Statistical Machine Learning

7주차

담당: 15기 염윤석

1. What is Ensemble?

2. Ensemble Methods

3. Ensemble Models

1. What is Ensemble Learning?

Ensemble

Ensemble learning

- 다수의 기본 분류 모델(base classifier, weak classifier)의 예측 결과를 종합하여, 정확한 예측 성능을 얻도록 하는 방법론

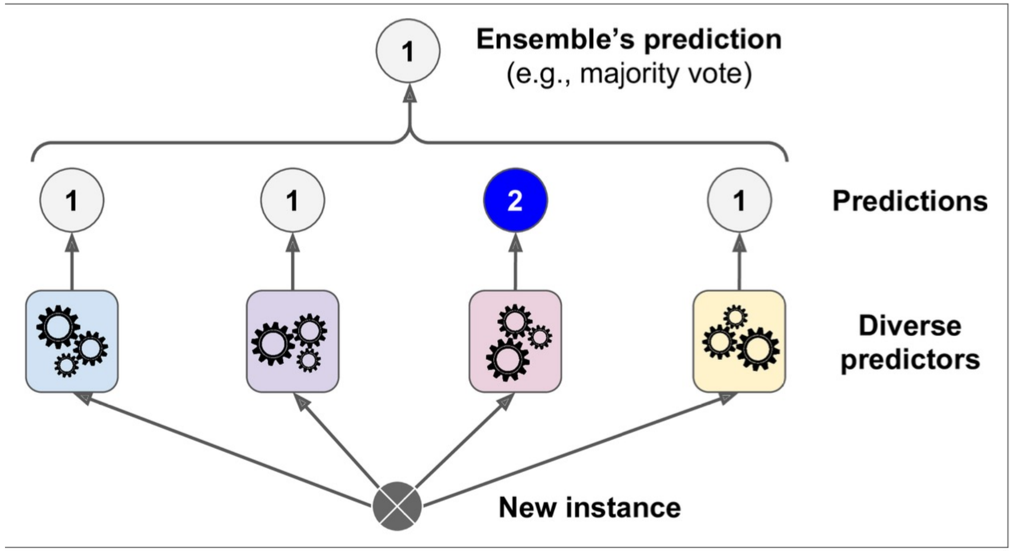
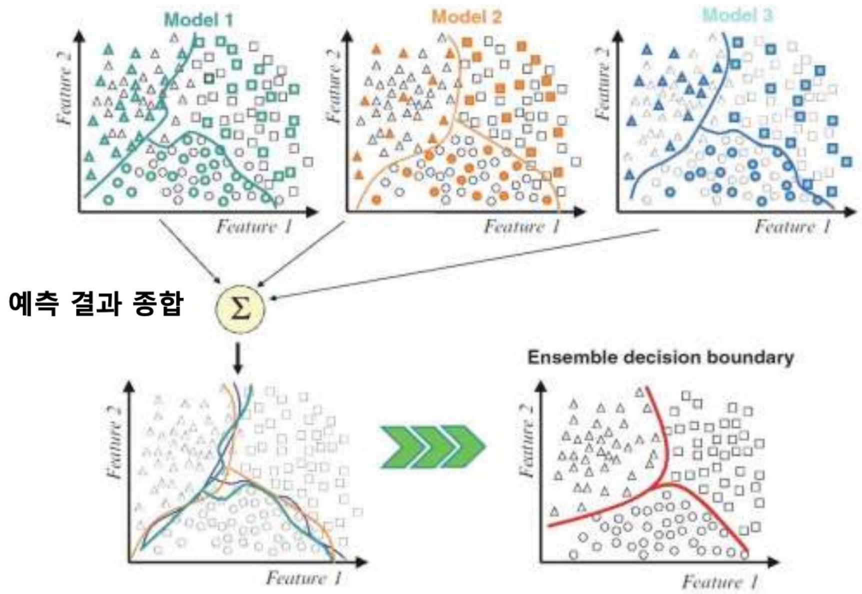


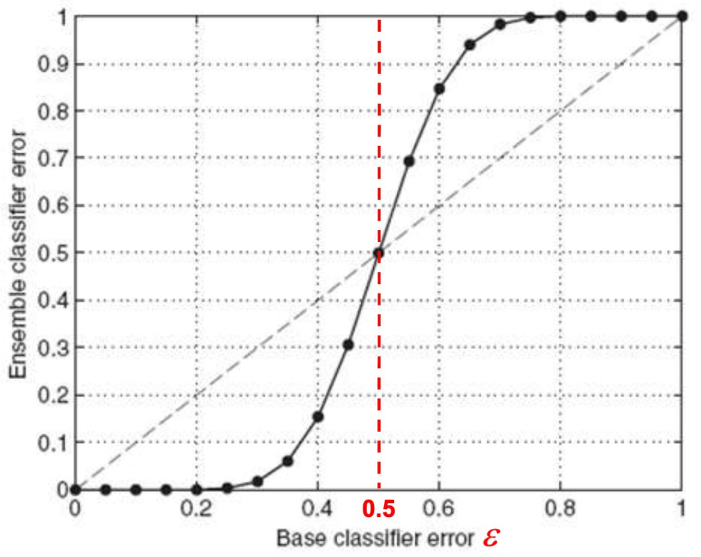
Figure 7-2. Hard voting classifier predictions



Ensemble

Example

- 25 base classifiers
- Error rate $\varepsilon = 0.35$
- Each independent
- Ensemble classifier : Majority vote



$$e_{ensemble} \sim Binomial(25, 0.35)$$

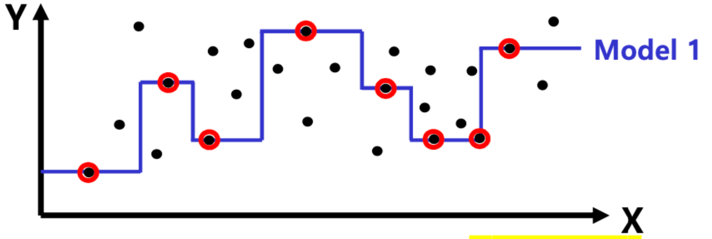
$$P(incorrect\ classifier\ \geq 13) = e_{ensemble} = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$

Probability that the ensemble classifier makes wrong prediction

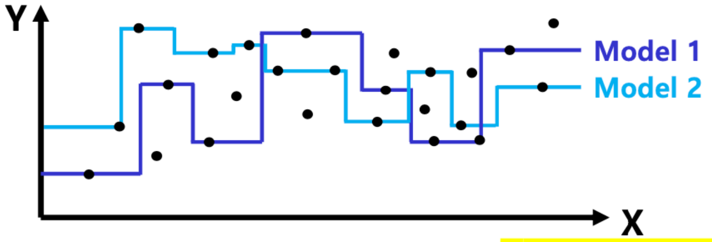
Ensemble

Ensemble Learning

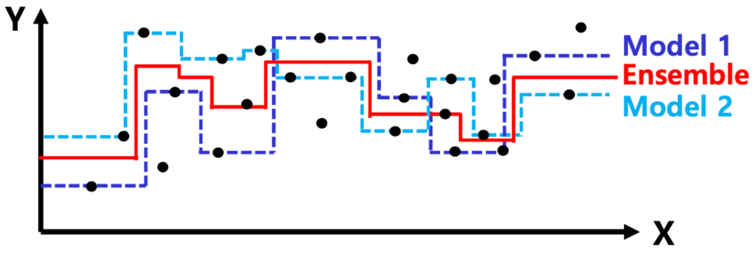
- Reduce Learning error
- Reduce Bias
- Reduce Variance



Sample 데이터에 대해 잘 예측 " Low bias "



Model 1과 Model 2 예측 값이 서로 다름 " High variance "



Model 1과 Model 2 예측 값의 평균 사용: Ensemble

"Low variance"

Ensemble

Voting

Hard Voting

Soft Voting

Weighted Voting

Stacking

Meta level Learning

Blending

Bagging

Bootstrap + Aggregating

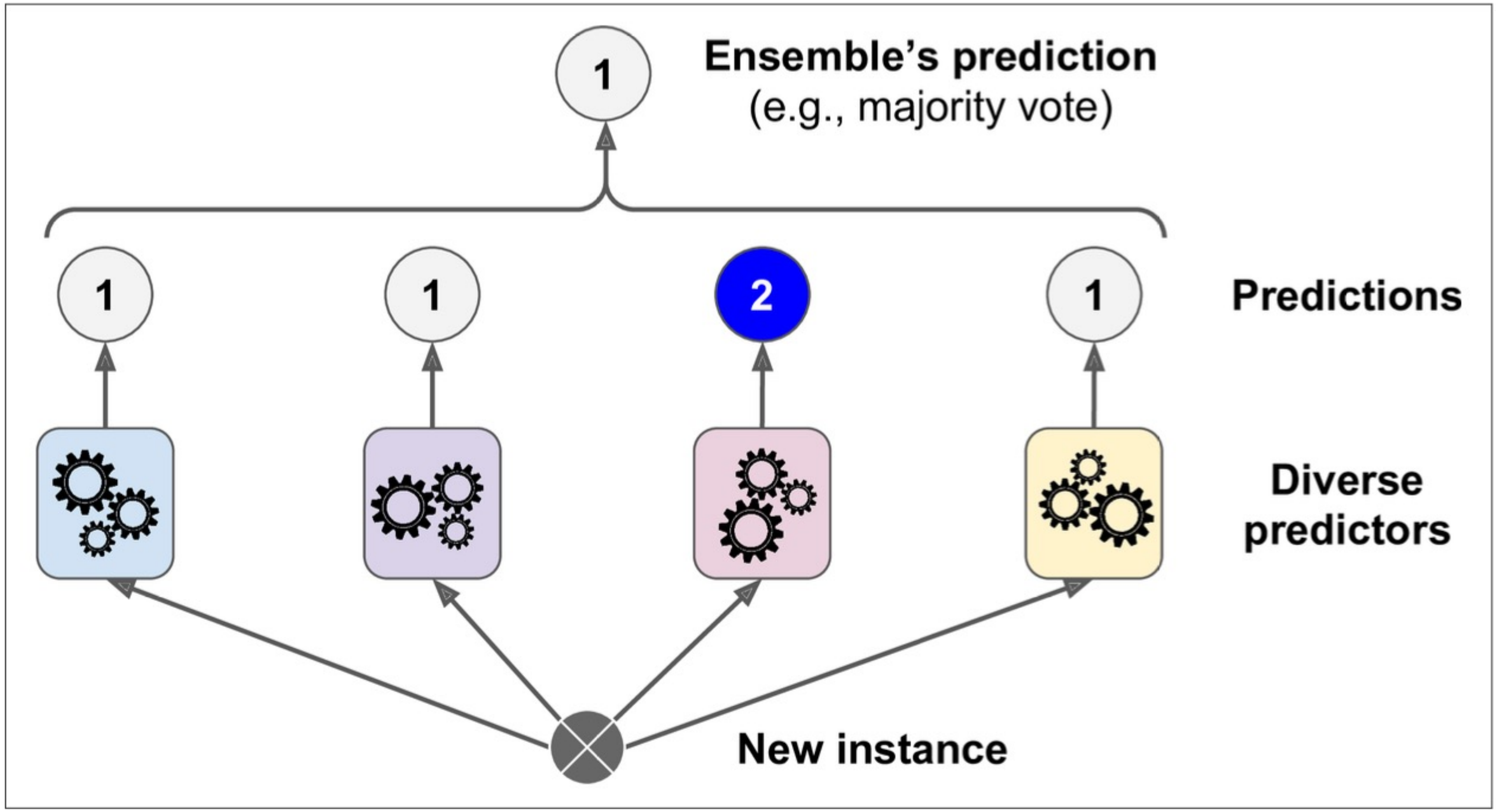
Boosting

Error learner

2. Ensemble Methods

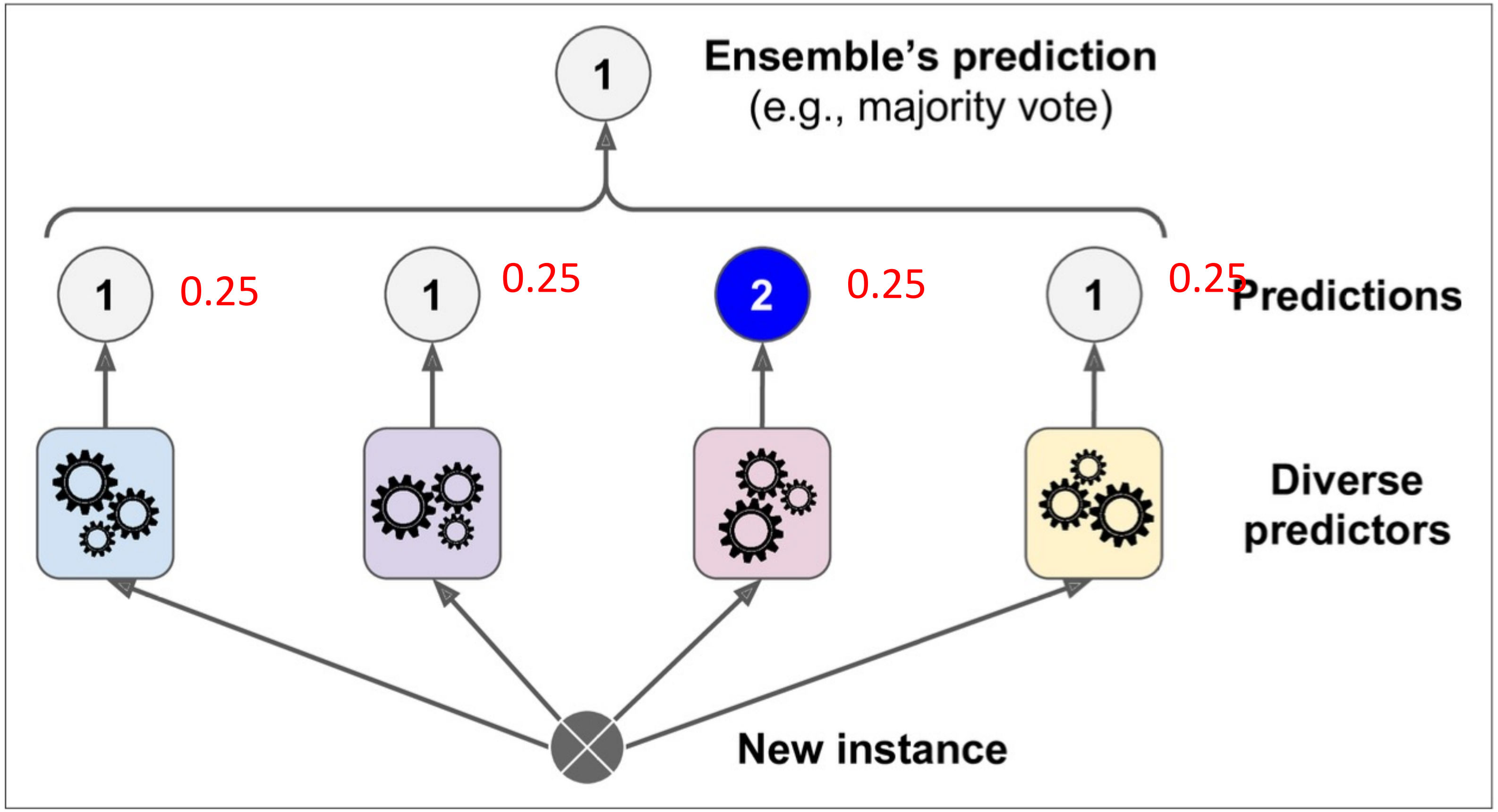
Voting

Hard Voting : Majority Voting



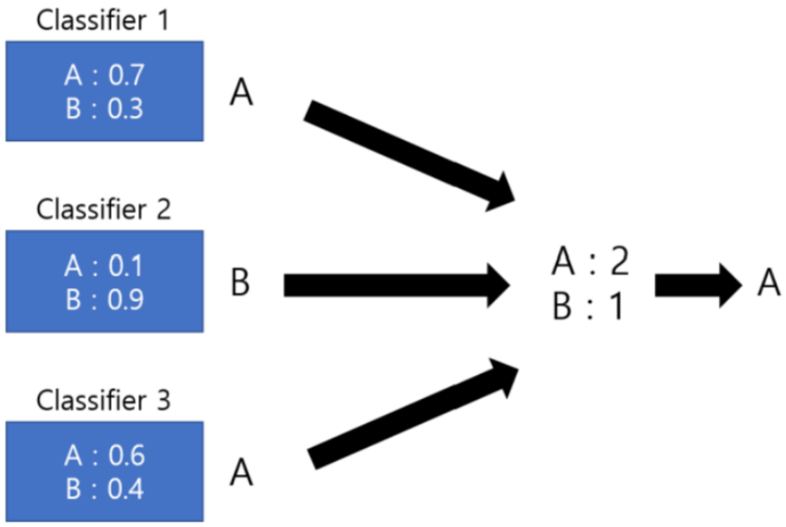
Voting

Soft Voting : Average Voting

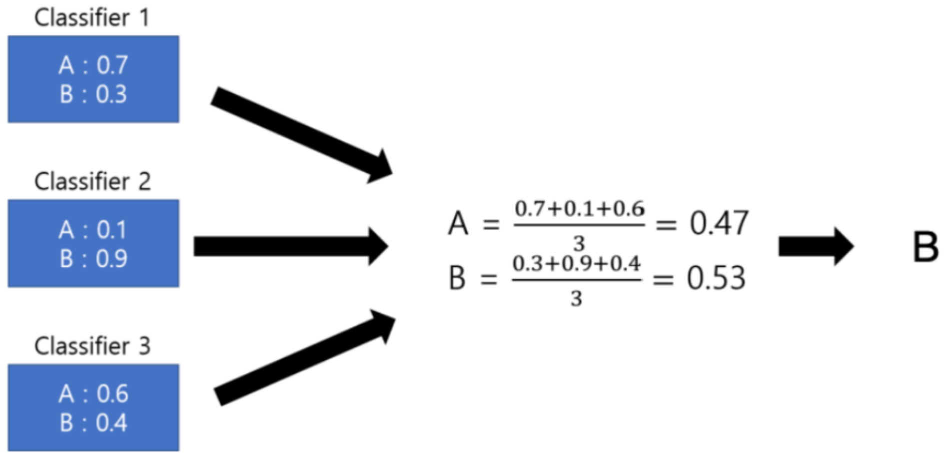


Voting

Hard Voting

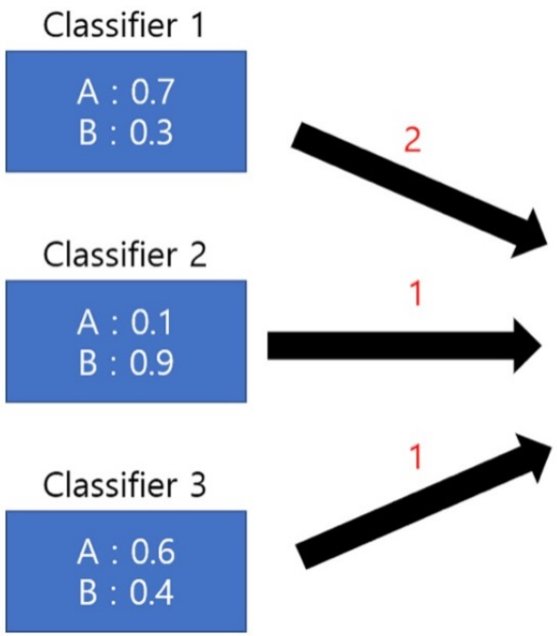


Soft Voting



Voting

Soft + Weighted



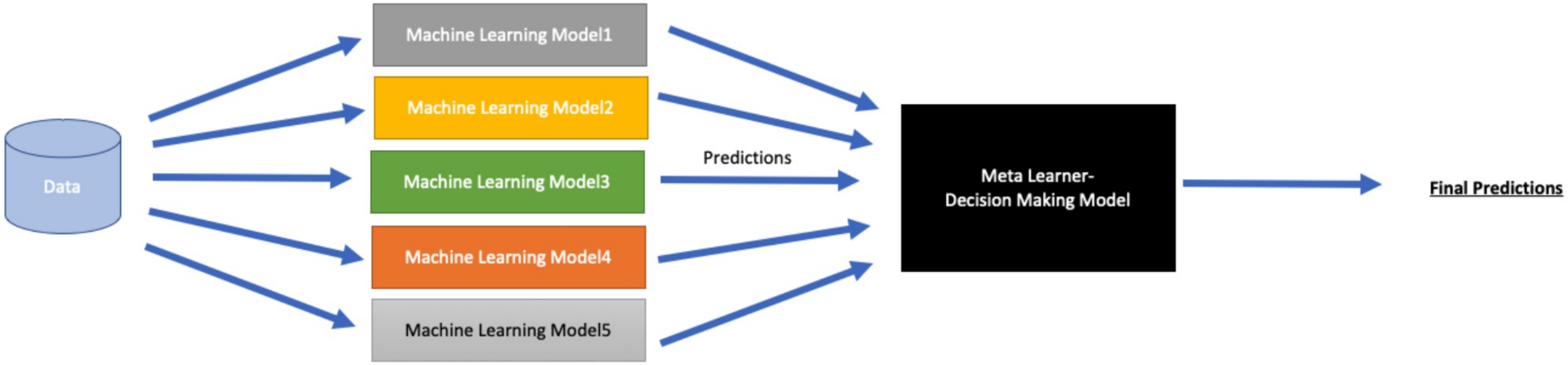
$$A = \frac{0.7\times2+0.1\times1+0.6\times1}{4} = 0.53$$

$$B = \frac{0.3\times2+0.9\times1+0.4\times1}{3} = 0.47$$

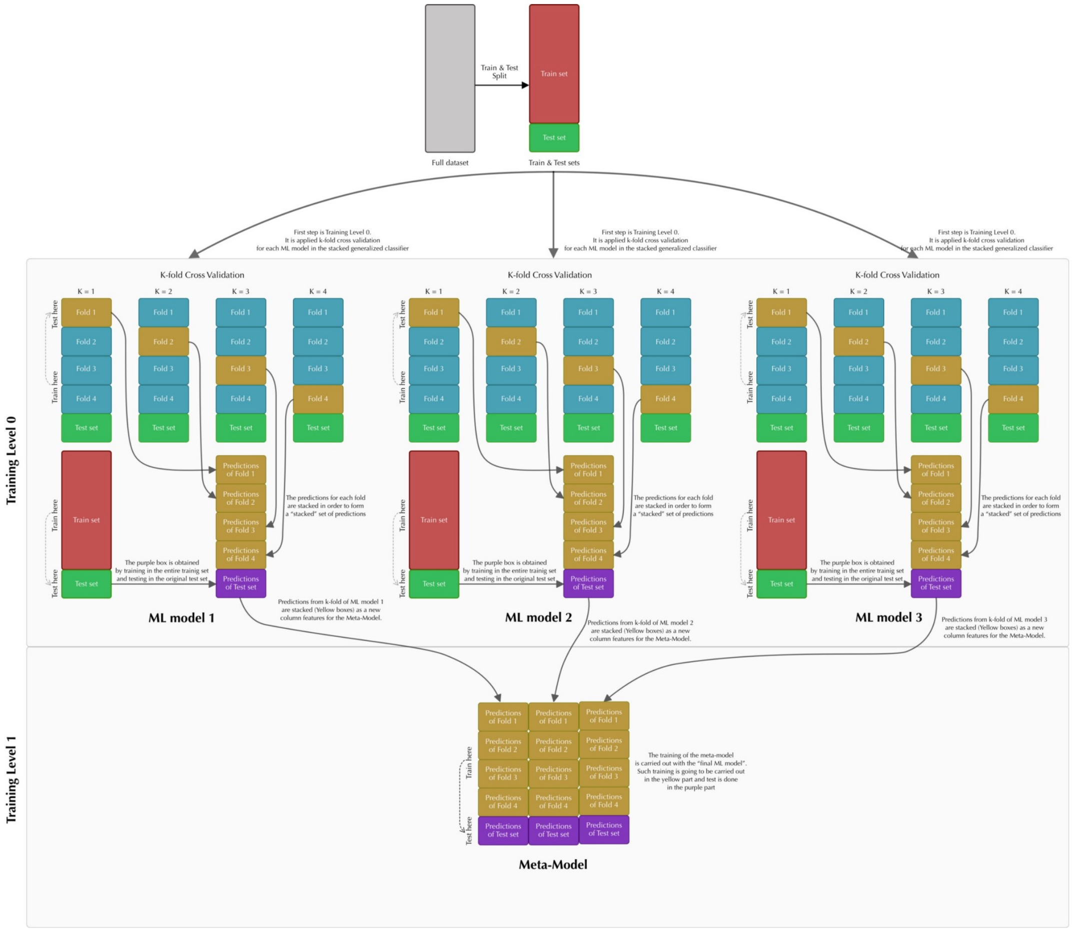
Stacking

Stacking Generalization

- Meta-learning model
- 개별 모델의 예측값을 다시 input으로 사용
- K-fold cv
- Step 0 : 각 weak model에 k-fold cv를 적용하여 예측 데이터를 형성
- Step 1 : step 0에서 만든 예측 데이터를 stack.하여 meta-model을 train 및 예측



Stacking



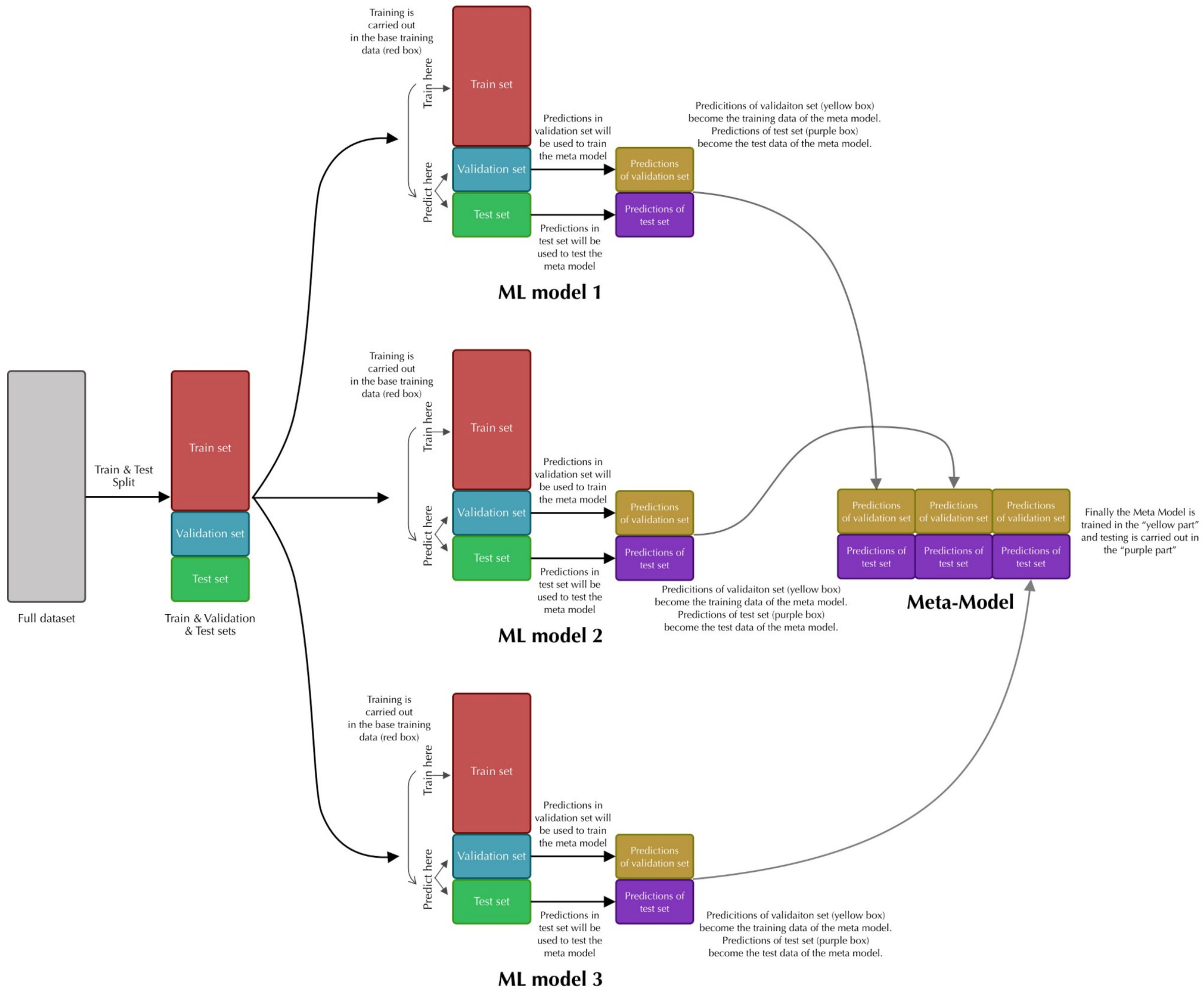
Full dataset
Train & Test Split
Train set
Test set
Train & Test sets
First step is Training Level 0. It is applied k-fold cross validation for each ML model in the stacked generalized classifier
Training Level 0
K-fold Cross Validation
K = 1
K = 2
K = 3
K = 4
Fold 1
Fold 2
Fold 3
Fold 4
Test here
Train here
Predictions of Fold 1
Predictions of Fold 2
Predictions of Fold 3
Predictions of Fold 4
Predictions of Test set
The predictions for each fold are stacked in order to form a "stacked" set of predictions
The purple box is obtained by training in the entire training set and testing in the original test set
ML model 1
ML model 2
ML model 3
Predictions from k-fold of ML model 1 are stacked (Yellow boxes) as a new column features for the Meta-Model.
Predictions from k-fold of ML model 2 are stacked (Yellow boxes) as a new column features for the Meta-Model.
Predictions from k-fold of ML model 3 are stacked (Yellow boxes) as a new column features for the Meta-Model.
Training Level 1
The training of the meta-model is carried out with the "final ML model". Such training is going to be carried out in the yellow part and test is done in the purple part
Meta-Model

Blending

Blending Generalization

- Meta-learning model
- 개별 모델의 예측값을 다시 input으로 사용
- One-hold out

Blending


Full dataset
Train & Test Split
Train set
Validation set
Test set
Train & Validation & Test sets
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions of validation set
Predictions of test set
Predictions in test set will be used to test the meta model
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 1
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions of validation set
Predictions of test set
Predictions in test set will be used to test the meta model
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 2
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions of validation set
Predictions of test set
Predictions in test set will be used to test the meta model
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 3
Predictions of validation set
Predictions of validation set
Predictions of validation set
Predictions of test set
Predictions of test set
Predictions of test set
Meta-Model
Finally the Meta Model is trained in the "yellow part" and testing is carried out in the "purple part"

Bagging

Bagging = Bootstrap + Aggregating(Average)

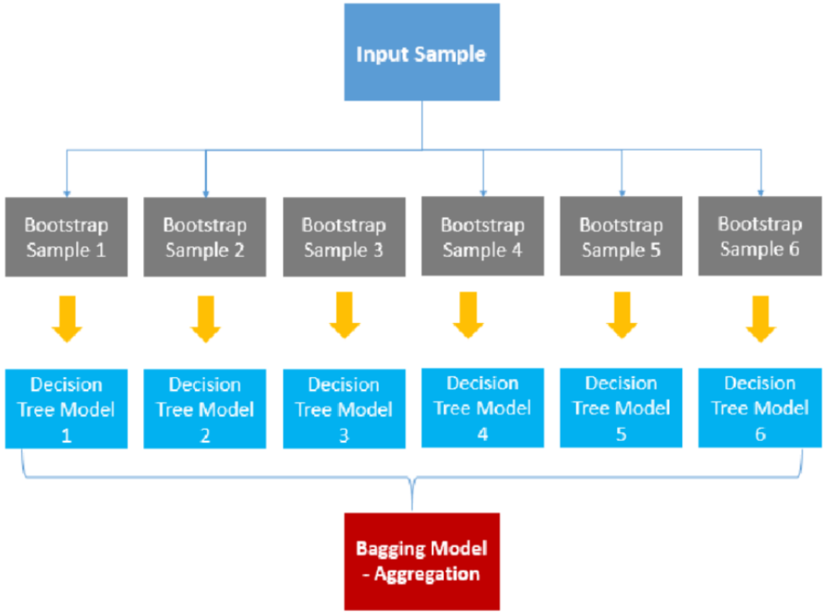
Bootstrap : sampling with Replacement → Variance 개선

Probability that a record is chosen by bootstrap
(N records, N sample size)

$= 1 - (1 - \frac{1}{N})^N$

If N is large enough, then $\lim_{N \to \infty} 1 - \left(1 - \frac{1}{N}\right)^N = 1 - e^{-1} = 0.632$

63.2% of original train dataset



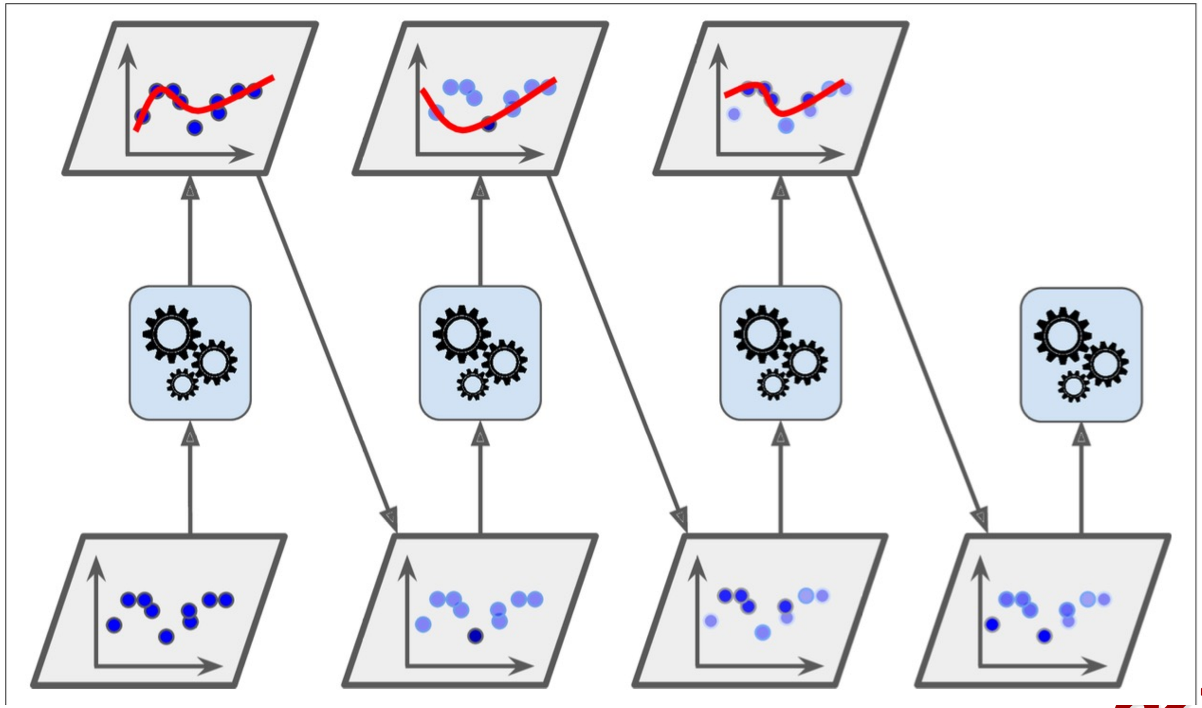
Aggregating : Majority Voting, Weighting, Soft voting

Boosting

Boosting

- 오답을 다시 학습
- 예측이 틀린 데이터가 다시 뽑힐 가중치가 높아진다.
- 이전 모델이 잘못 예측한 부분을 집중적으로 학습

→ Bias 개선



Bagging & Boosting

<Normal>	<Bagging>	<Boosting>
10개년 6,9 수능 문제 1회독	10개년 6,9, 수능 전체 문제에서 랜덤 복원 추출 → 10번 반복	10개년 6,9,수능 전체문제에서 랜덤 복원 추출 이때, 틀린 문제는 반드시 포함해서 추출 → 10번 반복

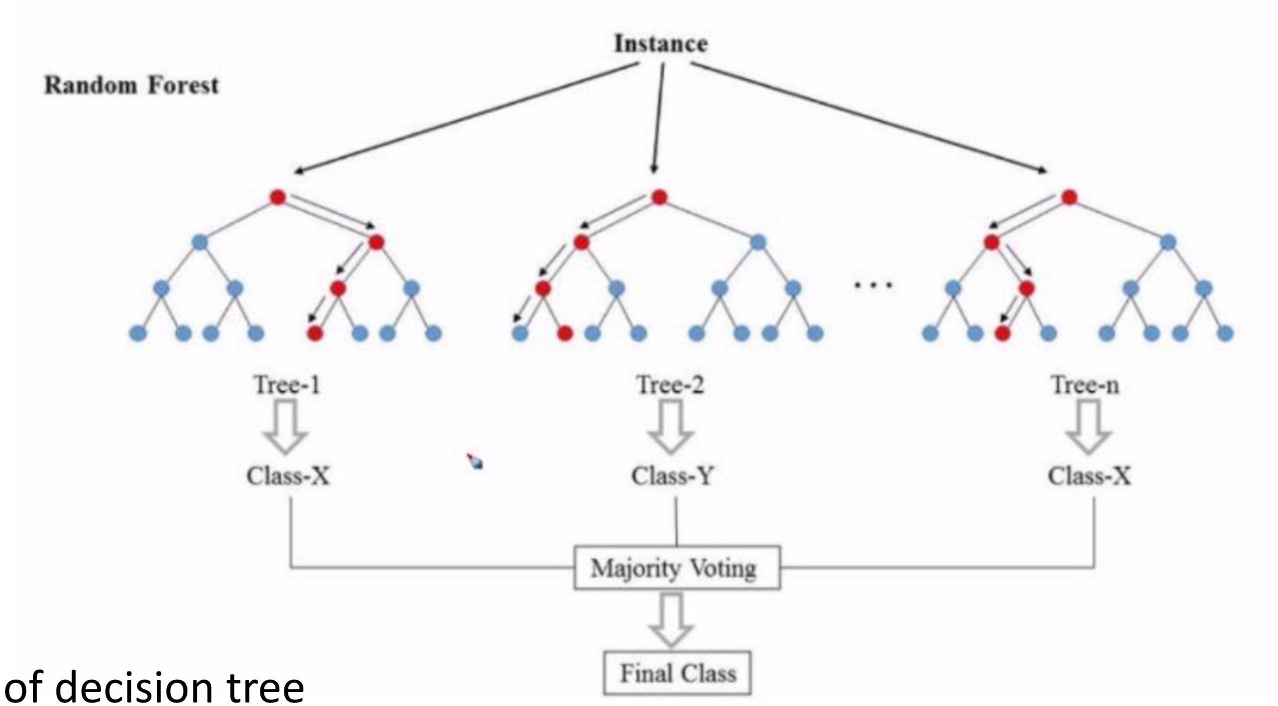
10개년 6,9,수능 문제 1회독
→ 틀린 문제만 뽑아서 다시 1회독
→ 다시 틀린 문제만 뽑아서 1회독
→ 다시 틀린 문제만 뽑아서 1회독
→ (반복)

3. Ensemble Models

RandomForest

Feature Bagging → RandomForest

<u>RandomForest Decision Tree Generation</u>



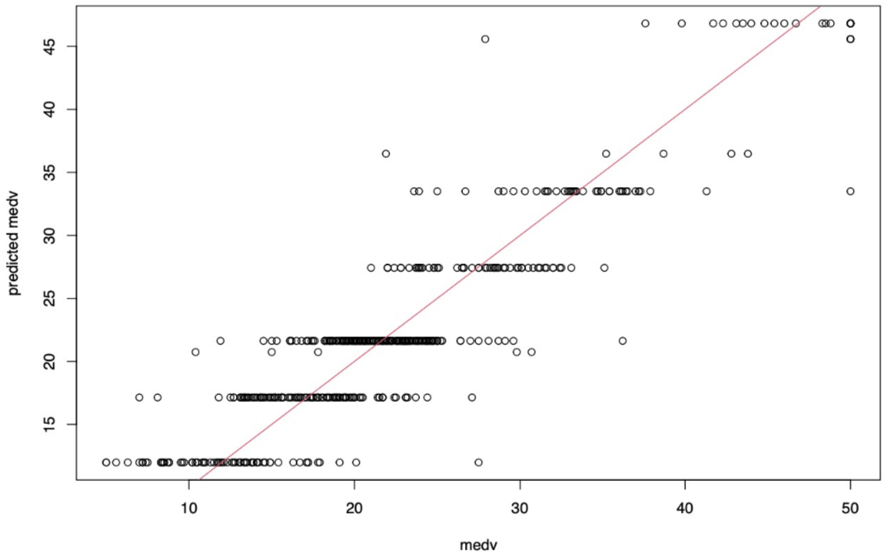
- **Forest-RI(random input)**
  randomly select F features
  to split each node of decision tree

- **Forest-RC(randomly combined)**
  F randomly combined new features
  (F linear combination)

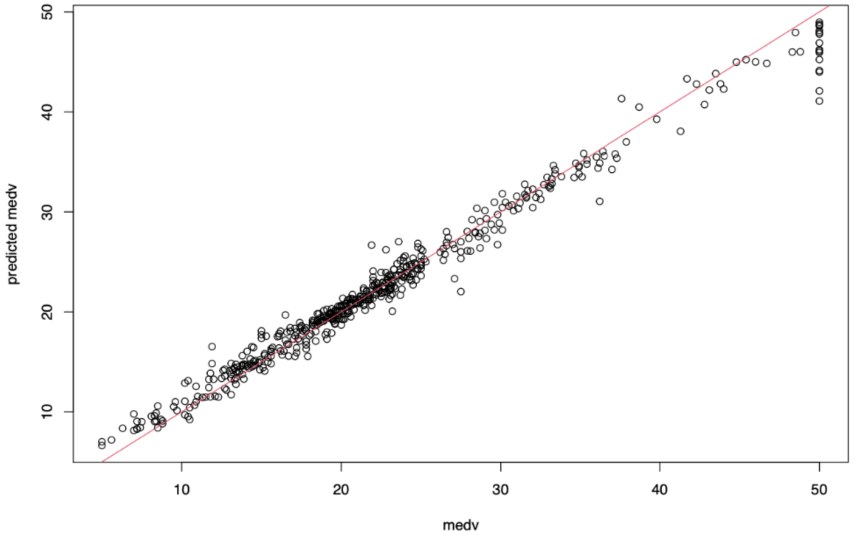
- **Randomly select**
  one of the F best splits at each node of decision tree

RandomForest

Single Tree



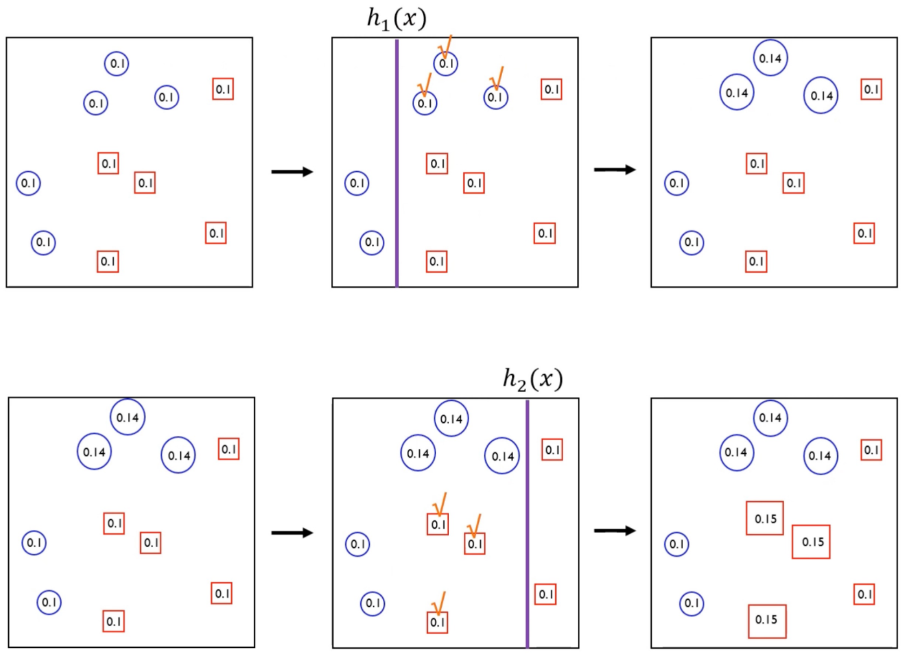
Random Forest



Adaboost

Adaboost : Adaptive + Boosting

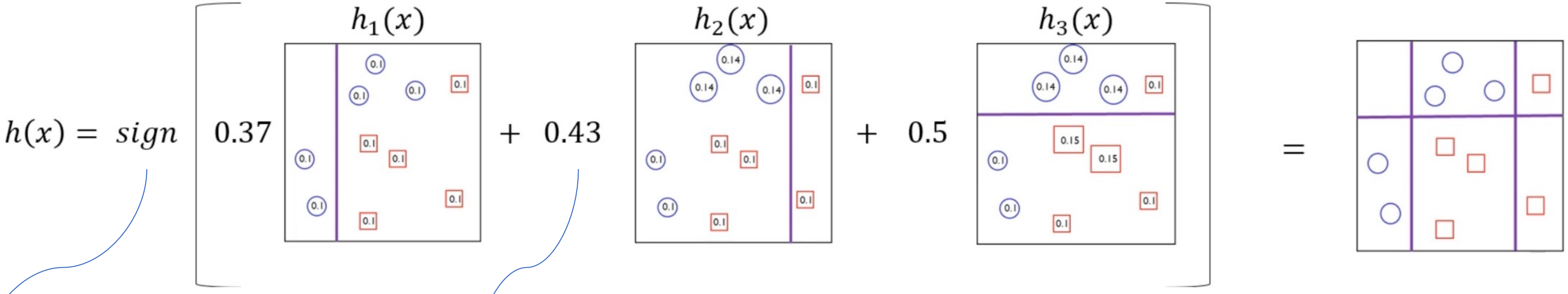
- Adaptive : 이전 모델이 잘못 분류한 데이터의 가중치를 adaptive하게 변경
- Boosting : 이전 모델이 잘못 분류한 데이터들을 중심으로 학습



정분류 sample : 그대로
오분류 sample : 가중치 ↑

Adaboost

$$h(x) = sign\left[\sum_{i=1}^{m=3} \alpha_j h_j(x)\right]$$



aggregating

Model importance based on error rate

Ensemble models

- RandomForest
- ExtraTrees
- Adaboost
- GradientBoost
- XGBoost
- LightGBM
- CatBoost

수고하셨습니다!

해당 세션자료는 KUBIG Github에서 보실 수 있습니다!

Ensemble 및 autoML 패키지 소개가 있습니다.