

# 머신러닝(Machine Learning)

앙상블 기법 - GradientBoosting 모델

# 목 차

01 앙상블 기법

02 앙상블 기법 - 랜덤 포레스트

03 앙상블 기법 - Gradient Boosting 기법

04 앙상블 기법 - LightGBM

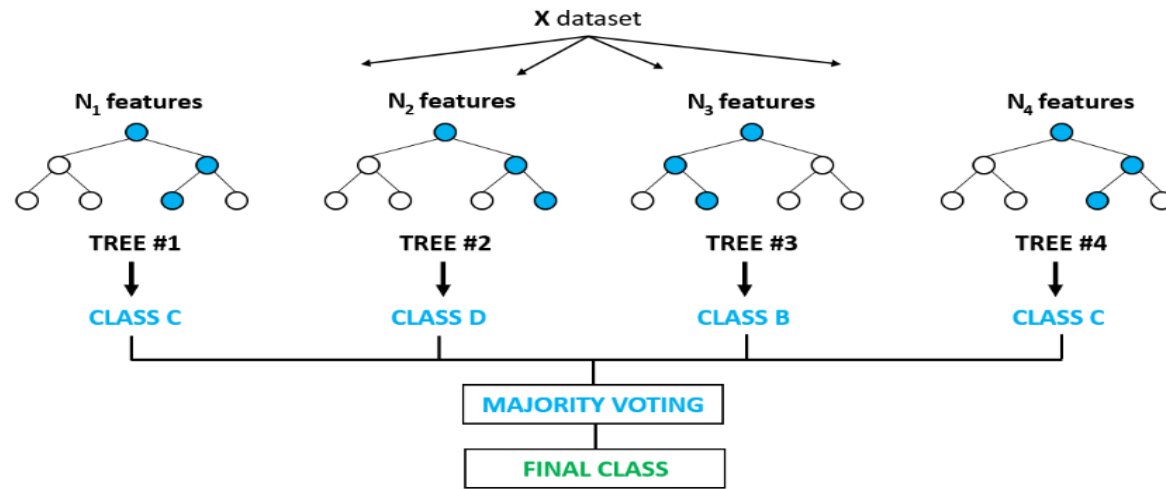
05 여러가지 모델

## 02 앙상블 기법-랜덤 포레스트

- ▶ 결정 트리의 주요 단점 - 훈련 데이터에 **과대 적합**되는 경향이 있음.

A. 랜덤 포레스트는 이 문제를 회피할 수 있는 방법.

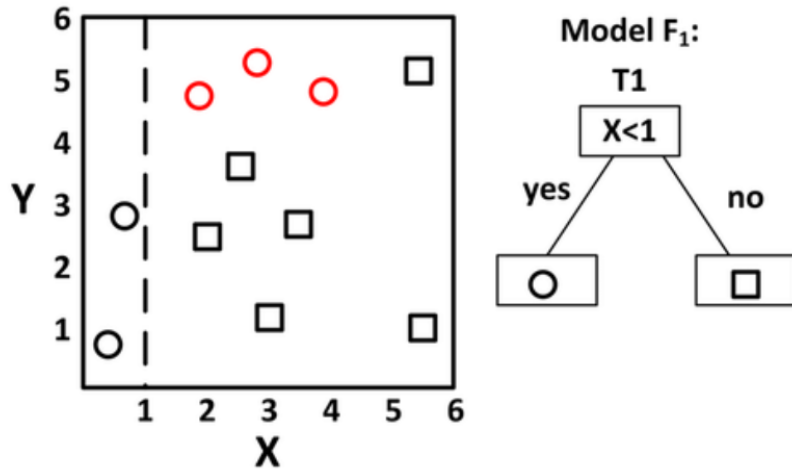
- ▶ 아이디어 : 조금씩 다른 여러 결정 트리의 묶음.



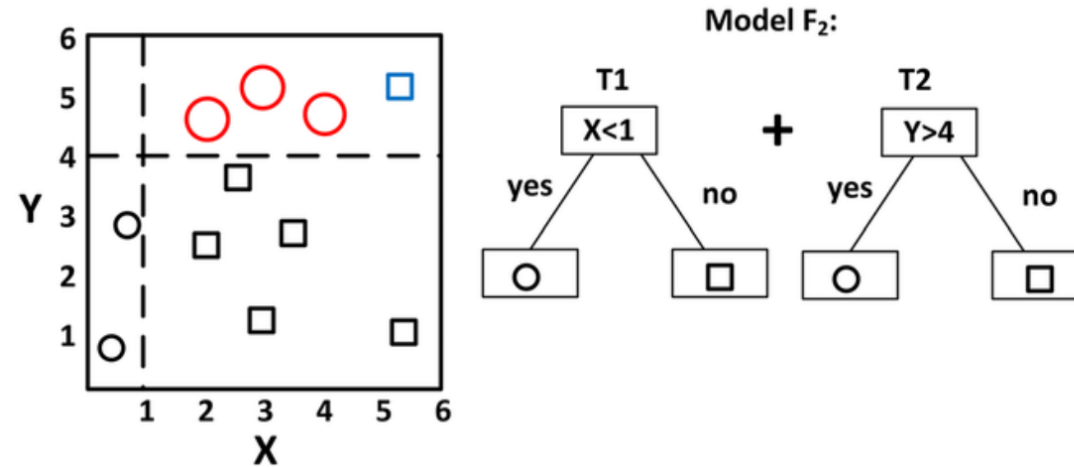
하나 하나의 트리는 데이터의 일부에 과대적합을 하는 경향이 있다.

# 03 앙상블 기법 - Gradient Boosting 기법

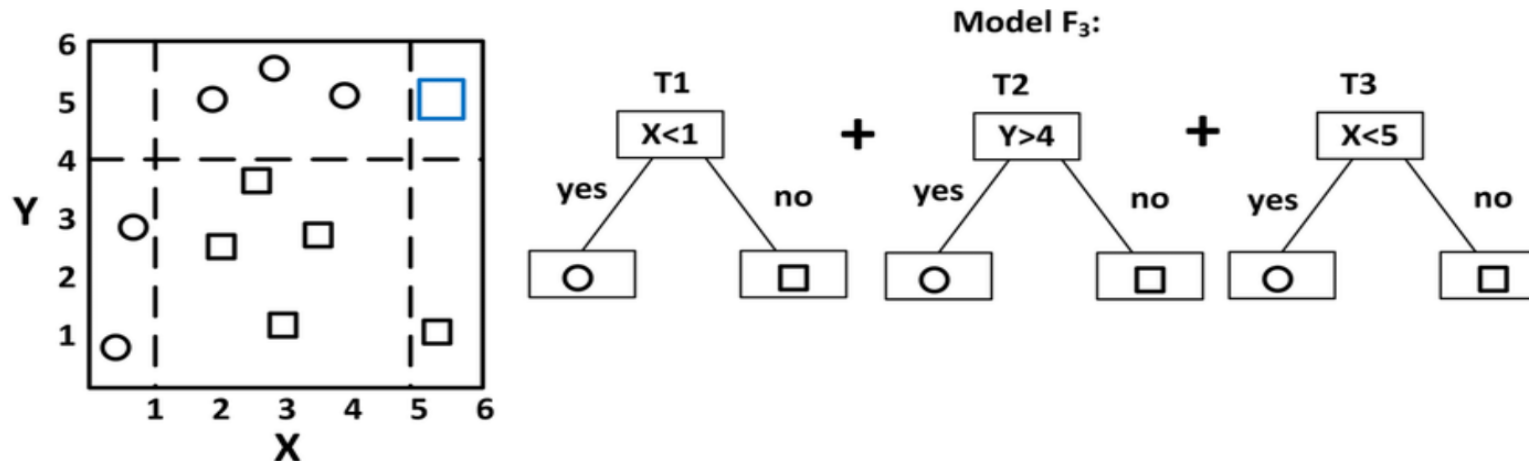
Iteration 1



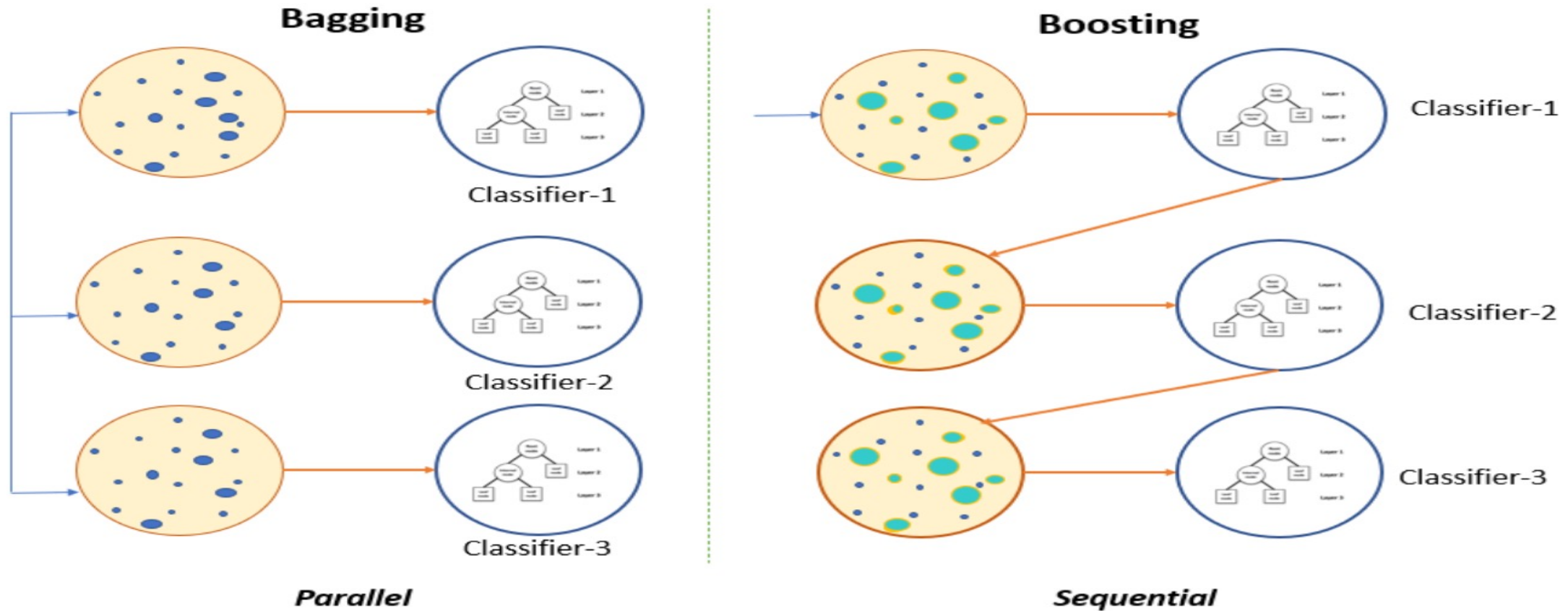
Iteration 2



Iteration 3



# 03 앙상블 기법 - Gradient Boosting 기법



Source: [https://pluralsight2.imgix.net/guides/81232a78-2e99-4ccc-ba8e-8cd873625fdf\\_2.jpg](https://pluralsight2.imgix.net/guides/81232a78-2e99-4ccc-ba8e-8cd873625fdf_2.jpg)

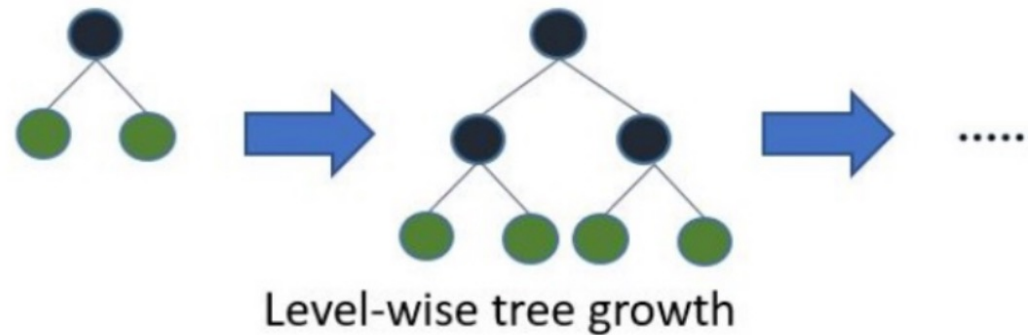
## 04 앙상블 기법 - LightGBM 알고리즘

- ▶ LightGBM은 XGBoost와 함께 부스팅 알고리즘에서 가장 각광을 받고 있다.
- ▶ LightGBM의 장점은 XGBoost보다 학습에 걸리는 시간이 훨씬 적다.
- ▶ LightGBM의 장점은 또한 메모리 사용량도 상대적으로 적다.

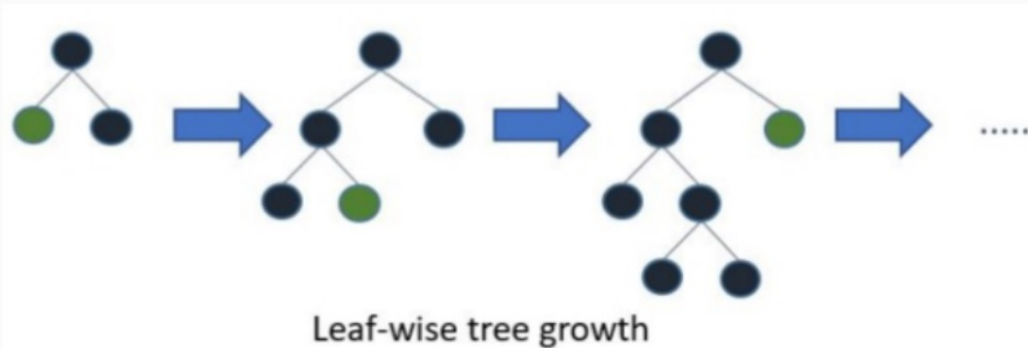
# 04 앙상블 기법 - LightGBM 알고리즘

## ▶ LightGBM와 XGBoost 알고리즘의 차이점

XGBoost:



LightGBM:



<https://rohitgr7.github.io/lightgbm-another-gradient-boosting/>

# 04 앙상블 기법 - LightGBM 알고리즘

## ▶ 파라미터

- (A) `num_iterations` (default = 100) : 반복 수행하려는 트리의 개수
- (B) `learning_rate` (default = 0.1) : 부스팅 스텝을 반복적으로 수행할 때, 업데이트 되는 학습값.
- (C) `max_depth` (default = -1)  
깊이 지정 없음. LightGBM은 **leaf-wise** 기반으로 깊이가 상대적으로 더 깊다.
- (D) `num_leaves` (default = 31) : 하나의 트리가 가질 수 있는 최대 리프 개수
- (E) `boosting` (default = gbdt) : 부스팅 트리를 생성하는 알고리즘  
gbdt, rf



# 04 앙상블 기법 - LightGBM 알고리즘

## ▶ 파라미터

- A. bagging\_fraction (default = 1.0) : 데이터를 샘플링하는 비율 지정. (과적합 제어)
- B. feature\_fraction (default = 0.1) : 개별 트리를 학습할 때마다 무작위로 선택하는 피처의 비율
- C. lambda\_l1(default = 0.0), lambda\_l2(default=0.0)  
L1 규제, L2 규제를 위한 값.

# 04 앙상블 기법 - LightGBM 알고리즘

## ▶ 파라미터 조정

- num\_leaves의 개수를 중심으로 min\_child\_samples, max\_depth를 함께 조정하면서 모델의 복잡도를 줄이기
- max\_depth는 num\_leaves, min\_data\_in\_leaf와 결합하면서 과적합을 제어해 간다.