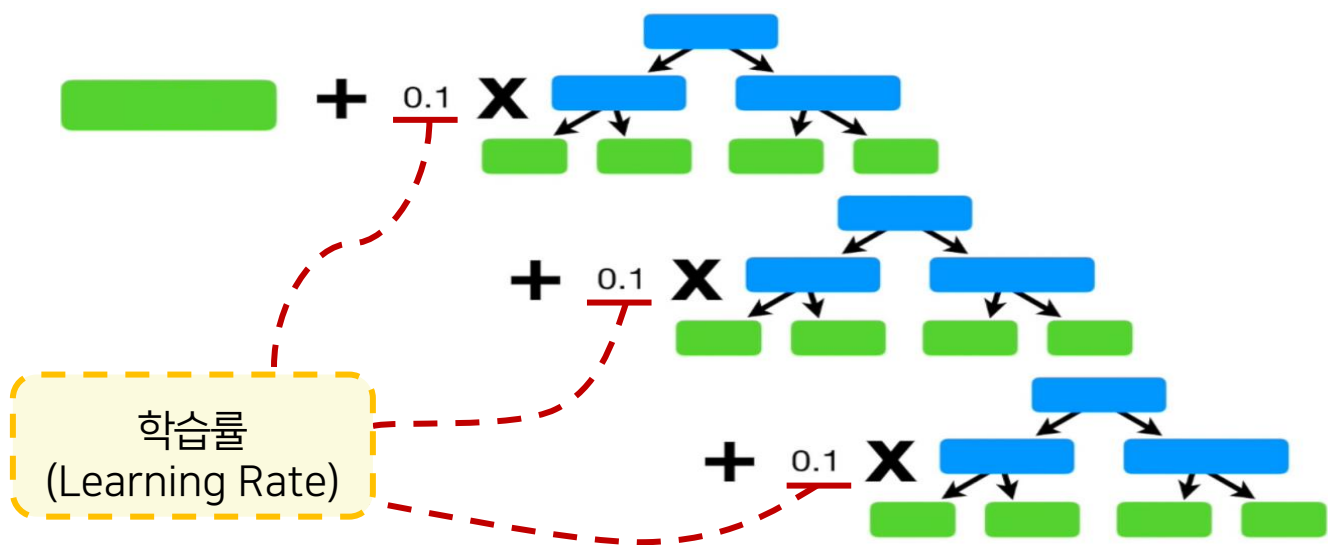


Residual 값을 그대로 받아 들여 예측을 하게 되면
학습데이터 값과 완전히 일치하는 과적합 발생 가능



GBM (Gradient Boosting Method)

GBM의 예측방식



0-1 사이의 **학습률**을 곱해
residual의 반영 비율을 낮추고 과적합을 방지

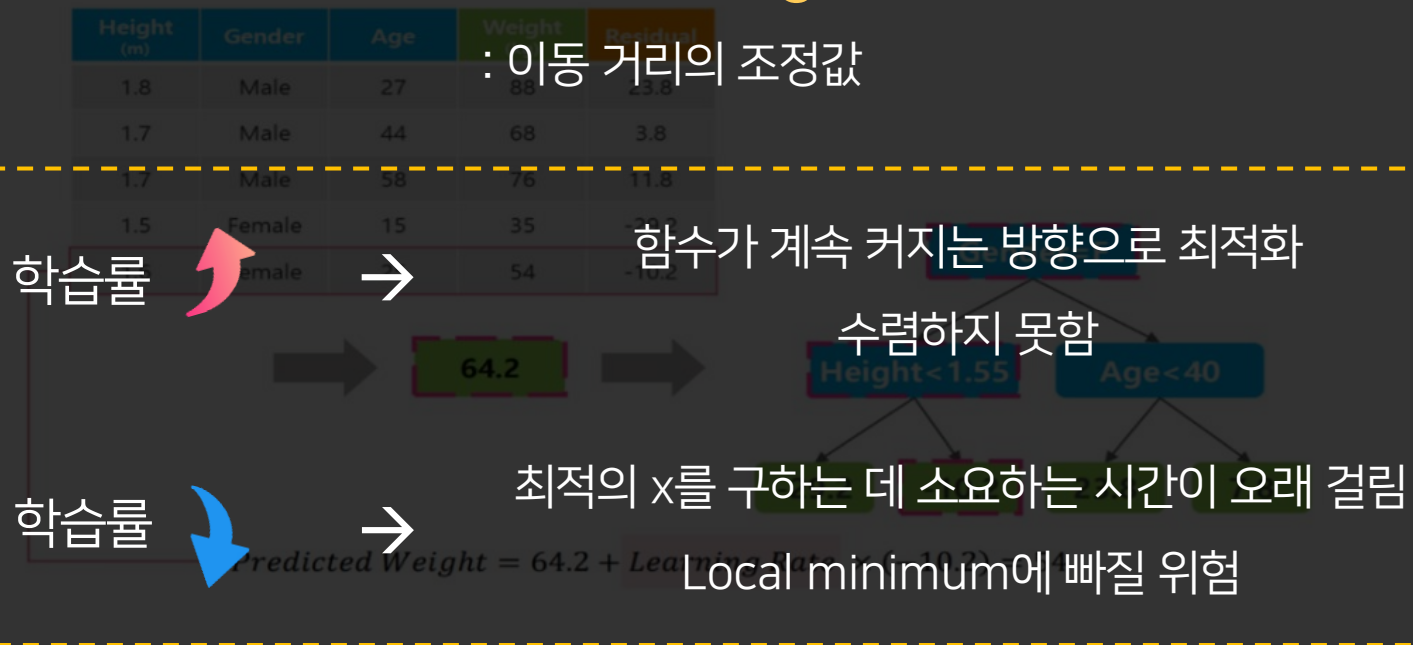


GBM (Gradient Boosting Method)

GBM의 예측방식

학습률 (Learning Rate)

: 이동 거리의 조정값



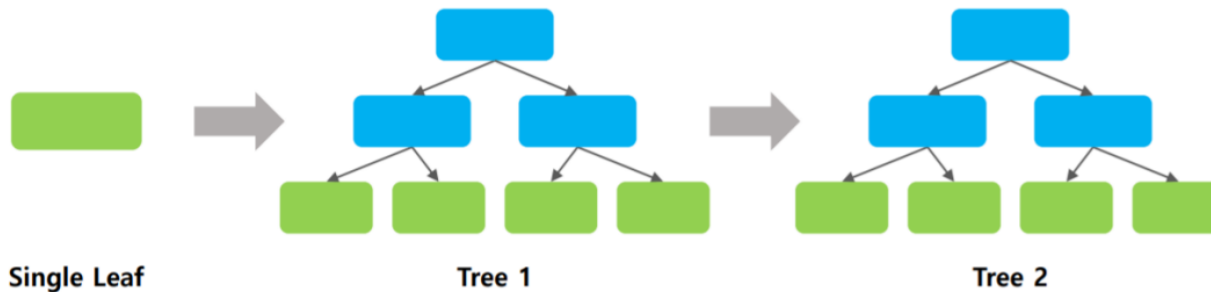
0-1 사이의 학습률을 곱해
학습률의 값에 따라 모델의 성능이 달라짐
residual의 반영 비율을 낮추고 과적합을 방지

GBM (Gradient Boosting Method)

GBM의 예측방식

Height (m)	Gender	Age	Weight (kg)	Residual	Residual	Residual
1.8	Male	27	88			
1.7	Male	44	68			
1.7	Male	58	76			
1.5	Female	15	35			
1.6	Female	25	54			

...



손실 함수를 미분 가능한 metric으로 설정하면
회귀 문제가 아닌 **분류 문제**에도 적용 가능

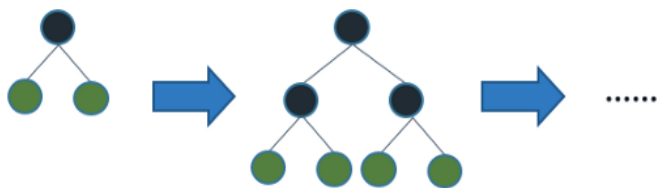


GBM (Gradient Boosting Method)

GBM의 예측방식

XGboost

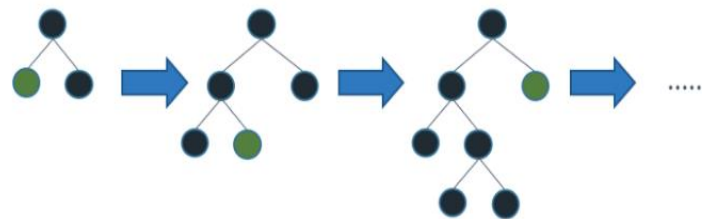
Level-wise tree growth



균형한 형태로 노드 분할

LightGBM

Leaf-wise tree growth



불균형한 형태로 노드 분할

2

비선형 모델

비선형 모델(Non-Linear Model)이란?



데이터를 어떻게 변형하더라도 파라미터를
선형결합식으로 표현할 수 **없는** 모델

현실에서 접하는 많은 데이터는 선형적으로 생성되지 않기 때문에
선형 회귀식은 예측 성능이 떨어짐

비선형 모델(Non-Linear Model)이란?



데이터를 어떻게 변형하더라도 파라미터를
선형결합식으로 표현할 수 **없는** 모델

현실에서 접하는 많은 데이터는 선형적으로 생성되지 않기 때문에
선형 회귀식은 예측 성능이 떨어짐

⇒ 비선형 모델을 통해 **선형성의 가정을 완화**하면서
가능한 **많은 해석 가능성**을 유지하는 것이 목표 !

비선형 모델(Non-Linear Model)

비선형 모델의 학습과정



Input matrix에 해당되는 X 를 비선형적으로 **transform**

$$X \Rightarrow X^*$$



변형된 Input matrix X^* 와 Y 를 **linear**한 모델로 학습

$$Y = \beta_0 + \beta_1 * X_1^* + \beta_2 * X_2^* + \dots + \beta_n * X_n^*$$

비선형 모델(Non-Linear Model)

비선형 모델의 학습과정



선형결합으로 계산이 가능한 형태로
만들어주는 것이 **기저함수**

$$X \Rightarrow X^*$$



변형된 Input matrix X^* 와 Y 를 **linear**한 모델로 학습

$$Y = \beta_0 + \beta_1 * X_1^* + \beta_2 * X_2^* + \dots + \beta_n * X_n^*$$

비선형 모델(Non-Linear Model)

비선형 모델의 학습과정



Input matrix에 해당되는 X 를 ✨ 비선형적으로 transform

✨ 기저함수

$$X \Rightarrow X^*$$

기저선대팀 클린업 참고 바람!



변형된 Input matrix X^* 와 Y 를 **linear**한 모델로 학습

$$Y = \beta_0 + \beta_1 * X_1^* + \beta_2 * X_2^* + \dots + \beta_n * X_n^*$$

Piecewise Polynomials

Piecewise Polynomial Model이란?

주어진 매듭점(Knot)으로 나누어진 영역에서
설명변수와 다항 함수로 회귀 함수를 추정하는 방법

Ex. Piecewise constant model, Piecewise linear model

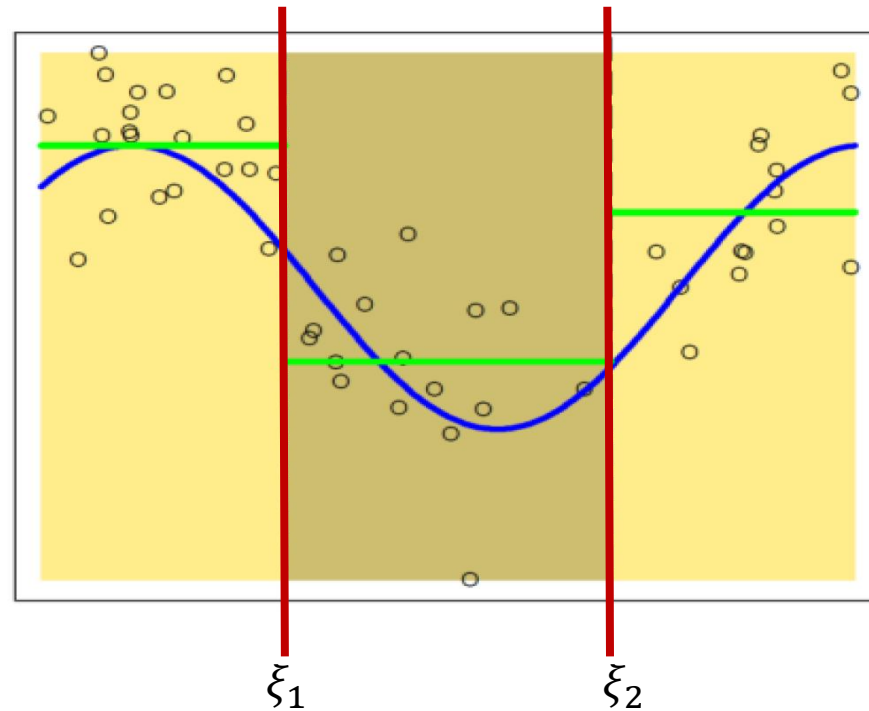
“ Knot ”

X값으로 구분된 관찰값들의 영역

ξ_n : nth knot

Piecewise Polynomials

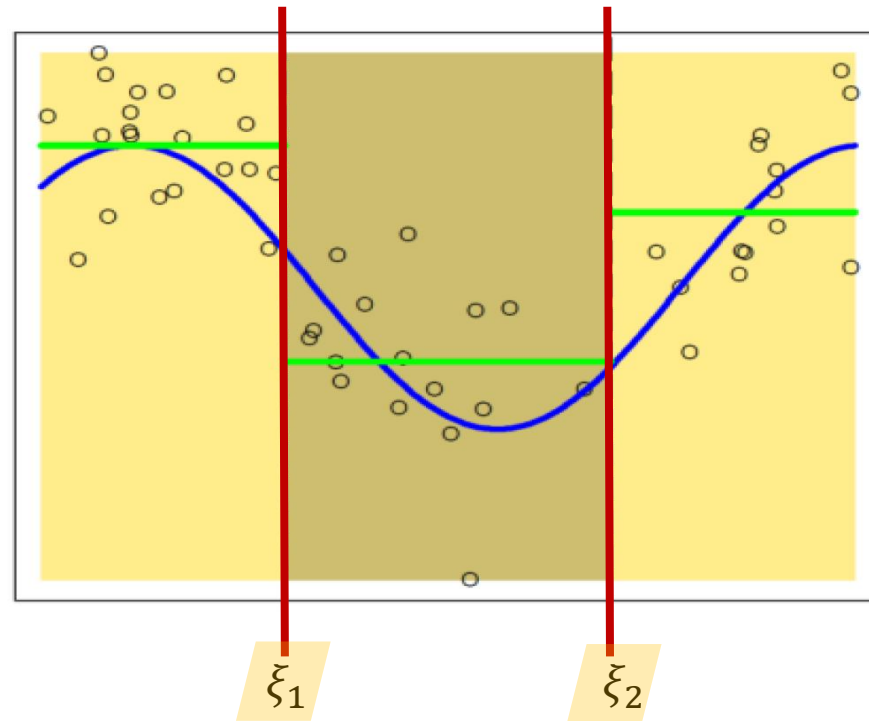
Piecewise Constant Model



knot로 나뉜 영역마다 서로 다른 모델을 적합시킬 때
예측값이 상수로 나타나는 모델

Piecewise Polynomials

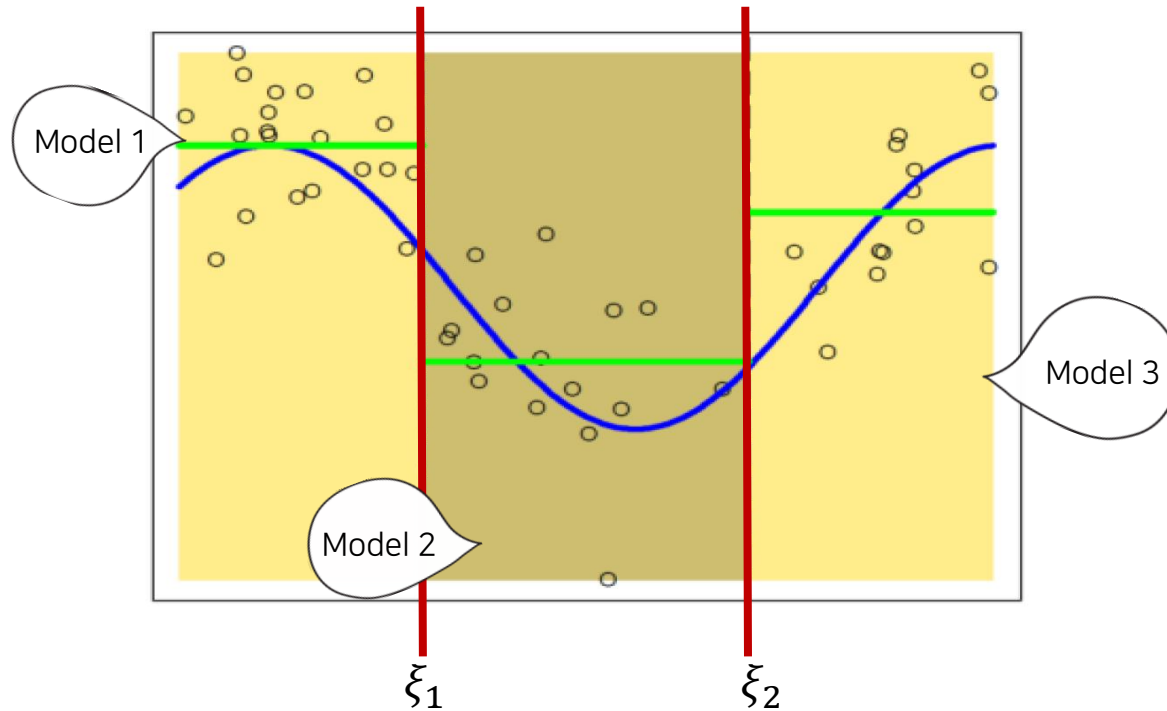
Piecewise Constant Model



주어진 Knot로 데이터 영역 분할

Piecewise Polynomials

Piecewise Constant Model

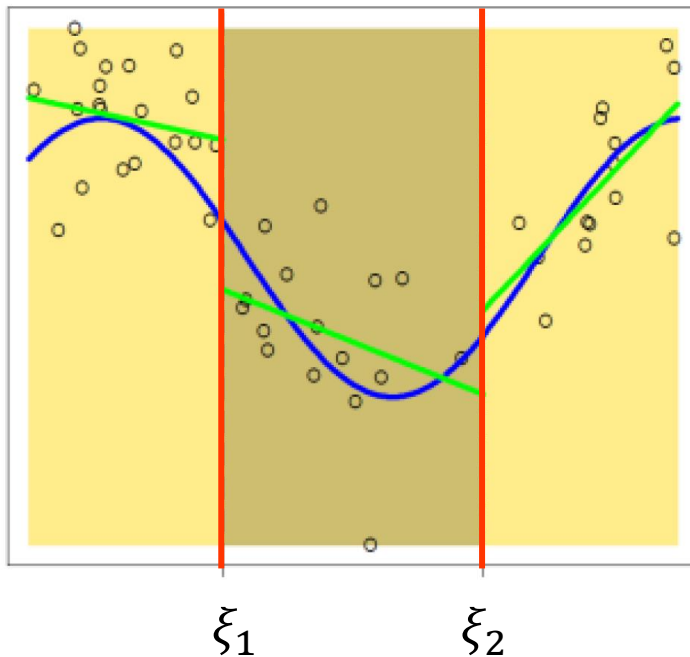


각각의 영역마다 서로 다른 모델을 적합

Piecewise Polynomials

Piecewise Linear Model

knot로 나뉜 영역마다 서로 다른 모델을 적합시킬 때
예측값이 선형으로 나타나는 모델



Region

$$x < \xi_1$$

$$\xi_1 < x < \xi_2$$

$$x > \xi_2$$

Model

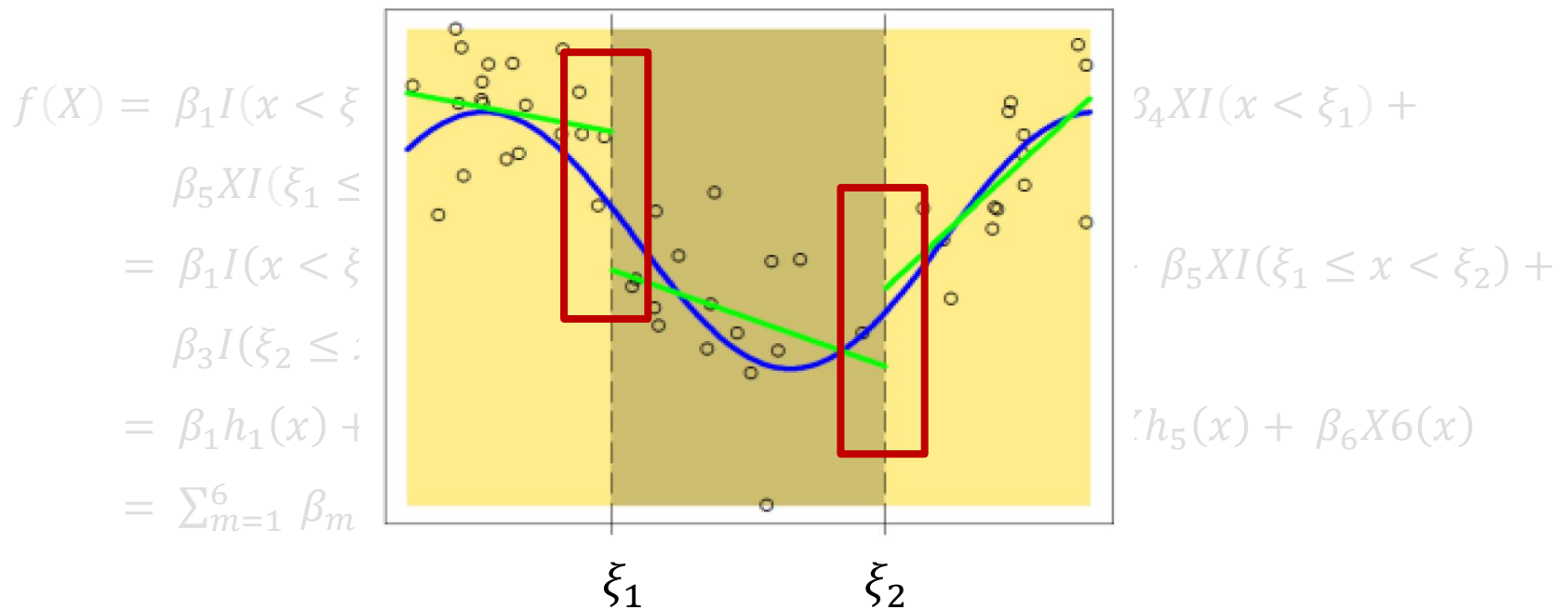
$$Y = \beta_1 + \beta_4 * X$$

$$Y = \beta_2 + \beta_5 * X$$

$$Y = \beta_3 + \beta_6 * X$$

Piecewise Polynomials

Piecewise Linear Model

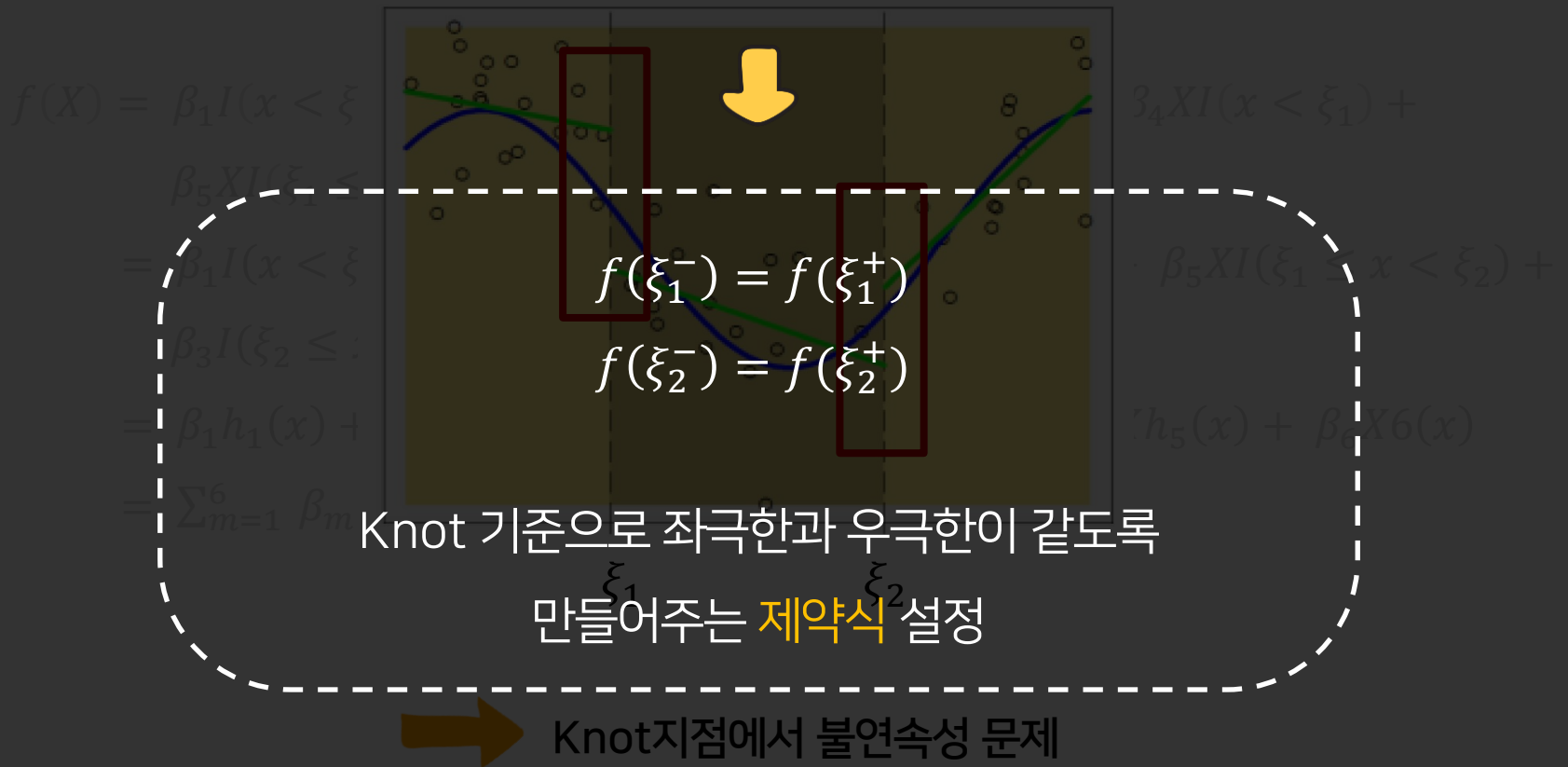


Knot지점에서 불연속성 문제

Piecewise Polynomials **WHY??**

Piecewise Linear Model

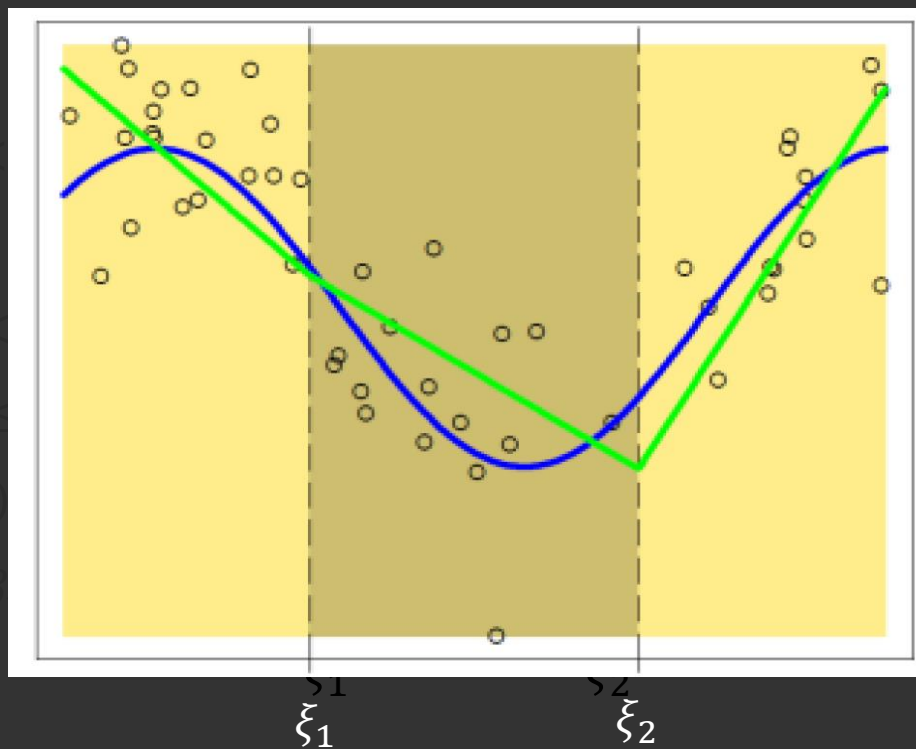
Knot 기준으로 좌극한 \neq 우극한이므로 불연속





Piecewise Polynomials

Piecewise Linear Model



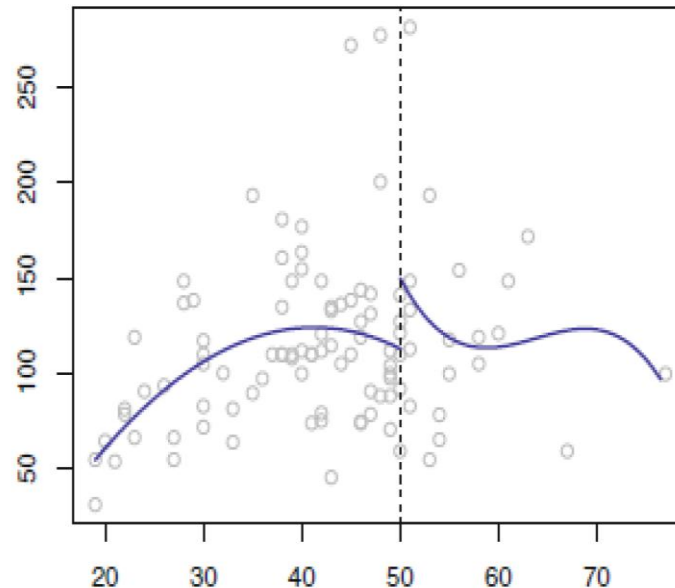
→ Knot지점에서 불연속성 문제
제약식에 의해 연속적인 형태

Cubic Spline

Cubic Spline이란?

Piecewise 모델의 일종으로, 3차 다항식을 적합시킨 모델

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$



Cubic Spline

Cubic Spline이란?



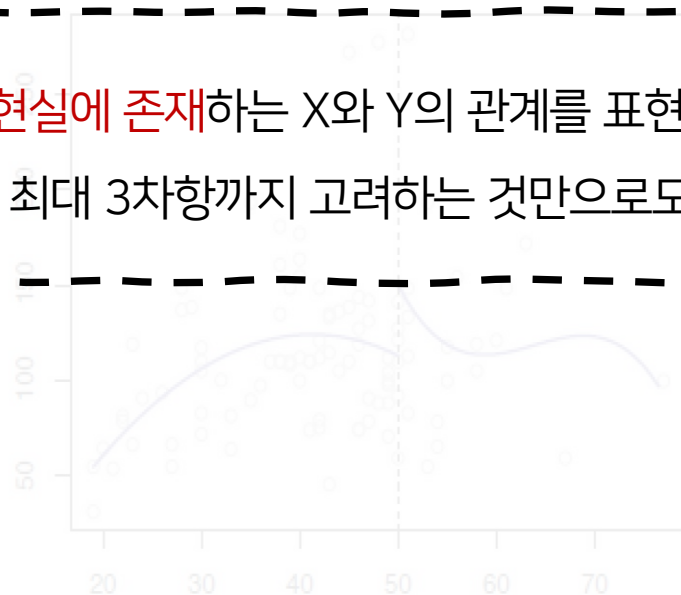
piecewise 모델의 일종으로, 3차 다항식을 적합시킨 모델

왜 3차 다항식일까?

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

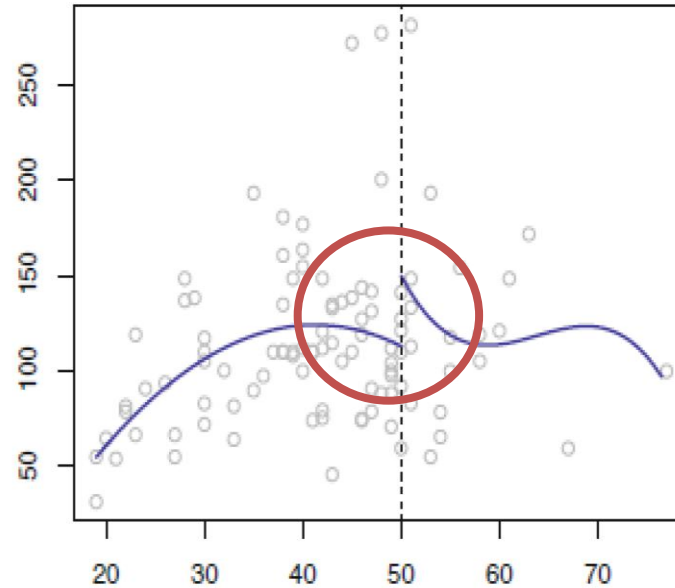
현실에 존재하는 X와 Y의 관계를 표현하기에

최대 3차항까지 고려하는 것만으로도 충분



Cubic Spline

Cubic Spline



Age=50라는 knot에서 선이 연결되지 않는 비합리적인 모델

“ 불연속성 문제 ”



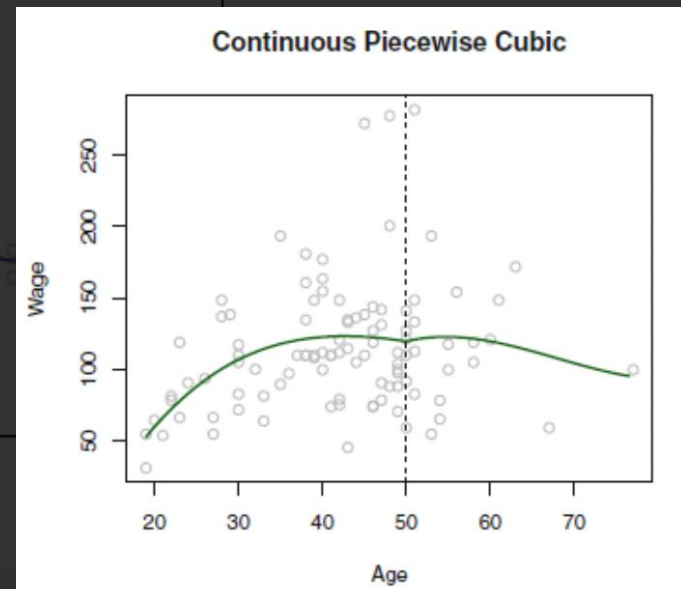
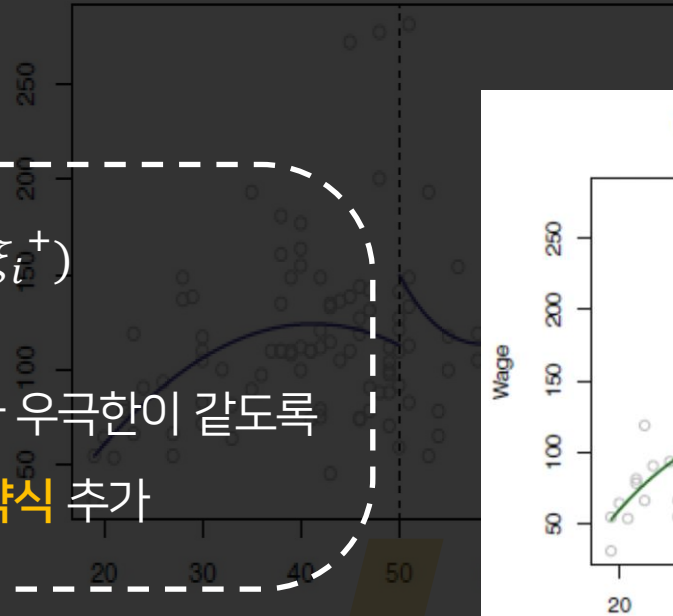
Cubic Spline

Cubic Spline

연속으로 만드는 방법은?

$$f(\xi_i^-) = f(\xi_i^+)$$

Knot 기준으로 좌극한과 우극한이 같도록
만들어주는 **제약식** 추가

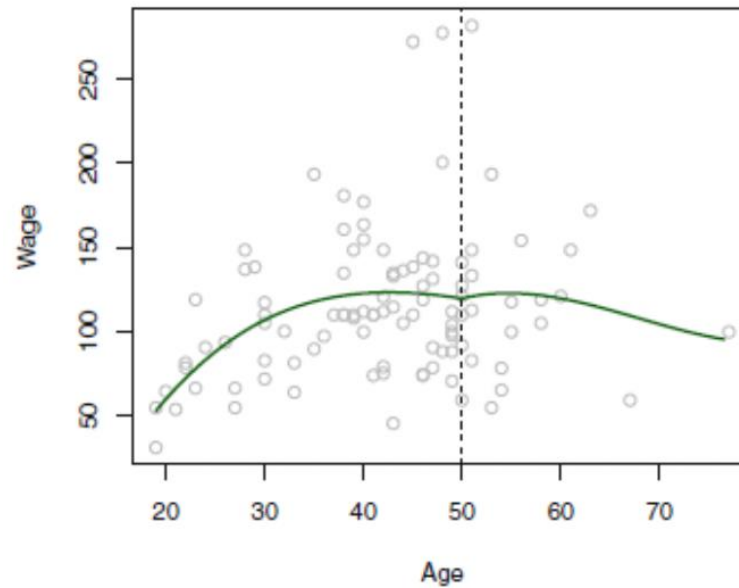


Age=50라는 knot에서 선이 **연결되지 않는** 비합리적인 모델

“적합된 곡선이 **Continuous**!”
Discontinuity 문제점

Cubic Spline

Cubic Spline



여전히 knot를 기준으로 **미분불가능**하다는 문제점

미분불가능 \rightarrow 좌미분계수 \neq 우미분계수

Cubic Spline

Cubic Spline



좌미분계수 = 우미분계수 제약식 추가

$$f'(\xi_i^-) = f'(\xi_i^+)$$

$$f''(\xi_i^-) = f''(\xi_i^+)$$

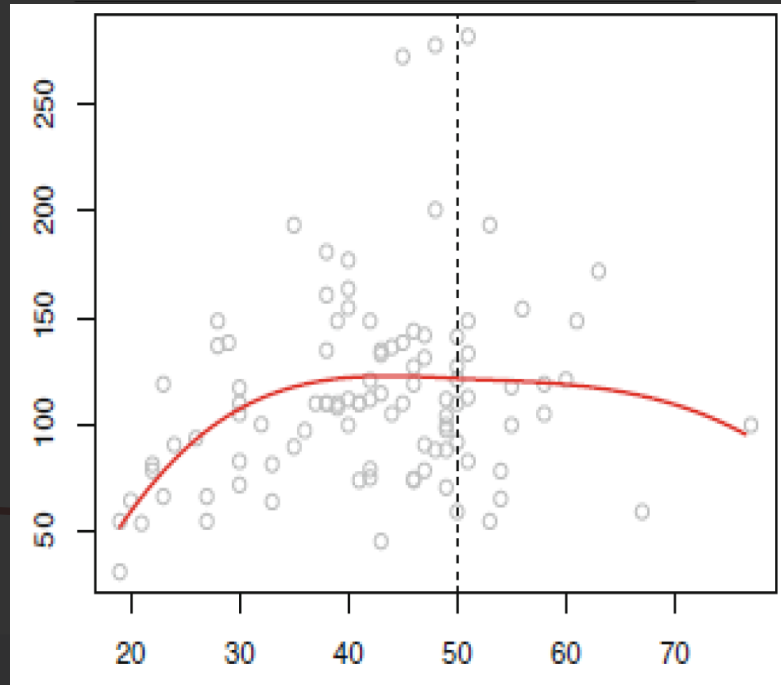
여전히 knot를 기준으로 미분불가능하다는 문제점

미분불가능 → 좌미분계수 \neq 우미분계수



Cubic Spline

Cubic Spline 왜 2계 미분이 연속이라는 조건까지 추가할까?



여전히 knot를 기준으로 미분불가능하다는 문제점

곡선의 곡률까지 연속으로 만들어

함수를 보다 smooth하게 만들기 위함



Cubic Spline

장점

✓ 데이터의 보간법(interpolation)

주어진 데이터의 **trend**를 반영해 새로운 데이터 지점 구성 가능

단점

✓ 차원의 저주(Curse of Dimensionality) 문제

설명변수가 3개 이상으로 늘어날 경우

Model의 **Variance**가 급격히 증가해 좋은 성능을 기대할 수 없음

GAM(Generalized Additive Model)

GAM이란?

기존의 선형 모델에서 **가법성은 유지**하면서도
각 변수에 **Non-Linear한 적합**을 가능하게 하는 방법

다중선형회귀의 확장으로 볼 수 있음 !



GAM(Generalized Additive Model)

GAM이란?

$$y = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon$$



$$y = \beta_0 + \sum_{j=1}^p f_j(x_{ij}) + \epsilon_i$$

$$y = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon$$

일반적인 다중선형회귀 모형을 각 선형결합 $\beta_j x_{ij}$ 가 아닌

smooth한 비선형함수 $f_j(x_{ij})$ 로 대체

GAM(Generalized Additive Model)

GAM이란?



가법적인 모형 (Additive Model)



$$y = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon$$

이때 여전히 각 변수 x_{ij} 을 각각의 함수 f_j 에 부과하고 있기에 **가법성**을 유지

각각의 Y에 대한 각각의 예측변수들의 기여를 '더하여' 이를 표현

GAM(Generalized Additive Model)

GAM의 장단점

장점

일반적인 선형 회귀모델이 놓칠 수 있는 **비선형적인 관계**를 반영
다른 변수들은 모두 고정하고서 Y에 대한 **X 각각의 영향**을 **개별적**으로 조사

단점

모델이 **가산적**이어야 한다는 제한
변수가 많은 경우 **중요한 상호작용**을 놓칠 수 있음

감사 감사



THANK YOU