Machine learning 03

Byung Chang Chung

Gyeongsang National University

bcchung@gnu.ac.kr

Contents

- Decision tree
- Ensemble learning
- Random forest

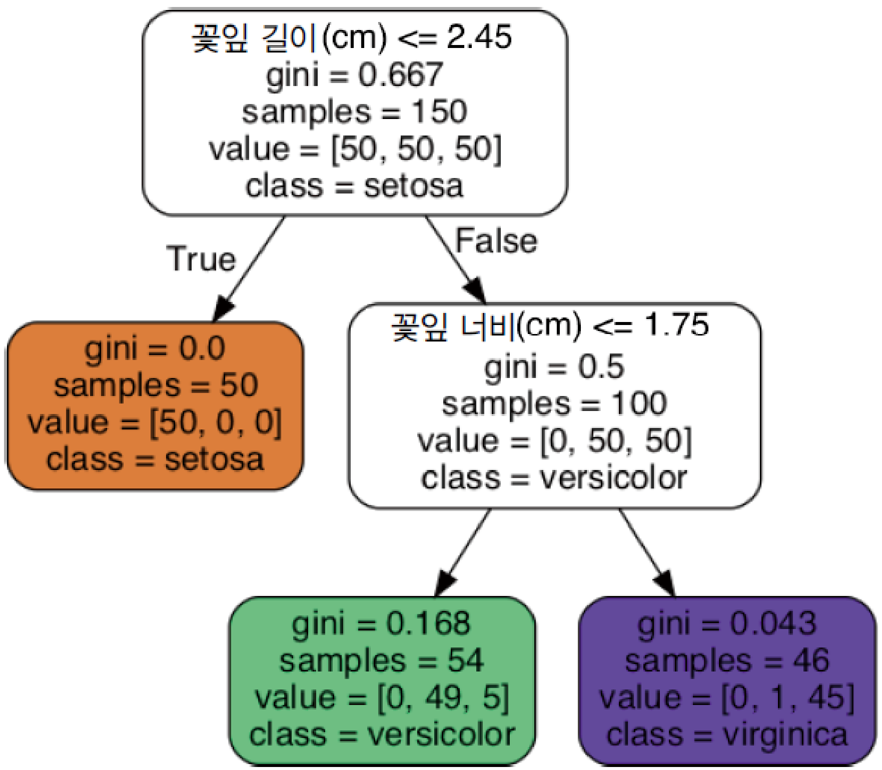
Gyeongsang
National
University
경상국립대학교

Decision tree

- Definition
  - a non-parametric supervised learning method used for classification and regression
  - create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features

Decision tree

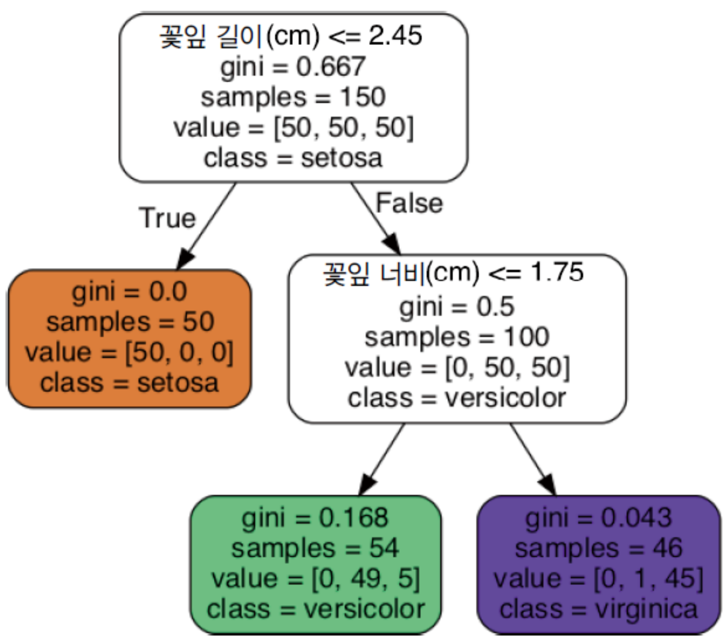
- Example



Gyeongsang National University 경상국립대학교

Decision tree

- How to predict?
  - Starting from root node
  - Sequentially follow the tree



GNU
Gyeongsang
National
University
경상국립대학교

Decision tree

- What is gini?
  - impurity
  - how many different types of samples are in one node?
  - mathematically,

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

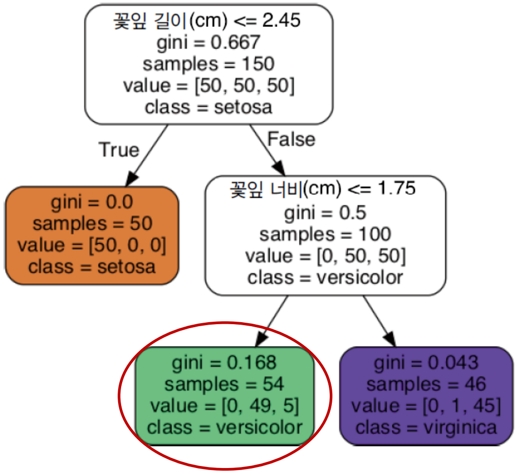
Gyeongsang National University 경상국립대학교

Decision tree

- Other metrics for impurity
  - entropy
  - the value of information

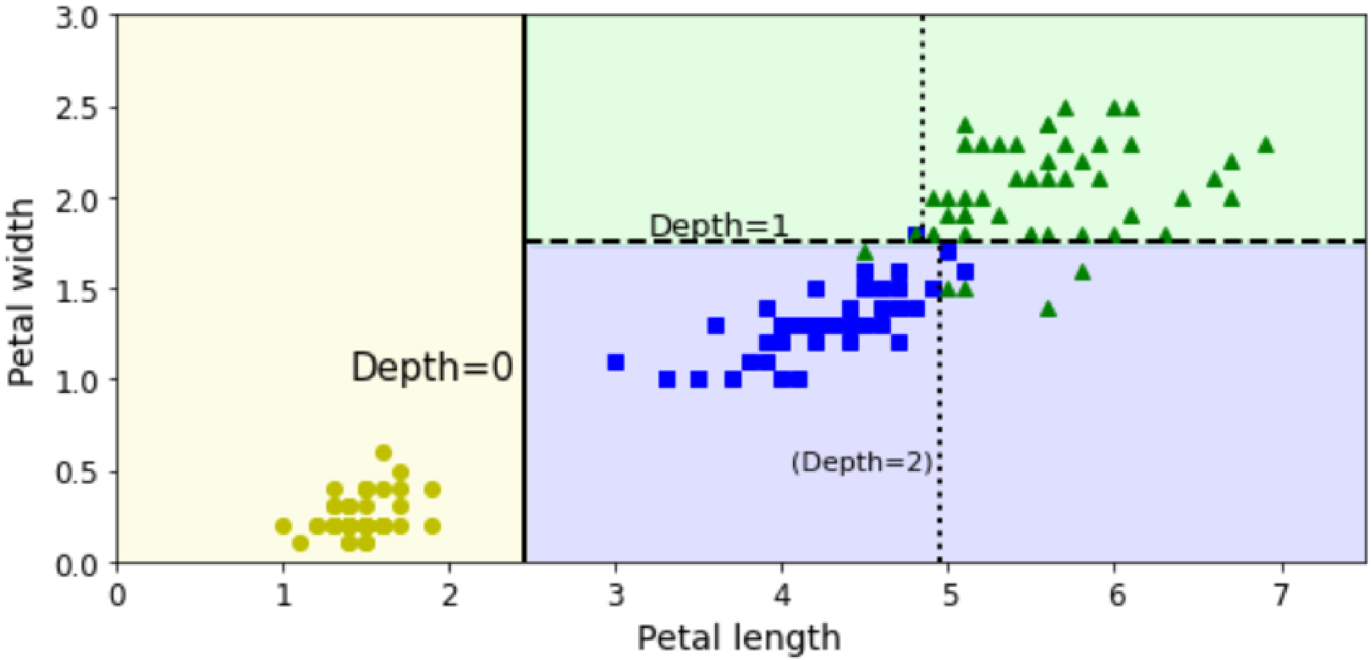
$$H_i = -\sum_{\substack{k=1 \\ p_{i,k}\neq 0}}^{n} p_{i,k}\log_2(p_{i,k})$$

  - for example,
    - $-\frac{49}{54}\log_2\frac{49}{54}-\frac{5}{54}\log_2\frac{5}{54}\cong 0.445$



Decision tree

- Visualization of borderline



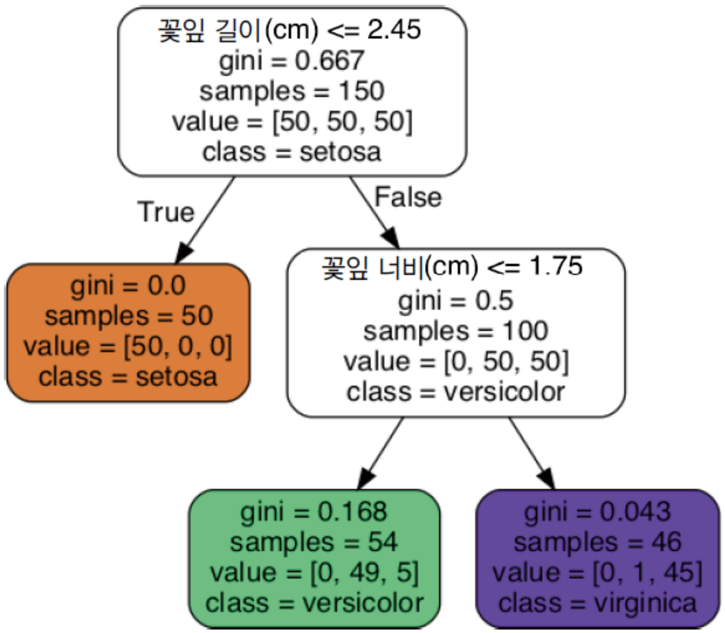
Gyeongsang National University 경상국립대학교

Decision tree

- Whitebox model
  - intuitively understand the outcome of a decision
- Blackbox model
  - cannot explain the outcome of a decision

Decision tree

- Inference for probability




```
tree_clf.predict_proba([[5, 1.5]])
```

```
array([[0.        , 0.90740741, 0.09259259]])
```

```
tree_clf.predict([[5, 1.5]])
```

```
array([1])
```

Gyeongsang National University 경상국립대학교

Decision tree

- CART training algorithm
  - find feature k and its threshold t_k

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

  - notations
    - $G_{\text{left/right}}$: impurity of left/right subset
    - $m_{\text{left/right}}$: numbers of samples of left/right subset

GNU
Gyeongsang
National
University
경상국립대학교

Decision tree

- CART training algorithm
  - find feature k and its threshold t_k which minimizes

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

  - notations
    - $G_{\text{left/right}}$: impurity of left/right subset
    - $m_{\text{left/right}}$: numbers of samples of left/right subset

Gyeongsang National University 경상국립대학교

Decision tree

- Computing complexity
  - for prediction
    - searching from root node to leaf node
    - $O(\log_2 m)$
  - for training
    - comparing all samples
    - $O(n \times m \log_2(m))$

Gyeongsang National University 경상국립대학교

Decision tree

- Regulation
    - no constraint for data properties
    - overfitting when there is no regulation
    - because decision tree is a non-parametric model
        - number of parameters not determined before training

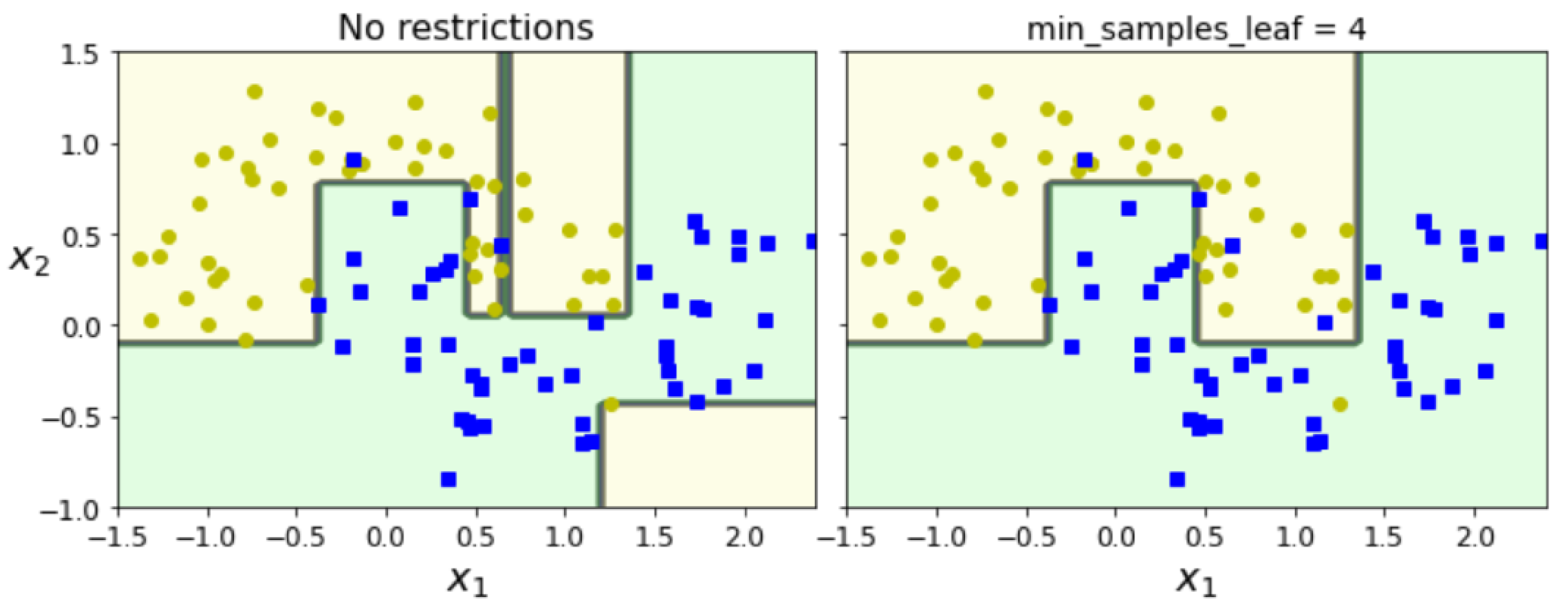
GNU
Gyeongsang
National
University
경상국립대학교

Decision tree

- Parameters for regulation
  - max_depth
  - min_samples_split
  - max_leaf_nodes
  - max_features

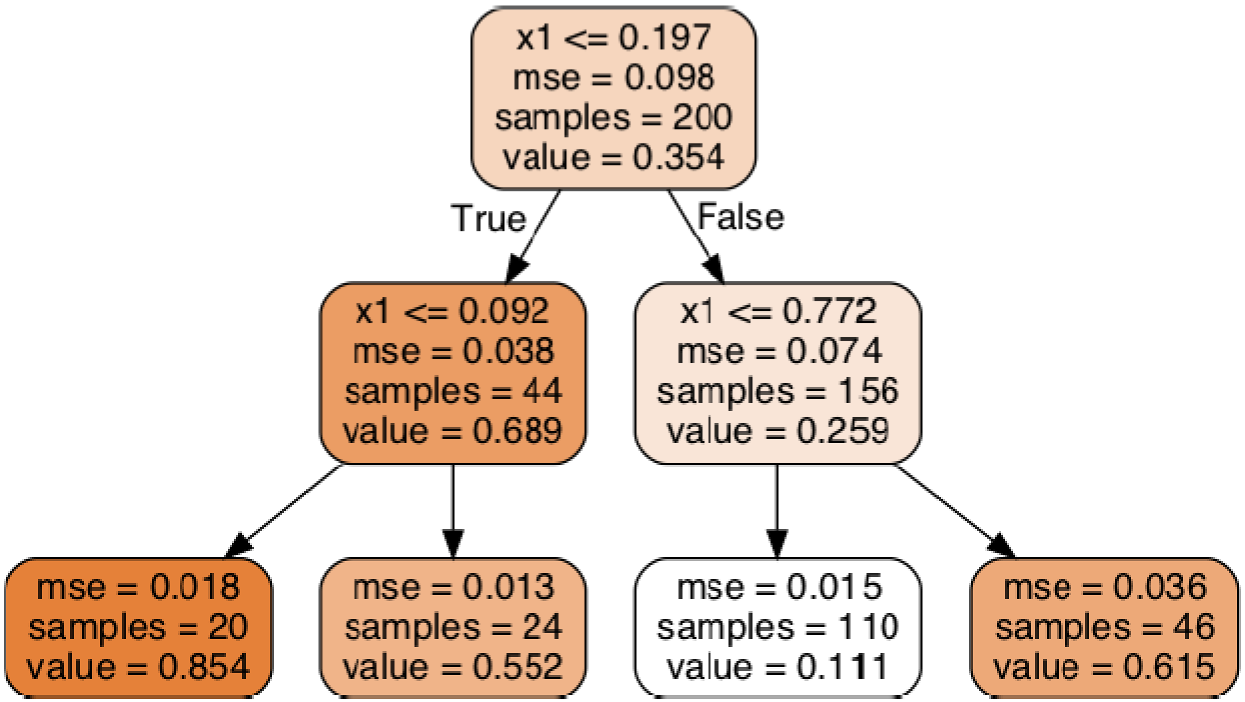
Decision tree

- Example of regulation in decision tree



Decision tree

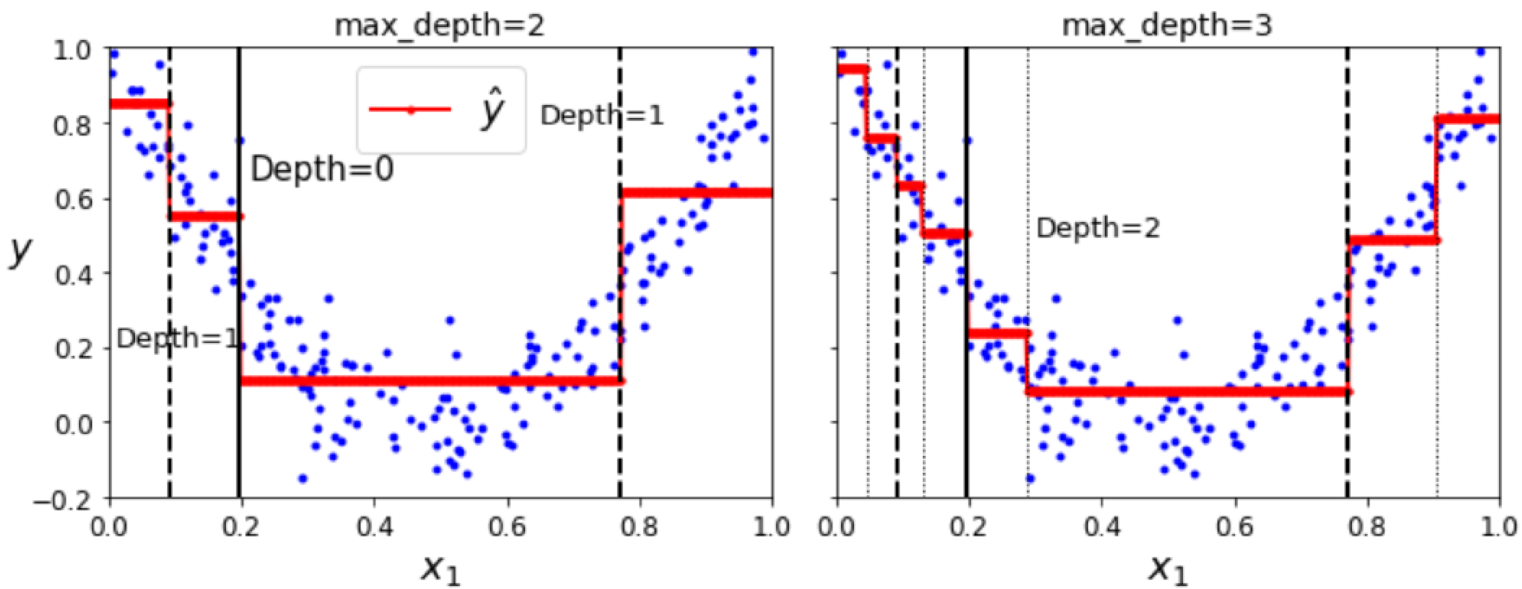
- Decision tree for regression



Gyeongsang National University 경상국립대학교

Decision tree

- Decision tree for regression



Decision tree

- Cost function for regression

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}}$$

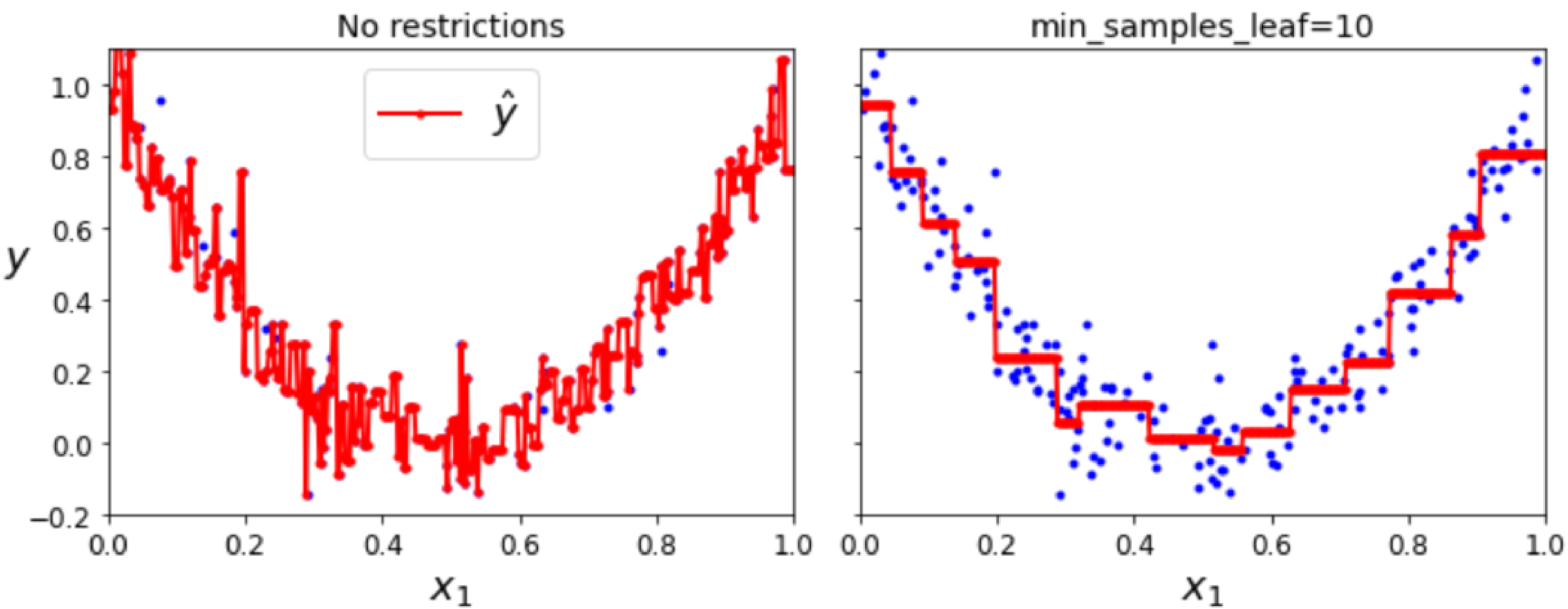
$$\text{MSE}_{\text{left}} = \sum_{i\in\text{node}}(\hat{y}_{\text{node}} - y^{(i)})^2$$

$$\hat{y}_{\text{node}} = \frac{1}{m_{node}}\sum_{i\in\text{node}} y^{(i)}$$

GNU
Gyeongsang
National
University
경상국립대학교

Decision tree

- Importance of regulation

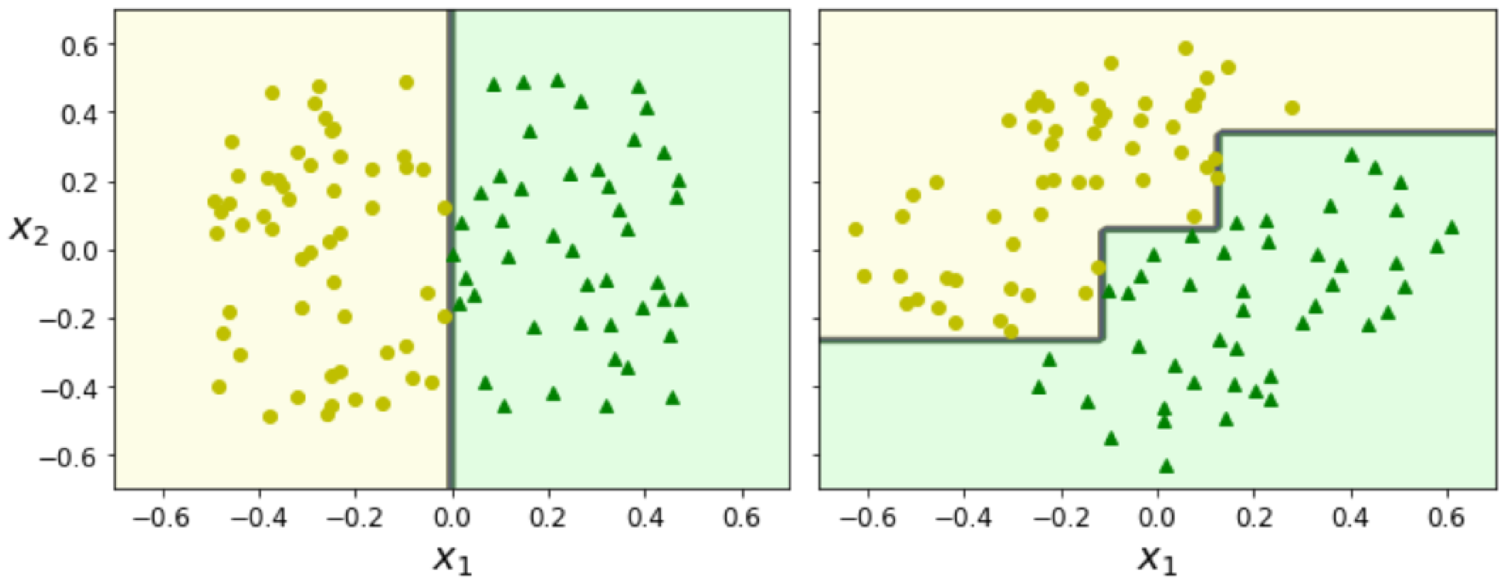


Decision tree

- Instability
  - decision tree makes stair-like line
  - sensitive when datasets are rotated
  - one of method for solving this issue is principal component analysis (PCA will be treated in later chapter)

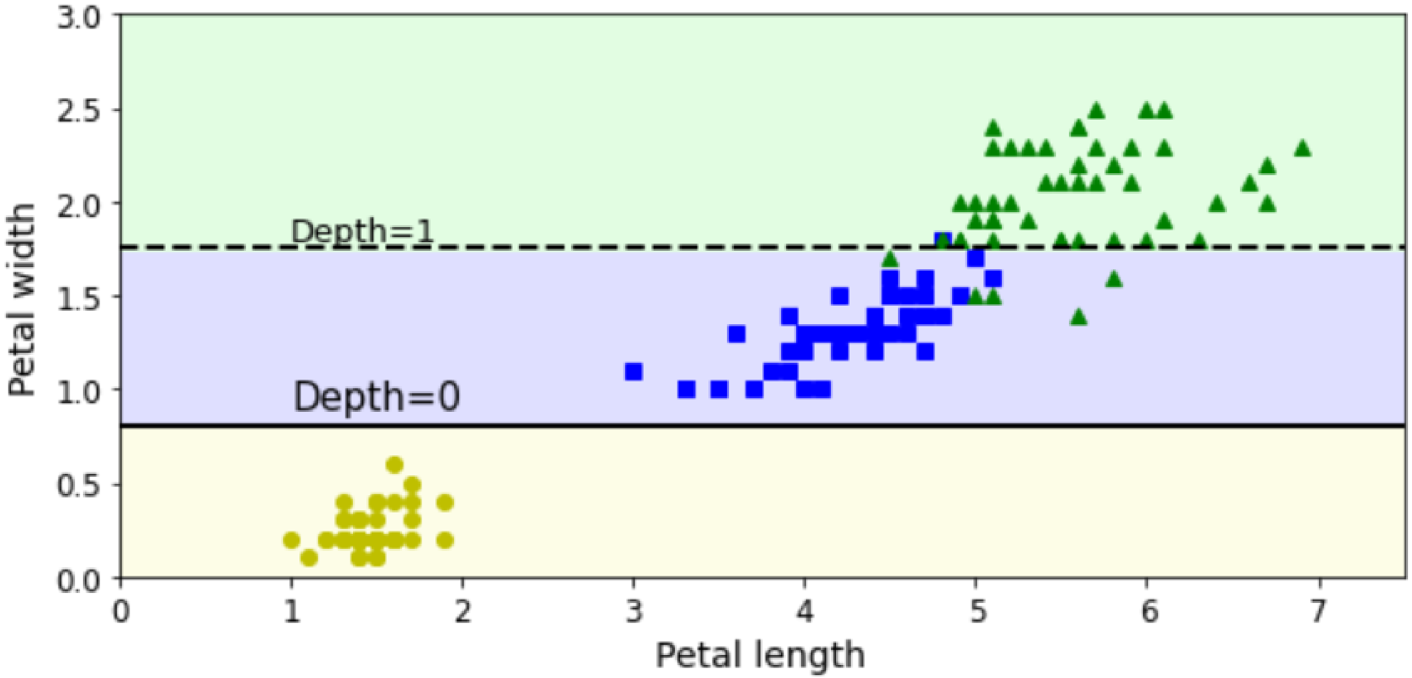
Decision tree

- Visualization of instability (rotation)



Decision tree

- Visualization of instability (data variation)



Gyeongsang National University 경상국립대학교

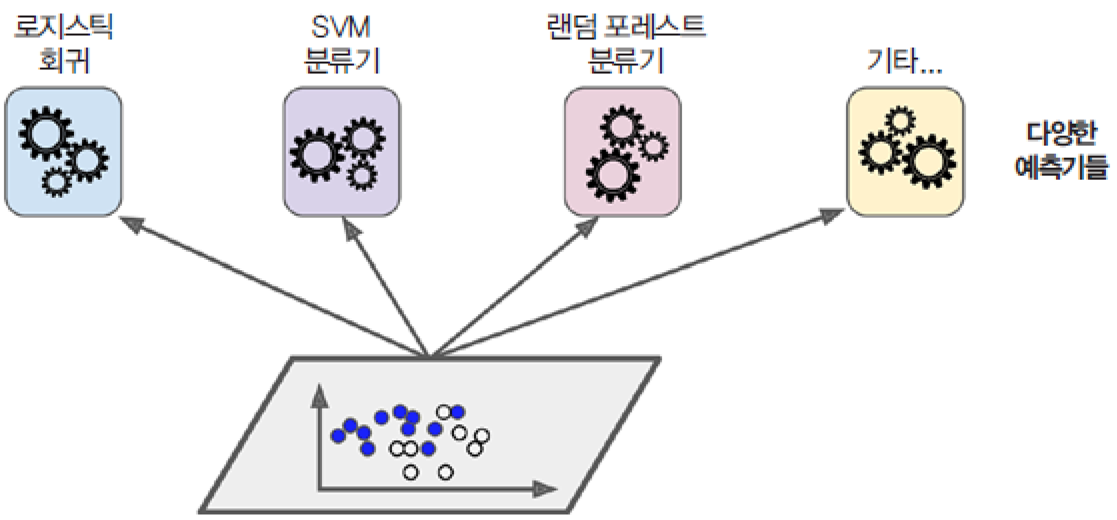
Ensemble learning

- Definition
  - use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone
  - wisdom of the cloud

Gyeongsang National University 경상국립대학교

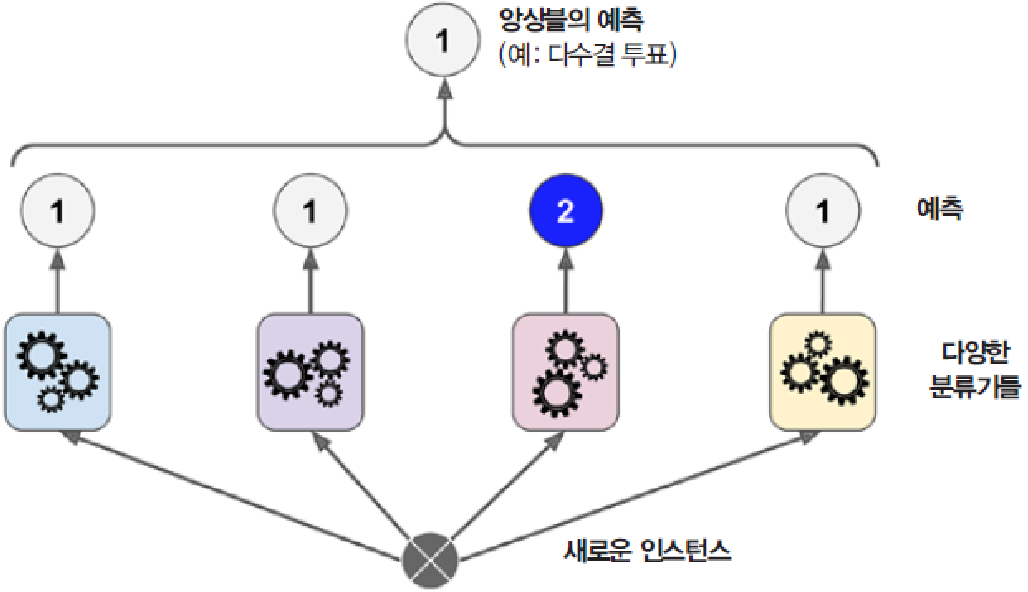
Ensemble learning

- Toy example
  - suppose you have multiple classifiers



Ensemble learning

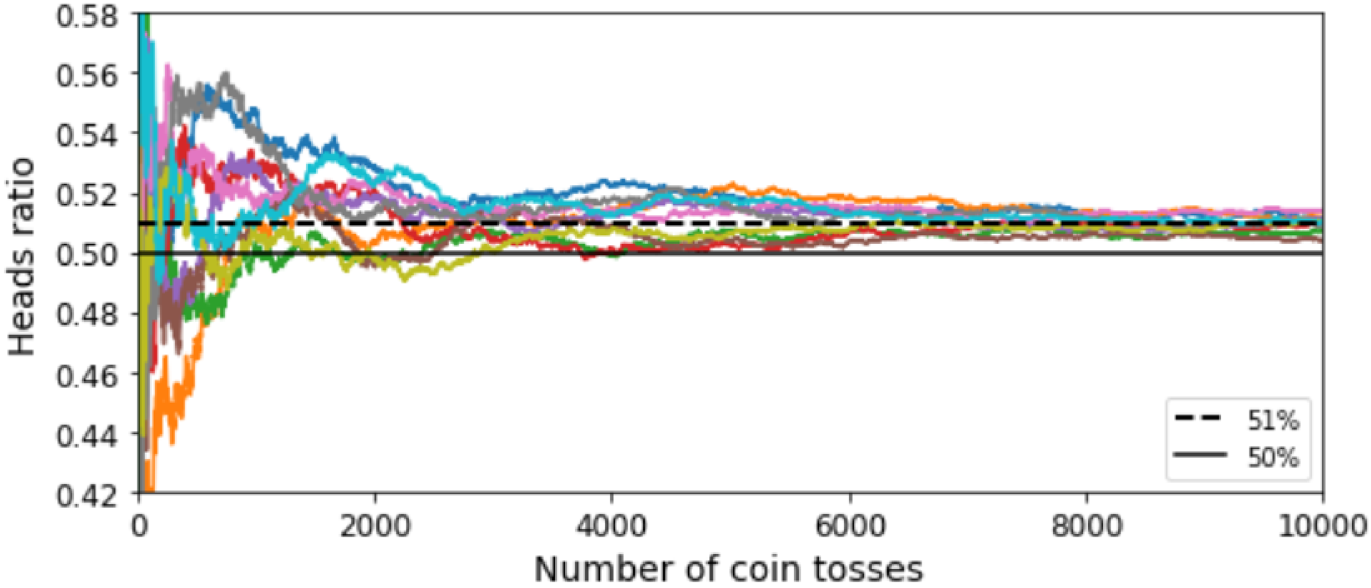
- Toy example
  - hard voting classifier



Gyeongsang National University 경상국립대학교

Ensemble learning

- Law of large numbers



Ensemble learning

- Soft voting
  - if all classifiers can predict the probability of classification, ensemble method can derive the ensemble probability

Ensemble learning

- Way of ensemble learning
  - usage of different algorithm
  - usage of different training set
    - bagging (bootstrap aggregating)
    - pasting

Ensemble learning

- Bagging
  - allow duplication in training set for sampling
- Pasting
  - without duplication in training set for sampling

Ensemble learning

- Data sampling
  - bias of individual classifiers is high
  - after ensemble method, bias and variance can be decreased
  - parallel computing is possible

Ensemble learning

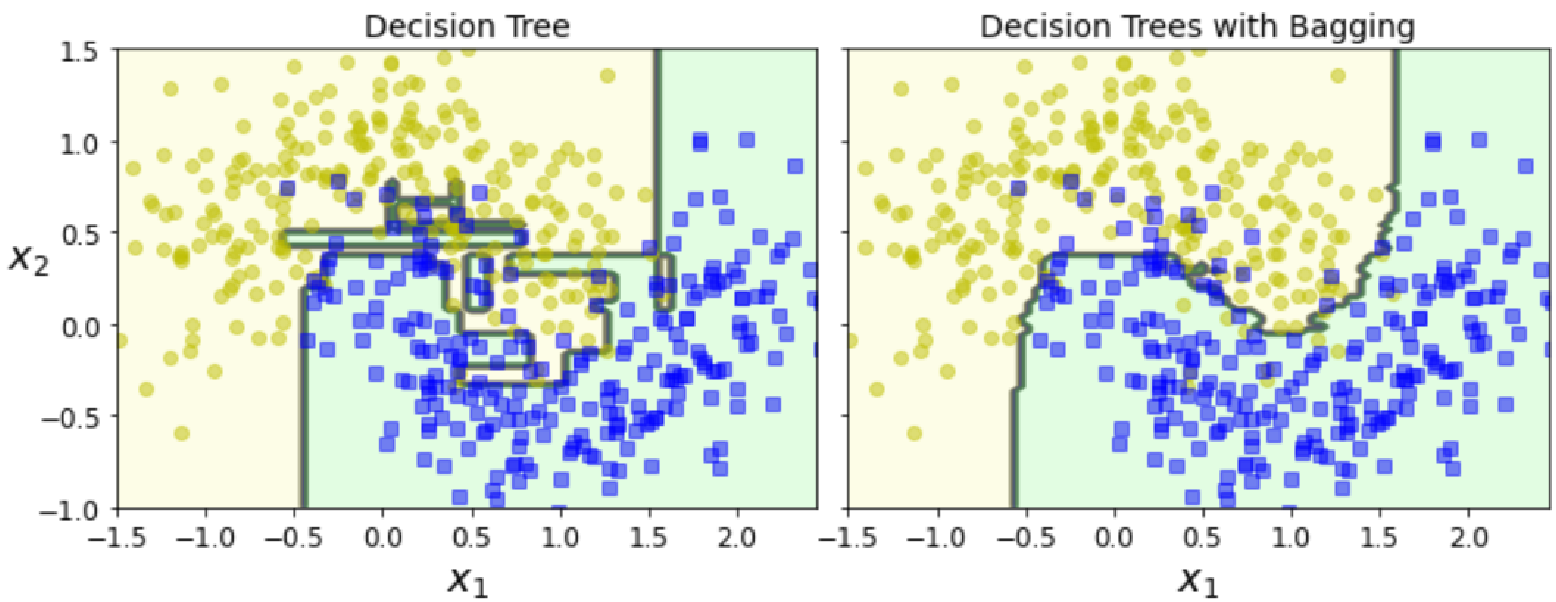
- Bagging in scikit-learn

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

bag_clf = BaggingClassifier(
    DecisionTreeClassifier(), n_estimators=500,
    max_samples=100, bootstrap=True, random_state=42)
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
```

Ensemble learning

- Bagging in scikit-learn



Ensemble learning

- out-of-bag sample
  - use only a fraction of training samples
  - mathematically, it is about 63.2%
  - remaining 36.8% samples can be used for validation

Ensemble learning

- Feature sampling
  - merits on higher-dimensional data processing such as image and video
  - parameters in scikit-learn
    - max_features
    - bootstrap_features

Ensemble learning

- Random patches method
  - sampling on both dimension
    - data sample
    - features
  - makes various classifiers

Gyeongsang National University 경상국립대학교

Ensemble learning

- Random forest
  - ensemble of decision tree using bagging or pasting

```
from sklearn.ensemble import RandomForestClassifier

rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16, random_state=42)
rnd_clf.fit(X_train, y_train)

y_pred_rf = rnd_clf.predict(X_test)
```

Gyeongsang National University 경상국립대학교

Ensemble learning

- Extremely randomized tress (extra trees)
  - randomly determine feature criteria without finding optimal thresholds
  - increasing bias, decreasing variance

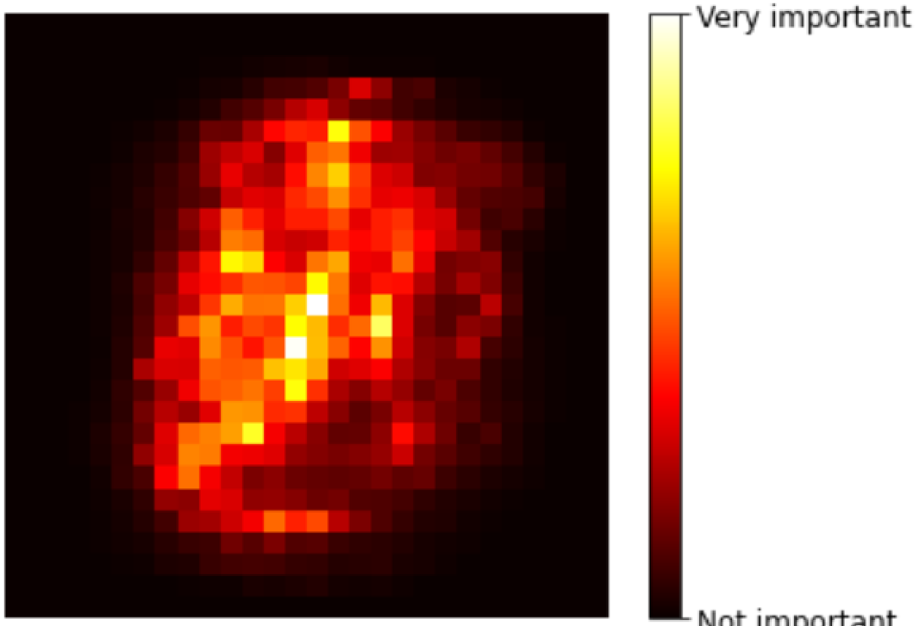
Gyeongsang National University 경상국립대학교

Ensemble learning

- Importance of individual features
  - checking the difference of impurities when utilizing particular feature
  - scikit-learn automatically checks the score

Ensemble learning

- Importance of individual features
  - in MNIST dataset



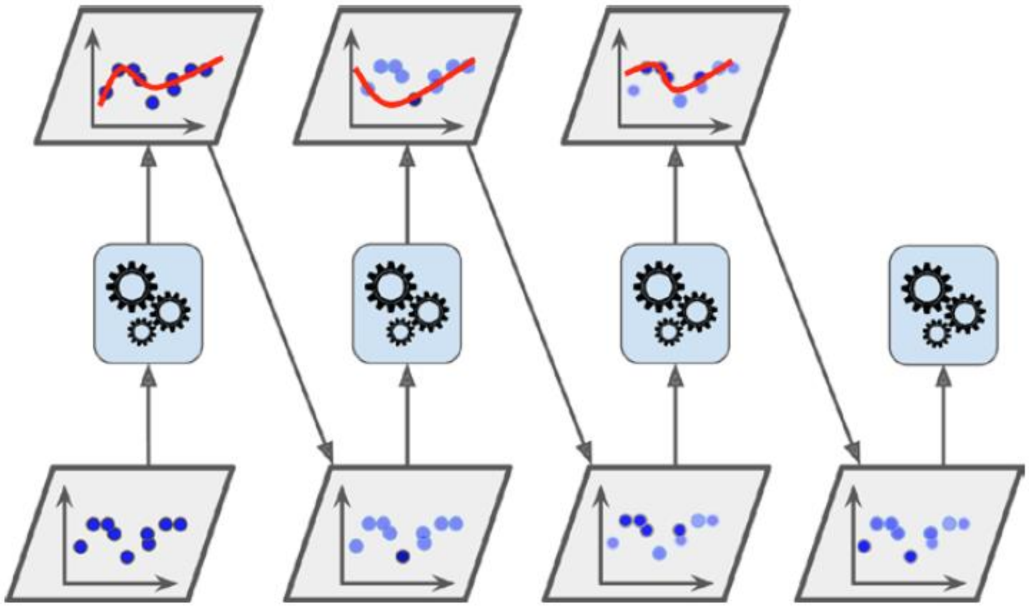
Ensemble learning

- Hypothesis boosting
  - connection of weak learner
  - learners that complement previous models
  - famous algorithm
    - adaptive boosting (AdaBoost)
    - gradient boosting

GNU Gyeongsang National University 경상국립대학교

Ensemble learning

- AdaBoost
  - increasing the weight of the underfitting part of the previous model



Ensemble learning

- AdaBoost
  - finding error rate in j-th classifier

$$r_j = \frac{\sum_{i=1,\, \hat{y}_j^{(i)} \neq y^{(i)}}^{m} w^{(i)}}{\sum_{i=1}^{m} w^{(i)}}$$

  - weights for classifier

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

GNU
Gyeongsang
National
University
경상국립대학교

Ensemble learning

- AdaBoost
  - updating the weight of samples

$$w^{(i)} = \begin{cases} w^{(i)} & \text{when } \hat{y}_j^{(i)} = y^{(i)} \\ w^{(i)} \exp(\alpha_j) & \text{when } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases}$$

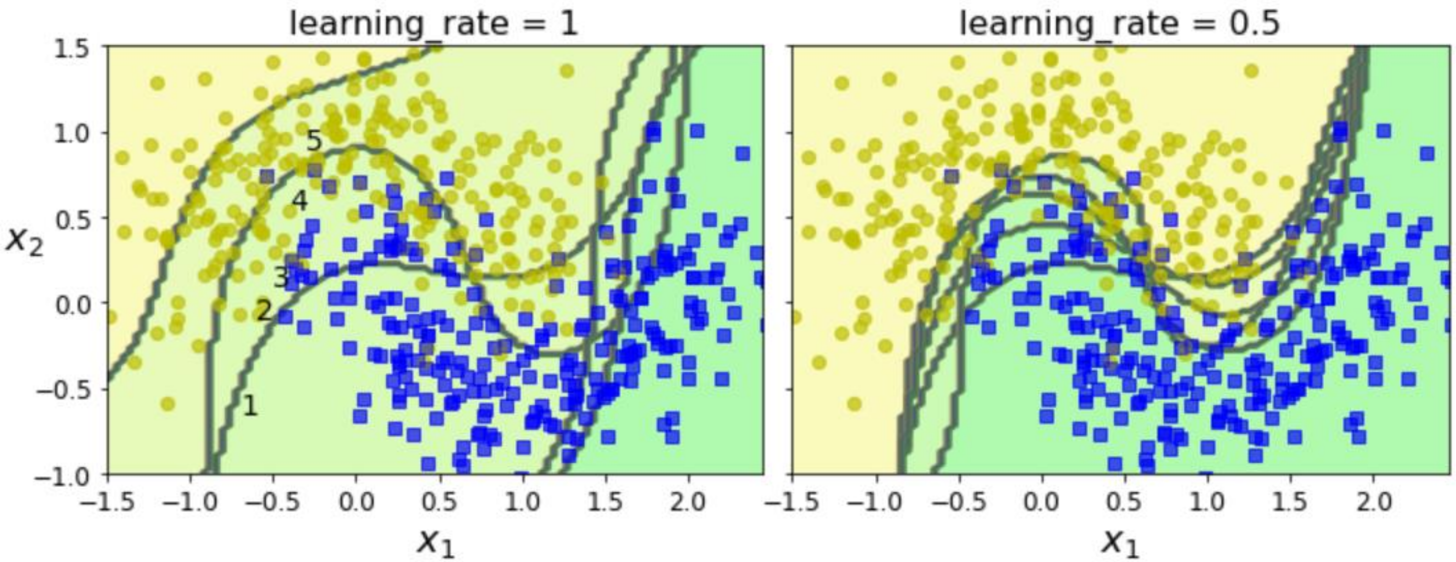
  - prediction of AdaBoost

$$\hat{y}(\mathbf{x}) = \arg\max_k \sum_{j=1,\hat{y}(\mathbf{x})=k}^{N} \alpha_j$$

GNU
Gyeongsang
National
University
경상국립대학교

Ensemble learning

- AdaBoost
  - implementation on scikit-learn

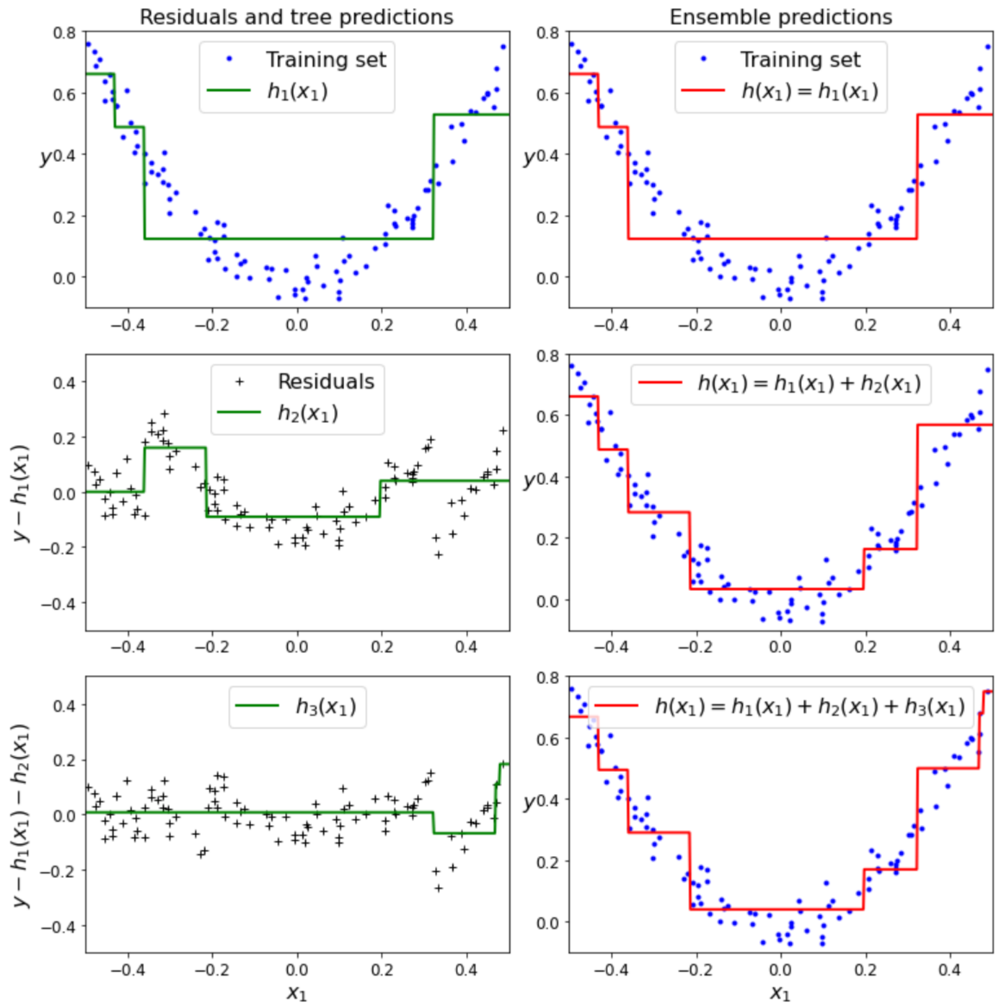


Ensemble learning

- Gradient boosting
  - not modifying the weights of samples
  - learning the residual error for next learner

Ensemble learning

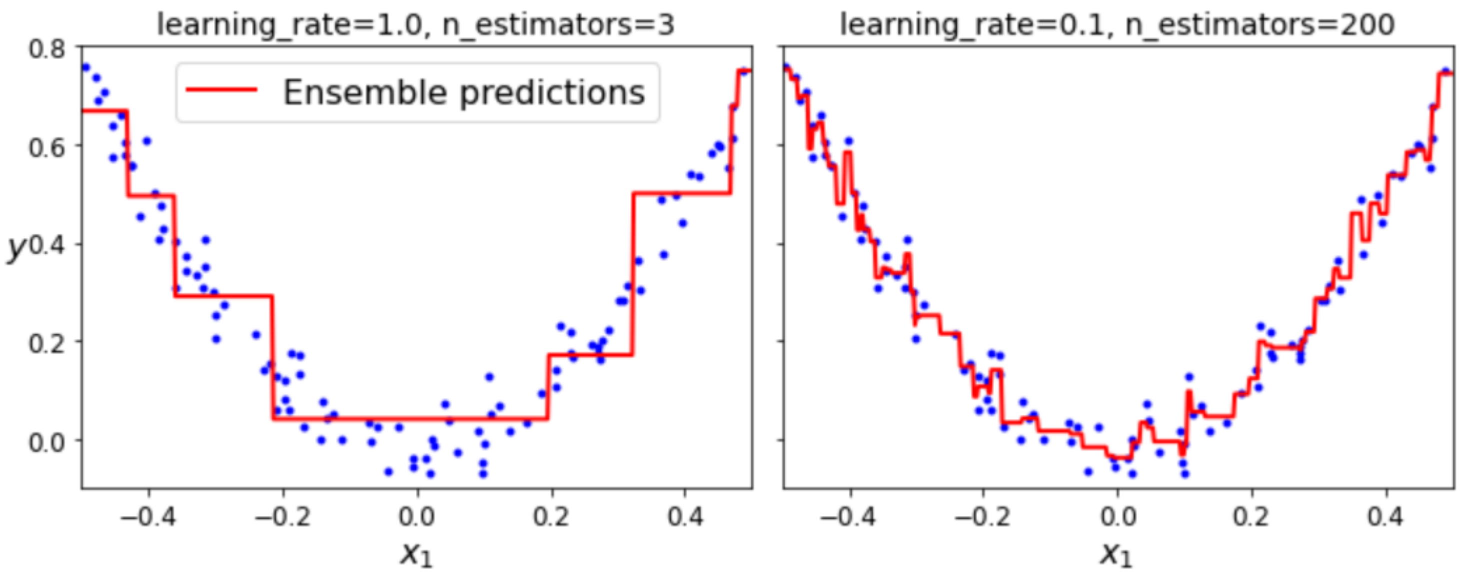
- Gradient boosting
  - residual error



Gyeongsang National University 경상국립대학교

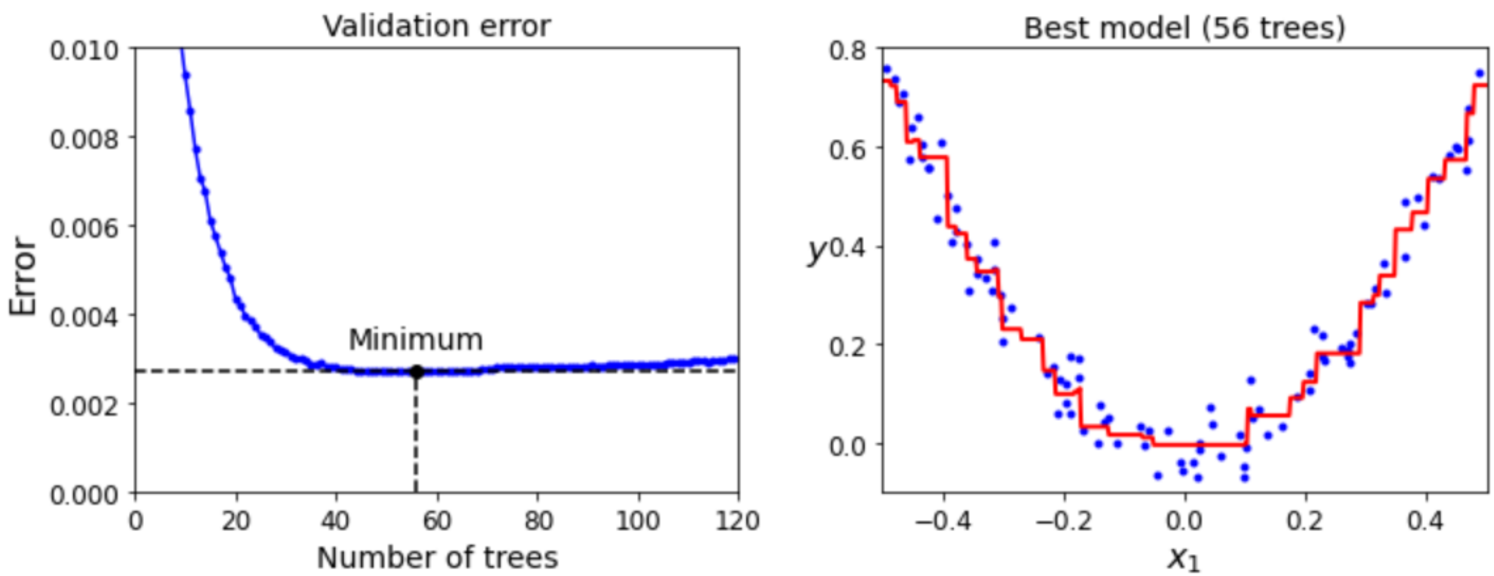
Ensemble learning

- Gradient boosting
  - regression result when number of estimator increases



Ensemble learning

- Gradient boosting
  - early stopping



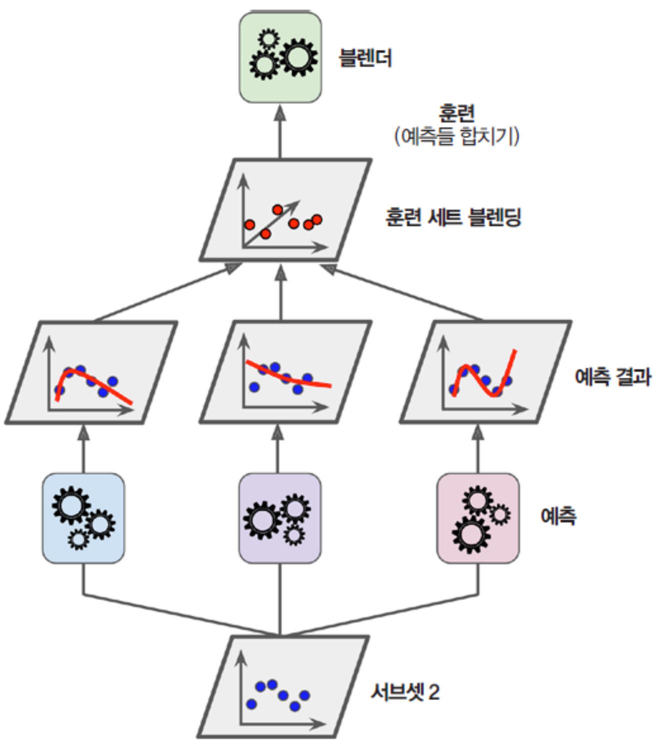
Ensemble learning

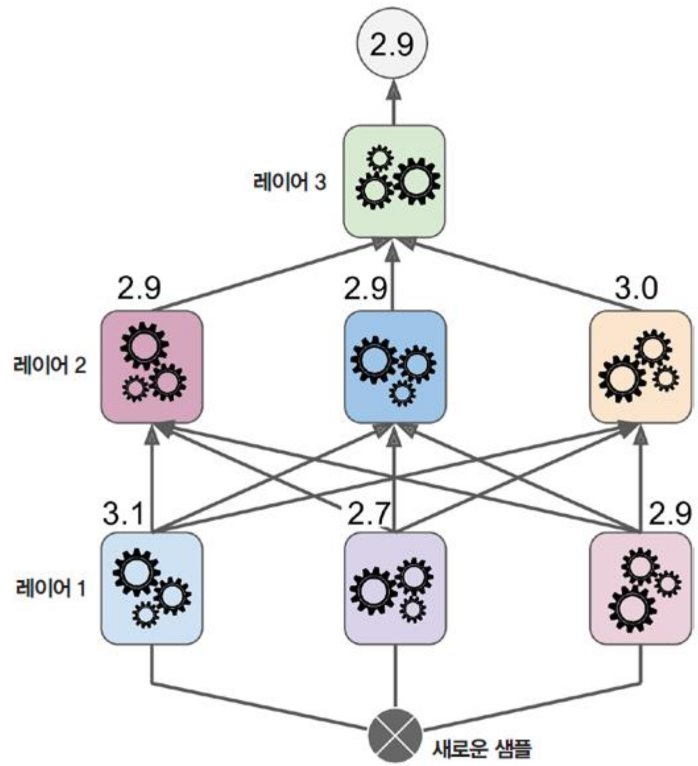
- Staking (stacked generalization)
  - learning form ensembled prediction
  - called blender or meta-learner

Gyeongsang
National
University
경상국립대학교

Ensemble learning

- Staking (stacked generalization)





Feel free to question

Through e-mail & LMS

본 자료의 연습문제는 수업의 본교재인
한빛미디어, Hands on Machine Learning(2판)에서 주로 발췌함