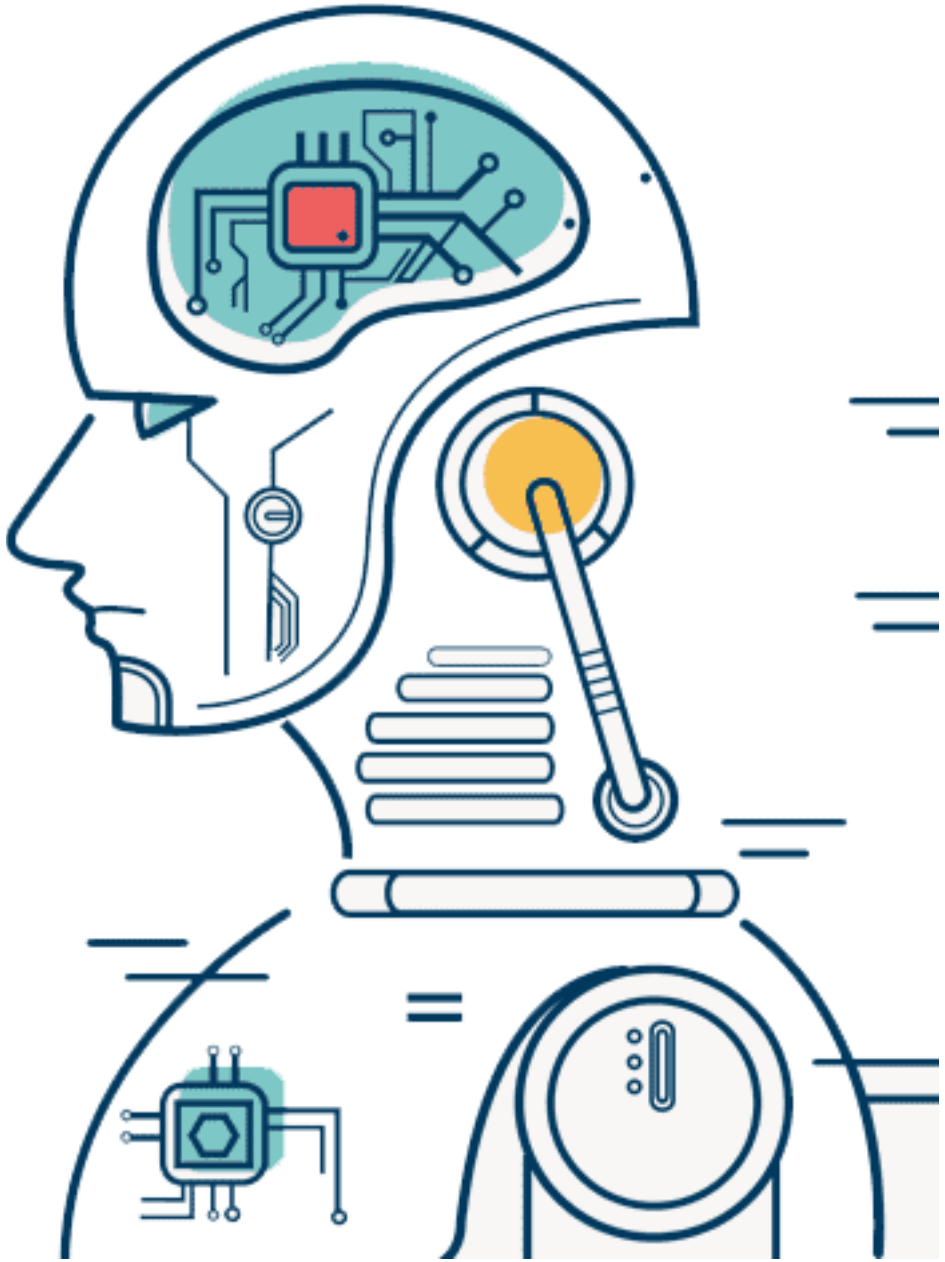
스마트인재개발원
Smart Human Resources Development

Machine Learning

Chapter_7 지도학습
(Ensemble Model)



김은영

학습목표

- Ensemble 개념을 이해 할 수 있다.
- Ensemble 모델의 유형별 학습방법을 이해하고 사용할 수 있다.

Ensemble?

스마트인재개발원
Smart Human Resources Development

여러 개의 모델이 예측한 값을 결합하여
정확한 최종 예측을 도출하는 기법

Ensemble을 사용하는 이유

- 단일 모델에 비해 높은 성능과 신뢰성을 얻을 수 있음
- 데이터의 양이 적은 것에 대비 충분한 학습 효과를 거둘 수 있음

Ensemble 유형에 따른 분류

- 보팅(Voting)

  여러 개의 다른 종류의 모델이 예측한 결과를 투표 혹은 평균을 통해 최종 결정하는 방식

- 배깅(Bagging)

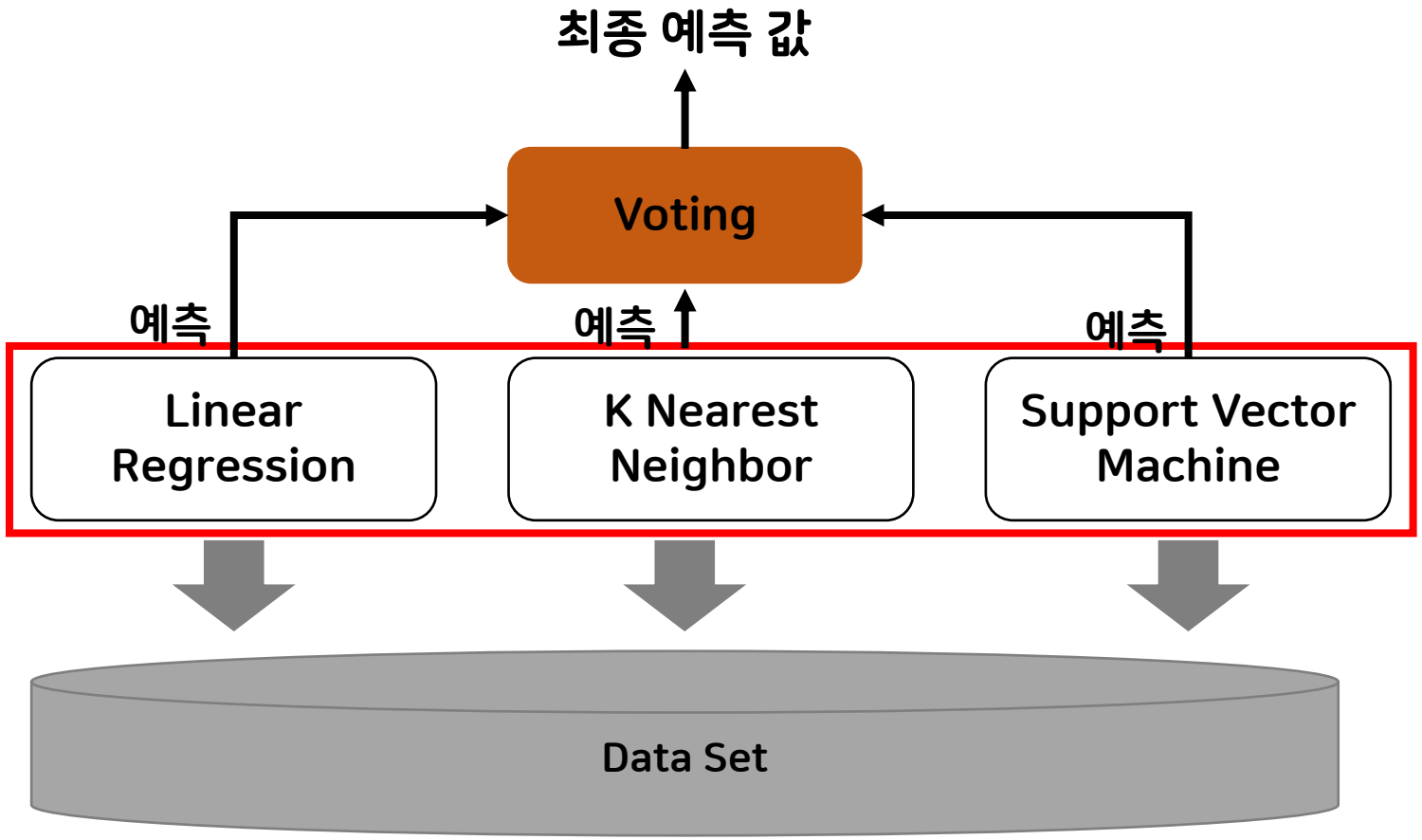
  여러 개의 같은 종류의 모델이 예측한 결과를 투표 혹은 평균을 통해 최종 결정하는 방식

- 부스팅(Boosting)

  여러 개의 같은 종류의 모델이 순차적으로 학습-예측하며 가중치를 부여해 오류를 개선하는 방식

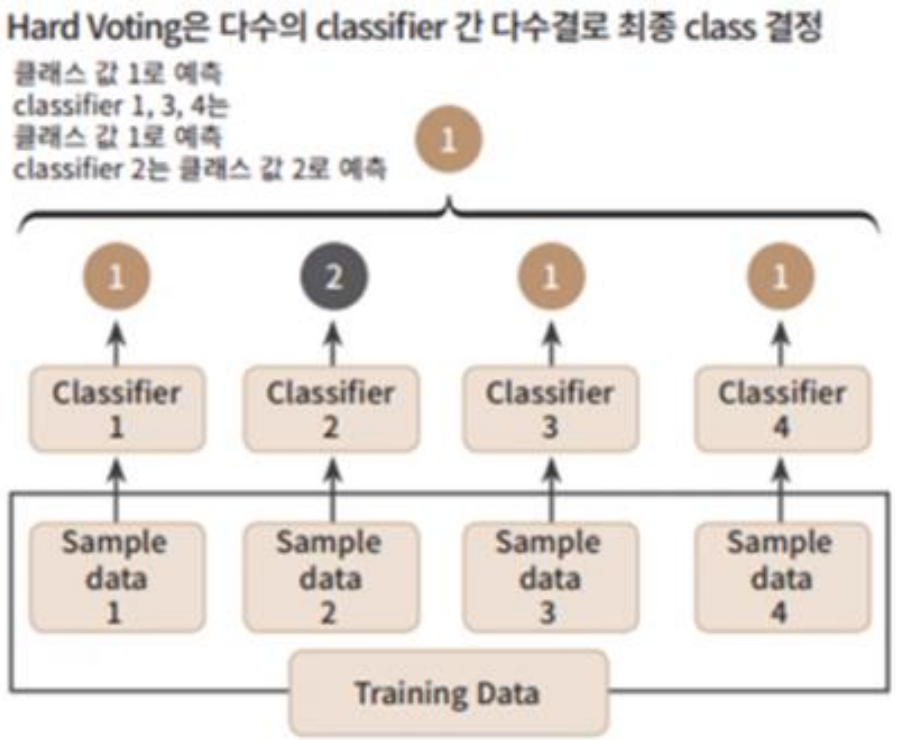
보팅(Voting)

여러 개의 다른 모델이 예측한 결과를 투표 혹은 평균을 통해 최종 예측 결과를 선정

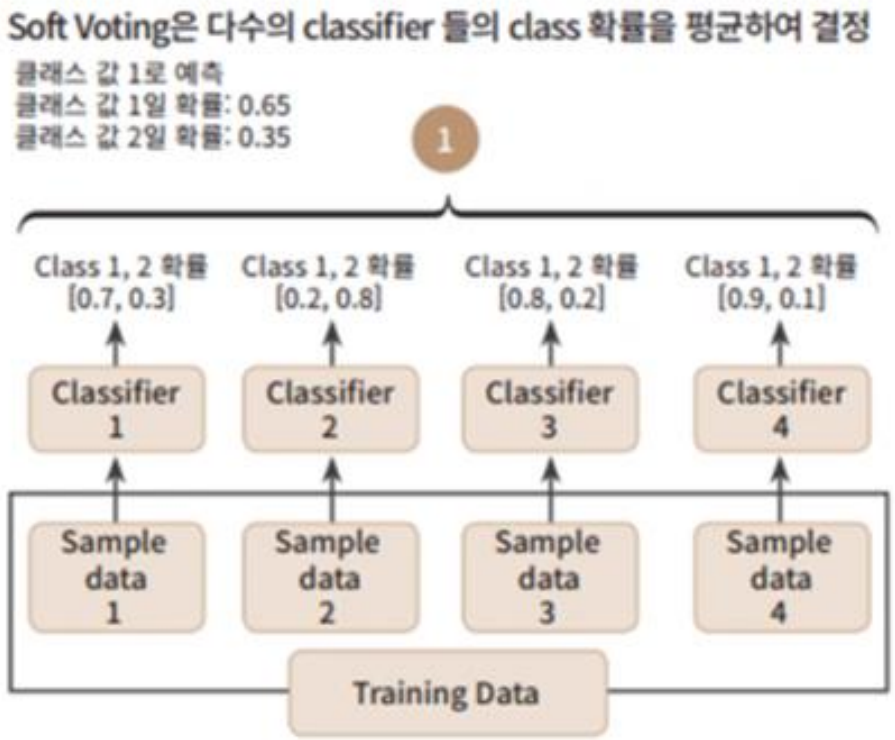


보팅(Voting)

1. 하드 보팅(Hard Voting) : 다수결
2. 소프트 보팅(Soft Voting) : 각 확률의 평균



<하드 보팅>



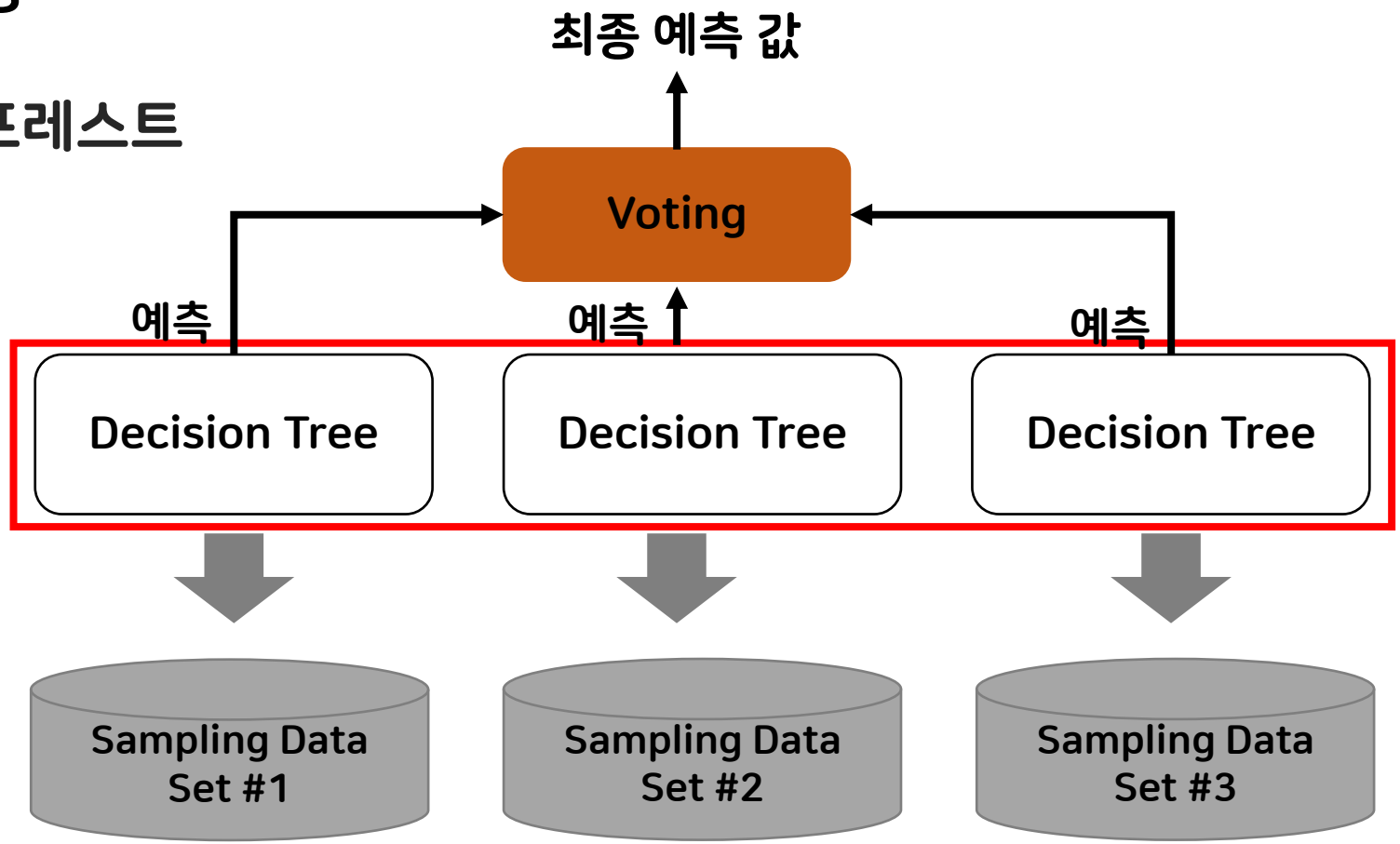
<소프트 보팅>

https://m.blog.naver.com/wnstn0154/221823380552

배깅(Bagging)

여러 개의 같은 종류의 모델이 예측한 결과를 투표 혹은 평균을 통해 최종 예측 결과를 선정

예 : 랜덤 포레스트



보팅(Voting) vs 배깅(Bagging)

공통점

- 여러 개의 모델이 투표 또는 평균을 통해 최종 예측 결과를 결정

다른점

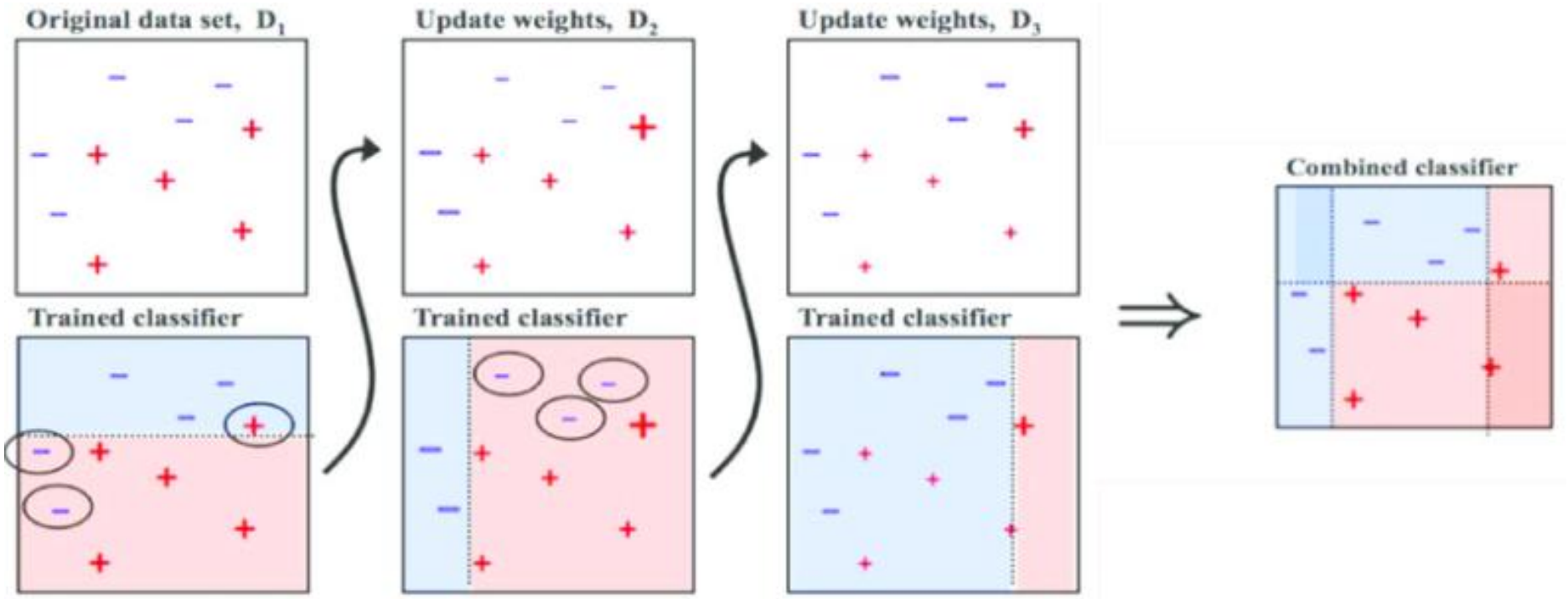
- 보팅(Voting) : 서로 다른 모델을 결합
- 배깅(Bagging) : 같은 종류의 모델을 결합

데이터 샘플링을 다르게, 중첩 허용 ➔ Bootstrap 분할 방식

부스팅(Boosting)

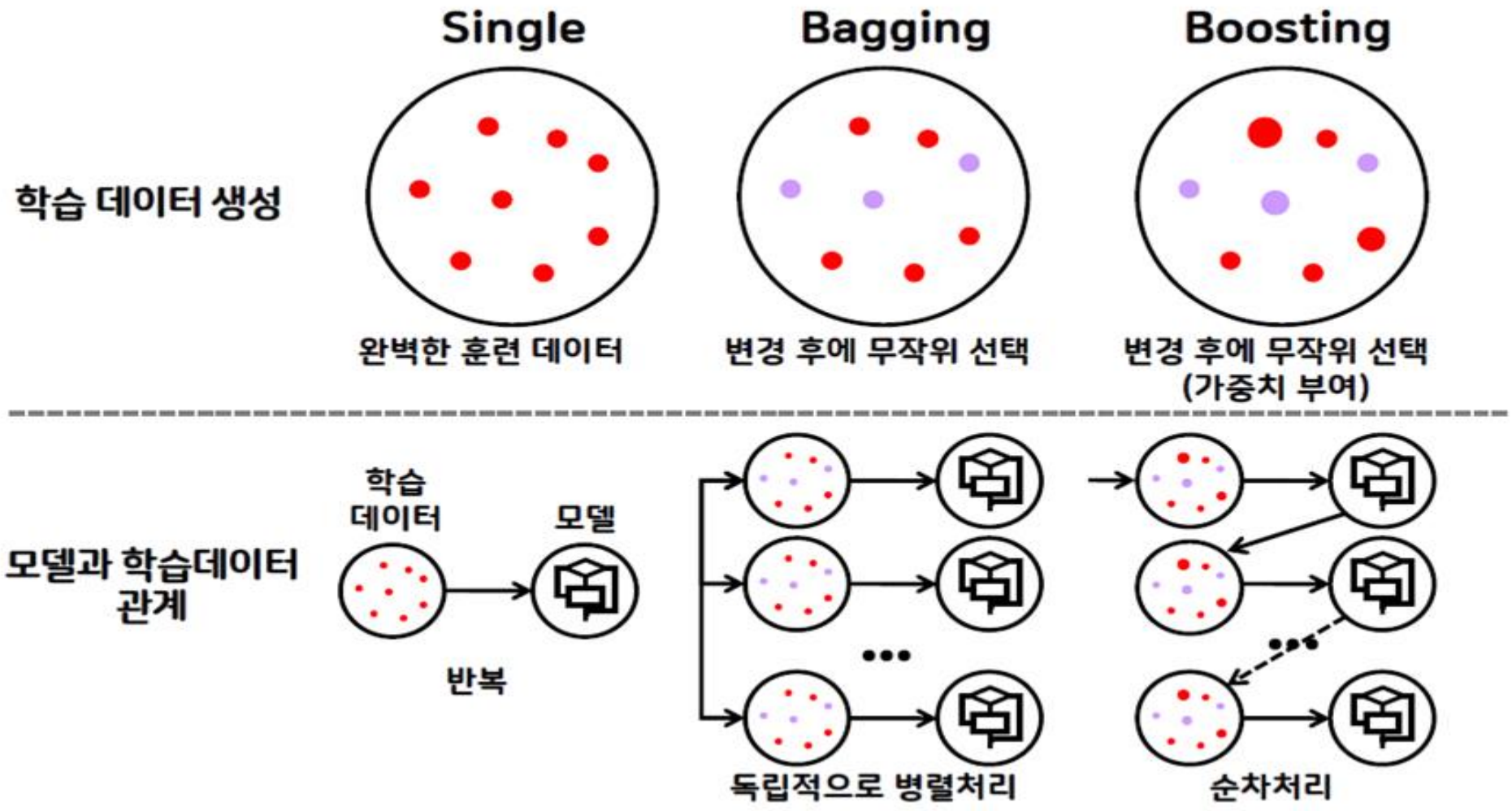
여러 개의 모델이 순차적으로 학습-예측하며 잘못 예측한 데이터에 가중치(weight)를 부여해 오류를 개선해 나가면서 학습하는 방식

예 : AdaBoost, Gradient Boosting, XGBoost, LightGBM



Medium (Boosting and Bagging explained with examples)

배깅(Bagging) vs 부스팅(Boosting)



https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting

배깅(Bagging) vs 부스팅(Boosting)

비교	Bagging	Boosting
특징	병렬 앙상블 모델 (각 모델이 서로 독립적)	연속 앙상블 (이전 모델의 오차를 고려)
목적	일반적으로 좋은 모델을 만들기 위해 과대적합 방지(편향된 학습을 방지)	맞추기 어려운 문제를 풀기 위해 과소적합 방지(부족한 학습을 방지)
적합한 상황	데이터 값들의 편차가 클 경우	학습 정확도가 낮거나 오차가 클 경우
대표 알고리즘	Random Forest	AdaBoost, Gradient Boosting, XGBoost, Light GBM
데이터 선택	무작위 선택	무작위 선택 (오류 데이터에 가중치 적용)

Ensemble Model - RandomForest

- 여러 개의 결정 트리 모델로 예측한 값을 소프트 보팅을 통해 최종 선택하는 배깅의 대표적 모델

- 서로 다른 방향(알고리즘)으로 과대적합된 트리를 많이 만들고 평균을 내어 일반화 시키는 모델

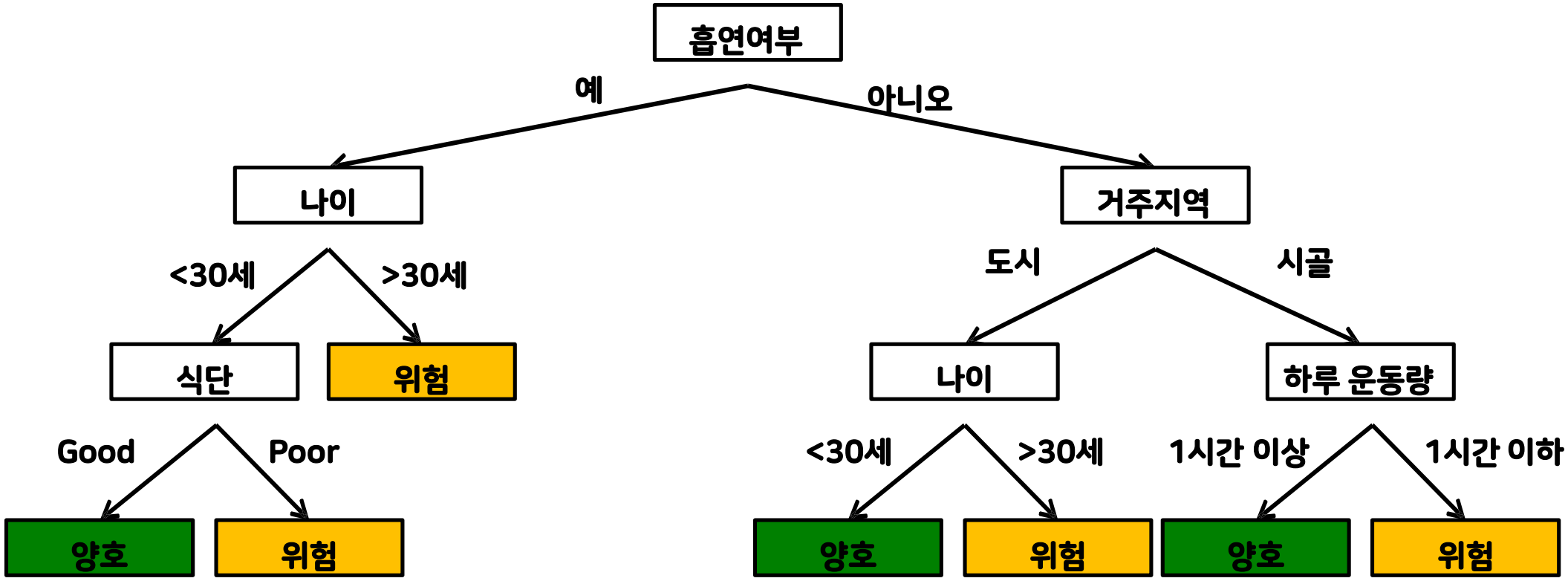
- 다양한 트리를 만드는 방법
  - 트리를 만들 때 사용하는 데이터를 무작위로 선택
  - 노드 구성 시 기준이 되는 특성을 무작위로 선택

- 분류와 회귀 모두 가능

Ensemble Model - RandomForest

스마트인재개발원
Smart Human Resources Development

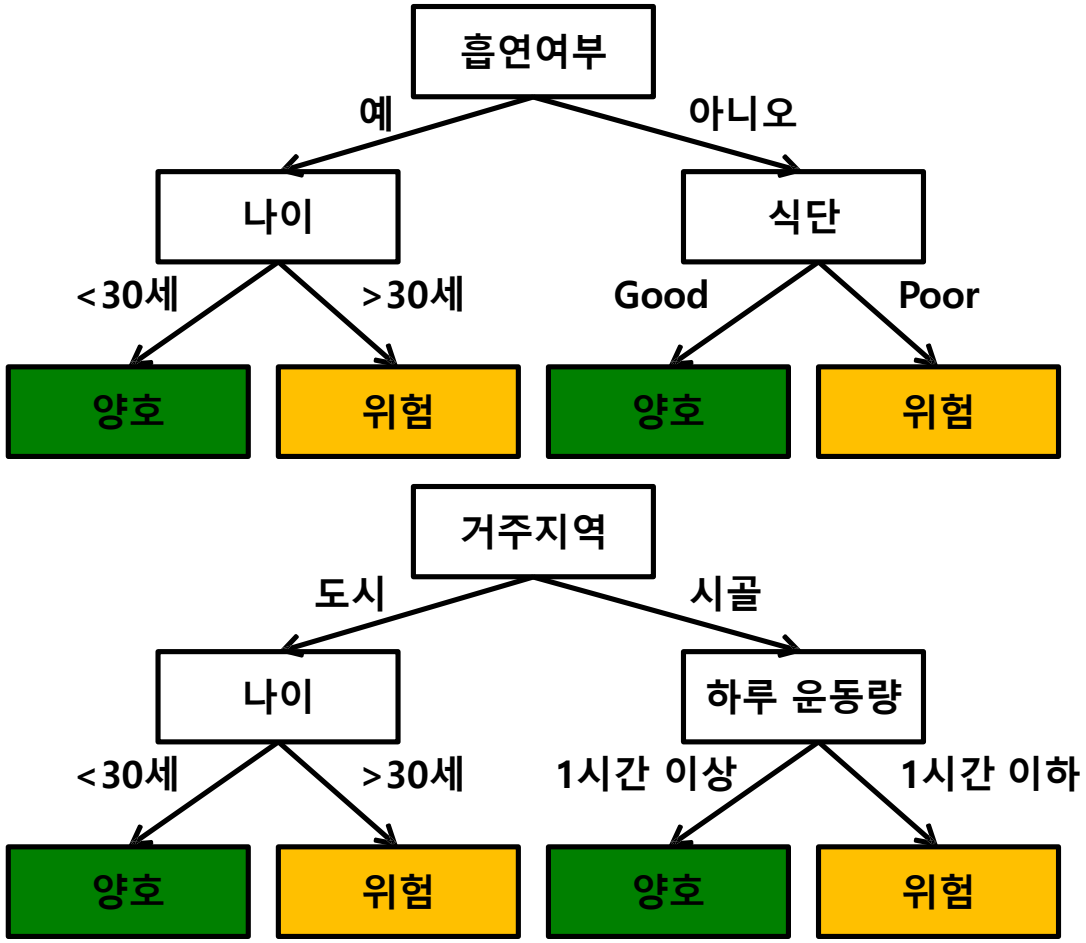
건강위험도를 예측하기 위한 결정트리

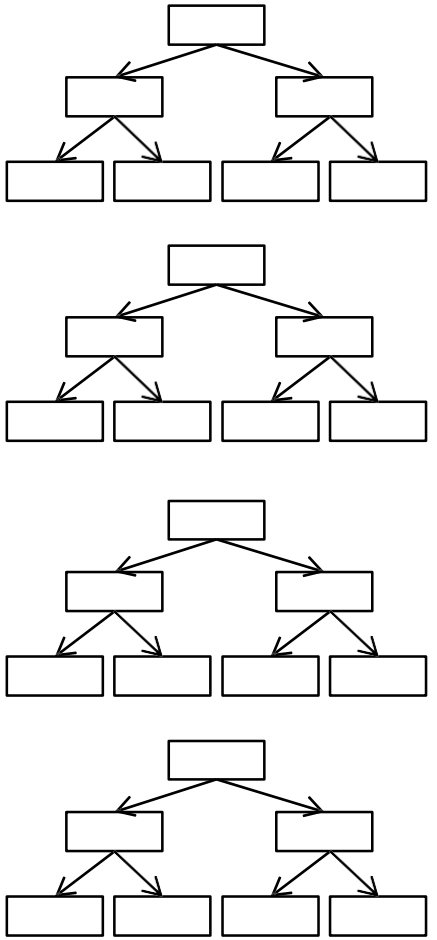


Ensemble Model - RandomForest

스마트인재개발원
Smart Human Resources Development

건강위험도를 예측하기 위한 랜덤포레스트





참고 : refactorthis.net의 포스트

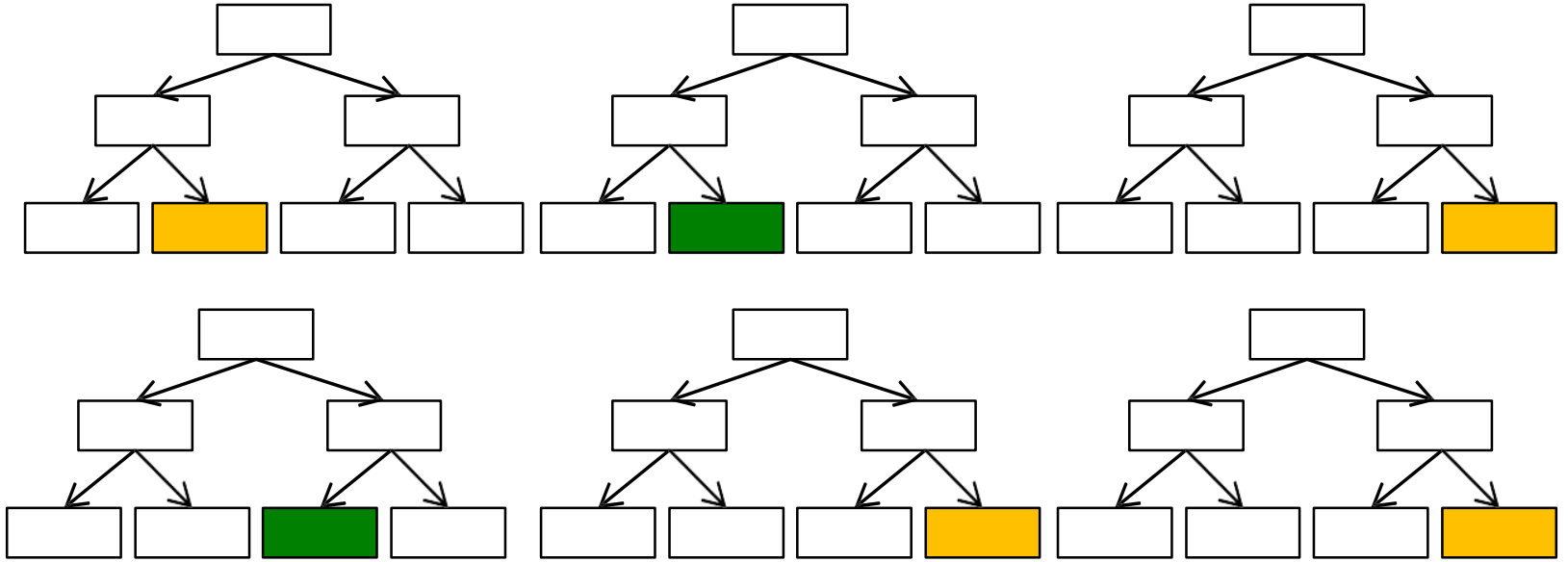
Ensemble Model - RandomForest

스마트인재개발원 Smart Human Resources Development

- 다수 결정트리 의견이 통합되지 않는다면? 투표에 의한 **다수결의 원칙**을 따름

  ➔**앙상블 방법** (Ensemble Methods)

  장점 : 실제값에 대한 추정값 오차 평균화, 분산 감소, 과대적합 감소



위험으로 판정 (양호 2, 위험 4)

Ensemble Model - RandomForest

스마트인재개발원
Smart Human Resources Development

주요 매개변수(Hyperparameter)

scikit-learn의 경우

```
RandomForestClassifier(n_estimators,
max_features, random_state)
```

- 트리의 개수 : n_estimators
- 선택할 특징의 최대 수 : max_features
  (1로 하면 특성을 고려하지 않으며 큰 값이면 DT와 비슷해짐)
- 선택할 데이터의 시드 : random_state

결정 트리 매개변수(Hyperparameter)

- 트리의 최대 깊이 : max_depth
- 말단 노드 최대 개수 : max_leaf_nodes
- 말단 노드가 되기 위한 최소 샘플 수 : min_samples_leaf

Ensemble Model – RandomForest

유방암 데이터를 이용하여

- 100개의 트리로 구성된 RandomForest 모델로 학습,
- 특성의 중요도(feature_importance_)를 활용하여
- bar차트로 표시해보자

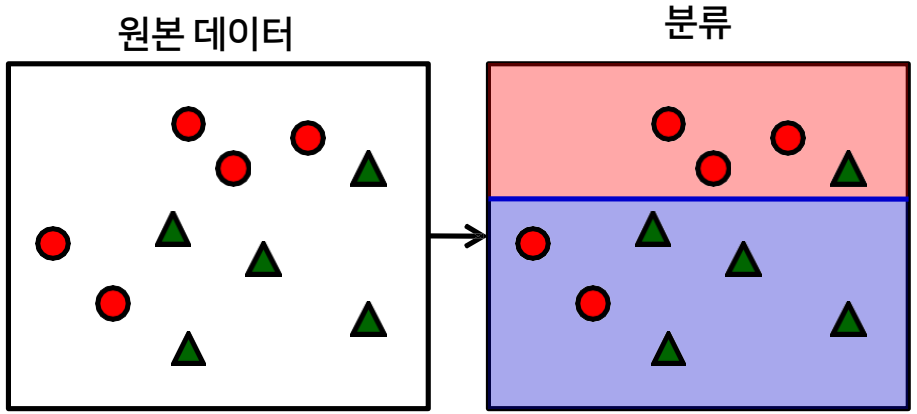
Ensemble Model – AdaBoost(Adaptive Boosting)

스마트인재개발원
Smart Human Resources Development

- 랜덤 포레스트처럼 결정 트리 기반의 모델 ➔ 각각의 트리들이 독립적으로 존재하지 않음.

- 동작 순서

(1) 첫 번째 의사결정 트리를 생성 → 위쪽 빨간 원이 3개 있는 곳을 대충 분류 시킴 → 2개의 빨간 원과 1개의 녹색 세모가 잘못 구분됨

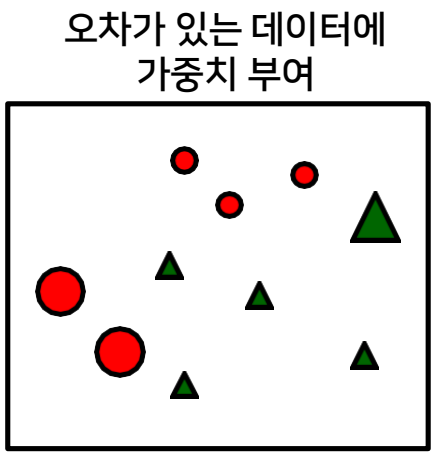


Ensemble Model – AdaBoost(Adaptive Boosting)

스마트인재개발원
Smart Human Resources Development

- 동작 순서

(2) 잘못된 2개의 빨간 원과 1개의 녹색 세모에 높은 가중치를 부여하고 맞은 것에는 빨간 원 3개와 녹색 세모 4개는 낮은 가중치 부여

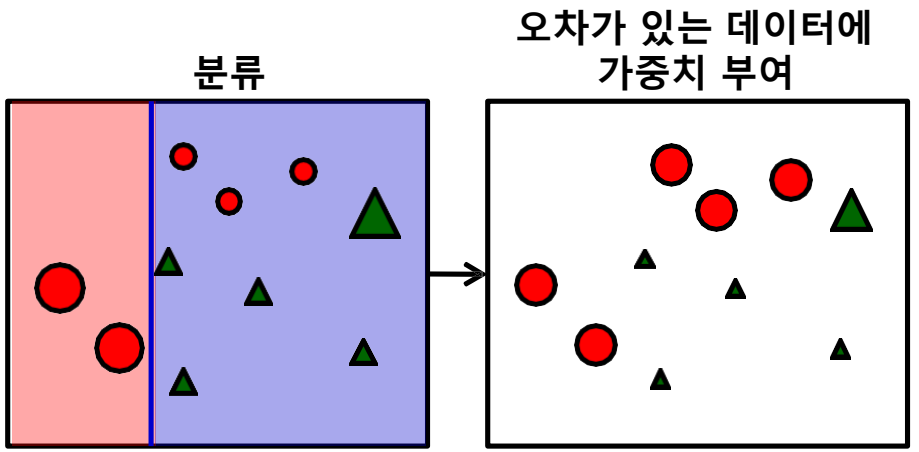


Ensemble Model – AdaBoost(Adaptive Boosting)

스마트인재개발원
Smart Human Resources Development

- 동작 순서

(3) 가중치를 부여한 상태에서 다시 분류 시킴 → 잘못된 3개의 빨간 원에 높은 가중치를 부여하고 맞은 5개의 녹색 세모는 낮은 가중치를 부여

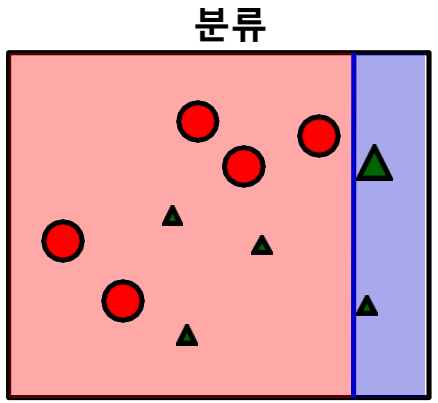


Ensemble Model – AdaBoost(Adaptive Boosting)

스마트인재개발원
Smart Human Resources Development

- 동작 순서

(4) 가중치를 부여한 상태에서 다시 분류 시킴

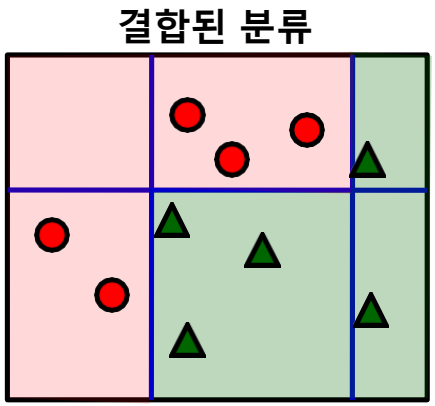


Ensemble Model – AdaBoost(Adaptive Boosting)

스마트인재개발원
Smart Human Resources Development

- 동작 순서

(5) 진행한 분류들을 결합한다.



- 에이다 부스팅은 학습과 예측을 진행할수록 데이터들의 가중치가 달라짐
  - 잘못 분류된 데이터는 가중치↑, 잘 분류된 데이터는 가중치↓

Ensemble Model – AdaBoost(Adaptive Boosting)

주요 매개변수(Hyperparameter)

scikit-learn의 경우

```
AdaBoostClassifier(n_estimators, random_state)
```

- 트리의 개수 : n_estimators
- 선택할 데이터의 시드 : random_state

Ensemble Model – AdaBoost(Adaptive Boosting)

유방암 데이터를

- AdaBoost 모델로 학습,

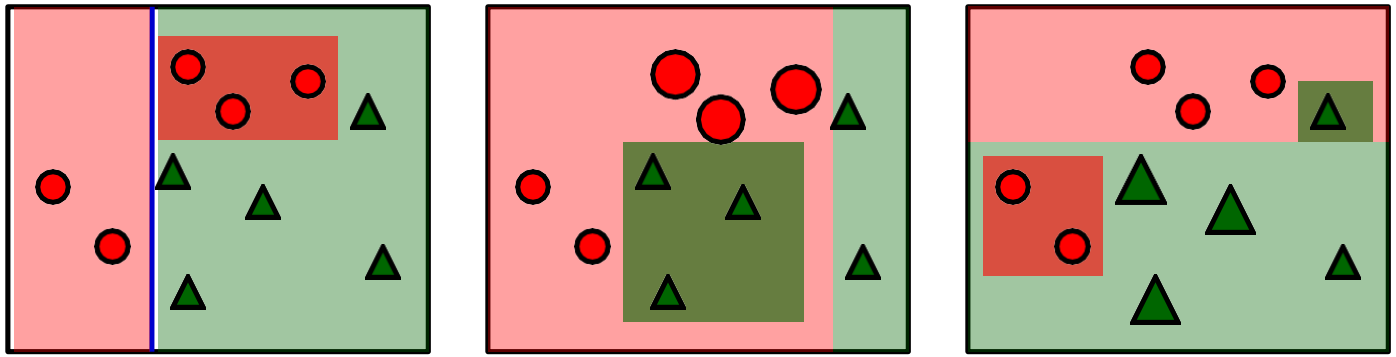
- 특성의 중요도(feature_importance_) 를 활용하여

- bar차트로 표시해 보자.

Ensemble Model – GBM(Gradient Boosting Machine)

스마트인재개발원
Smart Human Resources Development

- AdaBoost와 기본 개념은 동일하고 가중치를 계산하는 방식에서 경사하강법을 이용하여 최적의 가중치(파라미터)를 찾아냄



https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting

Ensemble Model – GBM(Gradient Boosting Machine)

주요 매개변수(Hyperparameter)

scikit-learn의 경우

```
GradientBoostingClassifier(n_estimators,
                           learning_rate, max_depth, random_state)
```

- 트리의 개수 : n_estimators
- 학습률 : learning_rate (높을수록 오차를 많이 보정)
- 트리의 깊이 : max_depth
- 선택할 데이터의 시드 : random_state

Ensemble Model – GBM(Gradient Boosting Machine)

장단점

- 학습속도가 느림(부스팅의 일반적인 단점)
- 데이터 특성의 스케일을 조정할 필요가 없음(트리 기반 모델의 특성)

Ensemble Model – GBM(Gradient Boosting Machine)

유방암 데이터를

- GBM 모델로 학습,

- 특성의 중요도(feature_importance_) 를 활용하여

- bar차트로 표시해 보자.

Ensemble Model – XGBoost(eXtreme Gradient Boosting)

- 결정 트리 기반의 앙상블 모델에서 가장 각광받고 있는 알고리즘의 하나
- 분류에 있어 일반적으로 다른 머신 러닝 모델보다 뛰어난 성능을 나타냄
- GBM 기반이지만 Early Stopping 기능과 과대적합 방지를 위한 규제 포함

  ➔ GBM의 단점인 느린 학습시간, 과대적합 문제를 해결

- 분류와 회귀 모두 가능

주요 매개변수(Hyperparameter)

scikit-learn의 경우

```
XGBClassifier(n_estimators, learning_rate,
                                max_depth, random_state)
```

- 트리의 개수 : n_estimators
- 학습률 : learning_rate (높을수록 오차를 많이 보정)
- 트리의 깊이 : max_depth
- 선택할 데이터의 시드 : random_state

유방암 데이터를

- XGBoost 모델로 학습,

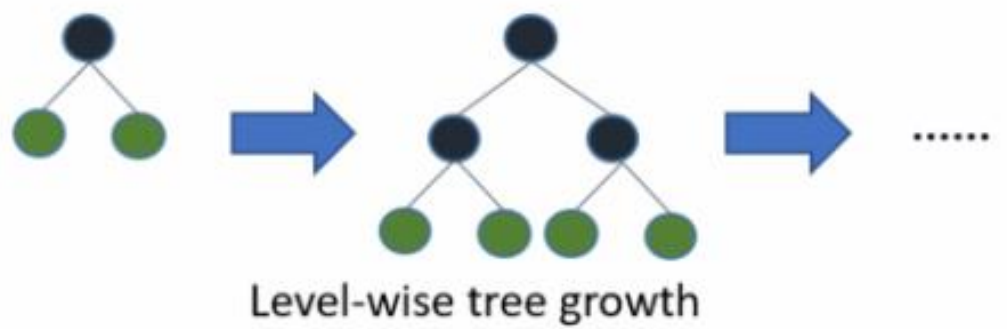
- 특성의 중요도(feature_importance_) 를 활용하여

- bar차트로 표시해 보자.

Ensemble Model – Light GBM

- XGBoost에 비해 가벼워 속도가 빠른 모델
- Leaf-wise(수직방향, 비대칭)로 트리를 성장 시킴(속도↑)
- Level-wise(수평방향, 깊이↓, 대칭)보다 오류가 더 적음(정확도↑)

Light GBM



Random Forest, XG Boosting

https://nurilee.com/2020/04/03/lightgbm-definition-parameter-tuning/

Ensemble Model – Light GBM

스마트인재개발원
Smart Human Resources Development

장단점

- 대량(1만개 이상)의 데이터를 병렬로 빠르게 학습 가능(low Memory, GPU 활용 가능)
  - XGBoost 대비 2~10배의 속도(동일 파라미터 설정 시)
  - 소량의 데이터에서는 제대로 동작하지 않음(과대적합 위험)

- 예측 속도가 빠름(Leaf-wise 트리의 장점)
  - Level-wise에 비해 과적합에 민감

Ensemble Model – Light GBM

스마트인재개발원
Smart Human Resources Development

주요 매개변수(Hyperparameter)

- 100개 이상

- max_depth : 트리의 최대 깊이
- early_stopping_rounds : validation 데이터 중 하나의 지표가 정해진 반복 수 만큼 향상되지 않았다면 학습을 중단
- lambda : lambda 값은 regularization 정규화를 합니다. 일반적인 값의 범위는 0 에서 1 사이
- Min_data_in_leaf : Leaf노드가 가지고 있는 최소한의 레코드 수(디폴트 값 : 20, 과적합을 해결할 때 사용되는 파라미터)
- feature_fraction : 0.8 의 의미는 Light GBM이 Tree를 만들 때 매번 각각의 반복 학습 시 파라미터 중에서 80%를 랜덤하게 선택하는 것을 의미
- bagging_fraction : 매번 iteration을 돌 때 사용되는 데이터의 일부를 선택하는데 트레이닝 속도를 높이고 과적합을 방지할 때 주로 사용
- num_boost_round : boosting 반복 학습 수로 일반적으로 100 이상
- min_gain_to_split : 이 파라미터는 분기하기 위해 필요한 최소한의 gain을 의미, Tree에서 유용한 분기의 수를 컨트롤하는데 사용
- max_cat_group : 카테고리 수가 클 때, 과적합을 방지하는 분기 포인트를 찾음(디폴트 값 : 64)
- Task : 데이터에 대해서 수행하고자 하는 임무를 구체화, train일수도 있고 predict 예측일 수도 있음
- application : 문제 타입 설정, 디폴트는 회귀(regression: 회귀분석, binary: 이진 분류, multiclass: 다중 분류)
- learning_rate : 학습률(일반적인 값은 0.1, 0.001, 0.003 등)
- num_leaves : 전체 Tree의 leave 수 이고, 디폴트 값은 31
- device : 디폴트 값은 CPU(GPU로 변경가능)