KUBIG
Data Science and Machine Learning

Week 7. Model Combining

Logistic Regression, Lasso, Ridge, Elastic Net,
Naive Bayes, QDA, LDA,
Support Vector Machine,
Decision Tree, Random Forest,
AdaBoost, GBM, LGBM, XgBoost, CatBoost...
and more?

모형 결합(Model Combining)

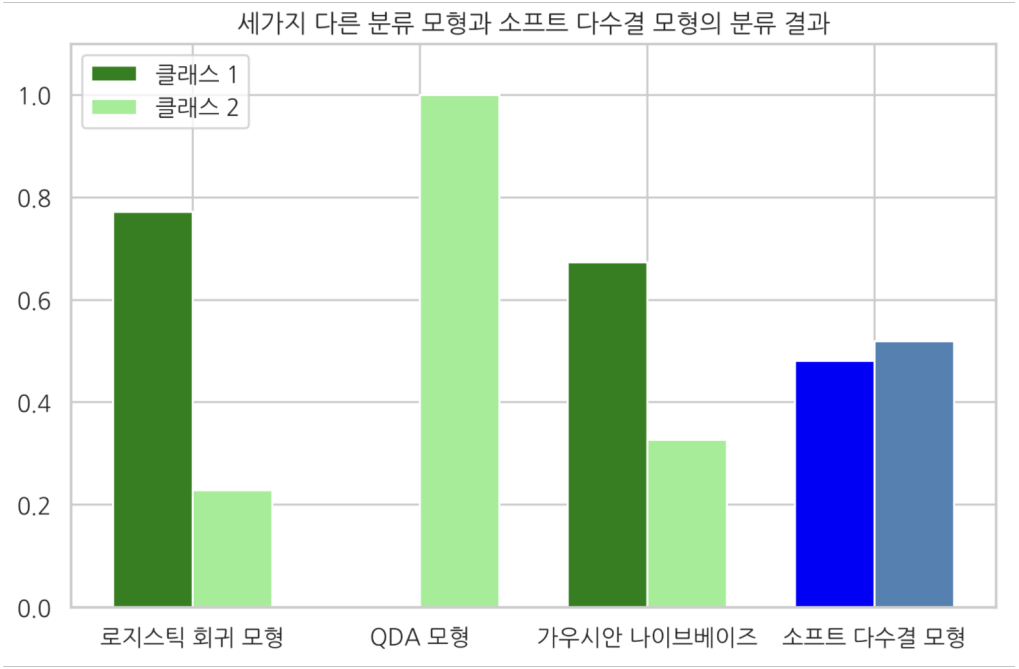
1. **취합 방법론(Aggregation)**
- 사용할 모형의 집합 고정
- 다수결 투표(Majority Voting)
- 배깅(Bagging)
- 랜덤 포레스트(Random Forest)
- 사용하는 독립적인 모형의 수가 늘어날 수록 성능이 증가

2. **부스팅(Boosting)**
- Weak learner를 순차적으로 학습시킴
- 에이다부스트(AdaBoost)
- 그레디언트 부스트(Gradient Boost)
  - XGBoost
  - LGBM
  - CatBoost

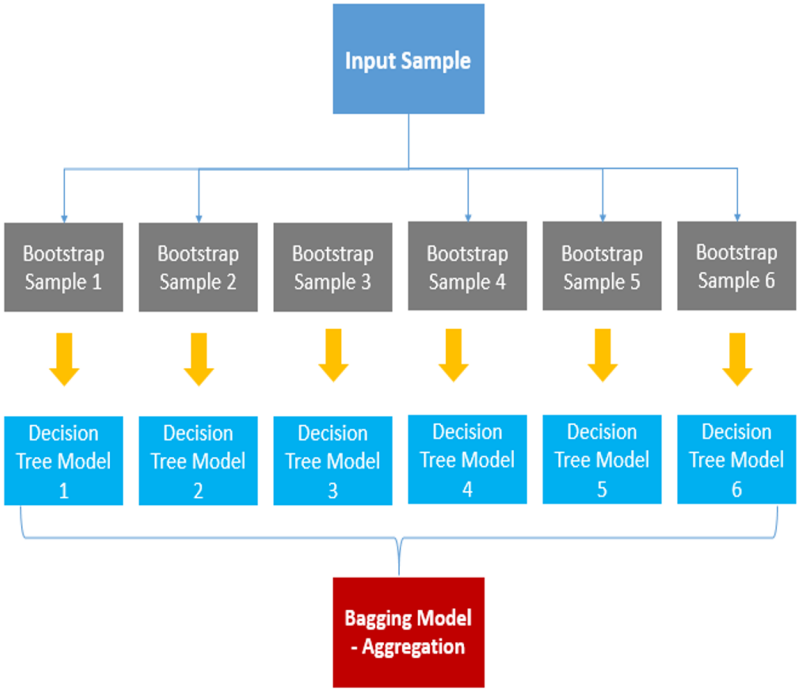
다수결 투표(Majority Voting)

1. **Hard Voting**
- 단순 투표
- 개별 모형의 결과 기준

2. **Soft Voting**
- 가중치 투표
- 개별 모형의 조건부 확률값 기준



배깅(Bagging): Bootstrap Aggregation



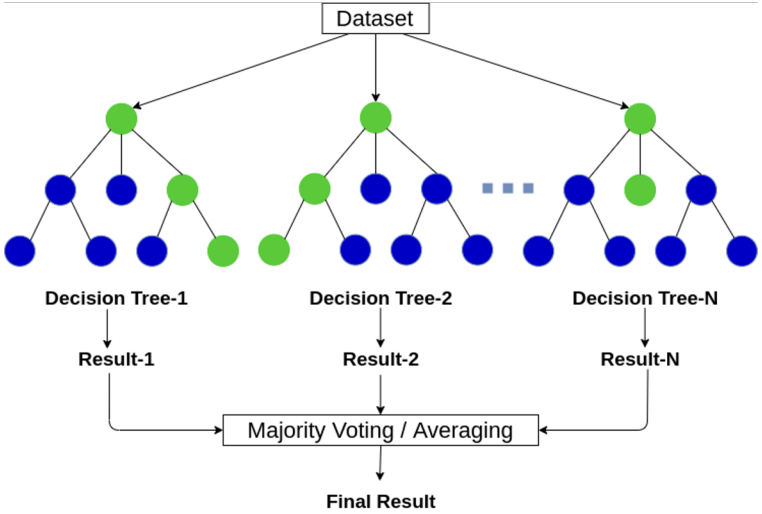
1. **Bootstrap**

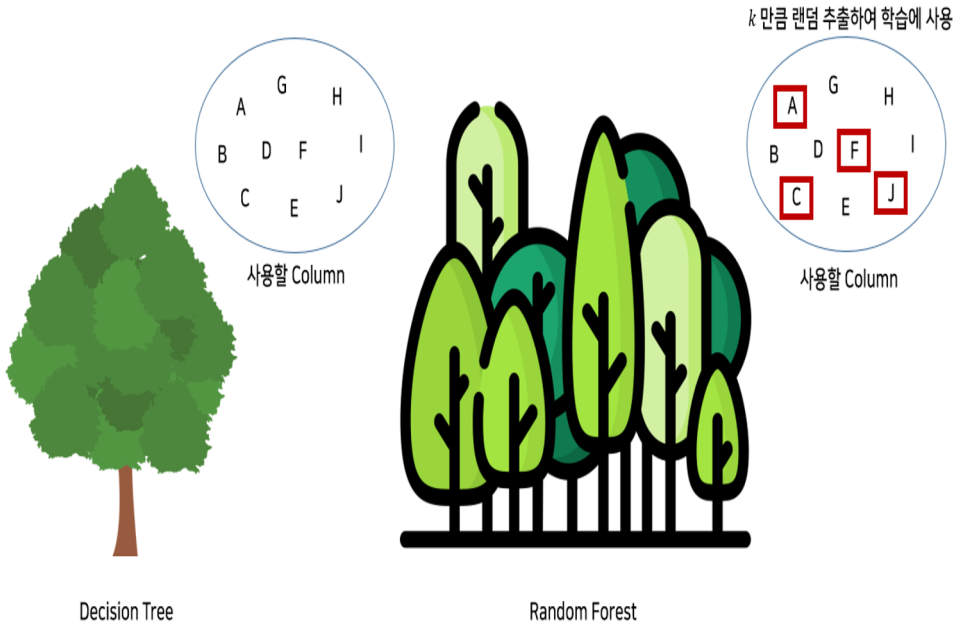
- Input Sample에서 복원추출 진행
  - bootstrap: 데이터 중복 사용 여부
  - max_samples: 선택할 샘플의 수/비율
  - bootstrap_features: 독립 변수의 중복 사용 여부
  - max_features: 다차원 독립 변수 중 선택할 차원의 수/비율
- 각 Bootstrap Sample에 대해 모델 학습

2. **Aggregation**

- Categorical Data는 투표 방식
- Continuous Data는 평균 방식

랜덤 포레스트(Random Forest)





부스팅(Boosting)

알고리즘	특징	비고
AdaBoost	Weak learner들이 상호 보완하며 순차적으로 학습	2003년
GBM	손실 함수의 Gradient를 통해 오답에 가중치 부여	2001년
XGBoost	Gradient Boosting 알고리즘을 분산환경에서 처리할 수 있게 구현	2014년
LGBM	빠른 처리 속도	2016년
CatBoost	범주형 데이터 처리에 탁월	2017년

AdaBoost(Adaptive Boosting)

Algorithm 4: AdaBoost algorithm

Input: Data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$;

Base learning algorithm $\mathcal{L}$;

Number of learning rounds T.

Process:

$D_1(i) = 1/m$. % Initialize the weight distribution

for $t = 1, \cdots, T$:

$h_t = \mathcal{L}(\mathcal{D}, D_t)$; % Train a weak learner h_t from $\mathcal{D}$ using distribution D_t

$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i)]$; % Measure the error of h_t

$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of h_t

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$ % Update the distribution, where Z_t is

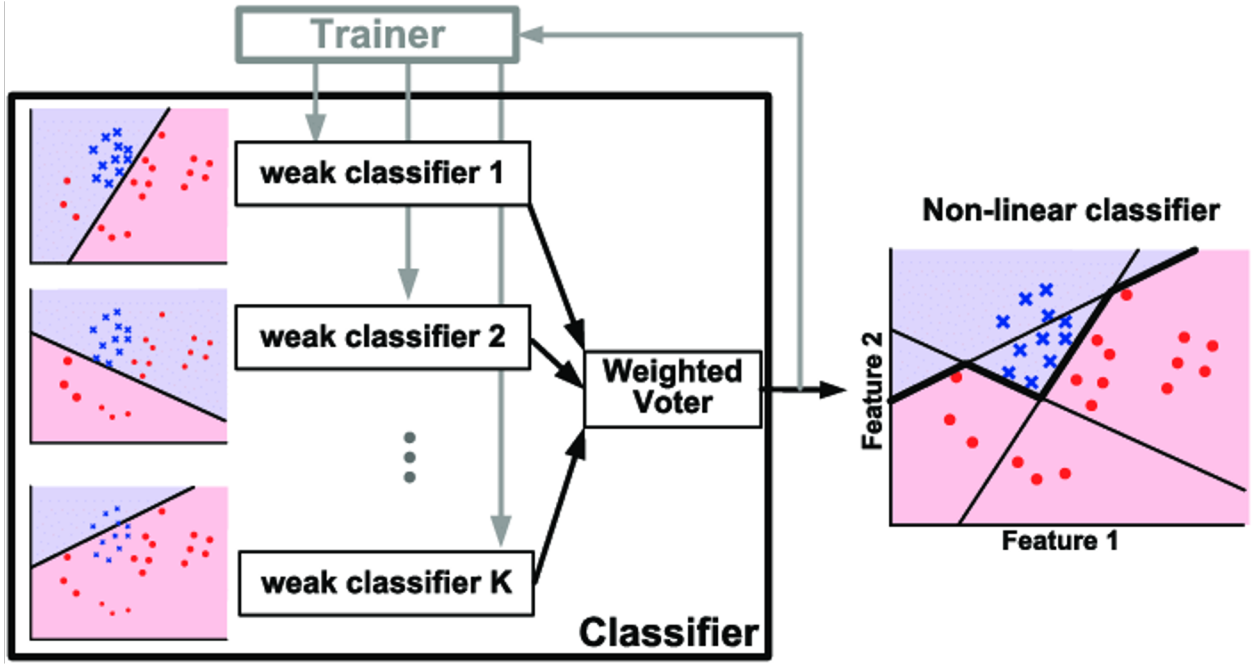
% a normalization factor which enables D_{t+1} be a distribution

end.

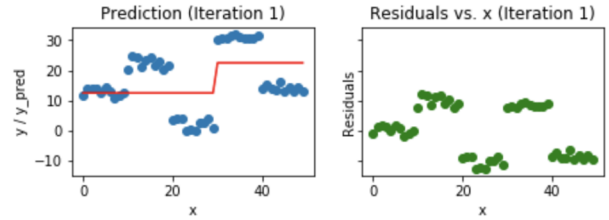
Output: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

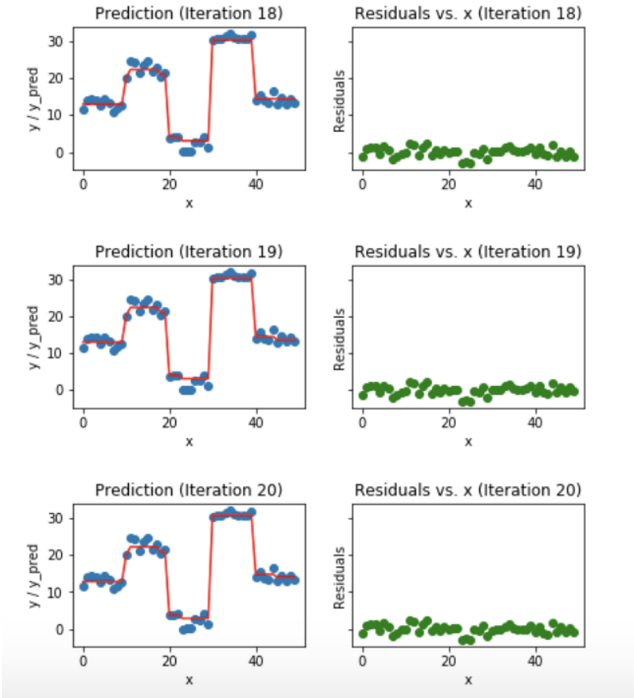
1. **데이터셋에서 샘플 i가 선택될 확률 초기화**
2. **한 번의 분류만을 사용하는 Stump Tree를 사용해 모델 훈련**
3. **훈련한 모델의 오분류 비율을 계산**
4. **모델의 신뢰도 계산**
   a. 오분류 확률이 0.5에 가까울 때, 신뢰도 ≈ 0
   b. 오분류 확률이 0에 가까울 때, 신뢰도 ↑
5. **샘플링 확률 업데이트**
   a. 오답이면 exp안 지수 양수 → 확률 증가
   b. 정답이면 exp안 지수 음수 → 확률 감소

AdaBoost(Adaptive Boosting)



GBM(Gradient Boosting Algorithm)





GBM(Gradient Boosting Algorithm)

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

$$minL = \frac{1}{2}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

$$\frac{\partial L}{\partial f(x_i)} = f(x_i) - y_i$$

$$y_i - f(x_i) = -\frac{\partial L}{\partial f(x_i)}$$

The end

고생 많으셨습니다 :)