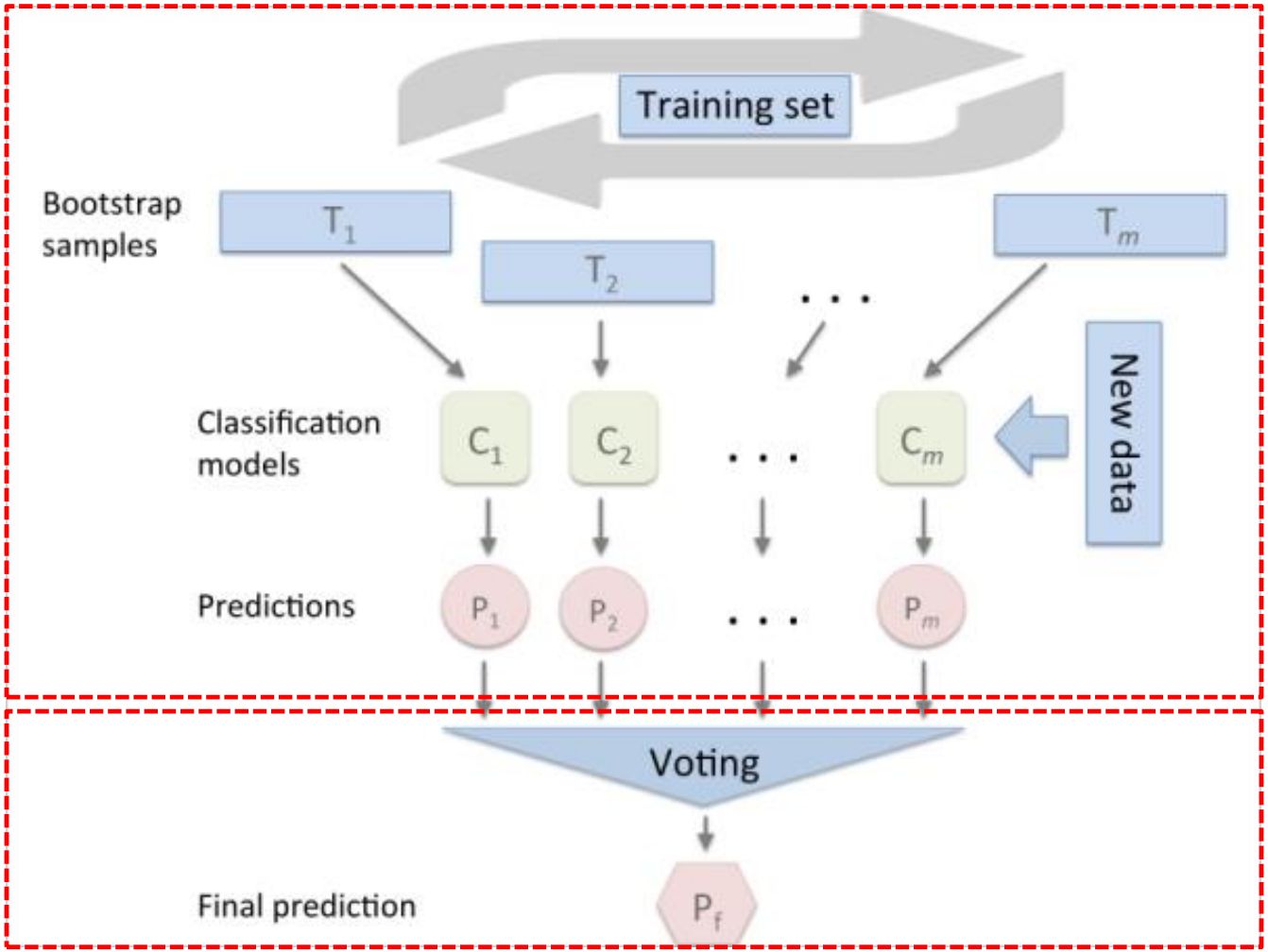
앙상블 (Ensemble)

앙상블

기계학습에서의 앙상블

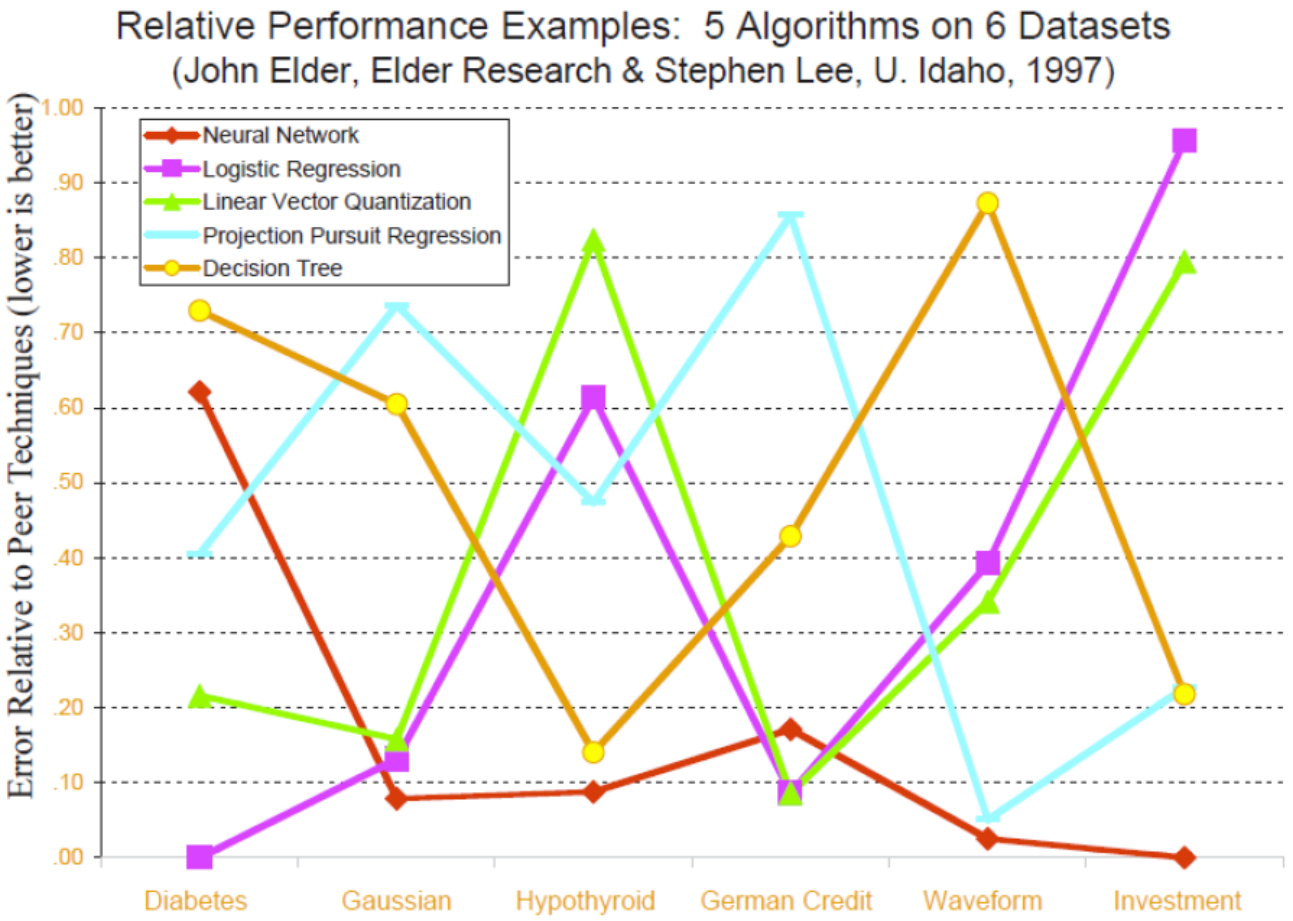

Training set
Bootstrap samples
T_1
T_2
T_m
Classification models
C_1
C_2
C_m
New data
Predictions
P_1
P_2
P_m
Voting
Final prediction
P_f

앙상블의 배경

- **No Free Lunch Theorem**
  - Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82.
  - **특정한 문제에 최적화된 알고리즘은 다른 문제에서는 그렇지 않다는 것**을 수학적으로 증명한 정리
- **기계학습에 적용**
  - 어떤 알고리즘도 모든 상황에서 다른 알고리즘보다 우월하다는 결론을 내릴 수 없음
  - 문제의 목적, 데이터 형태 등을 종합적으로 고려하여 최적의 알고리즘을 선택할 필요가 있음

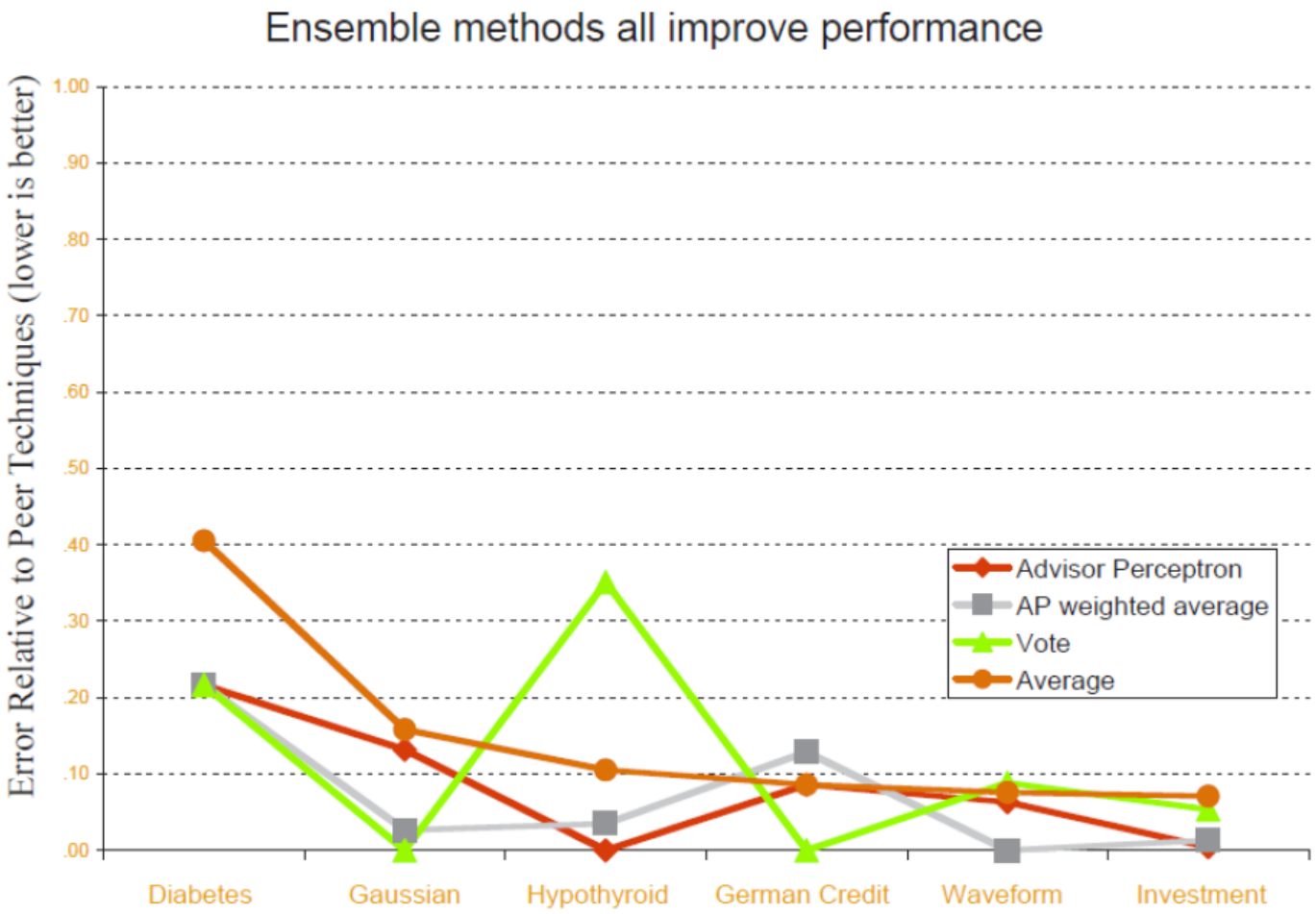
앙상블의 배경

- 모든 데이터에 좋은 성능을 내는 모델은 없음



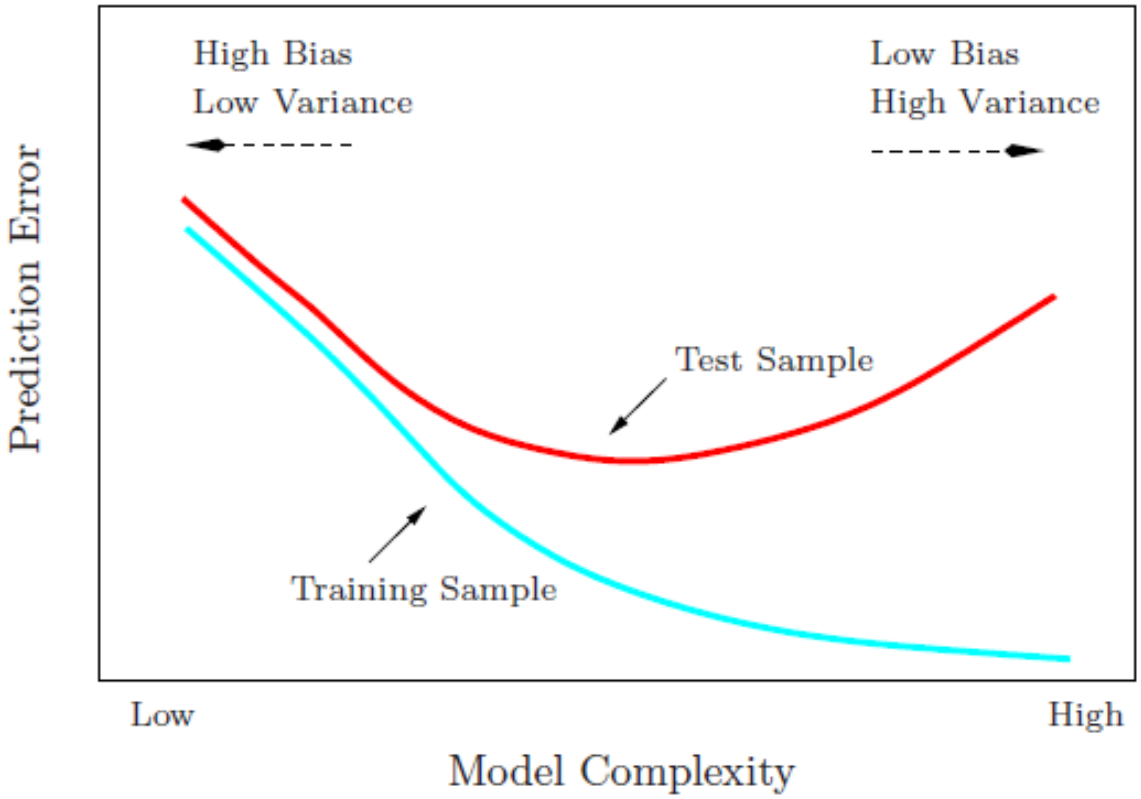
앙상블의 배경

- 개별 모델을 조합하여 더 좋은 Performance 달성이 가능



이론적 배경

- **Bias-Variance Tradeoff**



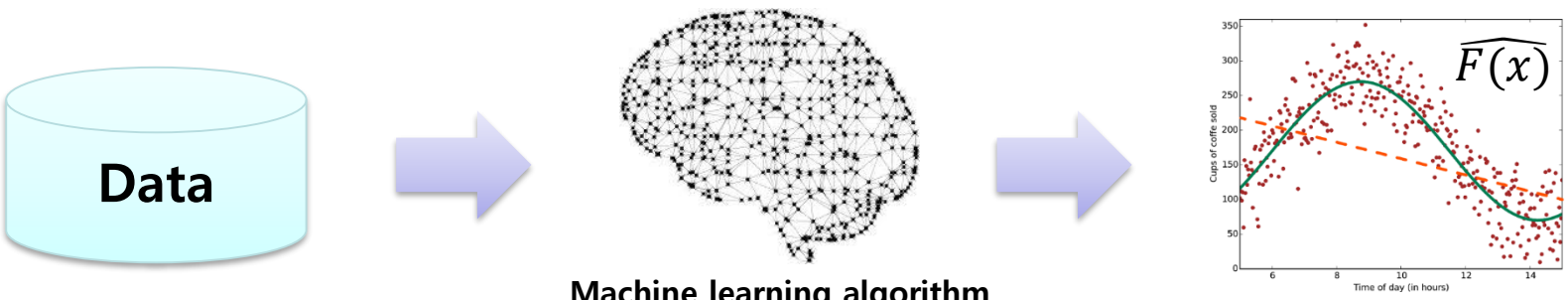
이론적 배경

- **실제 데이터는 항상 노이즈가 존재**

$$y = F^*(\mathbf{x}) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

  - ***F****(x) 는 우리가 학습하고 싶은 데이터 생성 함수
  - 노이즈로 인하여 정확한 추정은 현실적으로 불가능
  - 노이즈는 서로 독립적이고 일정한 분산을 갖는다고 가정

- **모델 학습**
  - 주어진 하나의 샘플 집합으로부터 **데이터 생성 함수**를 추론하는 것



이론적 배경

- 모델에 의한 오류는 **편향(Bias)**과 **분산(Variance)**로 구분될 수 있음
  - **편향(Bias)**: **추정 값의 평균**과 **참값들 간**의 차이

    → **실제값과 예측값의 거리**
  - **분산(Variance)**: **추정 값의 평균**과 **추정 값**들 간의 차이

    → **추정값들의 흩어진 정도**

$$Err(\mathbf{x}_0) = E\left[y - \hat{F}(\mathbf{x})|\mathbf{x} = \mathbf{x}_0\right]^2$$

$$= \left[\hat{F}^*(\mathbf{x}_0) - \bar{F}(\mathbf{x}_0)\right]^2 + E\left[\bar{F}(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

$$= Bias^2\left(\hat{F}(\mathbf{x}_0)\right) + Var\left(\hat{F}(\mathbf{x}_0)\right) + \sigma^2$$

이론적 배경 (optional)

- The MSE for a particular data point

$$Err(\mathbf{x}_0) = E\left[y - \hat{F}(\mathbf{x})|\mathbf{x} = \mathbf{x}_0\right]^2 \qquad (y = F^*(\mathbf{x}) + \epsilon)$$

$$= E\left[\hat{F}^*(\mathbf{x}_0) + \epsilon - \hat{F}(\mathbf{x}_0)\right]^2$$

$$= E\left[\hat{F}^*(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

$$= E\left[\hat{F}^*(\mathbf{x}_0) - \bar{F}(\mathbf{x}_0) + \bar{F}(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

이론적 배경 (optional)

$$= E\left[\hat{F}^*(\mathbf{x}_0) - \bar{F}(\mathbf{x}_0) + \bar{F}(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

By the properties of the expectation operator

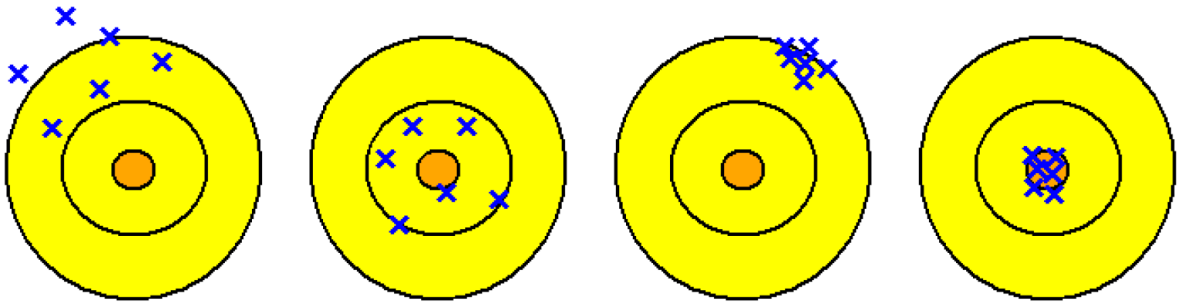
$$= E\left[\hat{F}^*(\mathbf{x}_0) - \bar{F}(\mathbf{x}_0)\right]^2 + E\left[\bar{F}(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

$$= \left[\hat{F}^*(\mathbf{x}_0) - \bar{F}(\mathbf{x}_0)\right]^2 + E\left[\bar{F}(\mathbf{x}_0) - \hat{F}(\mathbf{x}_0)\right]^2 + \sigma^2$$

$$= Bias^2\left(\hat{F}(\mathbf{x}_0)\right) + Var\left(\hat{F}(\mathbf{x}_0)\right) + \sigma^2$$

이론적 배경

- 편향(Bias)과 분산(Variance)의 크기에 따른 모델의 구분



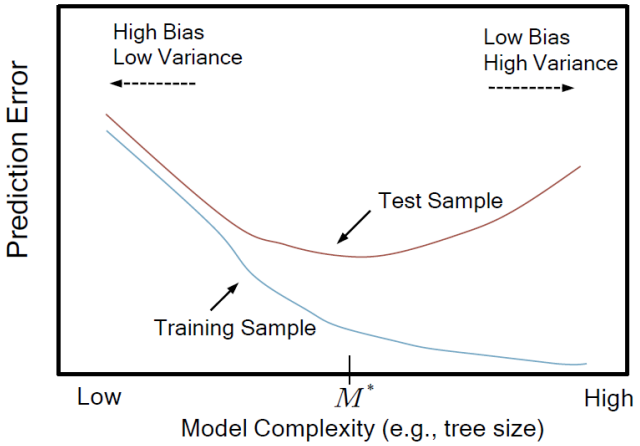
Bias	**High**	Low	**High**	Low
Variance	**High**	**High**	Low	Low

- **낮은 모델 복잡도: 높은 편향 & 낮은 분산**

  Logistic regression, LDA, k-NN with large k.

- **높은 모델 복잡도: 낮은 편향 & 높은 분산**

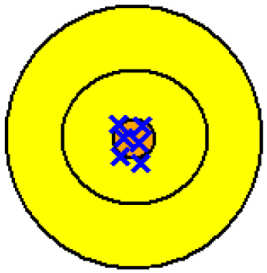
  DT, ANN, SVM, k-NN with small k.



이론적 배경

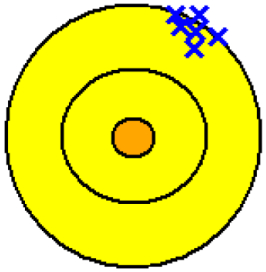
- Train / Test data 관점

Train data



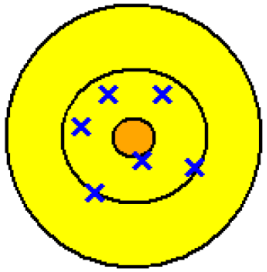
Bias	Low
Variance	Low

Test data(1)



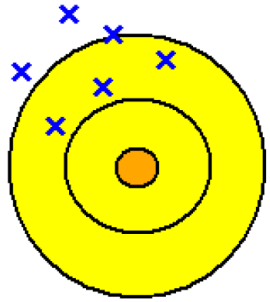
Bias	**High**
Variance	Low

Test data(2)



Bias	Low
Variance	**High**

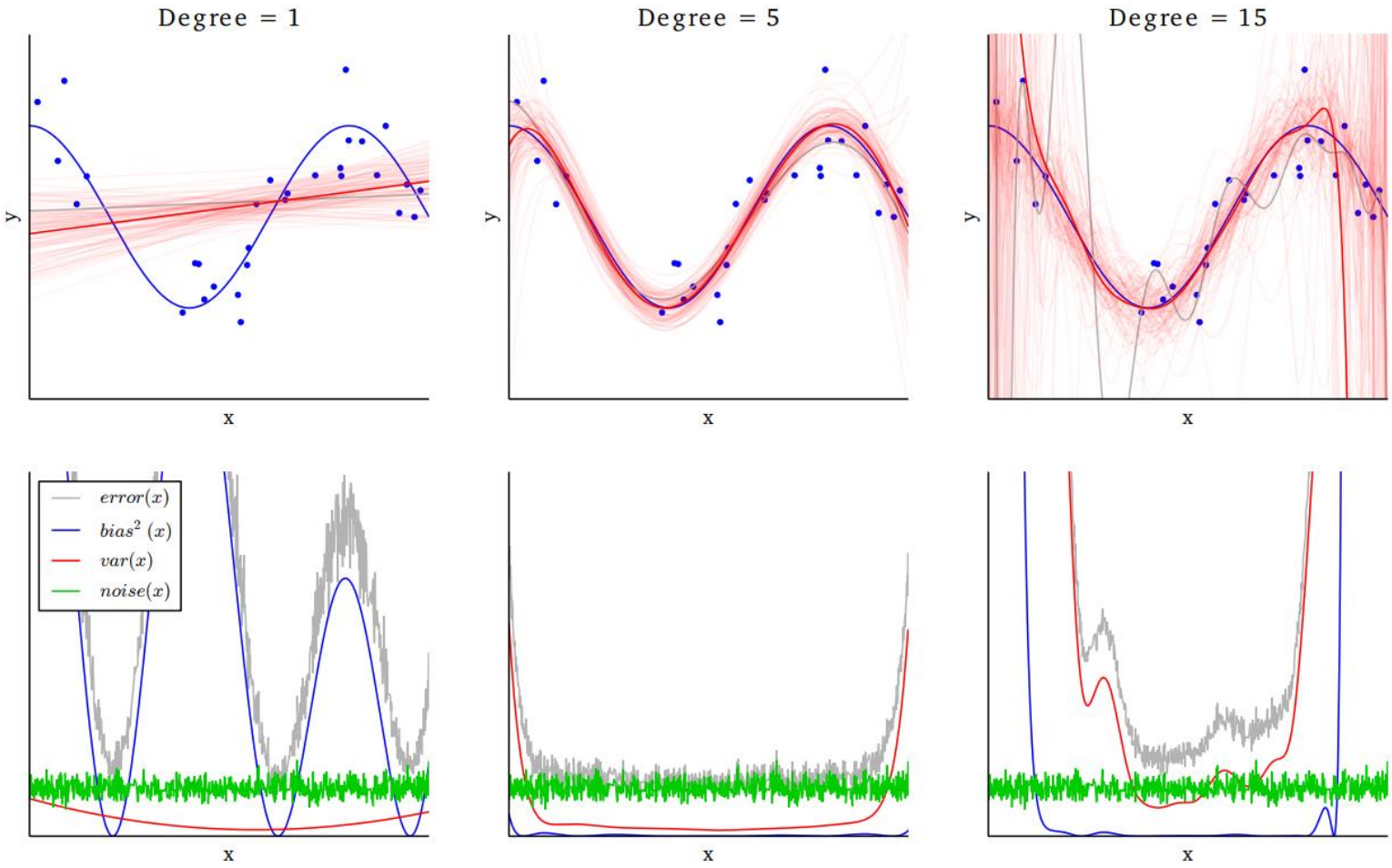
Test data(3)



Bias	**High**
Variance	**High**

이론적 배경

- Train / Test data 관점



앙상블

앙상블의 목적

- **앙상블의 목적: 다수의 모델을 학습하여 오류의 감소를 추구**
  - 분산의 감소에 의한 오류 감소: **배깅(Bagging) - 랜덤 포레스트(Random Forest)**
  - 편향의 감소에 의한 오류 감소: 부스팅(Boosting)
  - 분산과 편향의 동시 감소: Mixture of Experts

- **앙상블 구성의 두 가지 핵심 아이디어**
  - **다양성(diversity)**을 어떻게 확보할 것인가?
  - 최종 결과물을 어떻게 **결합(combine, aggregate)**할 것인가?

앙상블의 효과

- 이론적으로는 M개의 개별 모델을 결합한 앙상블의 경우
- M개의 개별 모델의 평균 오류의 1/M 수준으로 오류가 감소함

(가정: 각 모델은 서로 독립)

$$E_{Ensemble} = \frac{1}{M} E_{Avg}$$

- 위 가정은 현실세계에서 지켜지지 않는 경우가 많음
- 현실적으로는 M개의 개별 모델을 결합한 앙상블의 경우 개별 모델의 평균 오류보다는 최소한 같거나 낮은 오류를 나타내는 것을 증명할 수 있음

$$\Big[\sum_{m=1}^{M} \epsilon_m(\mathbf{x})\Big]^2 \leq M \sum_{m=1}^{M} \epsilon_m(\mathbf{x})^2 \Rightarrow \Big[\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\mathbf{x})\Big]^2 \leq \frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\mathbf{x})^2$$

$$E_{Ensemble} \leq E_{Avg}$$

- **즉, 1등 모델보다 성능이 우수함을 입증할 수는 없으나 개별 모델들의 평균치보다는 항상 우수하거나 같은 성능을 나타냄**

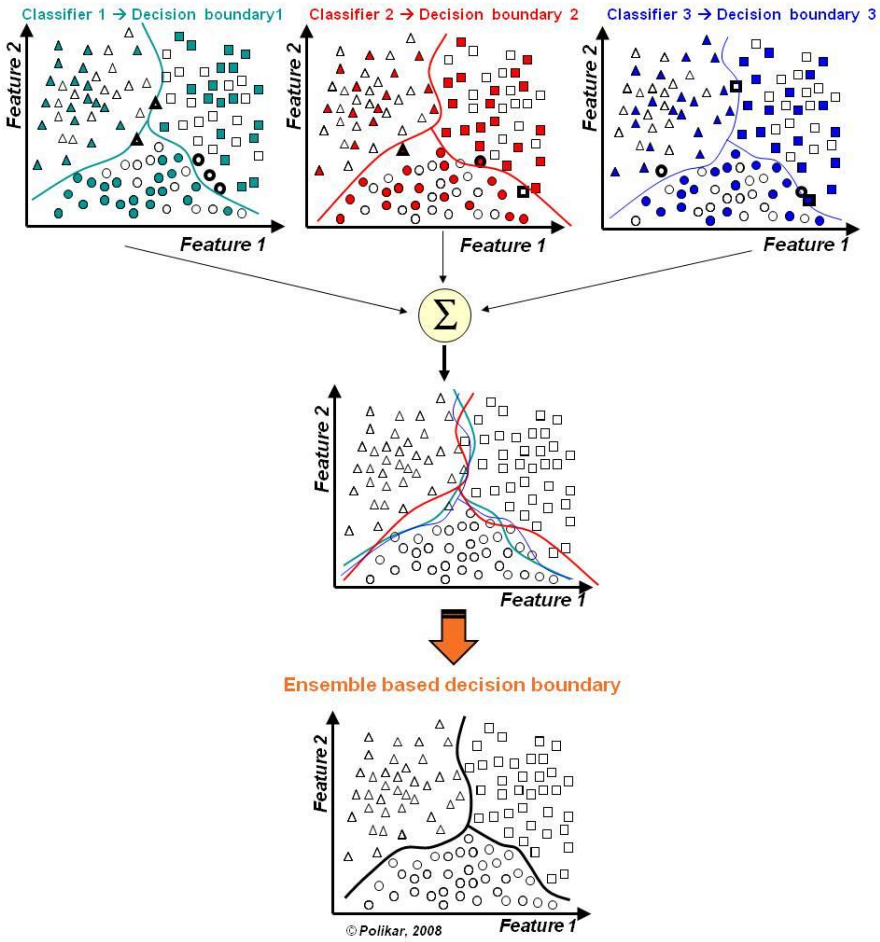
앙상블의 효과

- **동일한 모델을 여러 개 사용하는 것은 효과가 없음**
  - 개별 모델은 서로 **적절하게 달라야 앙상블의 효과를 볼 수 있음**
  - 개별적으로는 어느 정도 좋은 성능을 가지면서, 앙상블 내에서 **각각의 모델은 서로 다양한 형태를 나타내는 것이 가장 이상적**

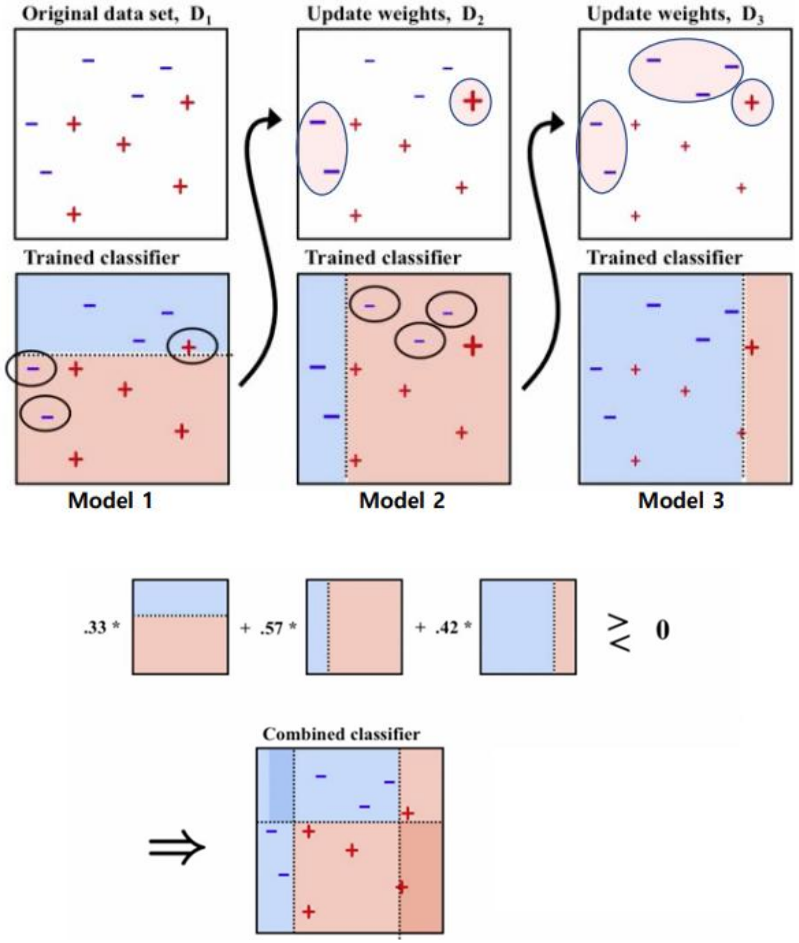
Diversity	**Implicit**	**Explicit**
Description	Provide different random subset of the training data to each learner	Use some measurement ensuring it is substantially different from the other members
Ensemble Algorithms	Instance: **Bagging** Variables: Random Subspaces, Rotation Forests Both: Random Forests	**Boosting**, Negative Correlation Learning

앙상블의 다양성 확보 방법

Bagging (Bootstrap Aggregating)



Boosting



배깅(Bagging)

- **Bagging: Bootstrap Aggregating**
  - 앙상블의 각 멤버(모델)은 서로 다른 학습 데이터셋을 이용
  - 각 데이터셋은 복원 추출(sampling with replacement)을 통해 **원래 데이터의 수만큼의 크기를 갖도록 샘플링**
  - 개별 데이터셋을 **붓스트랩(bootstrap)**이라 부름

Original Dataset

x^1	y^1
x^2	y^2
x^3	y^3
x^4	y^4
x^5	y^5
x^6	y^6
x^7	y^7
x^8	y^8
x^9	y^9
x^{10}	y^{10}

Bootstrap 1

x^3	y^3
x^6	y^6
x^2	y^2
x^{10}	y^{10}
x^8	y^8
x^7	y^7
x^7	y^7
x^3	y^3
x^2	y^2
x^7	y^7

Bootstrap 2

x^7	y^7
x^1	y^1
x^{10}	y^{10}
x^1	y^1
x^8	y^8
x^6	y^6
x^2	y^2
x^6	y^6
x^4	y^4
x^9	y^9

• • •

Bootstrap B

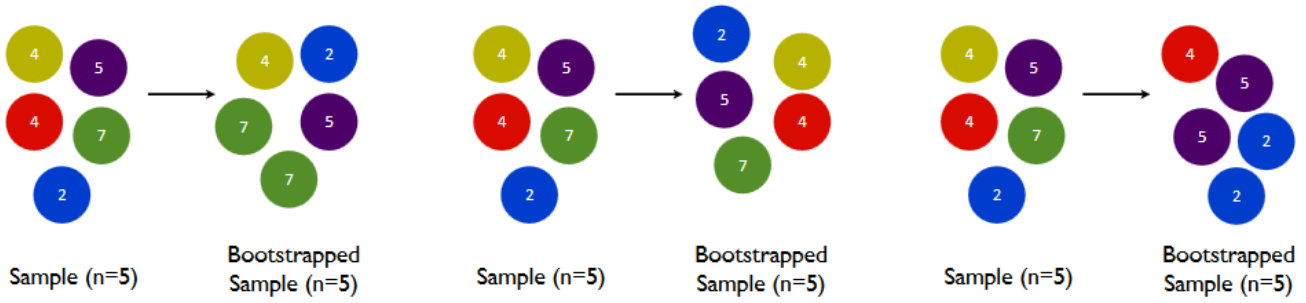
x^9	y^9
x^5	y^5
x^2	y^2
x^4	y^4
x^7	y^7
x^2	y^2
x^5	y^5
x^{10}	y^{10}
x^8	y^8
x^2	y^2

배깅(Bagging)

- **Bagging: Bootstrap Aggregating**
  - 이론적으로 한 관측치가 **하나의 붓스트랩에** 한번도 선택되지 않을 확률

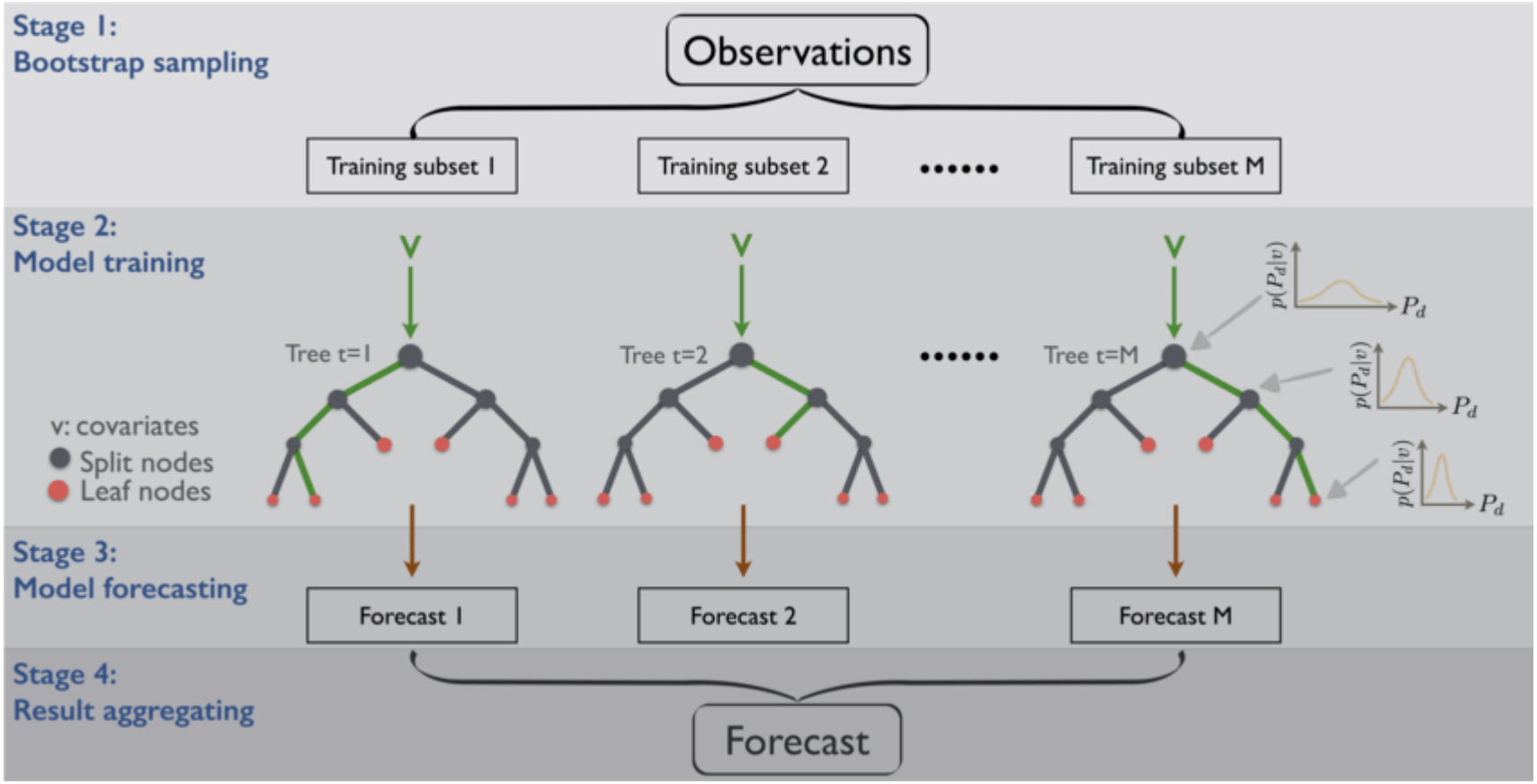
$$p = \left(1 - \frac{1}{N}\right)^N \rightarrow \lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = e^{-1} = 0.368$$

  - **개별 모델의 분산은 높고 편향이 낮은 알고리즘에 적절**함



배깅(Bagging)

- **Bagging with Decision Tree**



배깅(Bagging) : combine

- **Result Aggregating**
  - For classification problem
  - **Majority voting**

$$\hat{y}_{Ensemble} = arg\max_{i} \left(\sum_{j=1}^{n} \delta(\hat{y}_j = i), \quad i \in \{0, 1\} \right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

$$\sum_{j=1}^{n} \delta(\hat{y}_j = 0) = 4$$

$$\sum_{j=1}^{n} \delta(\hat{y}_j = 1) = 6$$

$$\hat{y}_{Ensemble} = 1$$

배깅(Bagging) : combine

- **Result Aggregating**
  - For classification problem
  - **Weighted voting (weight = training accuracy of individual models)**

$$\hat{y}_{Ensemble} = arg\max_{i} \left(\frac{\sum_{j=1}^{n} (TrnAcc_j) \cdot \delta(\hat{y}_j = i)}{\sum_{j=1}^{n} (TrnAcc_j)}, \quad i \in \{0, 1\} \right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

$$\frac{\sum_{j=1}^{n} (TrnAcc_j) \cdot \delta(\hat{y}_j = 0)}{\sum_{j=1}^{n} (TrnAcc_j)} = 0.424$$

$$\frac{\sum_{j=1}^{n} (TrnAcc_j) \cdot \delta(\hat{y}_j = 1)}{\sum_{j=1}^{n} (TrnAcc_j)} = 0.576$$

$$\hat{y}_{Ensemble} = 1$$

배깅(Bagging) : combine

- **Result Aggregating**
  - For classification problem
  - **Weighted voting (weight = predicted probability for each class)**

$$\hat{y}_{Ensemble} = arg\max_{i}\left(\frac{1}{n}\sum_{j=1}^{n} P(y=i), \quad i \in \{0,1\}\right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

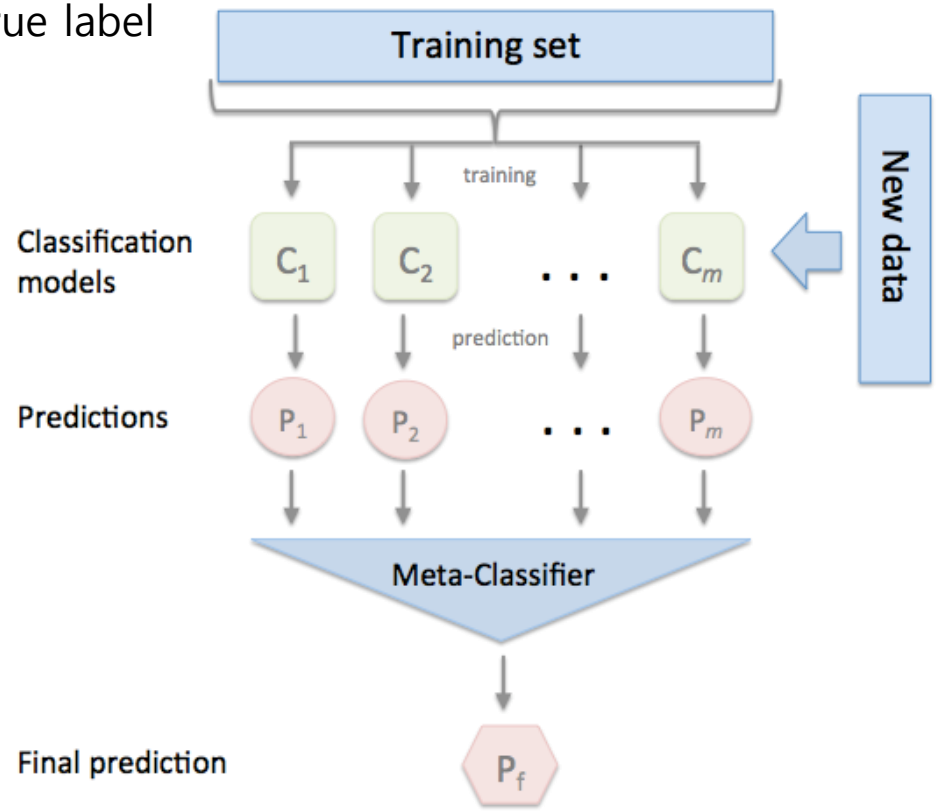
$$\sum_{j=1}^{n} P(y=0) = 0.375$$

$$\sum_{j=1}^{n} P(y=1) = 0.625$$

$$\hat{y}_{Ensemble} = 1$$

배깅(Bagging) : combine

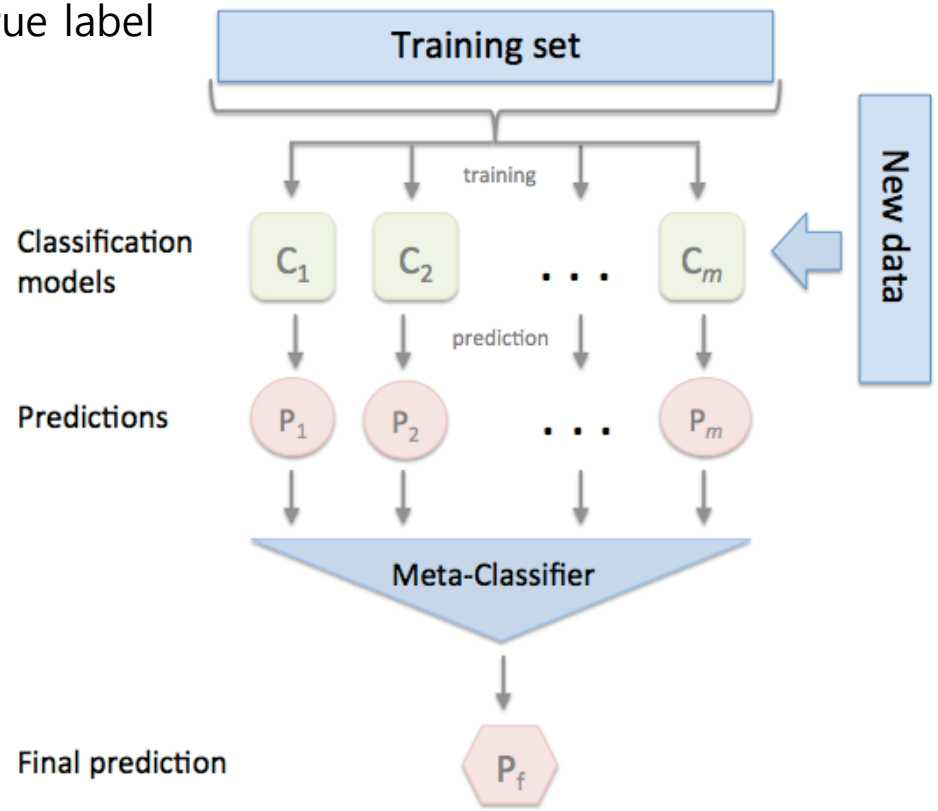
- **Result Aggregating: Stacking**
  - Use another prediction model to aggregate the results
  - Input: Predictions made by ensemble members
  - Target: Actual true label



https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

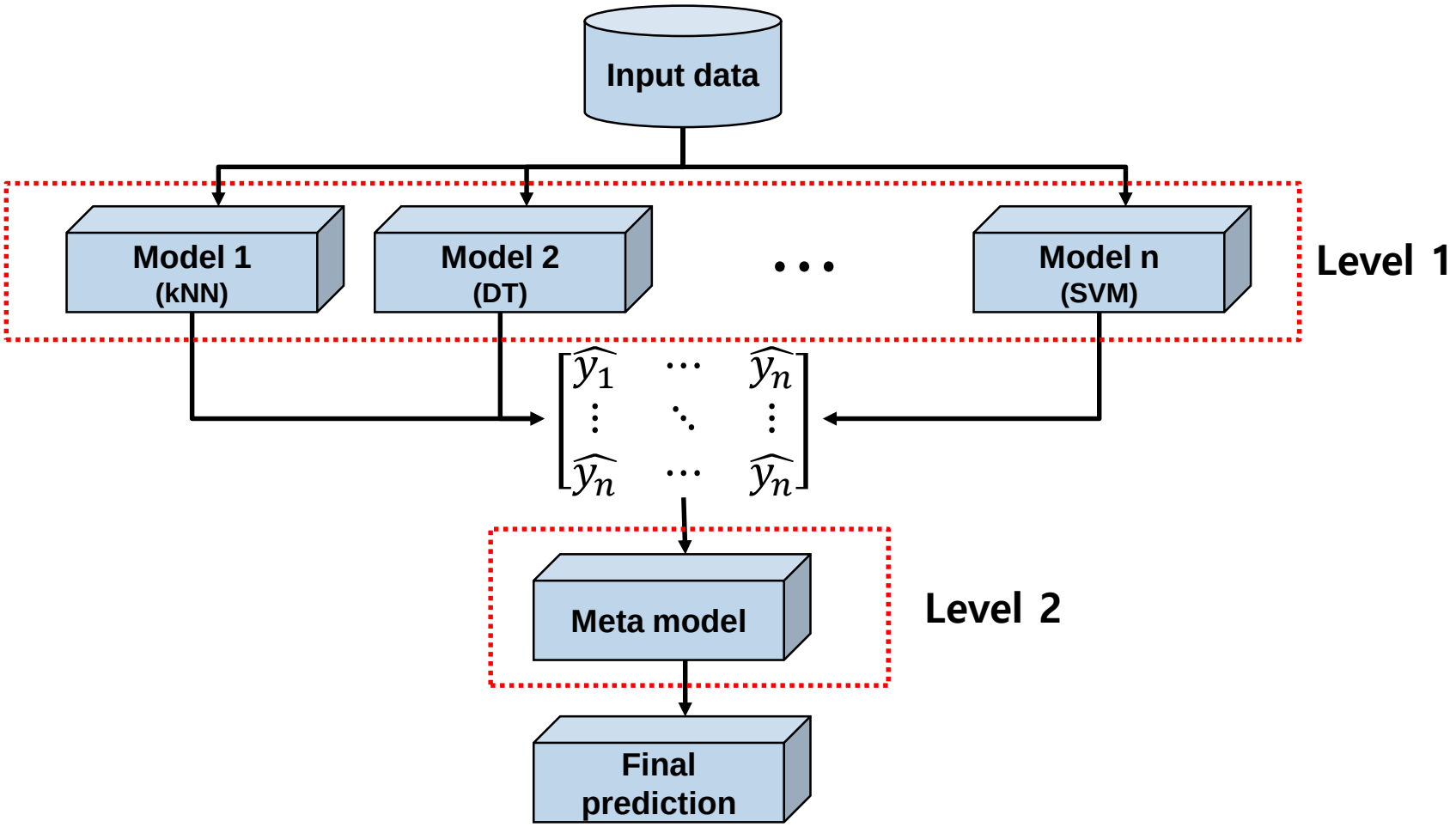
배깅(Bagging) : combine

- **Result Aggregating: Stacking**
  - Use another prediction model to aggregate the results
  - Input: Predictions made by ensemble members
  - Target: Actual true label



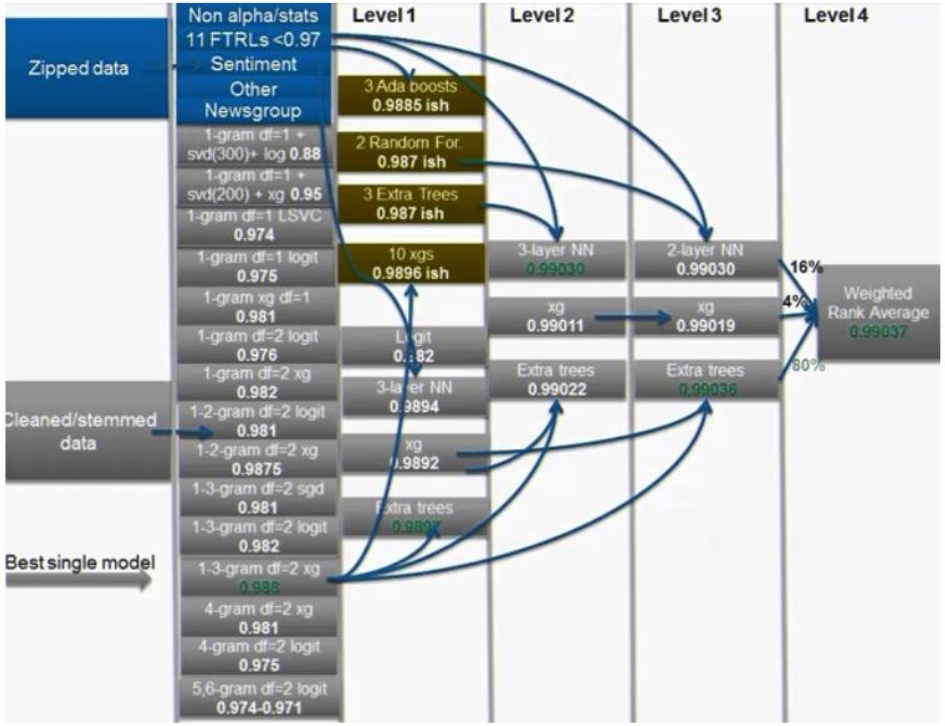
https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

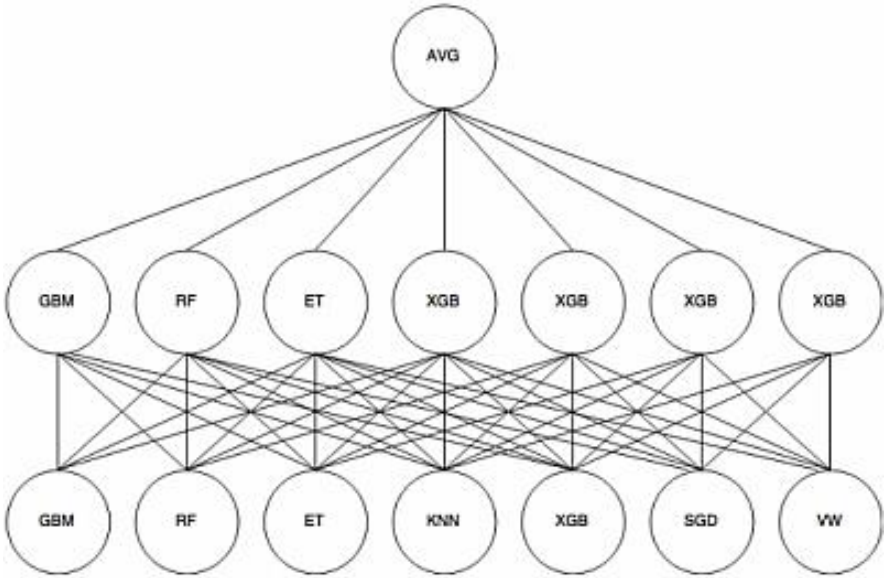
Stacking (optional)



Stacking (optional)

- 다양한 형태의 stacking 방법이 존재
- 각 Node는 개별 모델로 구성되며, Node로 구성된 여러 층의 layer를 쌓는 형태





배깅(Bagging)

- **Bagging: Algorithm**

Algorithm 1 Bagging

Input: Required ensemble size T

Input: Training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$

for $t = 1$ to T **do**

Build a dataset S_t, by sampling N items, randomly *with replacement* from S.

Train a model h_t using S_t, and add it to the ensemble.

end for

For a new testing point (x', y'),

If model outputs are continuous, combine them by averaging.

If model outputs are class labels, combine them by voting.

배깅의 효과

- **Bagging: Decision Boundary**

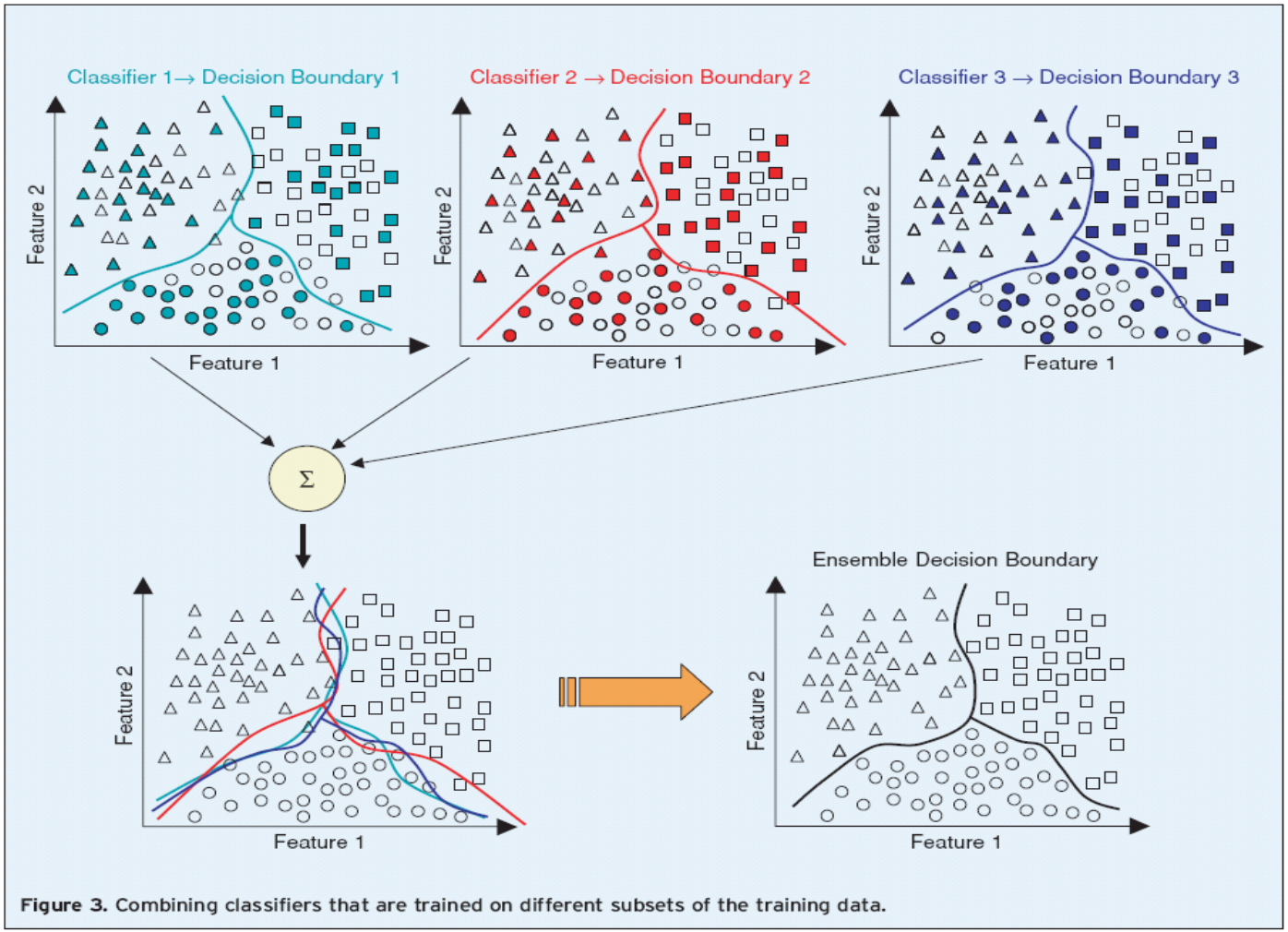
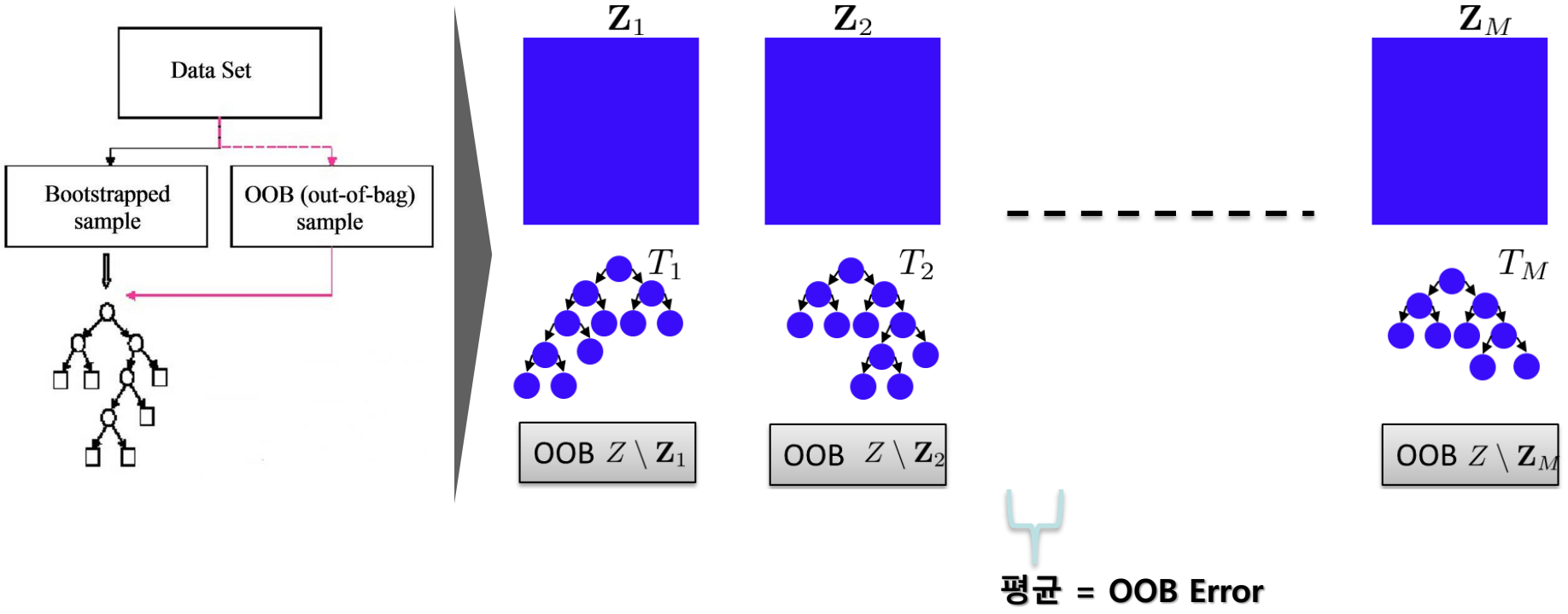


Figure 3. Combining classifiers that are trained on different subsets of the training data.

배깅: 성능 평가

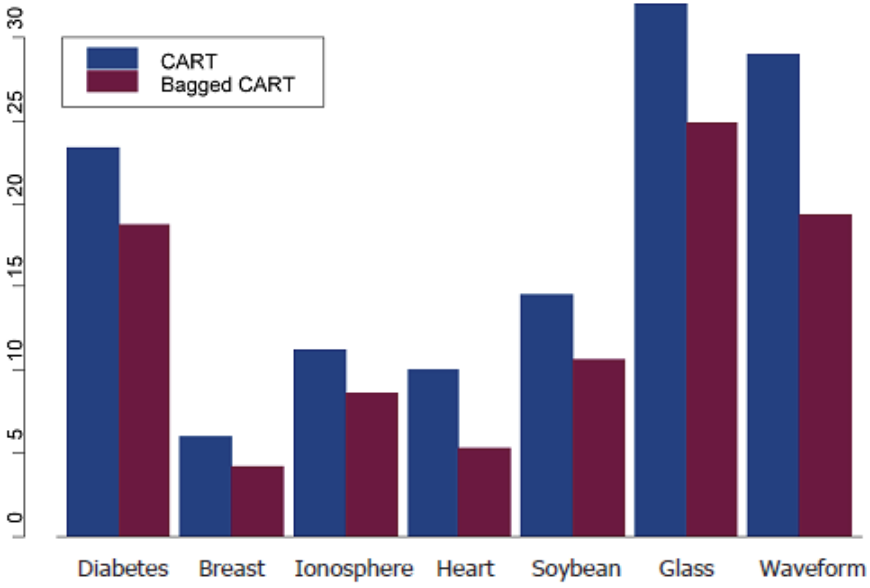
- **Out of bag error (OOB Error)**
  - 배깅을 사용할 경우, 학습/검증 집합을 사전에 나누지 않고 붓스트랩에 포함되지 않는 데이터들을 검증 집합으로 사용함
  - 앙상블 모델의 성능을 간접적으로 확인



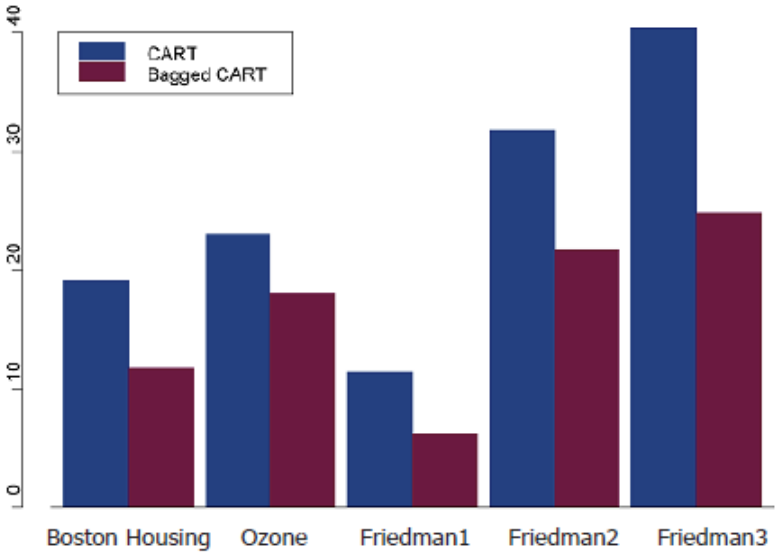
배깅의 성능

- **단일 모델과의 비교**
  - 대부분의 경우, 단일 모델에 비해 배깅을 수행하면 모델의 성능이 향상됨을 알 수 있음

Classification



Regression



Random Forest

- 179 Classifiers
- 121 datasets in UCI
- Random Forest가 최고

Journal of Machine Learning Research 15 (2014) 3133-3181 Submitted 11/13; Revised 4/14; Published 10/14

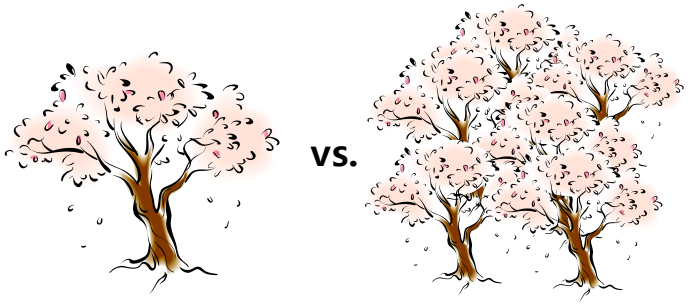
Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

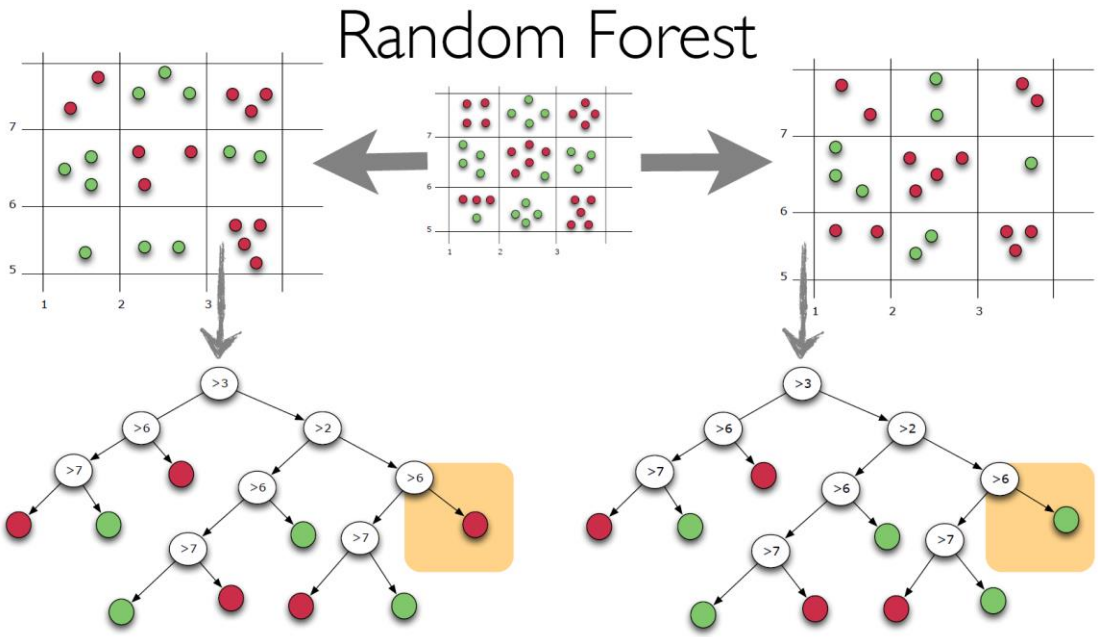
Manuel Fernández-Delgado MANUEL.FERNANDEZ.DELGADO@USC.ES
Eva Cernadas EVA.CERNADAS@USC.ES
Senén Barro SENEN.BARRO@USC.ES
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC
University of Santiago de Compostela
Campus Vida, 15872, Santiago de Compostela, Spain

Dinani Amorim DINANIAMORIM@GMAIL.COM
Departamento de Tecnologia e Ciências Sociais- DTCS
Universidade do Estado da Bahia
Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.305-680, Brasil

Random Forest

- **의사결정나무 기반 앙상블의 특수한 형태**
- 앙상블의 Diversity 확보를 위한 두 가지 장치
  - **Bagging**
  - **Randomly chosen predictor variables (Random Subspace)**





Random Forest

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

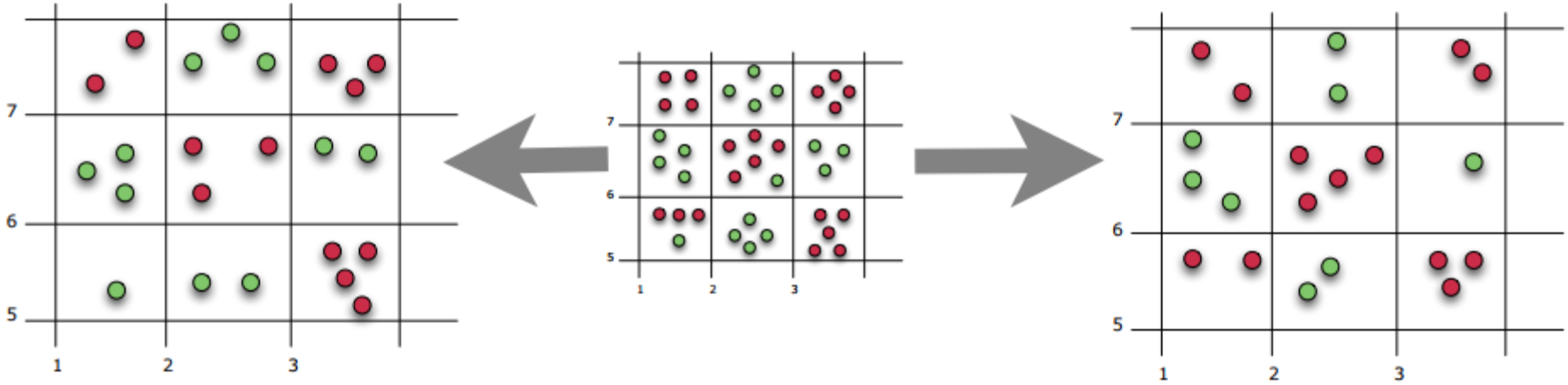
To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

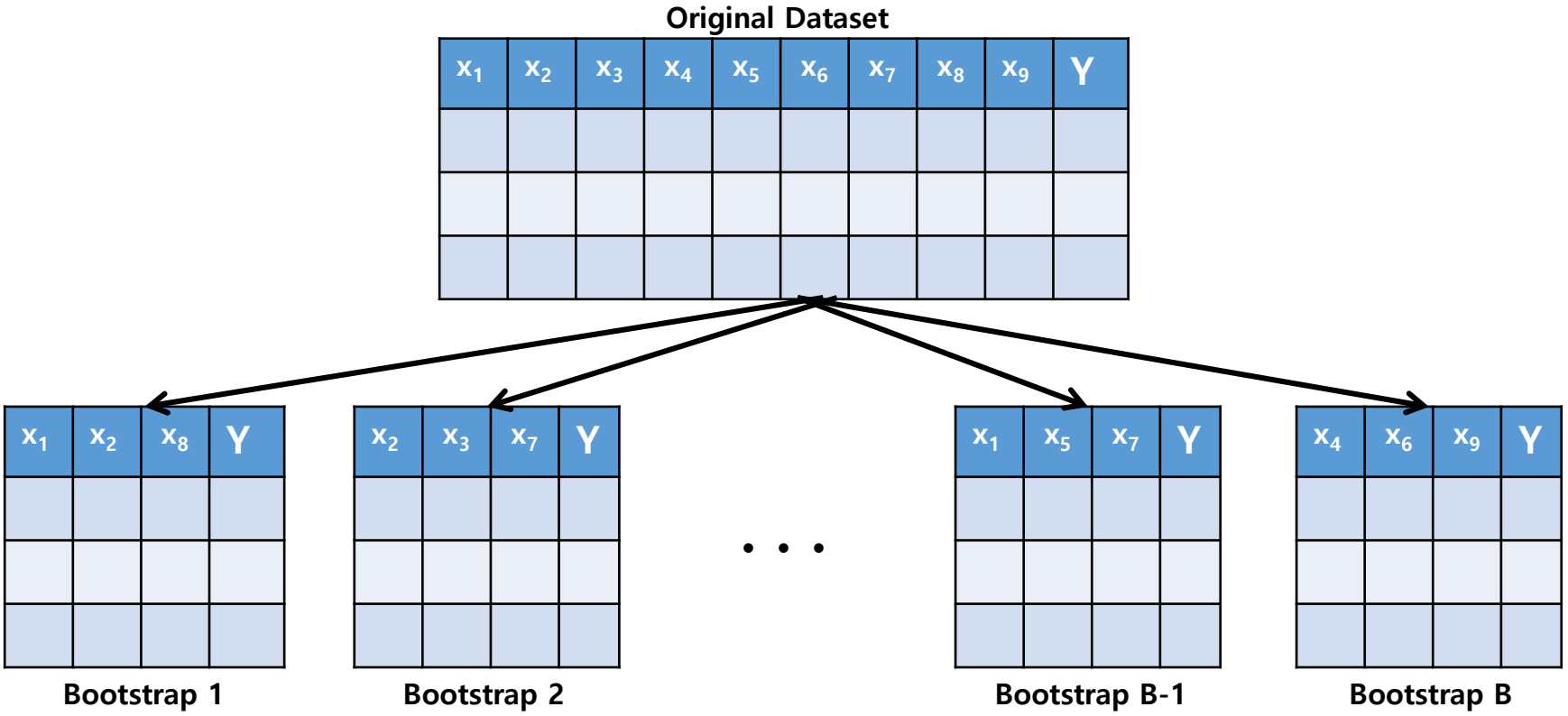
Random Forest

- **Bagging**
  - 복원추출 기법으로 원래 **학습데이터 개체 수 만큼을 샘플링**



Random Forest

- **Random Subspace**
  - 의사결정나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들 만을 고려 대상으로 함
  - 일반적으로 $\sqrt{\#\ of\ features}$ 개의 변수 사용



Random Forest

- **일반화 오류**
  - 랜덤 포레스트의 개별 트리는 가지치기를 하지 않으므로 과적합의 위험이 있음
  - 앙상블 구성모델의 수(population size)가 충분히 클 경우 랜덤 포레스트의 일반화 오류는 다음과 같은 상한을 가짐

$$Generalization\ Error \leq \frac{\bar{\rho}(1-s^2)}{s^2}$$

$\bar{\rho}$: 개별 나무들의 예측 결과물 사이의 상관계수 평균

s^2: 마진 함수 (이범주 분류 문제의 경우 개별 트리의 [Accuracy-Error Rate]의 평균)

  - **개별 나무들이 정확할 수록** 마진함수가 커지며, **개별 나무들의 상관관계가 낮을수록** 일반화 오류가 낮아짐
  - 따라서 일반화 오류가 낮아짐

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.
Ciss, S. (2015). Generalization Error and Out-of-bag Bounds in Random (Uniform) Forests.

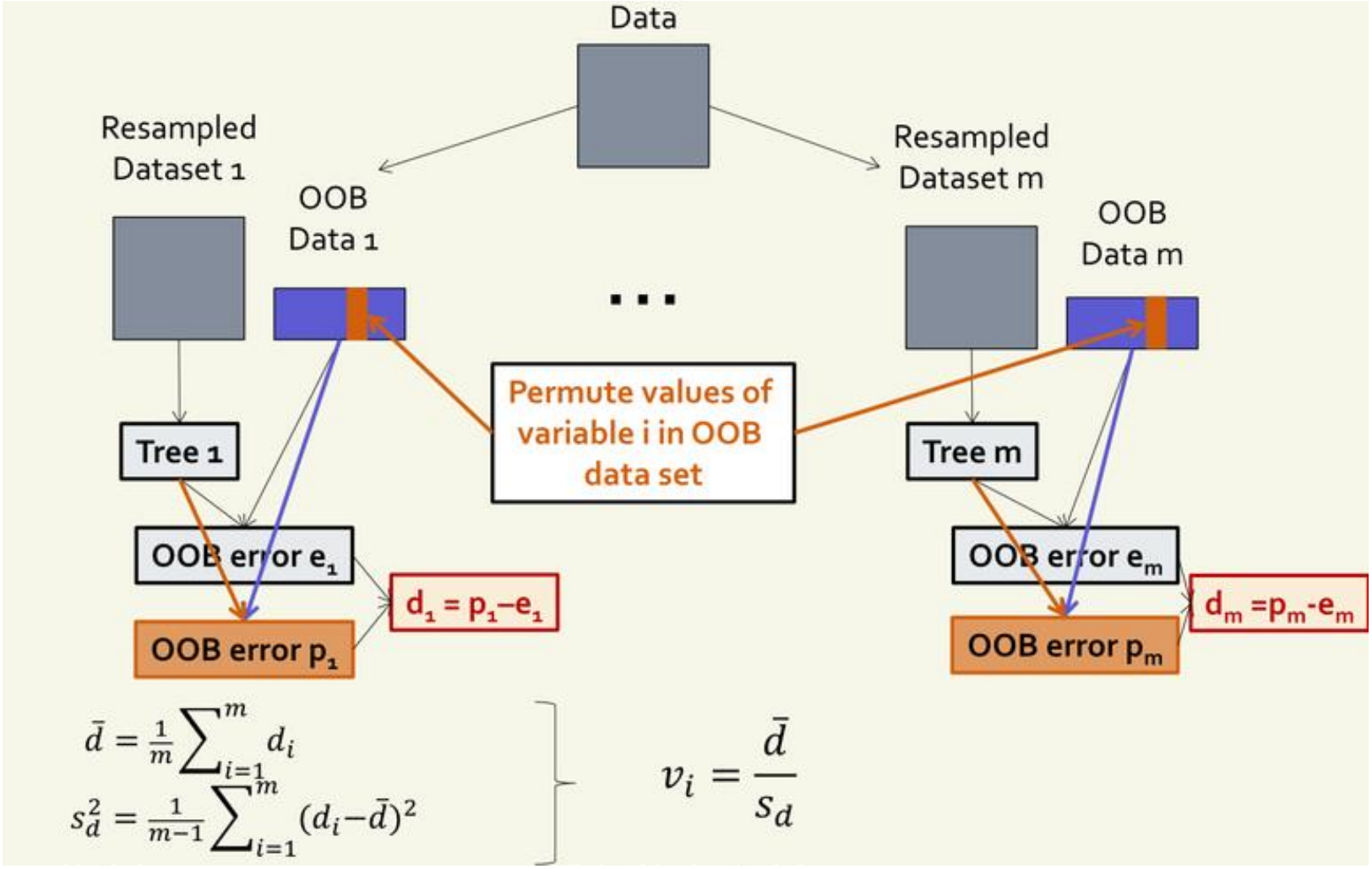
Random Forest

- **변수의 중요도**
  - 랜덤 포레스트는 다중선형 회귀분석/로지스틱 회귀분석과는 달리 **개별 변수가 통계적으로 얼마나 유의한지에 대한 정보를 제공하지 않음**
  - 대신 랜덤 포레스트는 다음과 같은 간접적인 방식으로 변수의 중요도를 추정함
    - 1단계: 원래 데이터 집합에 대해서 OOB Error를 구함
    - 2단계: 특정 변수의 값을 임의로 뒤섞은(random permutation) 데이터 집합에 대해서 OOB Error를 구함
    - 3단계: 개별 변수의 중요도는 2단계와 1단계 OOB Error의 평균과 분산을 고려하여 추정

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.
Ciss, S. (2015). Generalization Error and Out-of-bag Bounds in Random (Uniform) Forests.

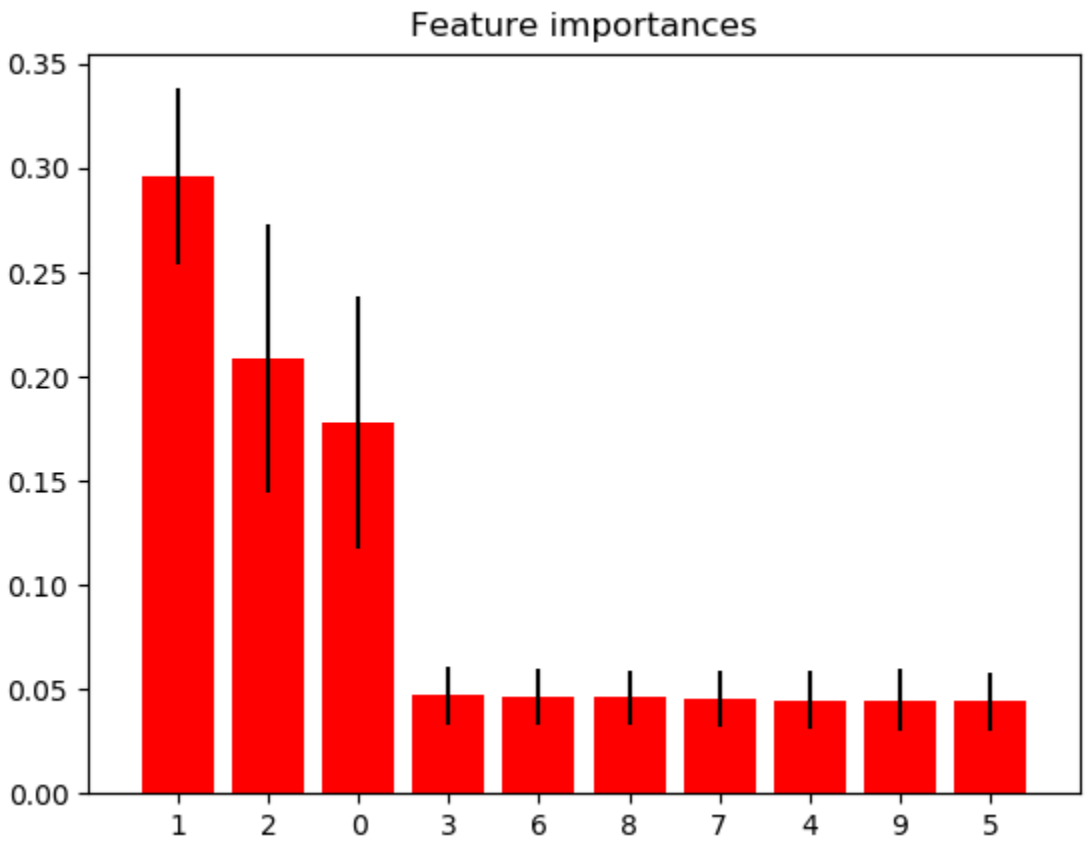
Random Forest

- 변수의 중요도



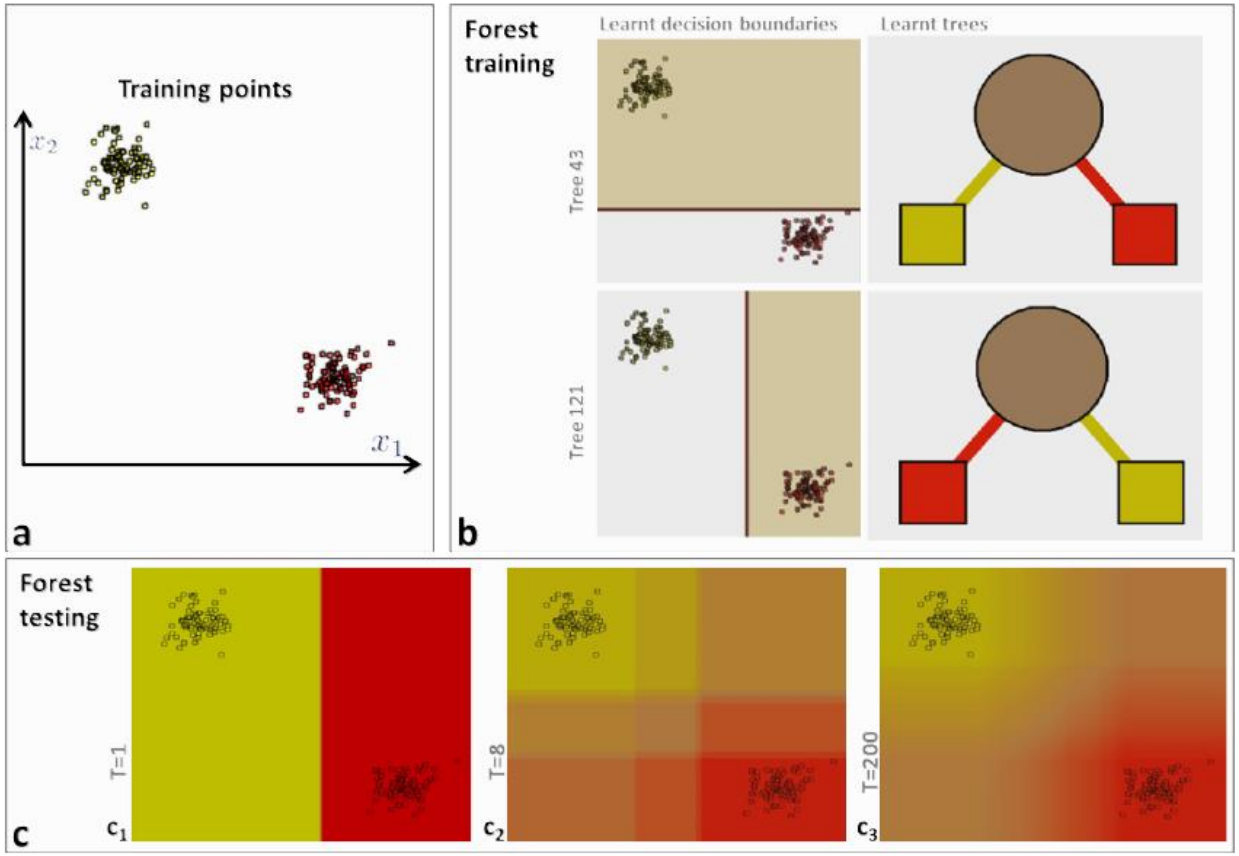
Random Forest

- 변수의 중요도



Random Forest

- **트리의 크기에 따른 분류 경계면의 변화**
  - 단일 트리는 이진 경계면을 생성
  - 랜덤 포레스트는 예측값이 보다 연속형의 스코어에 가깝게 생성됨



EOD