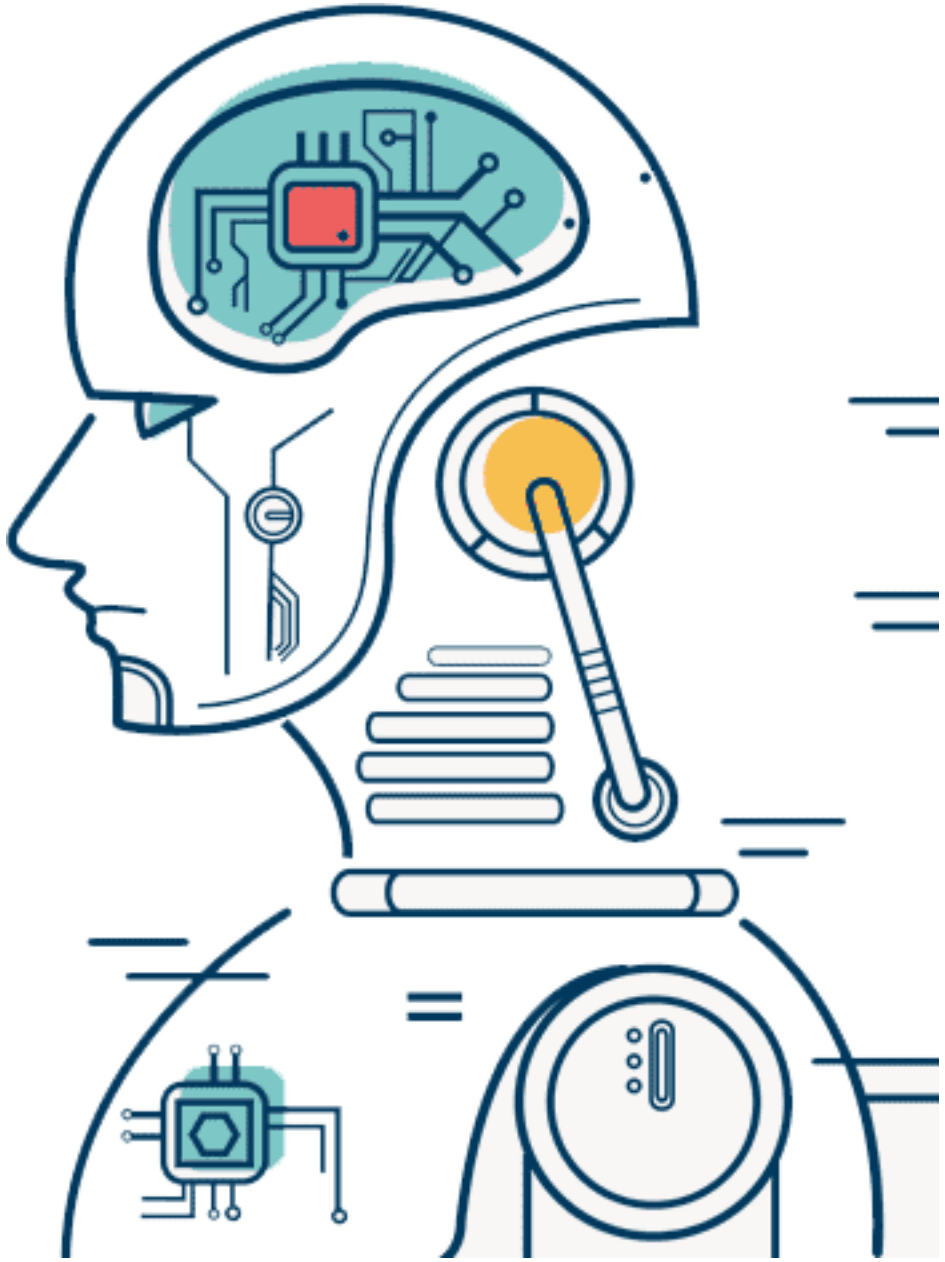
스마트인재개발원
Smart Human Resources Development

Machine Learning



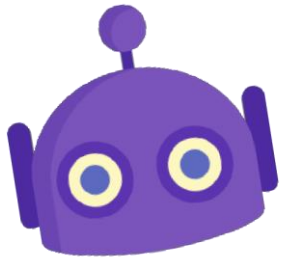
Chapter 7 지도 학습

(Bagging, Boosting, **Random Forest, GBM, Xgboost**)

학습목표

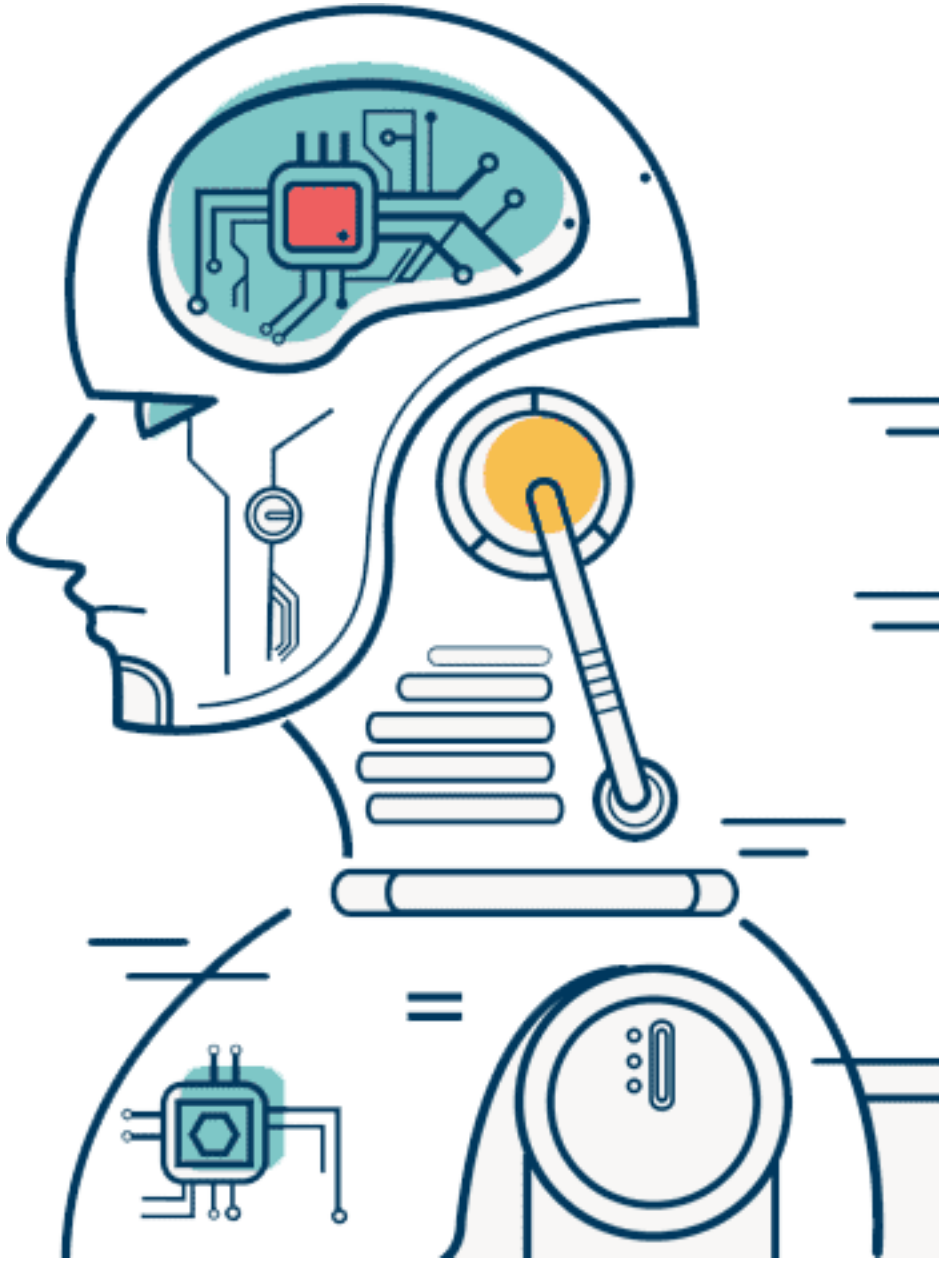
- Ensemble 개념을 이해할 수 있다.
- Ensemble 모델을 활용하여 학습하는 방법에 대해 이해하고 사용할 수 있다
- 특징선택방법에 대해 이해하고 사용할 있다.

스마트인재개발원
Smart Human Resources Development



Ensemble Model

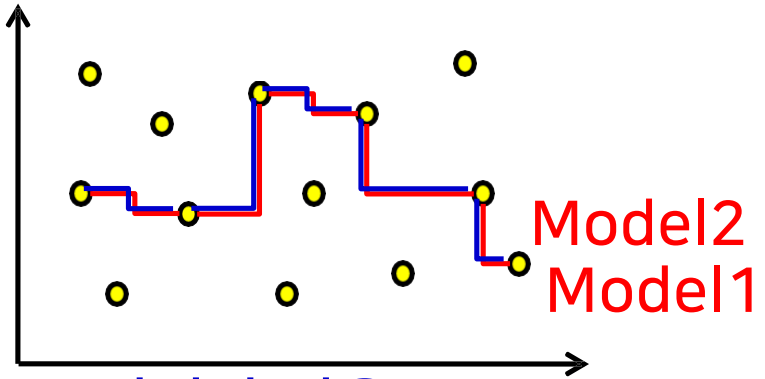
(bagging/boosting,
Random Forest, GBM, Xgboost)



Ensemble Model

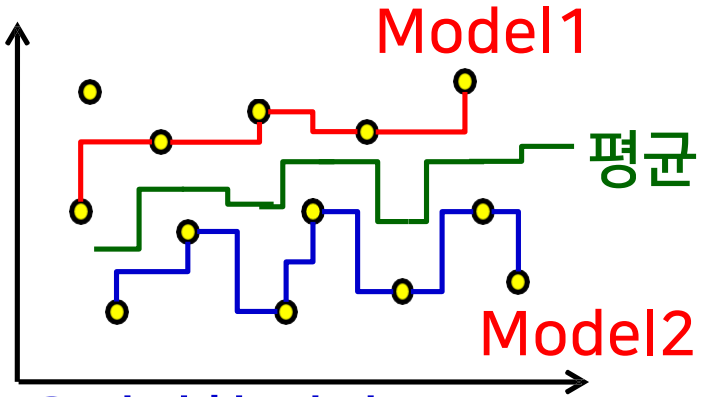
- 모델마다 결과가 다를 때 통합하는 방법의 일종 → 예측 모델들의 평균값을 사용

  → 모델이 많아지면 평균값이 모델의 실제 데이터와 유사해짐



큰 의미가 없음

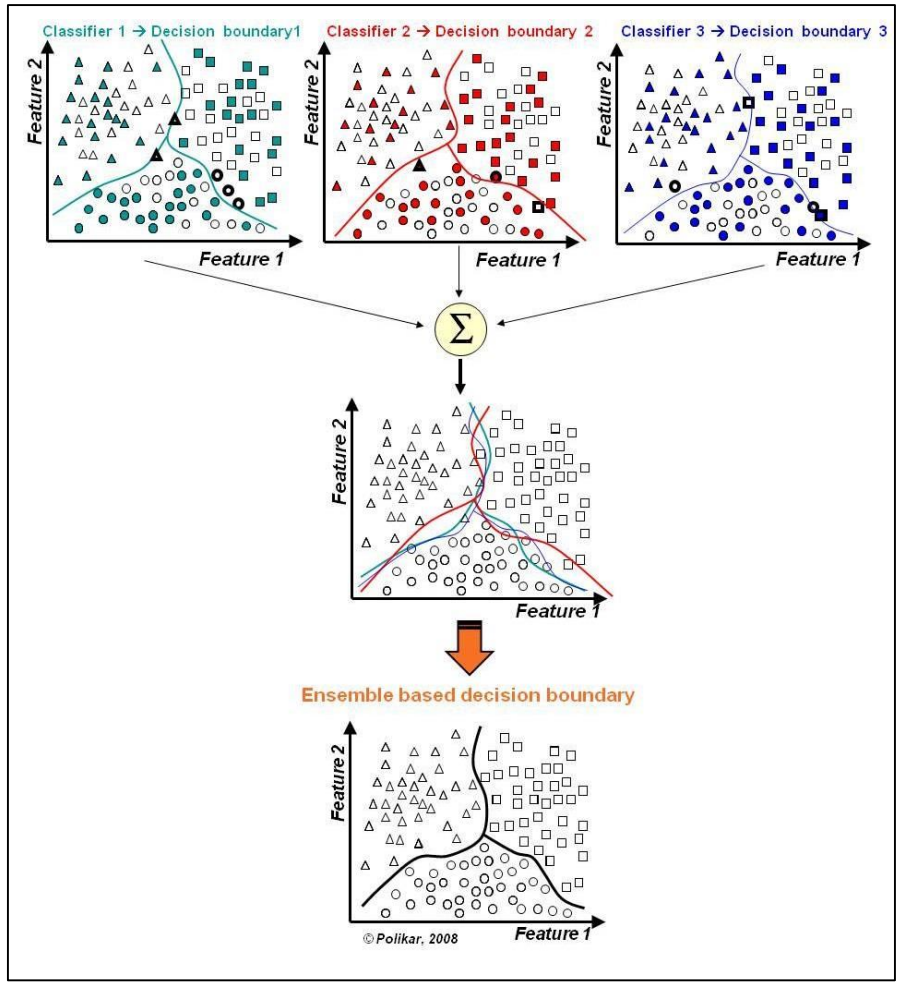
모델1과 2가 가까운 거리에위치



유의미한 결과

모델1과 2의예측 값이 차이가 크다

Ensemble Model



Ensemble Model

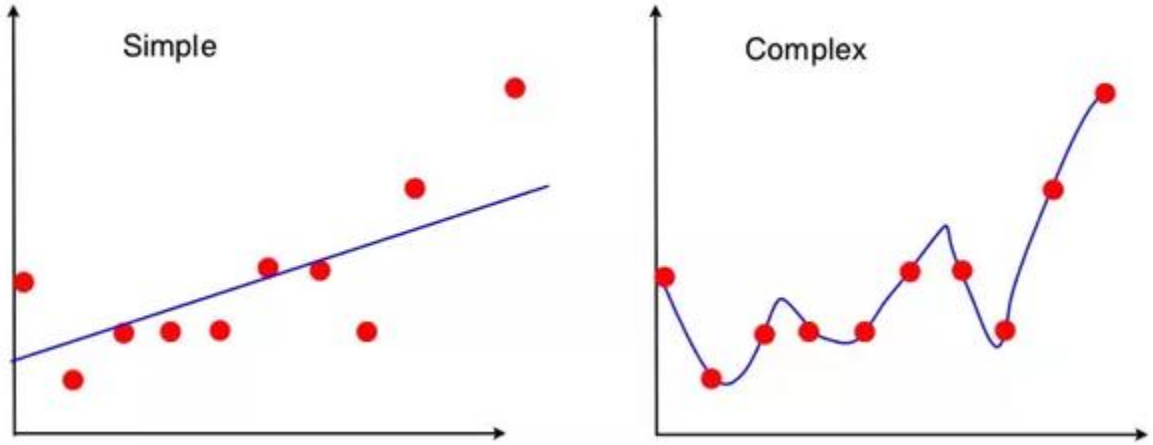
- 높은 bias로 인한 underfitting, 높은 variance로 인한 overfitting을 줄이기 위해 앙상블 기법을 사용 → 알고리즘의 안정성과 정확성 향상

- bias 오차 : 예측값과 실제값 간의 차이
- variance 오차 : 학습된 모델 간에 예측한 값들의 차이

Ensemble Model

- 높은 bias로 인한 underfitting, 높은 variance로 인한 overfitting을 줄이기 위해 앙상블 기법을 사용 → 알고리즘의 안정성과 정확성 향상

- bias 오차 : 예측값과 실제값 간의 차이
- variance 오차 : 학습된 모델 간에 예측한 값들의 차이



Bagging vs Boosting

- Boosting : 일반적으로 학습 데이터의 개수가 작을 때 다수의 학습 데이터 대신에 이전 데이터의 오차를 고려하여 주어진 학습 데이터를 가중치를 변경하면서 여러 번 반복 추출하여 활용 (AdaBoost, GradientBoost, XGBoost 등)

- Bagging (Boostrap Aggregation) : 약간씩 다른 독립된 훈련 데이터 세트를 여러 개 생성하는 것으로 통계적 학습 (머신러닝)의 결과 분산을 줄이는 알고리즘

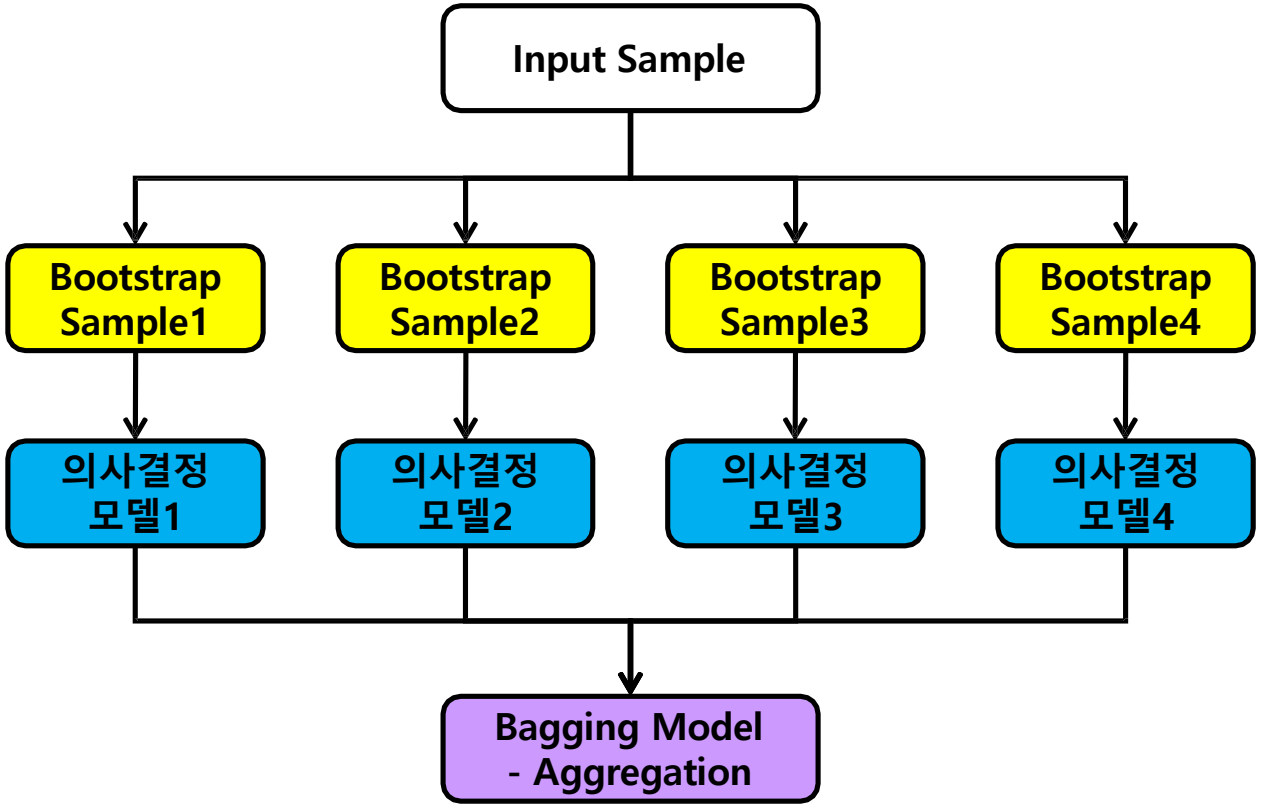
  → Overfitting 감소 → 이산 데이터인 경우는 투표(voting) 방식, 연속 데이터인 경우는 평균으로 집계 (RamdomForest 등)

Bagging vs Boosting

비교	Bagging	Boosting
특징	병렬 앙상블 모델 (각 모델이 서로 독립적)	연속 앙상블 (이전 모델의 오차를 고려)
목적	유사한 모델 생성 → Variance 감소	오차를 감소시킨 모델 생성 → Bias 감소
적합한 상황	복잡한 모델	단순한 모델
대표 알고리즘	Random Forest	AdaBoost, Gradient Boosting, Xgboost
데이터 선택	무작위 선택	무작위 선택 (오류 데이터에 가중치 적용)

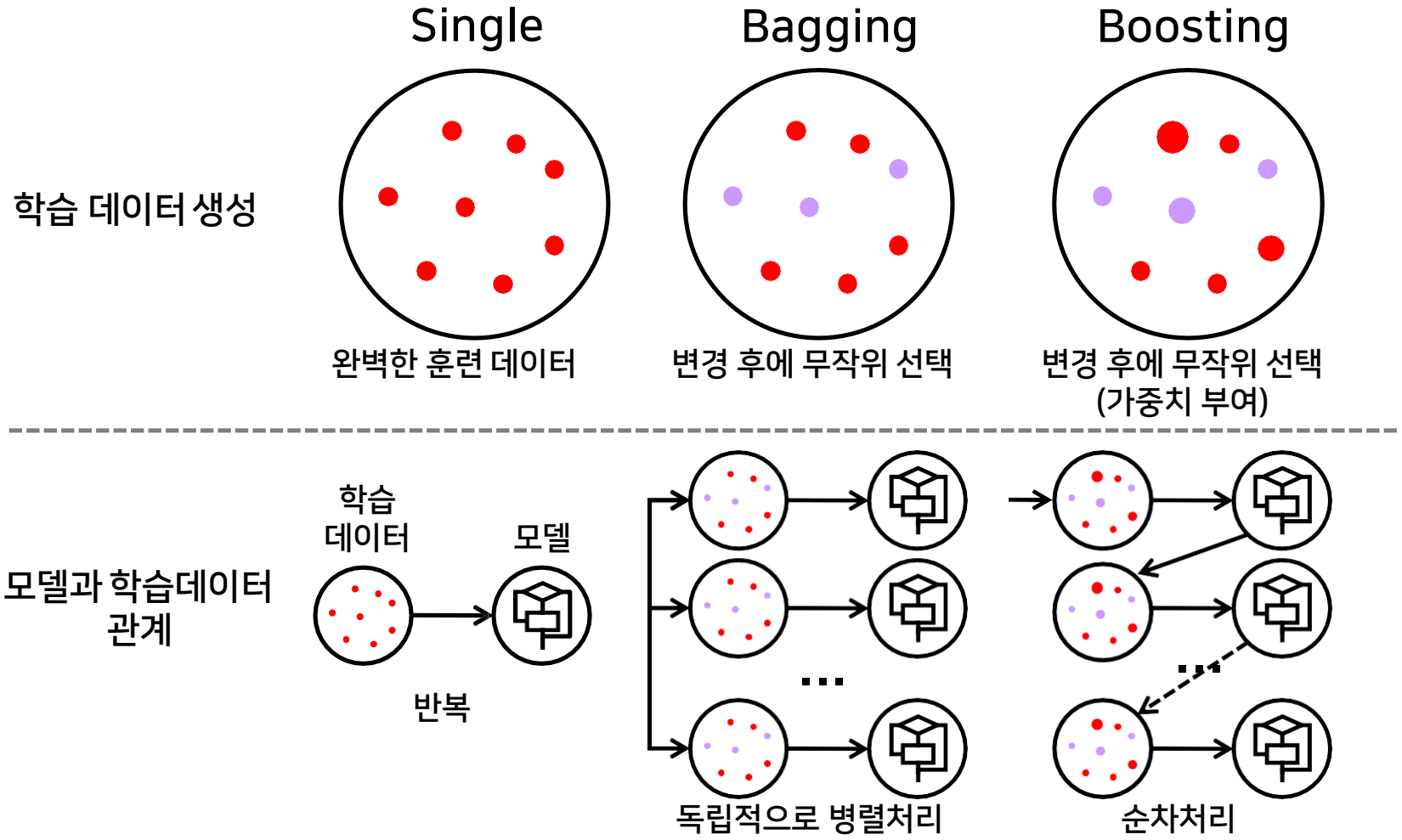
https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting

Bagging vs Boosting



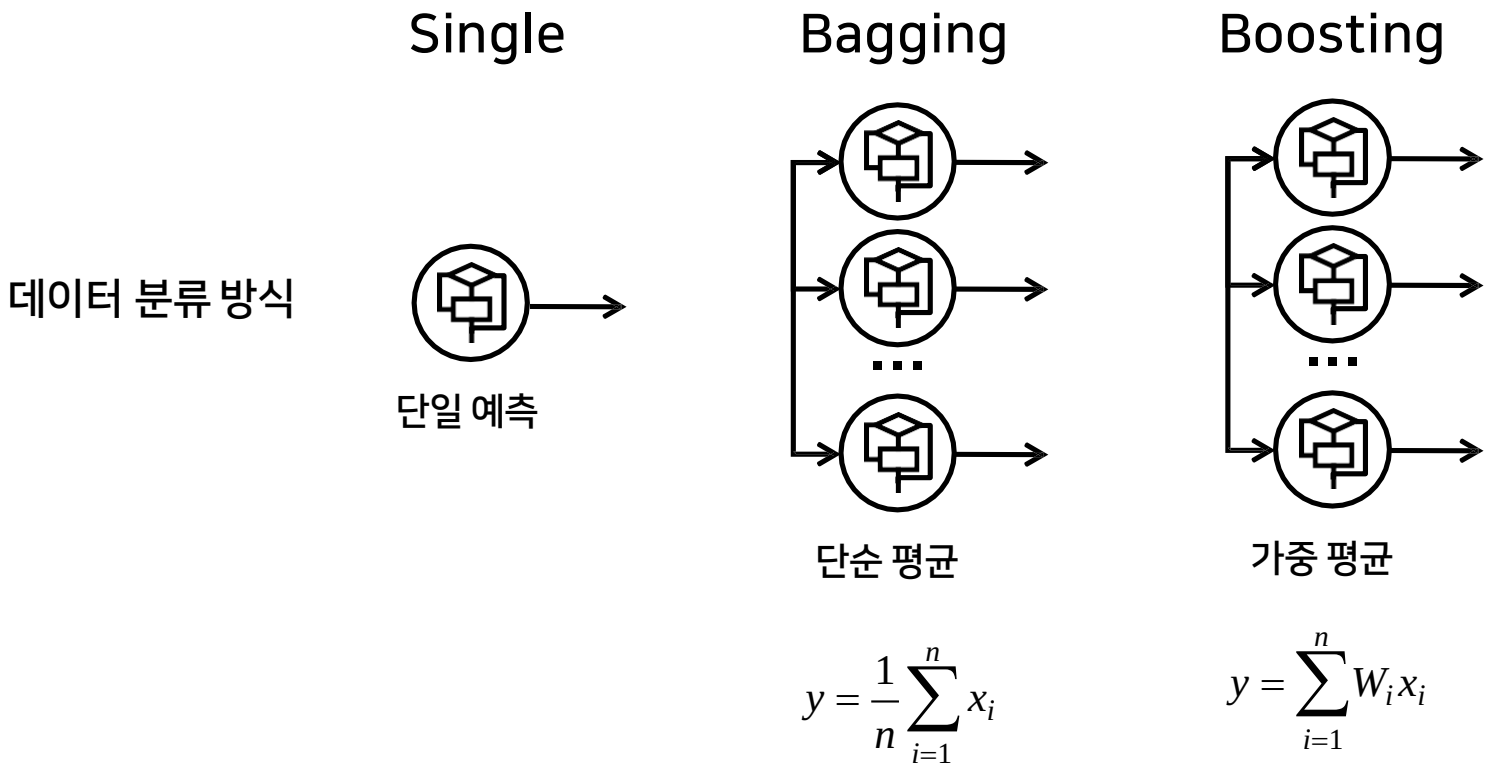
https://swalloow.github.io/bagging-boosting

Bagging vs Boosting



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

Bagging vs Boosting



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

Ensemble Model - RandomForest

서로 다른 방향으로 과대적합된 트리를 많이 만들고 평균을 내어 일반화 시키는 모델

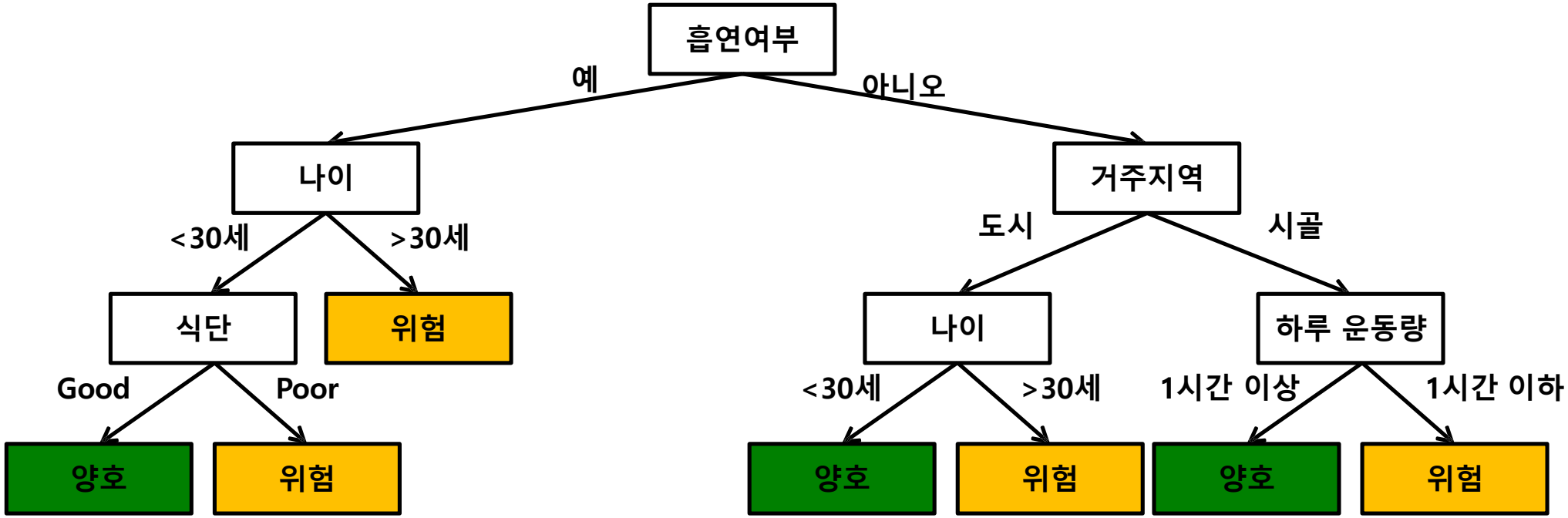
다양한 트리를 만드는 방법 두 가지

- 트리를 만들 때 사용하는 데이터 포인트 샘플을 무작위로 선택한다.

- 노드 구성시 기준이 되는 특성을 무작위로 선택하게 한다.

Ensemble Model - RandomForest

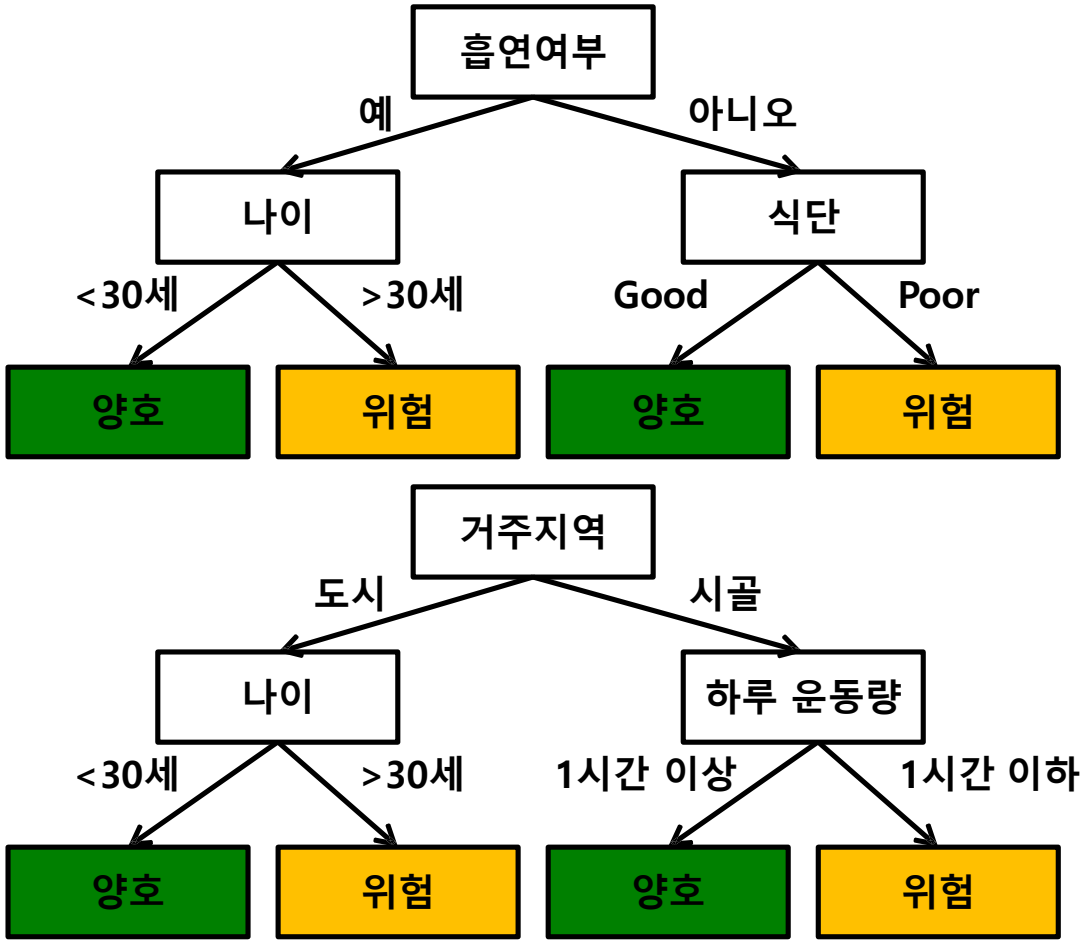
건강위험도를 예측하기 위한 의사결정트리

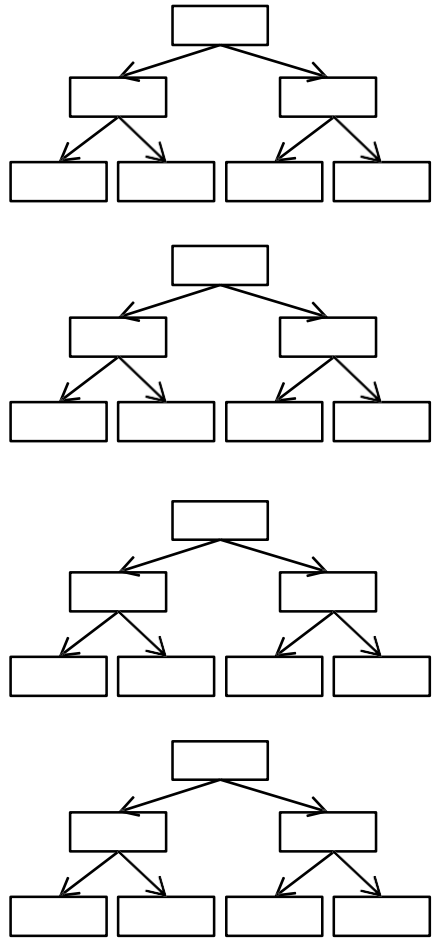


Ensemble Model - RandomForest

스마트인재개발원
Smart Human Resources Development

건강위험도를 예측하기 위한 랜덤포레스트

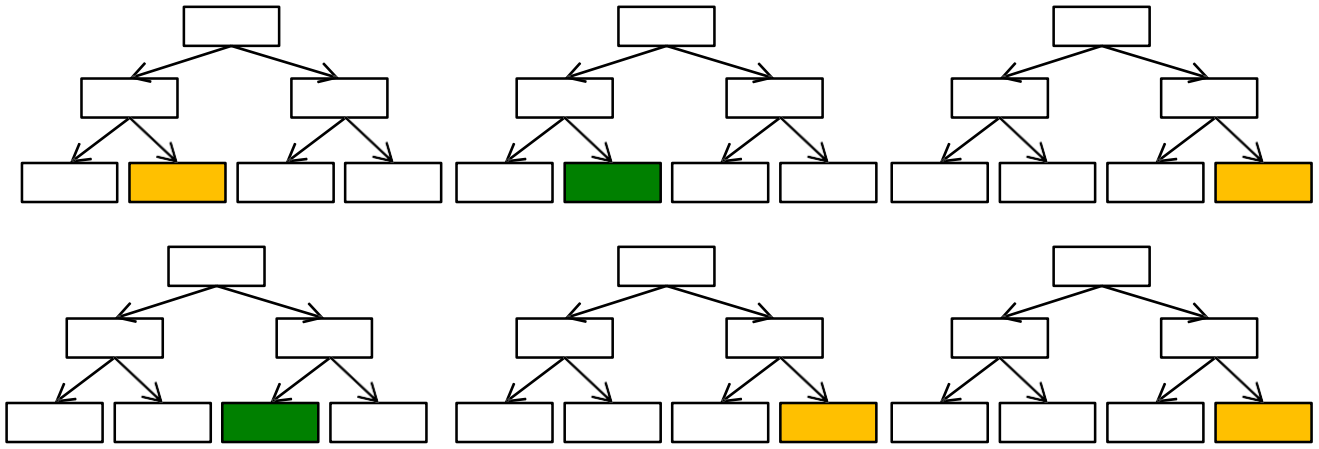




참고 : refactorthis.net의 포스트

Ensemble Model - RandomForest

- 다수의 의사결정트리의 의견이 통합되지 않는다면 → 투표에 의한 다수결의 원칙을 따름 → 앙상블 방법 (Ensemble Methods)
- 장점 : 실제값에 대한 추정값 오차 평균화, 분산 감소, 과적합 감소



위험으로 판정 (양호 2, 위험 4)

Ensemble Model - RandomForest

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

장단점

텍스트 데이터와 같은 희소한 데이터에는 잘 동작하지 않는다.

큰 데이터 세트에도 잘 동작하지만 훈련과 예측이 상대적으로 느리다.

트리 개수가 많아질 수록 시간이 더 오래 걸린다.

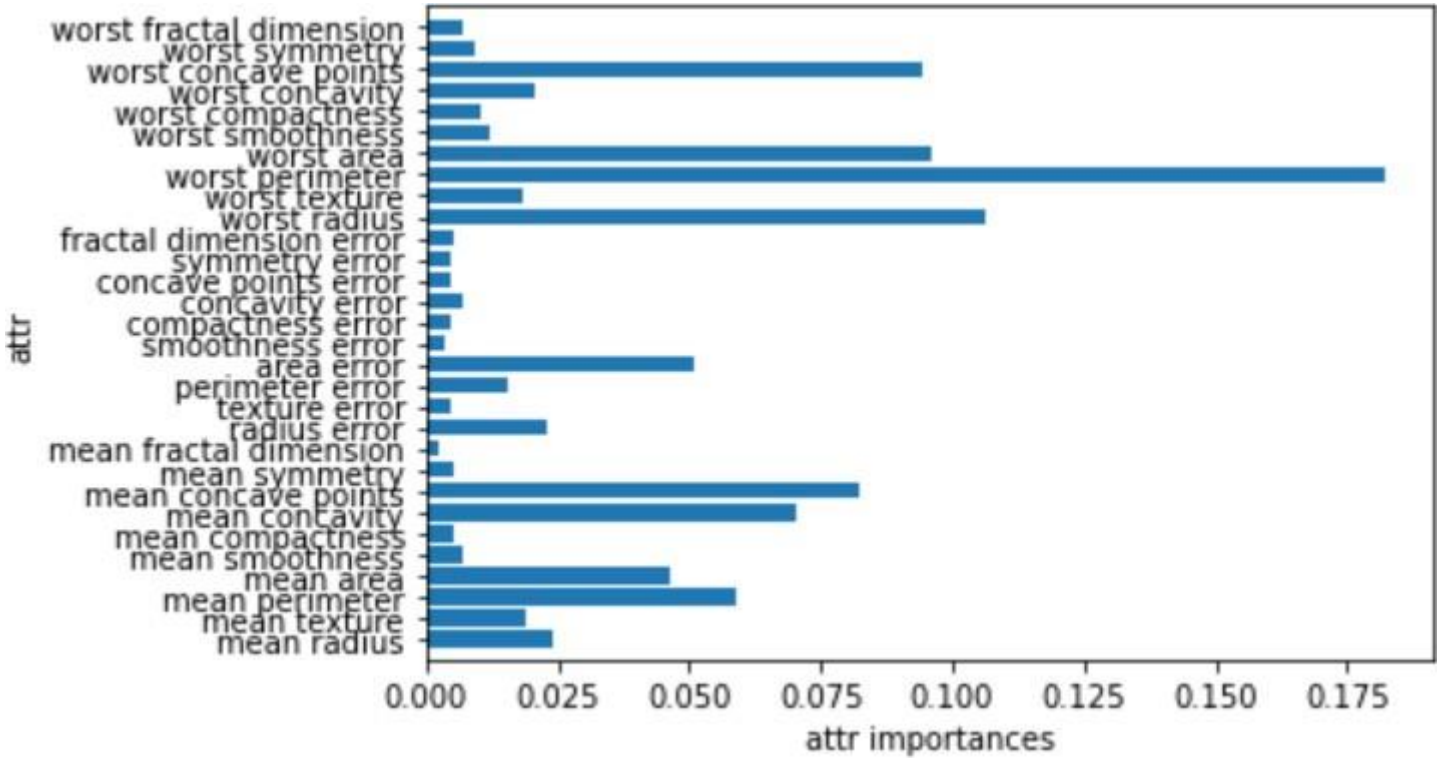
유방암 데이터를 100개의 트리로 만들어
RandomForest 모델로 학습해보자

feature_importance_

- 특성의 중요도를 관리
- Feature Selection에 활용

Ensemble Model - RandomForest

유방암 데이터에서 특징의 중요도를 feature_importance_를 활용하여 bar차트로 표시해보자

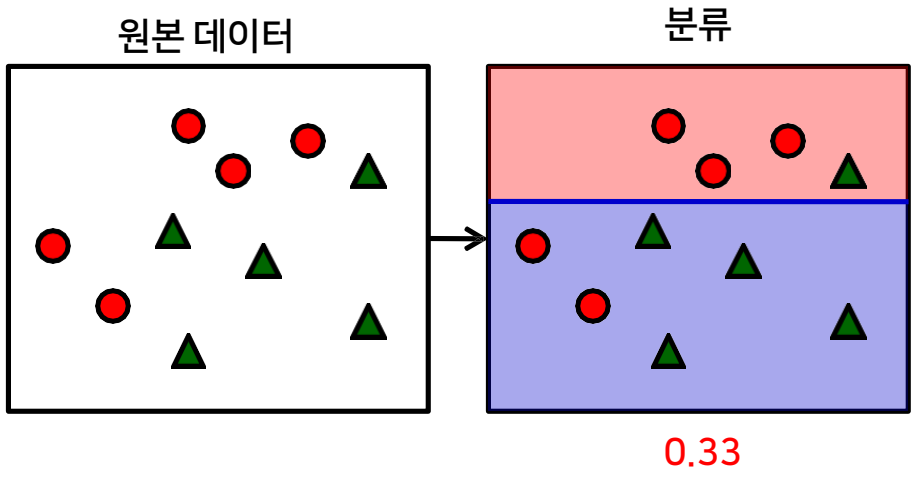


AdaBoost (Adaptive Boosting)

- RF처럼 의사결정 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

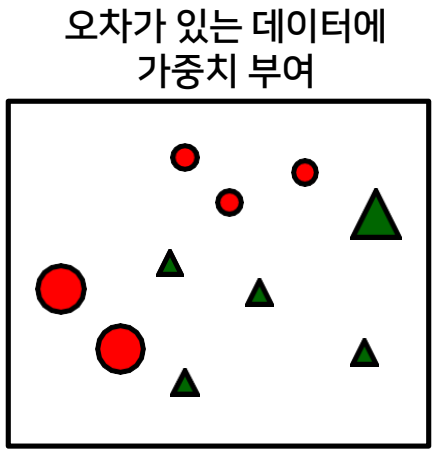
→ 2개의 빨간 원과 1개의 녹색 세모가 잘못 구분됨



AdaBoost (Adaptive Boosting)

- 동작 순서

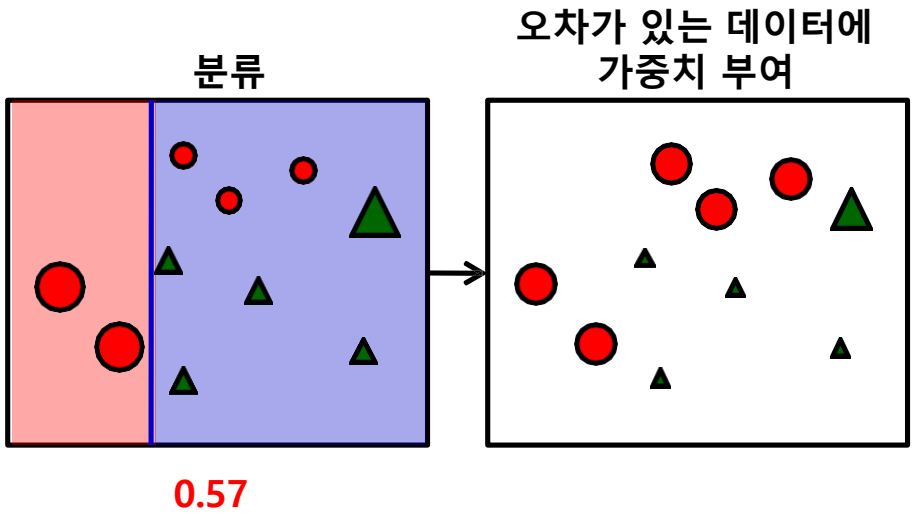
(2) 잘못된 2개의 빨간 원과 1개의 녹색 세모에 높은 가중치를 부여하고 맞은 것에는 빨간 원 3개와 녹색 세모 4개는 낮은 가중치 부여



AdaBoost (Adaptive Boosting)

- 동작 순서

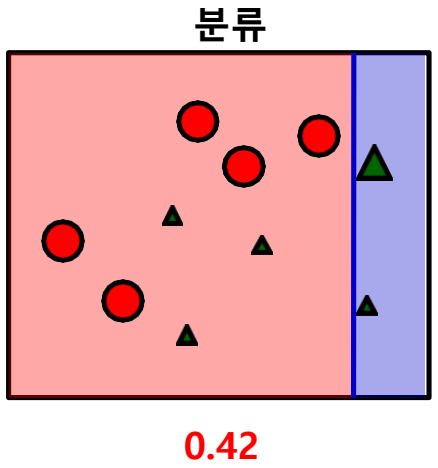
(3) 가중치를 부여한 상태에서 다시 분류 시킴 → 잘못된 3개의 빨간 원에 높은 가중치를 부여하고 맞은 5개의 녹색 세모는 낮은 가중치를 부여



AdaBoost (Adaptive Boosting)

- 동작 순서

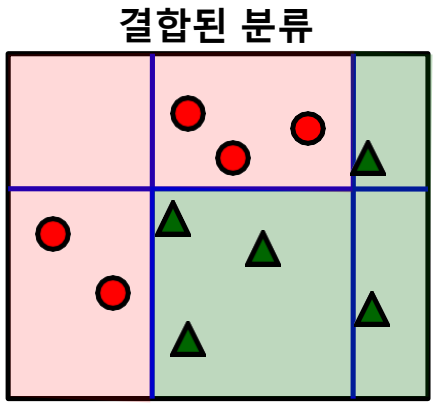
(4) 가중치를 부여한 상태에서 다시 분류 시킴



AdaBoost (Adaptive Boosting)

- 동작 순서

(5) 진행한 분류들을 결합한다.

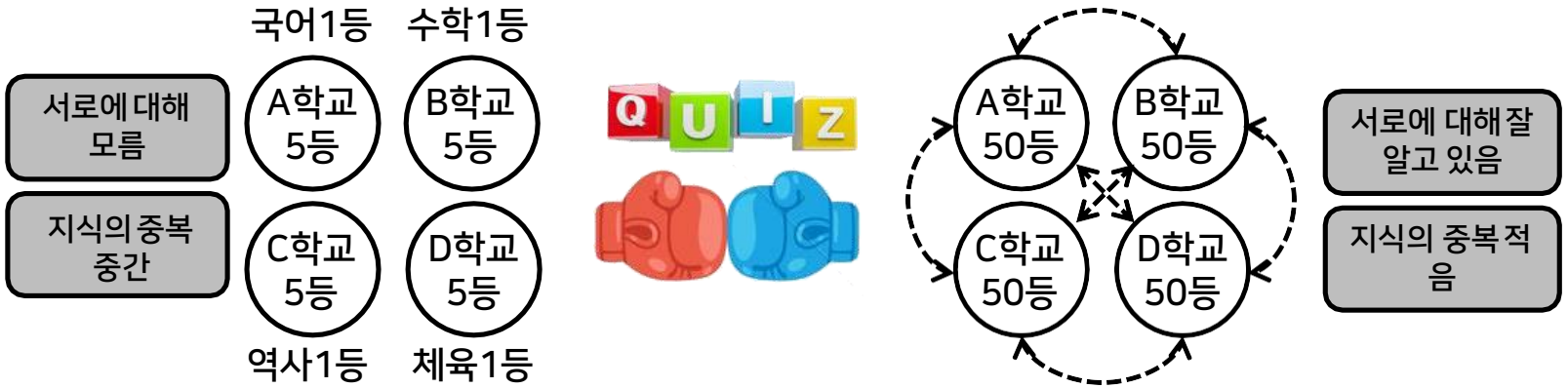


Ensemble Model - AdaBoost

스마트인재개발원
Smart Human Resources Development

Decision Trees vs Random Forest

Random Forest vs AdaBoost



http://medium.com/@deepvalidation/군중은-똑똑하다2-adaboost-ba0122500034 정리

주요 매개변수(Hyperparameter)

scikit-learn의 경우

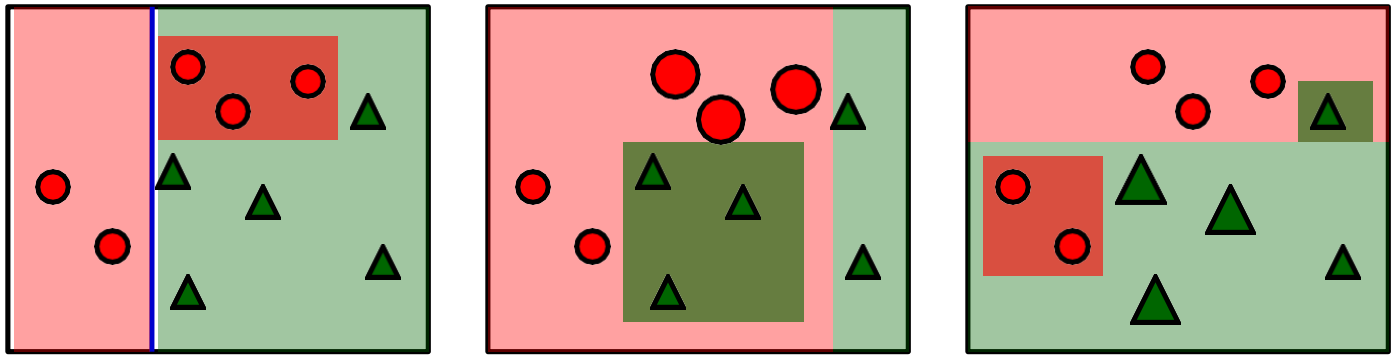
```
AdaBoostClassifier(n_estimators,
random_state)
```

- 트리의 개수 : n_estimators
- 선택할 데이터의 시드 : random_state

유방암 데이터를
Adaboost 모델로 학습해 보고
특징의 중요도를 feature_importance_를
활용하여 bar차트로 표시해보자

Ensemble Model - GBM (Gradient Booting Machine)

- AdaBoost와 기본 개념이 동일하고 가중치를 계산하는 방식에서 경사하강법를 이용하여 최적의 가중치(파라미터)를 찾아냄



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

Ensemble Model - RandomForest

장단점

보통 트리의 깊이를 깊게하지 않기 때문에 예측 속도는 비교적 빠르다.

이전 트리의 오차를 반영해서 새로운 트리를 만 들기 때문에 학습속도가 느리다.

특성의 스케일을 조정하지 않아도 된다.

희소한 고차원 데이터에는 잘 동작하지 않는다

머신러닝의 성능을 마지막까지 쥐어짜야 할 때 활용

Ensemble Model - GBM (Gradient Booting 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 Booting Machine)

유방암 데이터를
GBM 모델로 학습해 보고
특징의 중요도를 feature_importance_를
활용하여 bar차트로 표시해보자

Ensemble Model - Xgboost

GBM의 단점 : 느림, 과대적합 문제

대규모 머신러닝 문제에 그래디언트 부스팅을 적용하려면 xgboost 패키지를 사용 → 분산환경을 고려

GBM보다 빠름 → Early Stopping 제공

과대적합 방지를 위한 규제 포함

CART (Classification And Regression Tree)을 기반으로 함 → 분류 와 회귀가 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 학습해 보고
특징의 중요도를 feature_importance_를
활용하여 bar차트로 표시해보자