차세대융합기술연구원
ADVANCED INSTITUTE OF CONVERGENCE TECHNOLOGY

Boosting Regression model 설명 보완

2021.08.19 여지민

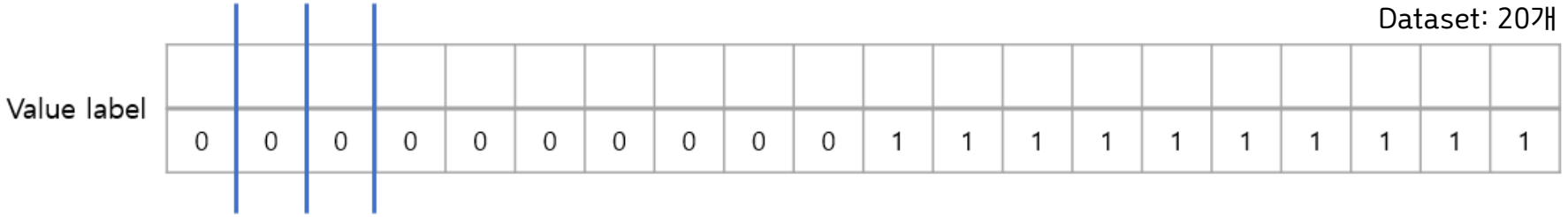
eXtreme Gradient Boosting(XGB)

근사적인 트리 학습(Approximate Tree Learning)

* 기존 Basic Exact Greedy Algorithm

: 모든 데이터 포인트를 기준으로 가능한 분할 열거하여 최적의 분할 탐색

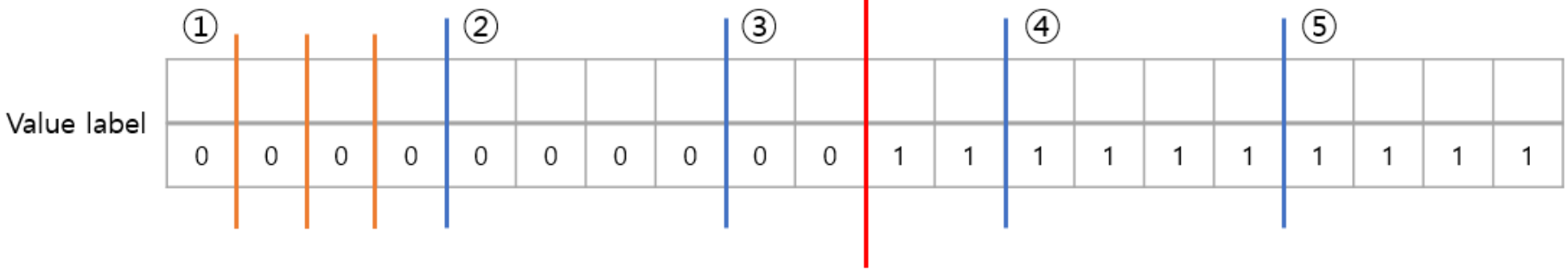
Dataset: 20개



- Approximate Algorithm

: 분위수를 기준으로 후보 분할 지점 제시 → 한 bucket 안에 있는 데이터를 기준으로 split 가정

Best split point → 얻는 정보량 최대



*각 split 지점에서 gradient 계산 횟수: EGA: 19회, AA: 3*5 =15회

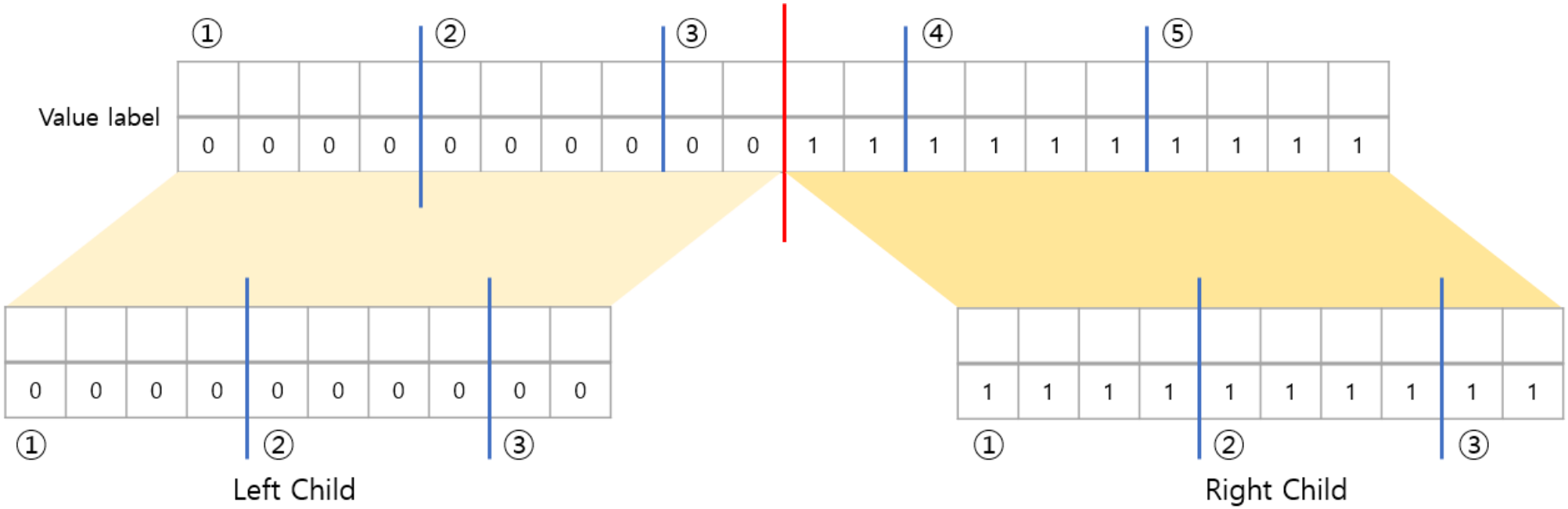
eXtreme Gradient Boosting(XGB)

근사적인 트리 학습(Approximate Tree Learning)

* candidate split point를 제안 시기에 따라 방법 2가지로 나뉨

- Global variant(per tree)

: 트리 구성 첫 단계에서 모든 후보 분할 제시, 모든 levels에서 동일한 후보 분할 제시



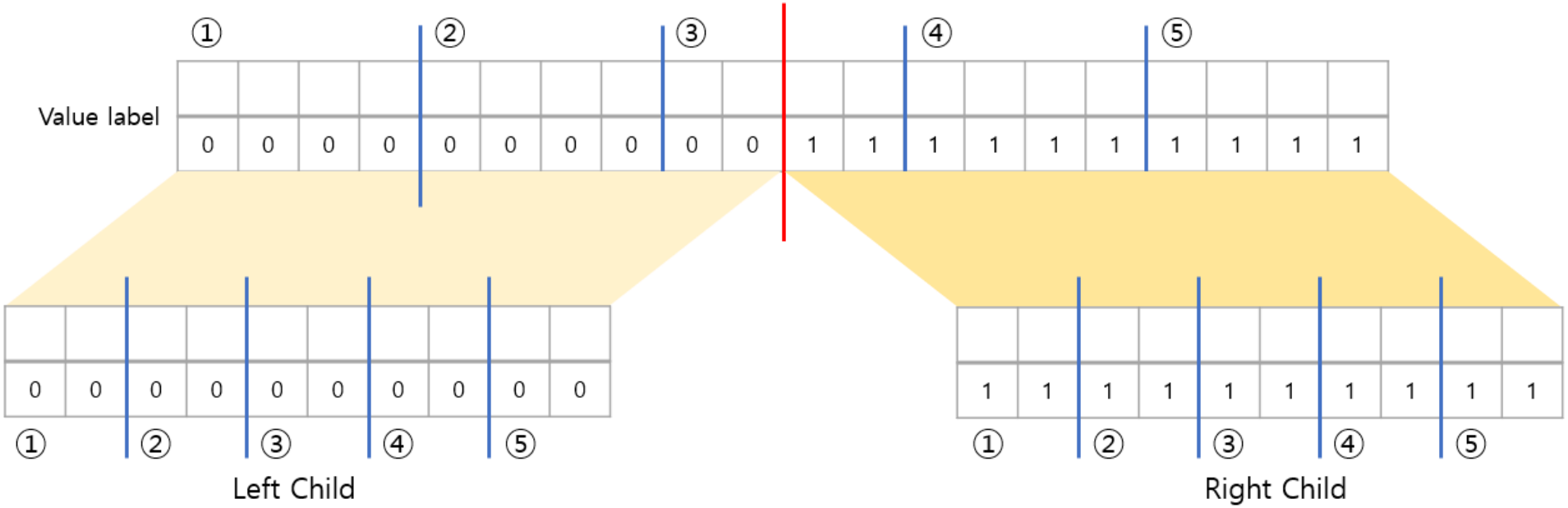
eXtreme Gradient Boosting(XGB)

근사적인 트리 학습(Approximate Tree Learning)

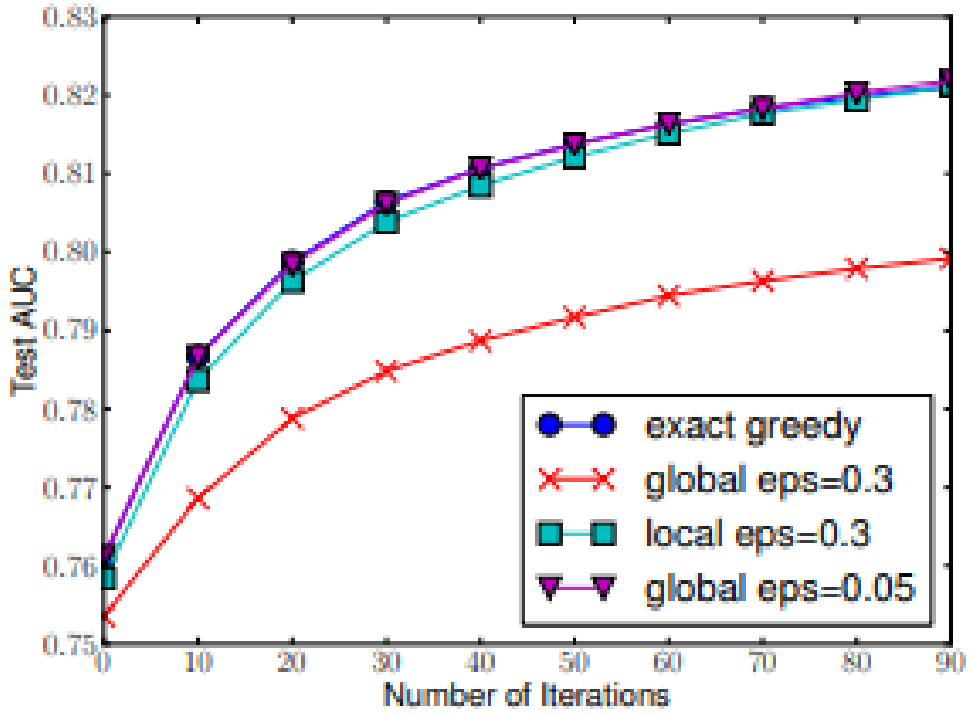
- Local variant(per split)

: 매번 분할 진행 후, 새롭게 후보 분할 지점 제시

percentile을 통해서 bucket 사이즈 유지



eXtreme Gradient Boosting(XGB)



- eps: percentile을 얼마나 잘게 나눌지에 대한 파라미터

  candidate split point 개수: 1/eps

→ 10/3=3.333 100/5=20

eXtreme Gradient Boosting(XGB)

희소성 인식 알고리즘(Sparsity-Aware Algorithm)

- 희소성 있는 데이터 효율적으로 처리

- 결측값
- 빈번한 0 항목
- One-Hot encoding같은 Feature Engineering

Value		1.1	0.2	0.8		1.3	1.5	0.5		1
Class	0	1	0	1	0	1	1	0	0	1.9

Best split point

Value	0.2	0.5	0.8	1.1	1.3	1.5	1.9			
Class	0	0	1	1	1	1	1	0	0	0

Best split point

Value				0.2	0.5	0.8	1.1	1.3	1.5	1.9
Class	0	0	0	0	0	1	1	1	1	1

→ 새로운 결측치 들어왔을 때, 해당 데이터 왼쪽으로 보냄

LightGBM(LGBM)

GOSS(Gradient-based One-Side Sampling) - 인스턴스 수 줄이기

- 더 큰 Gradient를 가진 인스턴스 유지, 나머지 작은 Gradient를 사용하는 인스턴스 무작위 샘플링

→ 잘못 예측된 인스턴스: 큰 가중치 부여 / 제대로 예측된 인스턴스: 작은 가중치 부여

*Gradient 기준: 미리 임계값이나, 상위 백분위수로 정의

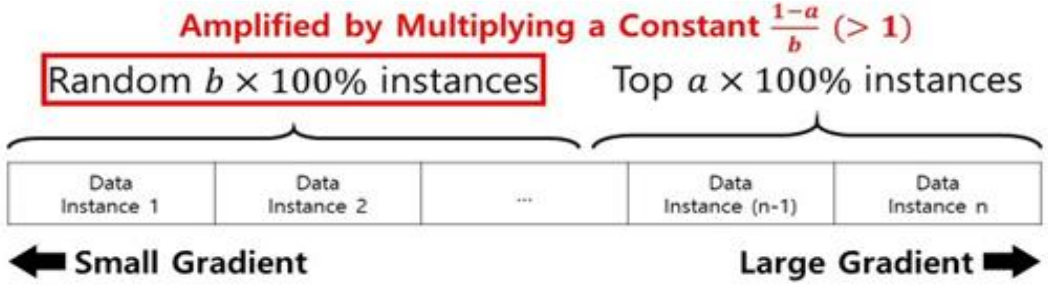
Algorithm 2: Gradient-based One-Side Sampling

Input: I: training data, d: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: $loss$: loss function, L: weak learner
models ← {}, fact ← $\frac{1-a}{b}$
topN ← a × len(I) , randN ← b × len(I)
for $i = 1$ ***to*** d **do**
  preds ← models.predict(I)
  g ← $loss$(I, preds), w ← {1,1,...}
  sorted ← GetSortedIndices(abs(g))
  topSet ← sorted[1:topN]
  randSet ← RandomPick(sorted[topN:len(I)], randN)
  usedSet ← topSet + randSet
  w[randSet] × = fact ▷ Assign weight $fact$ to the small gradient data.
  newModel ← L(I[usedSet], − g[usedSet], w[usedSet])
  models.append(newModel)

1. 인스턴스 절대값에 따라 정렬
2. Gradient 상위 a×100%개의 인스턴스 선택 (모두 사용)
3. 나머지 데이터에서 b×100%개의 인스턴스 무작위 샘플링
4. gradient가 작은 데이터 셋에 대해서는 Information Gain을 계산할 때,

정보이득에 상수 $\frac{1-a}{b}$를 곱하여 가중치 업데이트

* $\frac{1-a}{b}$는 1보다 크면 효과 극명



LightGBM(LGBM)

EFB(Exclusive Feature Bundling) - Feature개수 줄이기

- 변수 값 0에 대한 불필요한 계산 피할 수 있음

  상호배타적인 변수를 bundle로 제공하여 변수 개수 ↓

* 상호배타적인 변수: 동시에 0이 아닌 값을 가지는 관계의 변수

Graph-coloring 문제로 표현 가능

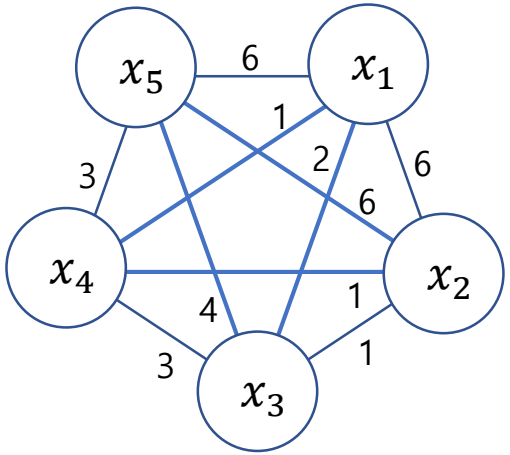
- 점: Feature
- 모서리: 두 Feature 사이의 conflict

conflict: 동시에 non-zero 값을 갖는 객체의 수

	x_1	x_2	x_3	x_4	x_5
l_1	1	1	0	0	1
l_2	0	0	1	1	1
l_3	1	2	0	0	2
l_4	0	0	2	3	1
l_5	2	1	0	0	3
l_6	3	3	0	0	1
l_7	0	0	3	0	2
l_8	1	2	3	4	3
l_9	1	0	1	0	0
l_{10}	2	3	0	0	2

	x_1	x_2	x_3	x_4	x_5
x_1	-	6	2	1	6
x_2	6	-	1	1	6
x_3	2	1	-	3	4
x_4	1	1	3	-	3
x_5	6	6	4	3	-

두 변수사이의 conflict 정도



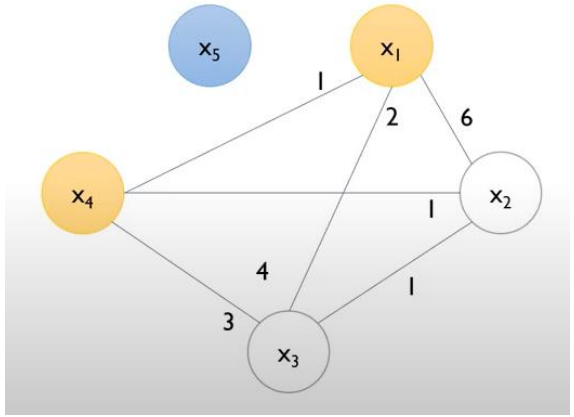
두 변수사이의 conflict 정도를 선으로 표현

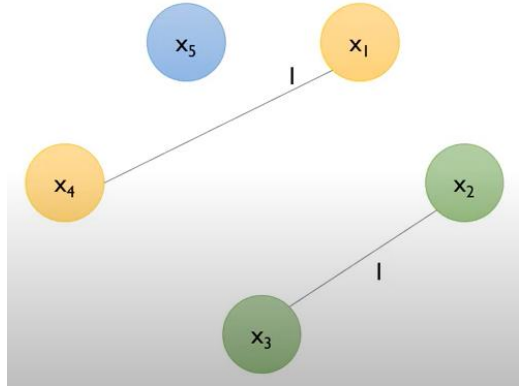
LightGBM(LGBM)

변수 별 degree 기준으로 bundling

	x_5	x_1	x_2	x_3	x_4
d	19	15	14	10	8

Max conflict count: 2





$\{x_5\},\ \{x_1, x_4\},\ \{x_2, x_3\}$

Categorical Boosting Machine(CATB)

Background : Gradient Boosting

example

	Height (m)	Favorite Color	Gender	Weight (kg)	Residual
x1	1.6	Blue	Male	88	16.8
x2	1.6	Green	Female	76	4.8
x3	1.5	Blue	Female	56	-15.2
x4	1.8	Red	Male	73	1.8
x5	1.5	Green	Male	77	5.8
x6	1.4	Blue	Female	57	-14.2

- 키, 좋아하는 색깔, 성별을 기반으로 몸무게를 예측하는 Gradient Boost 모델 생성
- Gradient Boost는 single leaf부터 시작

  ex) 실제 몸무게 값 : 88kg
  Leaf 초기 예측값 : 71.2kg
  (88 + 76 + 56 + 73 + 77 + 57) / 6 = 71.2

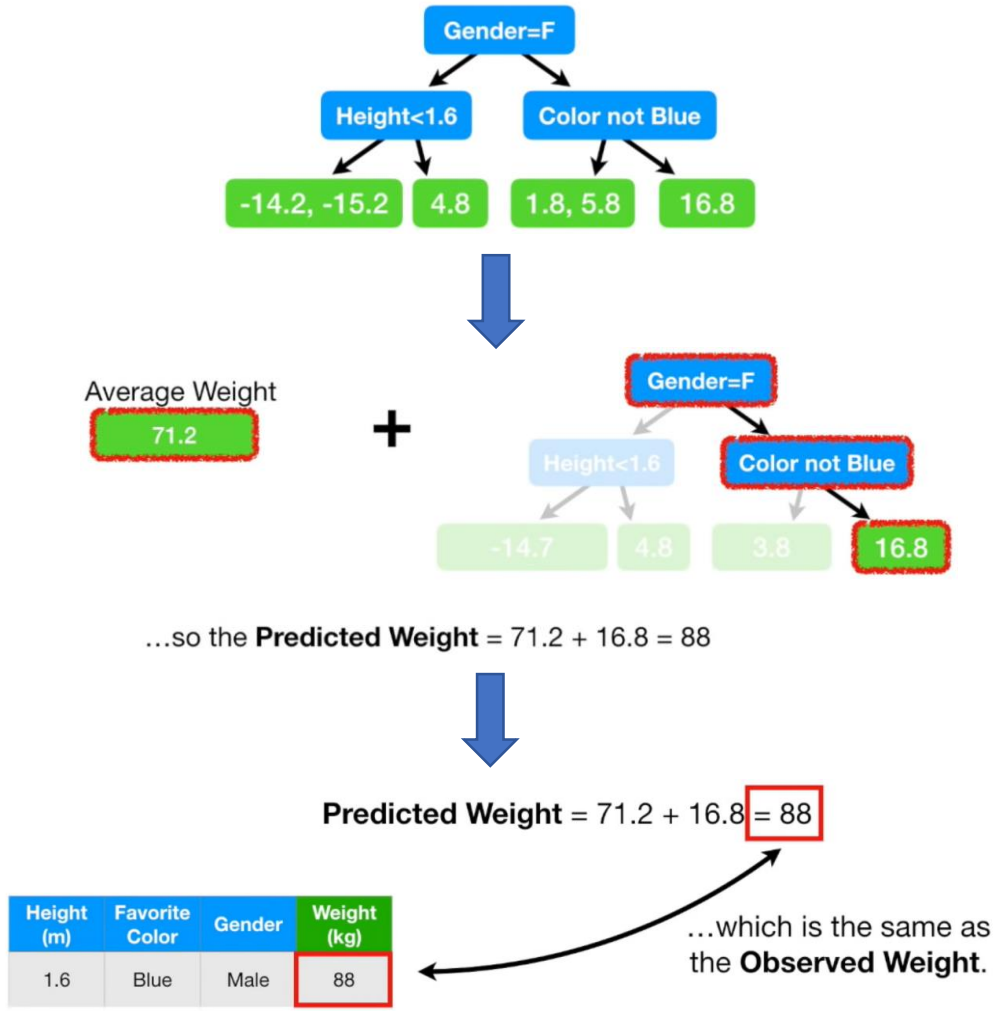
  따라서 x1의 잔차(Residual)는 16.8kg

Categorical Boosting Machine(CATB)

Background : Gradient Boosting

- 키, 좋아하는 색깔, 성별을 통해 Residual을 예측하는 트리 모델 생성
- 남자이며 좋아하는 색상이 파란색이 아닌 사람이라면 16.8이라는 예측값 부여

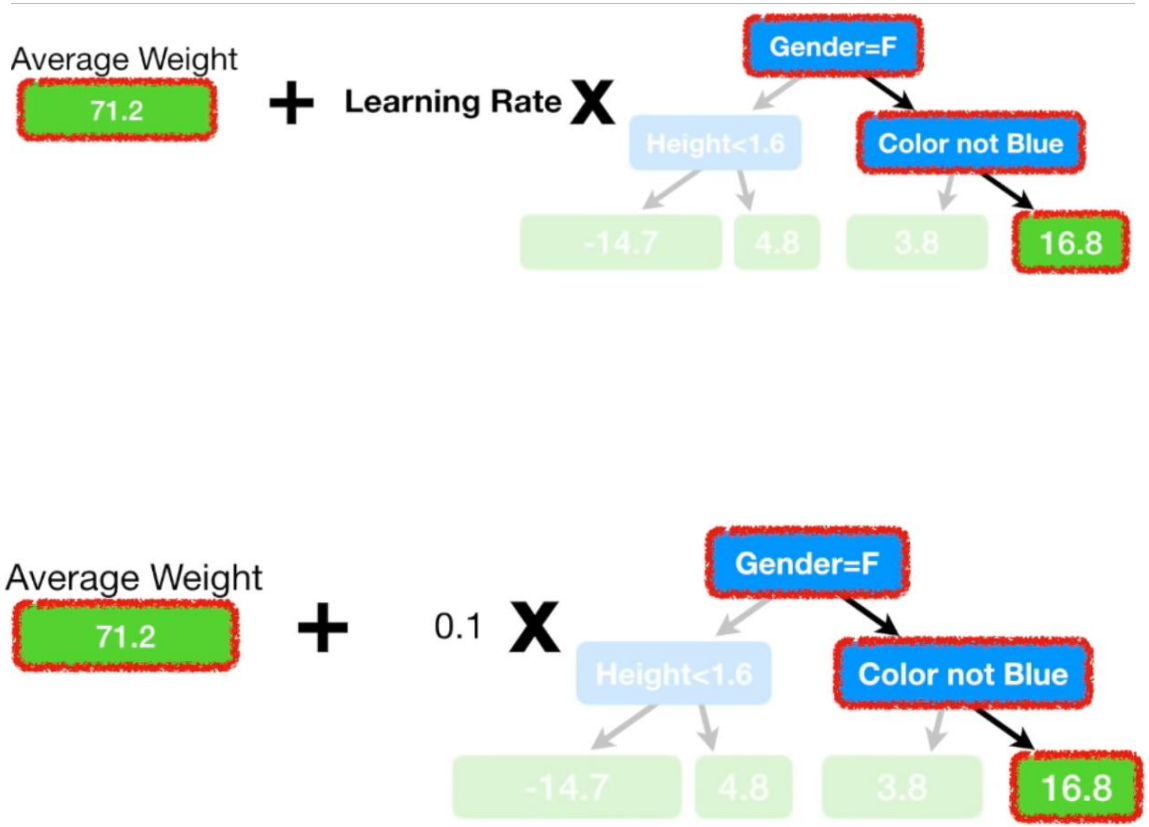
  71.2 + 16.8 = 88
  따라서 남자이며 좋아하는 색상이 파란색이 아닌 사람은 88kg으로 예측



Categorical Boosting Machine(CATB)

Background : Gradient Boosting

- 해당 모델은 현재 데이터에 과적합
- 학습률(Learning Rate) 활용
- 학습률을 곱해줌으로써 과적합을 해결
- 71.2 + (학습률(0.1) x 16.8) = 72.9
- 두 번째 모델에서 x1의 잔차: 88 - 72.9 = 15.1
  예측값: 71.2+(0.1*16.8)+(0.1*15.1) = 74.4
- 앞서 과정을 반복하여 일반화 성능 향상

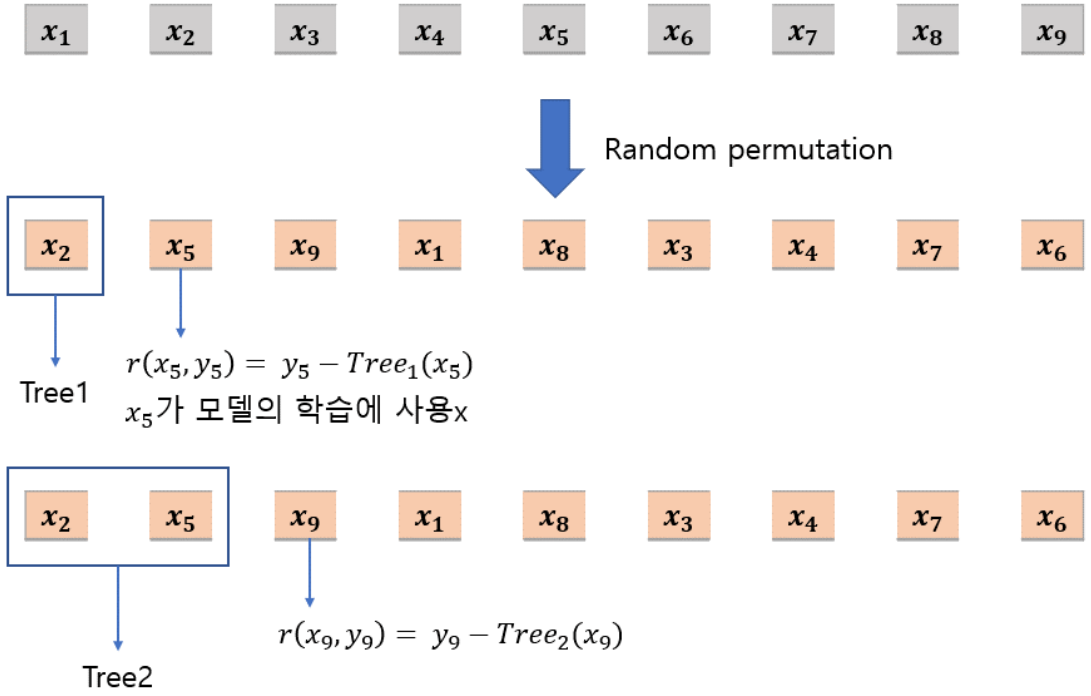


Categorical Boosting Machine(CATB)

Ordered Boosting

기존 부스팅 모델: 모든 훈련 데이터를 대상으로 Residual 계산
→ 각 부스팅 단계에서 학습에 사용하는 데이터셋은 독립적이어야함

Catboost: 일부만으로 잔차 계산을 한 뒤 모델을 만들고, 그 뒤에 데이터의 잔차는 이 모델로 예측한 값 사용



Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

일반적인 이미지 분류 : 각 입력 특성(픽셀)의 가중치로 이미지 분류

TCAV(Testing with Concept Activation Vectors)

: 예측 클래스에 대한 상위 개념(ex. 색, 성별, 인종)을 인식하여 이미지 분류

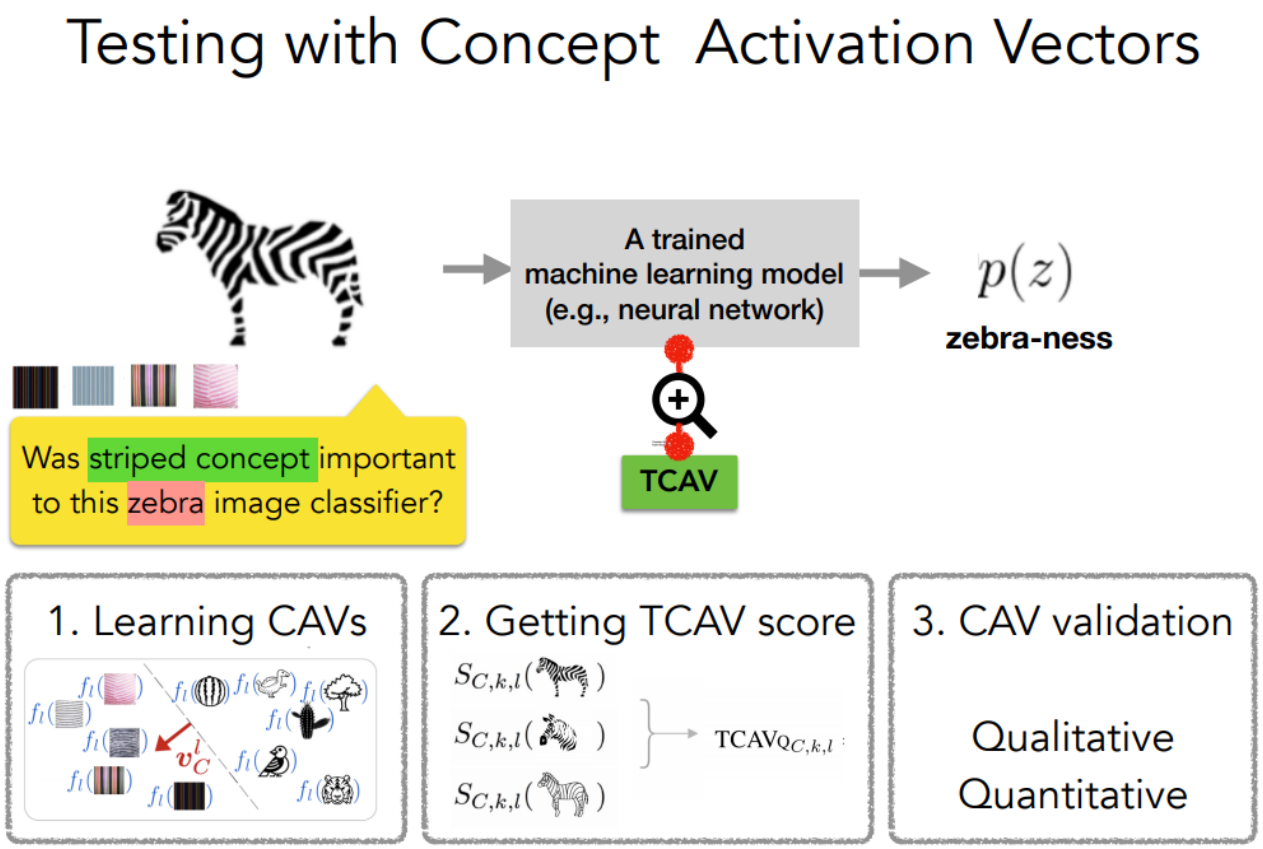
→ 관심 클래스에 대한 일반적인 사실을 설명으로 제공

변수의 개수가 많을수록 정보를 이해하기 어려워 사람이 의사소통하는 방식으로 설명 제공

사용자가 정의하는 높은 수준의 concept이 분류 결과에 중요한지 정량화

→ ex) 얼룩말 줄무늬를 얼마나 sensitive하게 분류할 수 있나 본 논문에서 수행

여름학교 1일차



* CAV(Concept Activation Vectors)

: 인간 관점에서 신경망 내부 상태를 해석

→ 컨셉과 랜덤 이미지를 주고, 특정 패턴과 이미지를 확실히 분리할 수 있는 vector