Ensemble Learning

Data Scientist
안건이

목차

Ensemble Overview

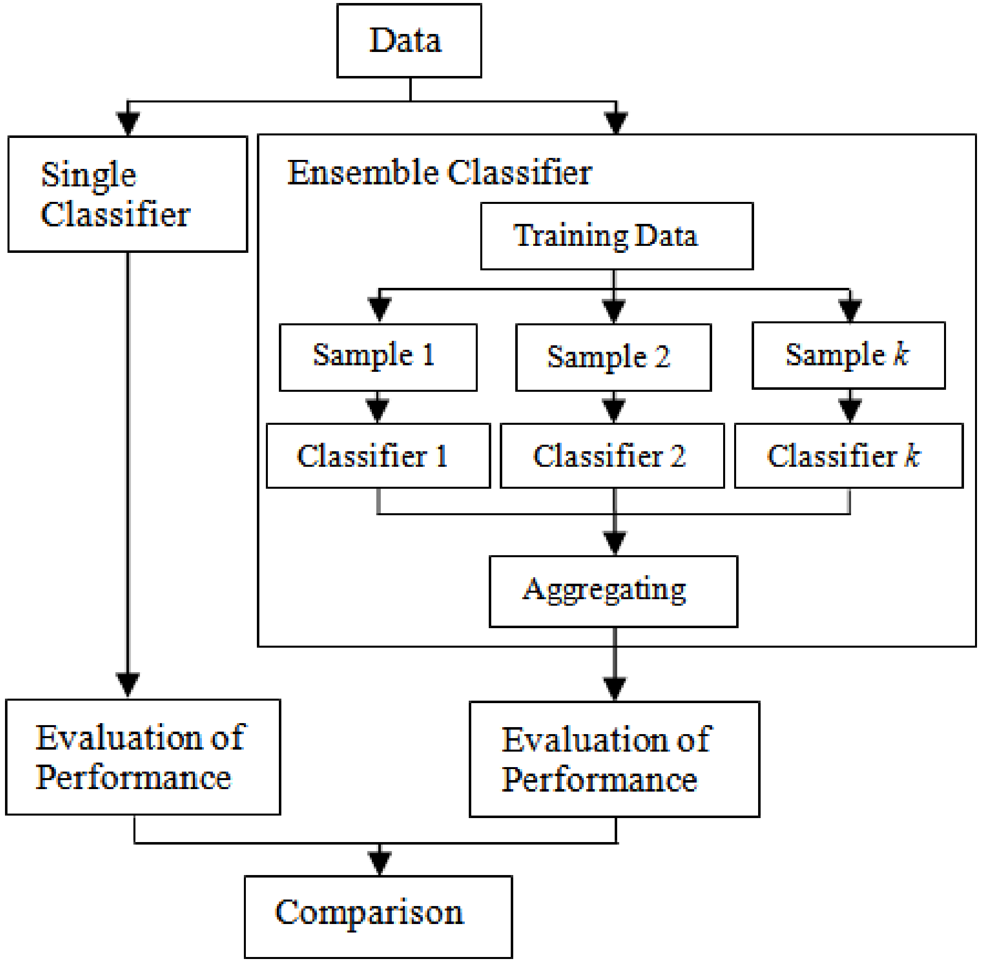
- Ensemble (앙상블)
  - 2인 이상에 의한 가창이나 연주
  - 조화 또는 통일을 의미함

- 어떤 데이터를 학습할 때, **여러 개의 모델을 조화롭게 학습**시켜 그 모델들의 예측 결과들을 이용하여 더 정확한 예측 값을 구할 수 있음

Ensemble Overview

- Ensemble (앙상블)
  - 2인 이상에 의한 가창이나 연주
  - 조화 또는 통일을 의미함

- 어떤 데이터를 학습할 때, **여러 개의 모델을 조화롭게 학습**시켜 그 모델들의 예측 결과들을 이용하여 더 정확한 예측 값을 구할 수 있음



✓Ensembles almost always work better

- Why Ensemble works?
  - ✓ True functions, estimations, and the expected error

$$y_m(\mathbf{x}) = f(\mathbf{x}) + \epsilon_m(\mathbf{x}). \quad \mathbb{E}_{\mathbf{x}}\left[\{y_m(\mathbf{x}) - f(\mathbf{x})\}^2\right] = \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right]$$

  - ✓ The average error made by M individual models vs. Expected error of the ensemble

$$E_{Avg} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right]$$

$$E_{Ensemble} = \mathbb{E}_{\mathbf{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x}) - f(\mathbf{x})\right\}^2\right]$$

$$= \mathbb{E}_{\mathbf{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\mathbf{x})\right\}^2\right]$$

Ensemble Overview

✓Ensembles almost always work better

- Why Ensemble works?
  - ✓True functions, estimations, and the expected error

$$y_m(\mathbf{x}) = f(\mathbf{x}) + \epsilon_m(\mathbf{x}). \quad \mathbb{E}_{\mathbf{x}}\left[\{y_m(\mathbf{x}) - f(\mathbf{x})\}^2\right] = \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right]$$

  - ✓The average error made by M individual models vs. Expected error of the ensemble

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

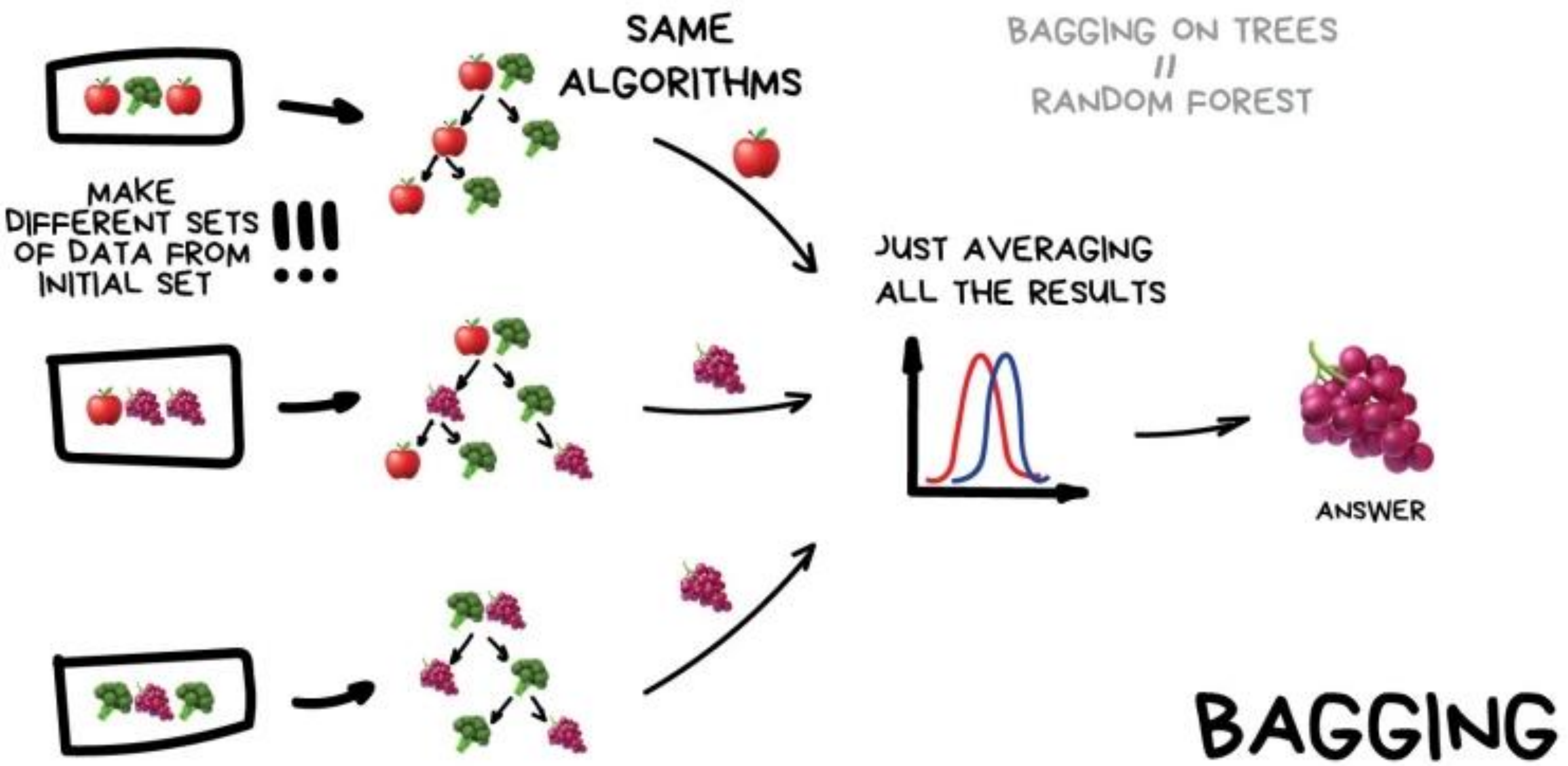
  - ✓In reality (errors are correlated), by the Cauchy's inequality

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

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

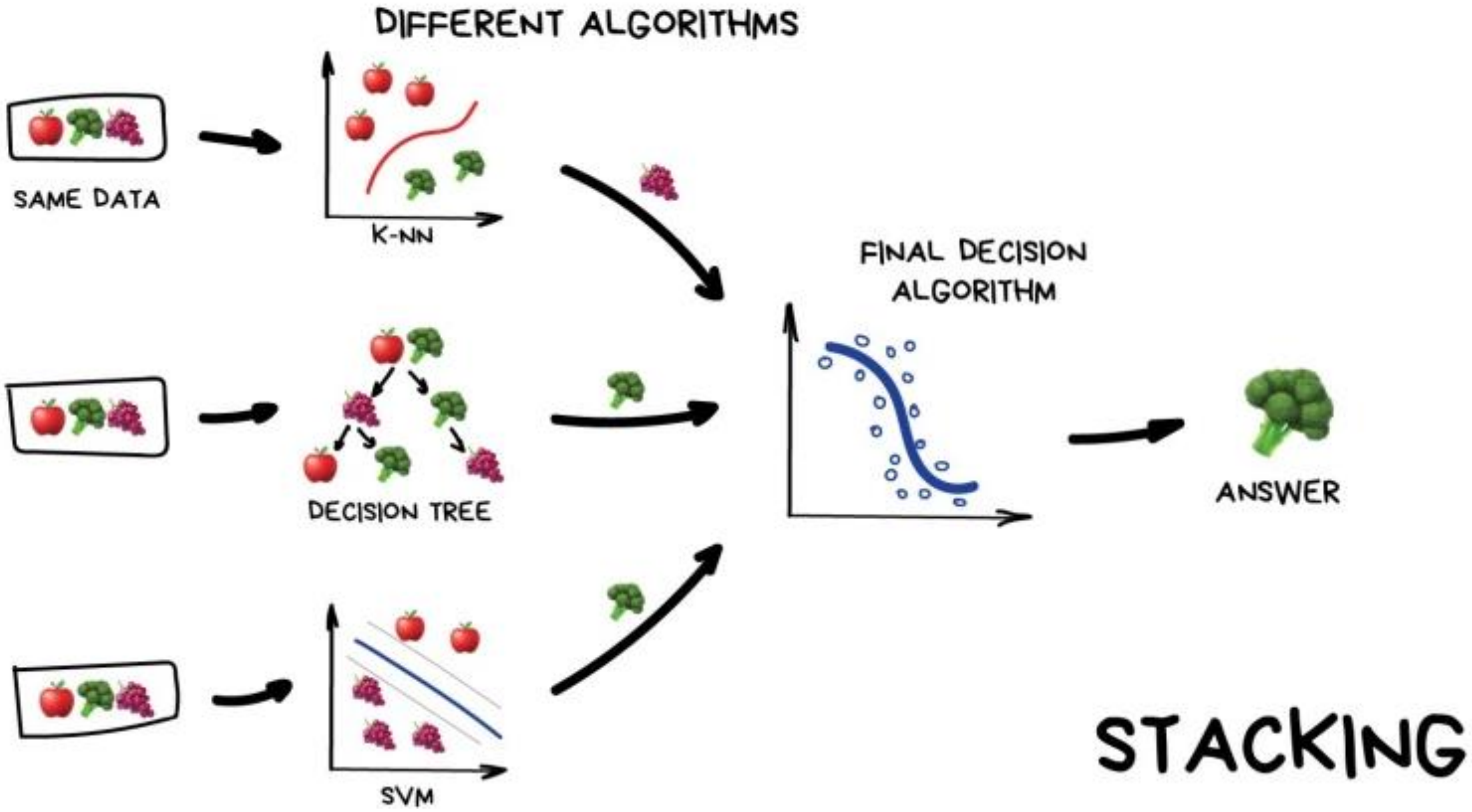
Ensemble Overview

- Ensemble (앙상블) 종류
  - Bagging : Reduce the Variance
    - Stacking : Use another prediction model
  - Boosting : Reduce the Bias



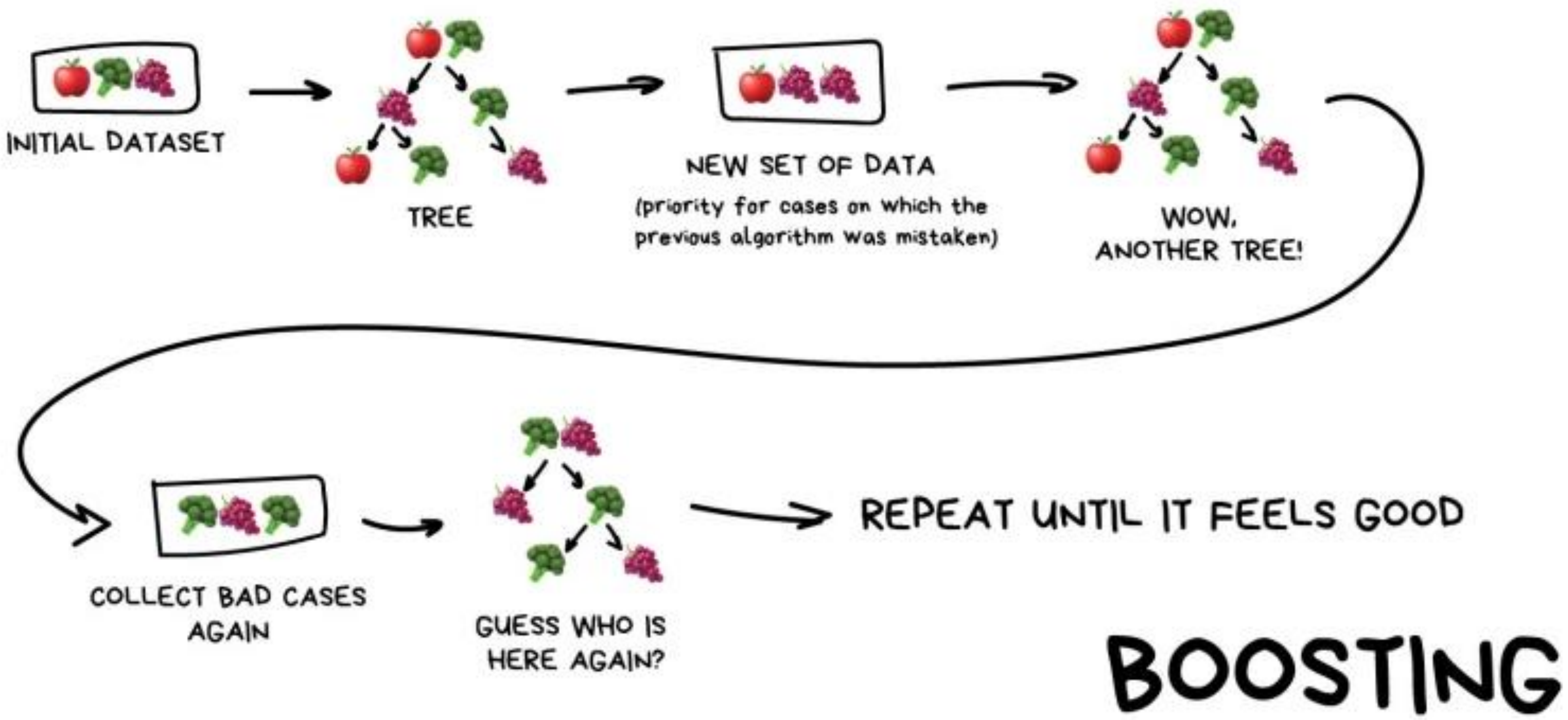
Ensemble Overview

- Ensemble (앙상블) 종류
  - Bagging : Reduce the Variance
    - Stacking : Use another prediction model
  - Boosting : Reduce the Bias



Ensemble Overview

- Ensemble (앙상블) 종류
  - Bagging : Reduce the Variance
    - Stacking : Use another prediction model
  - Boosting : Reduce the Bias



Bagging

- Bagging : **B**ootstrap **Agg**regat**ing**
  - Each member of the ensemble is constructed from a different training dataset
  - Each member is generated by sampling from the total N data examples, choosing N items uniformly at random with replacement
  - Each dataset sample is known as a bootstrap

Original Dataset		Bootstrap 1		Bootstrap 2			Bootstrap B	
x^1	y^1	x^3	y^3	x^7	y^7		x^9	y^9
x^2	y^2	x^6	y^6	x^1	y^1		x^5	y^5
x^3	y^3	x^2	y^2	x^{10}	y^{10}		x^2	y^2
x^4	y^4	x^{10}	y^{10}	x^1	y^1		x^4	y^4
x^5	y^5	x^8	y^8	x^8	y^8	…	x^7	y^7
x^6	y^6	x^7	y^7	x^6	y^6		x^2	y^2
x^7	y^7	x^7	y^7	x^2	y^2		x^5	y^5
x^8	y^8	x^3	y^3	x^6	y^6		x^{10}	y^{10}
x^9	y^9	x^2	y^2	x^4	y^4		x^8	y^8
x^{10}	y^{10}	x^7	y^7	x^9	y^9		x^2	y^2

Bagging

- Bagging : <u>**B**</u>ootstrap <u>**Agg**</u>regat<u>**ing**</u>

✓ 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

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

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

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

Bagging

- Bagging : **B**ootstrap **Agg**regat**ing**

✓ 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

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

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

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

Bagging

- Bagging : **B**ootstrap **Agg**regat**ing**

✓ 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

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

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

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

Bagging

- Bagging : **B**ootstrap **Agg**regat**ing**

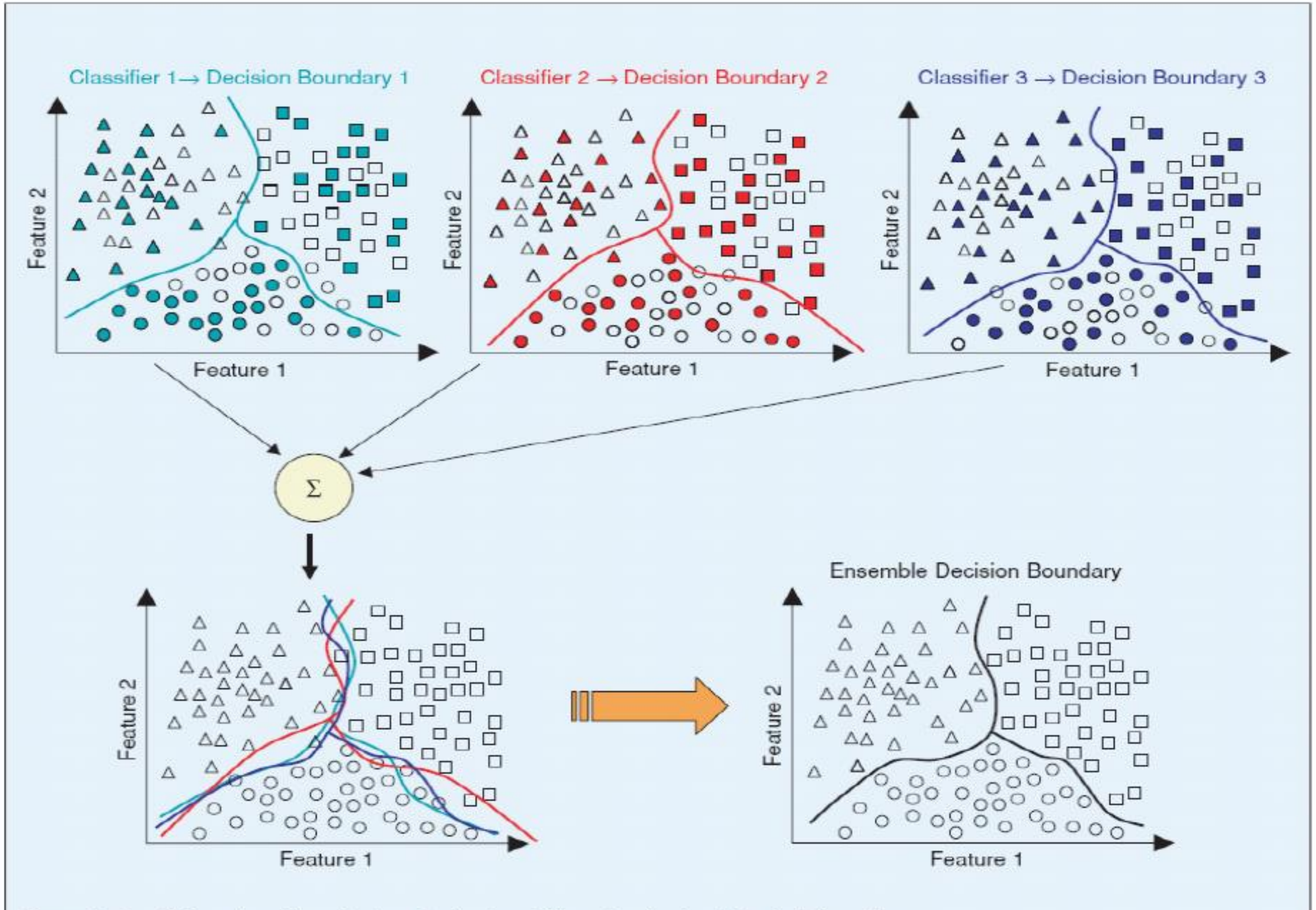


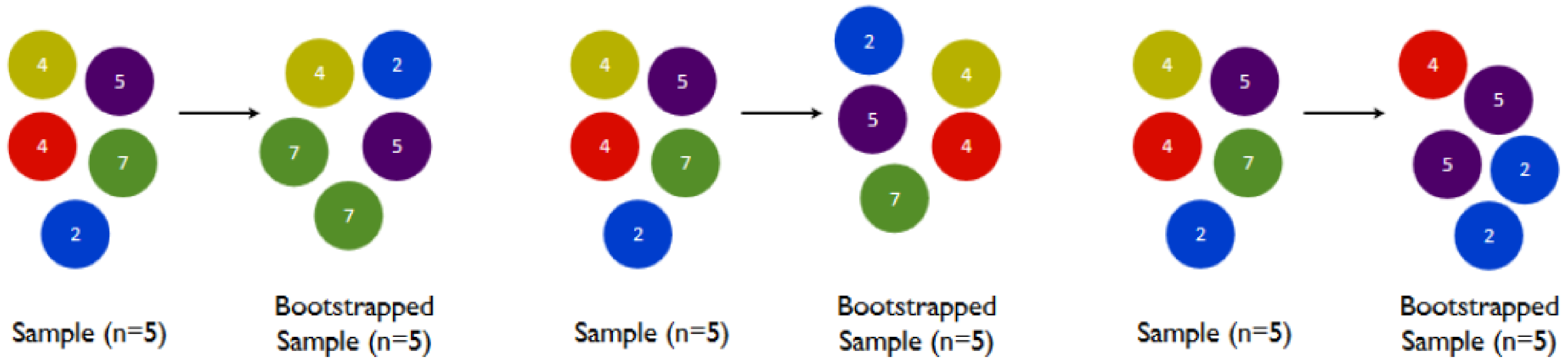
Figure 3. Combining classifiers that are trained on different subsets of the training data.

Bagging

- Bagging : <u>**B**</u>ootstrap <u>**Agg**</u>regat<u>**ing**</u>
  - **Out of Bag error**
    - **Use the training instances that are not sampled for validation**
    - **% of Not Sampled : 36.8%**

✓ Probability that an instance is not included in a bootstrap

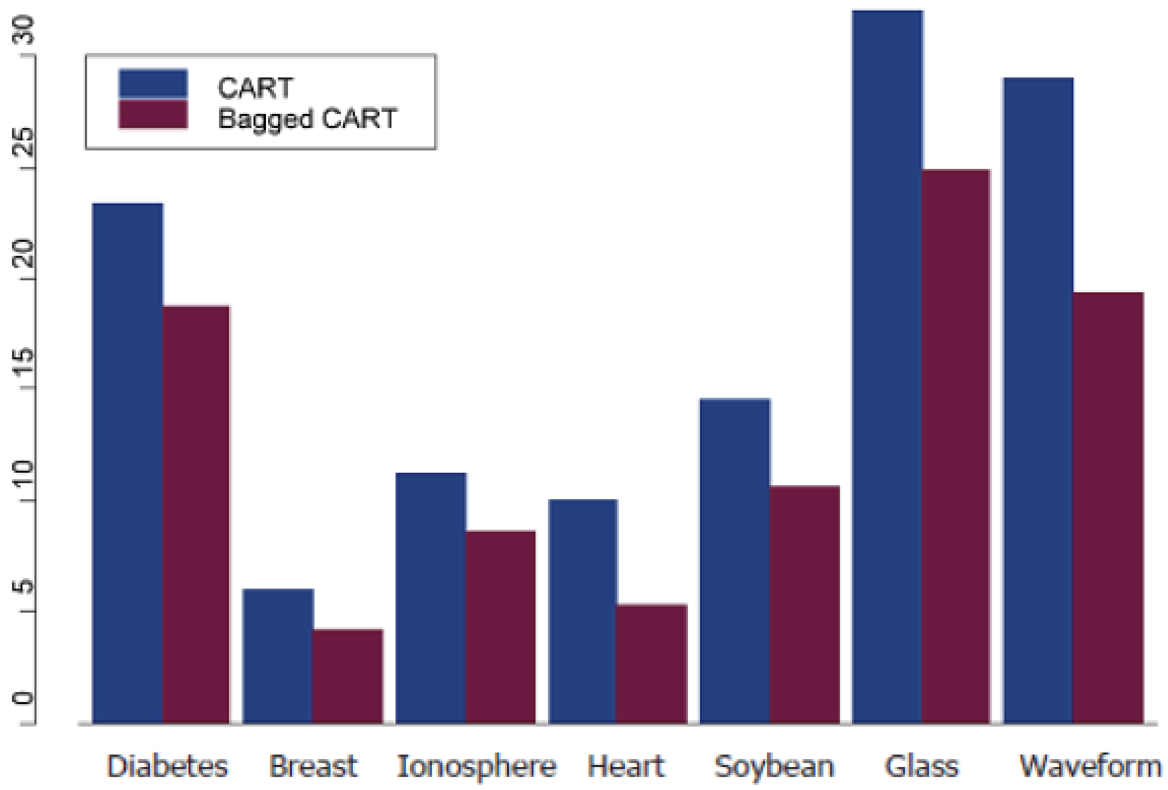
$$p = \left(1 - \frac{1}{N}\right)^N \quad \rightarrow \quad \lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = e^{-1} = 0.368$$



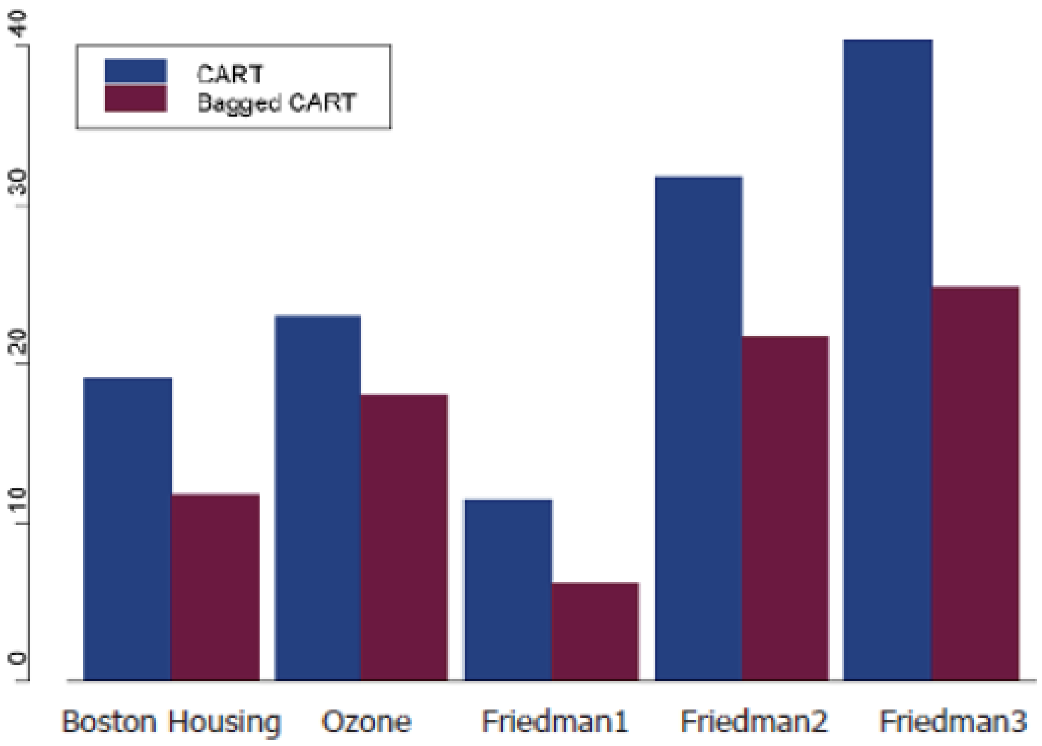
Bagging

- Bagged Trees vs Single Tree

Classification

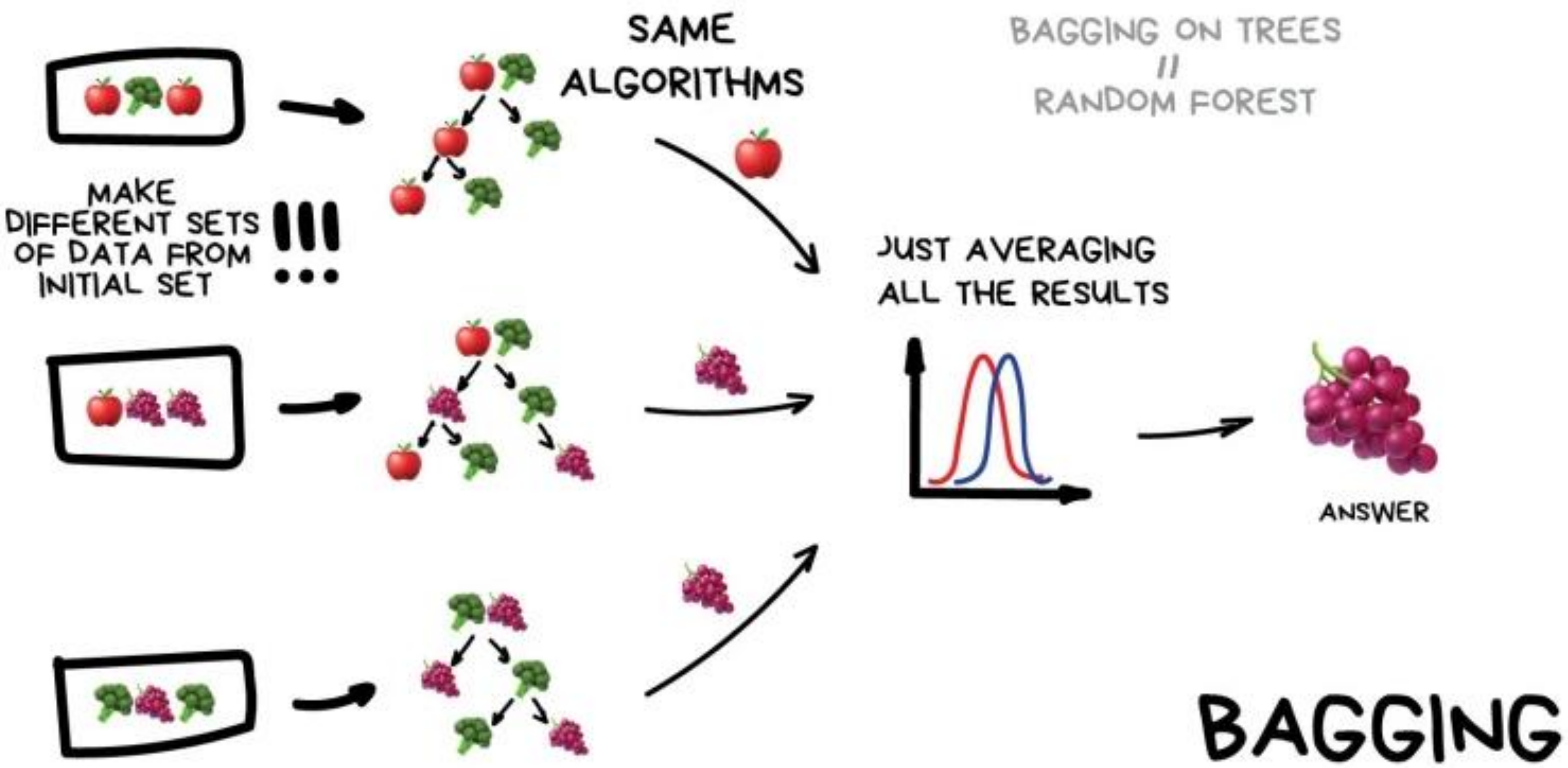


Regression



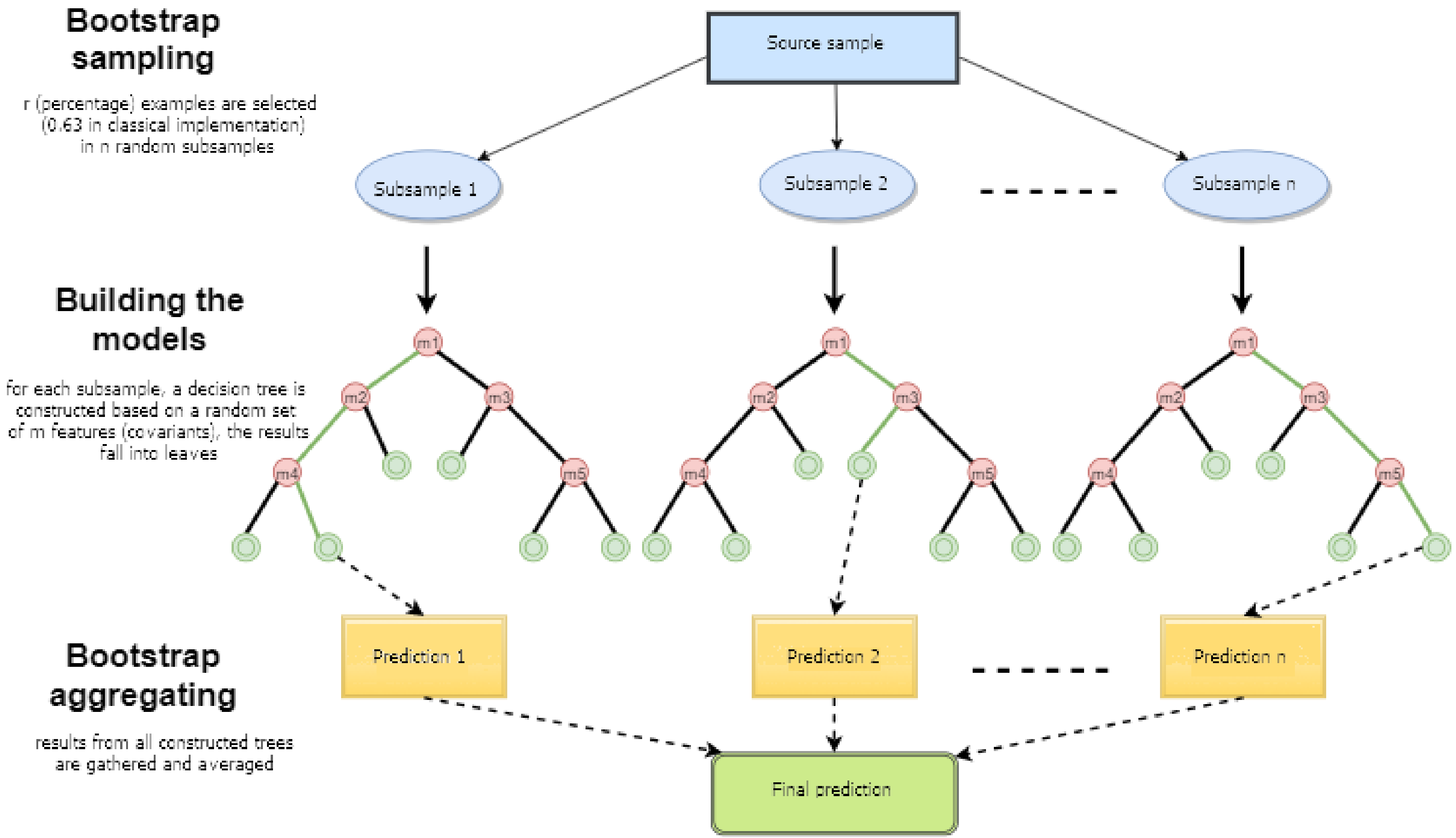
Ensemble Overview

- Ensemble (앙상블) 종류
  - Bagging : Reduce the Variance
    - Stacking : Use another prediction model
  - Boosting : Reduce the Bias



Random Forest

- Random Forest
  - A specialized bagging for decision tree algorithms
  - Two ways to increase the diversity of ensemble
    - Bagging
    - Randomly chosen predictor variables



Random Forest

- Random Forest
  - A specialized bagging for decision tree algorithms
  - Two ways to increase the diversity of ensemble
    - Bagging
    - Randomly chosen predictor variables

Algorithm 1: Pseudo code for the random forest algorithm

To generate c classifiers:
for $i = 1$ to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N , N_1 ,..., N_f , where F has f possible values (F_1 , … , F_f)
  for $i = 1$ to f **do**
    Set the contents of N_i to D_i , where D_i is all instances in N that match F_i
    Call BuildTree(N_i)
  end for
end if

Random Forest

- Random Forest
  - A specialized bagging for decision tree algorithms
  - Two ways to increase the diversity of ensemble
    - Bagging
    - Randomly chosen predictor variables → Why Random?

Generalization Error

❖ Tree는 작은 Bias와 큰 Variance를 갖기 때문에, 매우 깊이 성장한(Depth가 깊은) 트리는 훈련 데이터에 대해 Overfitting 하게 됨

❖ 한 개의 Tree의 경우 훈련 데이터에 있는 Noise에 대해 매우 민감함

  ❖ Tree들이 서로 상관화(correlated)되어 있지 않다면 여러 Tree들의 평균은 Noise에 대해 강인해짐

  ❖ 상관화를 줄이는 방법은 → **Randomly Chosen (N & P 모두)**

  ❖ 반면, Forest를 구성하는 모든 Tree들을 동일한 데이터 셋으로만 훈련시키게 되면, Tree들의 상관성은 커짐

❖ 따라서 Bagging은 서로 다른 데이터 셋들에 대해 훈련 시킴으로써, Tree들을 비상관화 시켜줌

  ❖ **Bias는 유지하면서 Variance를 낮춤**

Random Forest

- Random Forest
  - Variable Importance
    - Step 1 : Compute the OOB(Out of bag) error for the original dataset
    - Step 2 : Compute the OOB error for the dataset in which the variable x_i is permuted p_i
    - Step 3 : Compute the variable importance based on the mean and standard deviation of over all trees in the population

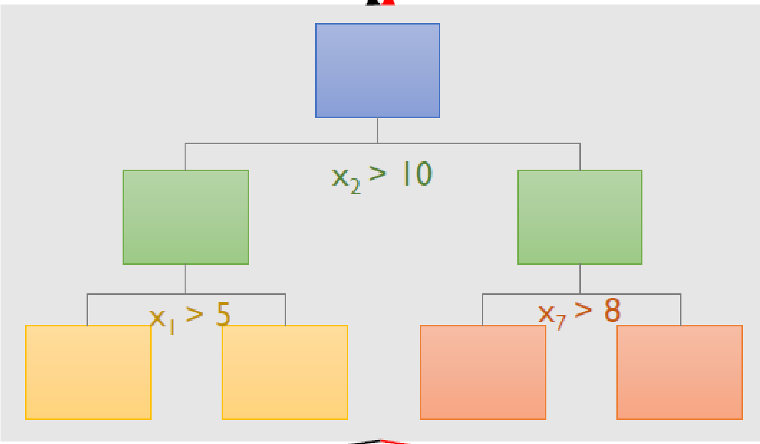
Original OOB Data

ID	X1	...	Xi	...	Xd	Y
1			0.1			
2			0.5			
3			1.1			
4			1.2			
5			0.4			
6			0.2			
7			0.7			
8			0.8			
9			1.4			
10			1.6			

i번째 변수에 대한 random permutation이 수행된 OOB Data

ID	X1	...	Xi	...	Xd	Y
1			1.1			
2			0.2			
3			0.1			
4			1.4			
5			1.2			
6			0.5			
7			1.6			
8			0.8			
9			0.7			
10			0.4			

변수 i가 Tree를 split하는데
한번도 사용되지 않았다면



OOB Error of the Original Data e_i = OOB Error of the Permuted Data p_i

Random Forest

- Random Forest
  - Variable Importance
    - Step 1 : Compute the OOB(Out of bag) error for the original dataset
    - Step 2 : Compute the OOB error for the dataset in which the variable x_i is permuted p_i
    - Step 3 : Compute the variable importance based on the mean and standard deviation of over all trees in the population

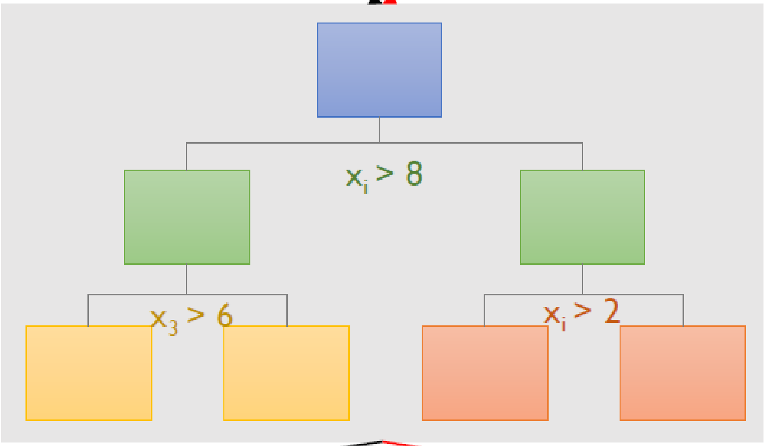
Original OOB Data

ID	X1	...	Xi	...	Xd	Y
1			0.1			
2			0.5			
3			1.1			
4			1.2			
5			0.4			
6			0.2			
7			0.7			
8			0.8			
9			1.4			
10			1.6			

i번째 변수에 대한 random permutation이 수행된 OOB Data

ID	X1	...	Xi	...	Xd	Y
1			1.1			
2			0.2			
3			0.1			
4			1.4			
5			1.2			
6			0.5			
7			1.6			
8			0.8			
9			0.7			
10			0.4			

변수 i가 Tree를 split하는데 중요하게 사용되었다면



OOB Error of the Original Data e_i < OOB Error of the Permuted Data p_i

Random Forest

- Random Forest
  - Variable Importance
    - Step 1 : Compute the OOB(Out of bag) error for the original dataset
    - Step 2 : Compute the OOB error for the dataset in which the variable x_i is permuted p_i
    - Step 3 : Compute the variable importance based on the mean and standard deviation of over all trees in the population

✓ 랜덤 포레스트에서 변수의 중요도가 높다면

- 1) Random permutation 전-후의 OOB Error 차이가 크게 나타나야 하며,
- 2) 그 차이의 편차가 적어야 함

- m번째 tree에서 변수 i에 대한 Random permutation 전후 OOB error의 차이

$$d_i^m = p_i^m - e_i^m$$

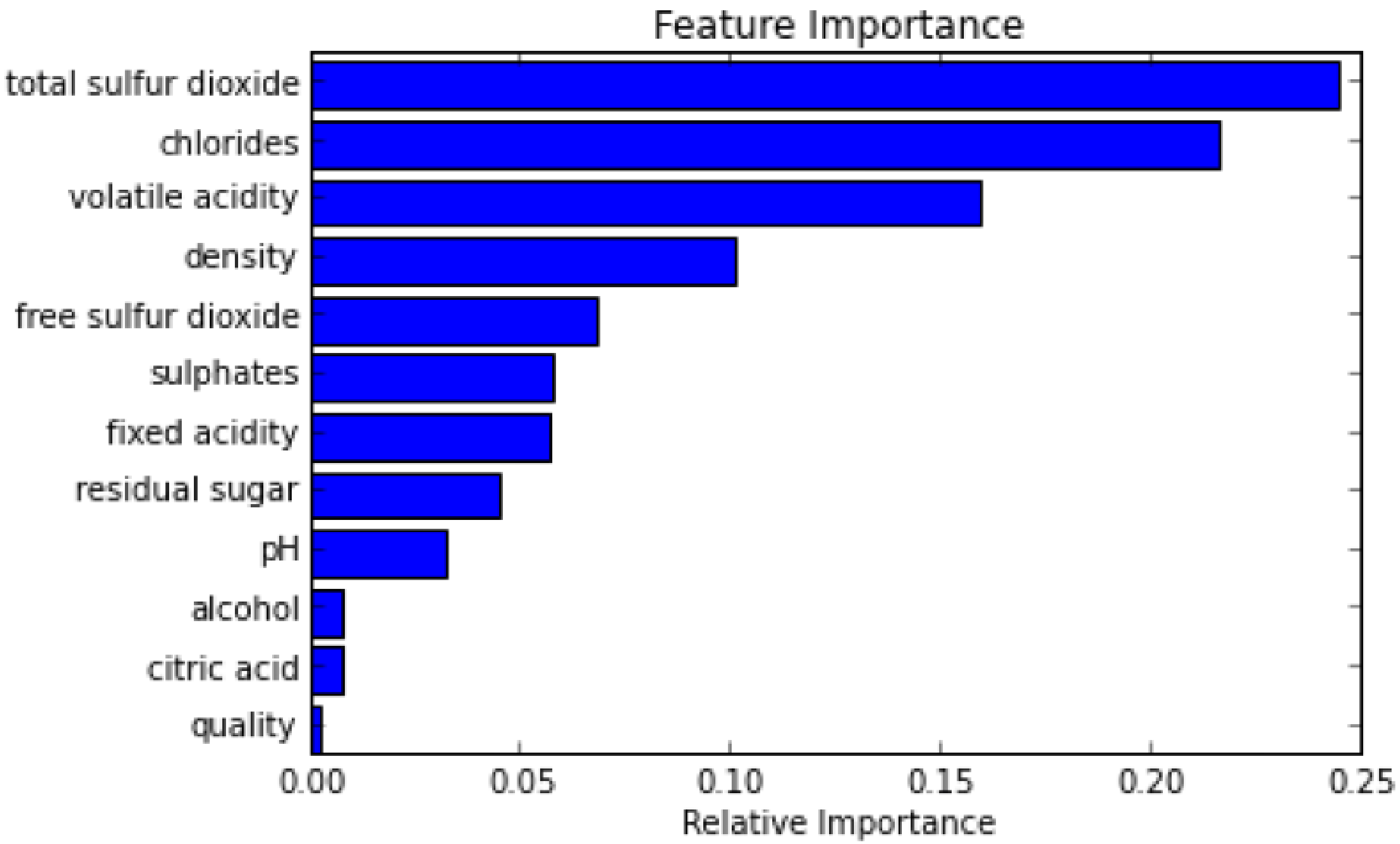
- 전체 Tree들에 대한 OOB error 차이의 평균 및 분산

$$\bar{d}_i = \frac{1}{m}\sum_{i=1}^{m} d_i^m, \quad s_i^2 = \frac{1}{m-1}\sum_{i=1}^{m}(d_i^m - \bar{d}_i)^2$$

- i번째 변수의 중요도: $v_i = \dfrac{\bar{d}_i}{s_i}$

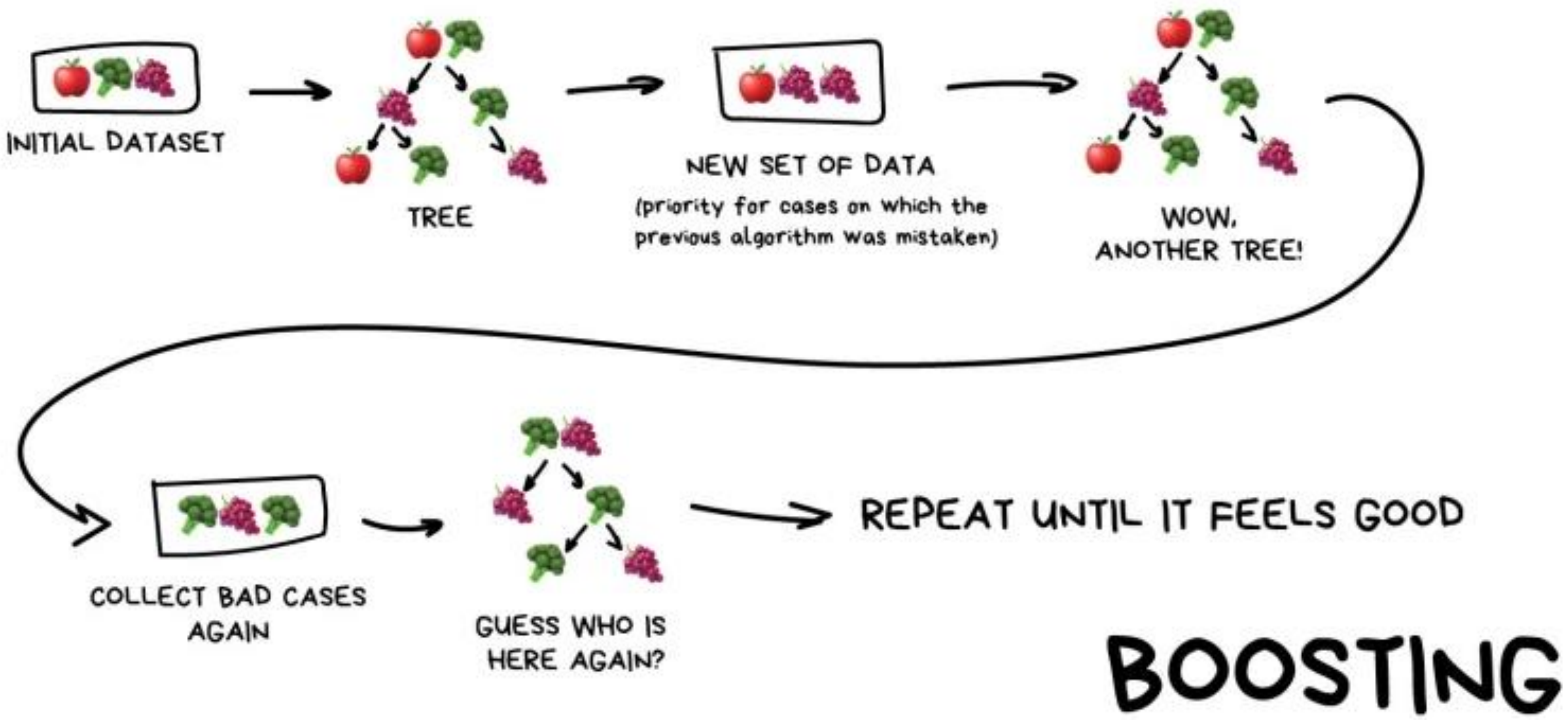
Random Forest

- Random Forest
  - Variable Importance
    - Step 1 : Compute the OOB(Out of bag) error for the original dataset
    - Step 2 : Compute the OOB error for the dataset in which the variable x_i is permuted p_i
    - Step 3 : Compute the variable importance based on the mean and standard deviation of over all trees in the population



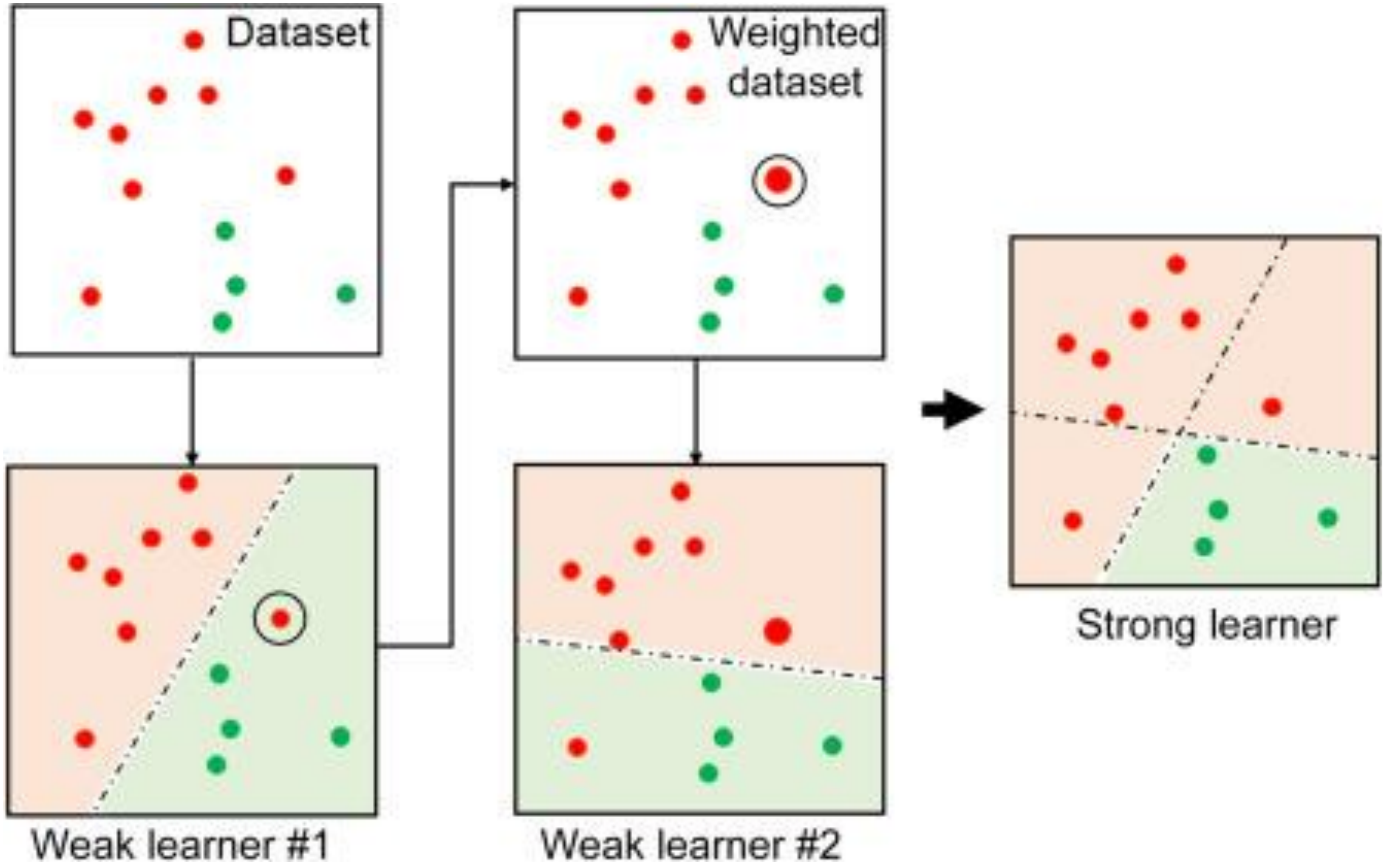
Ensemble Overview

- Ensemble (앙상블) 종류
  - Bagging : Reduce the Variance
    - Stacking : Use another prediction model
  - Boosting : Reduce the Bias



AdaBoost

- AdaBoost = Adaptive + Boosting
  - Boosting : An iterative procedure to adaptively change distribution of training data by focusing more on previously mis-classified records
  - Parallel한 과정이 아니라 **Sequential한 과정**
  - Strong Classifier vs Weak Classifier
    - Strong Classifier : 개별 약분류기들에 각각 가중치를 적용, 조합하여 얻을 수 있음
    - Weak Classifier : 개별 약분류기



AdaBoost

- AdaBoost = Adaptive + Boosting
  - Boosting : An iterative procedure to adaptively change distribution of training data by focusing more on previously mis-classified records
  - Parallel한 과정이 아니라 **Sequential한 과정**
  - Strong Classifier vs Weak Classifier
    - Strong Classifier : 개별 약분류기들에 각각 가중치를 적용, 조합하여 얻을 수 있음
    - Weak Classifier : 개별 약분류기

Algorithm 2 Adaboost

Input: Required ensemble size T

Input: Training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where $y_i \in \{-1, +1\}$

Define a uniform distribution $D_1(i)$ over elements of S.

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

  Train a model h_t using distribution D_t.

  Calculate $\epsilon_t = P_{D_t}(h_t(x) \neq y)$

  If $\epsilon_t \geq 0.5$ break

  Set $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

  Update $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

  where Z_t is a normalization factor so that D_{t+1} is a valid distribution.

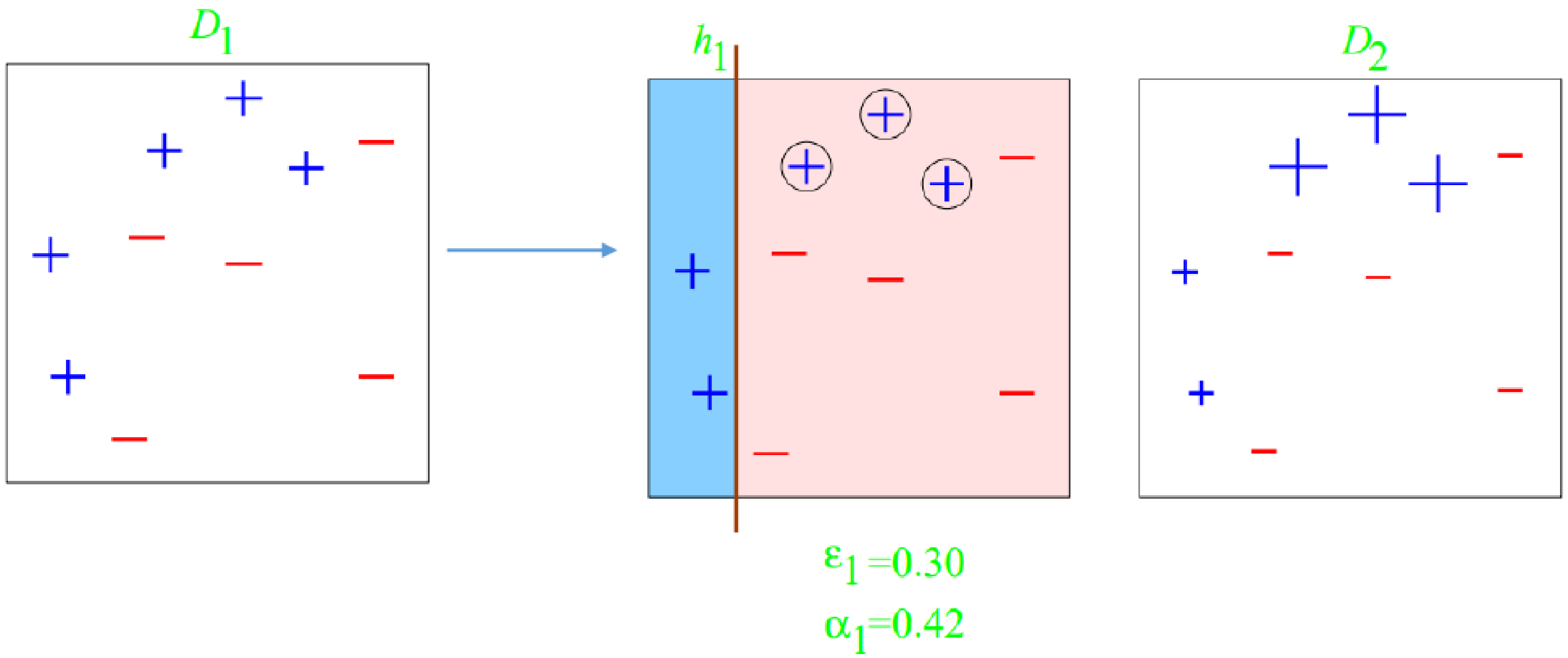
end for

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

$H(x') = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x')\right)$

AdaBoost

- AdaBoost = Adaptive + Boosting
  - Round 1



- 3 misclassifications out of 10: $\epsilon_i = 0.30$
- Model confidence: $\alpha_i = \frac{1}{2} \log \left(\frac{1 - \epsilon_i}{\epsilon_i} \right) = \frac{1}{2} \log \frac{1 - 0.3}{0.3} = 0.42$

AdaBoost

- AdaBoost = Adaptive + Boosting
  - Round 1

✓ The selection probability of x_i for the next training dataset

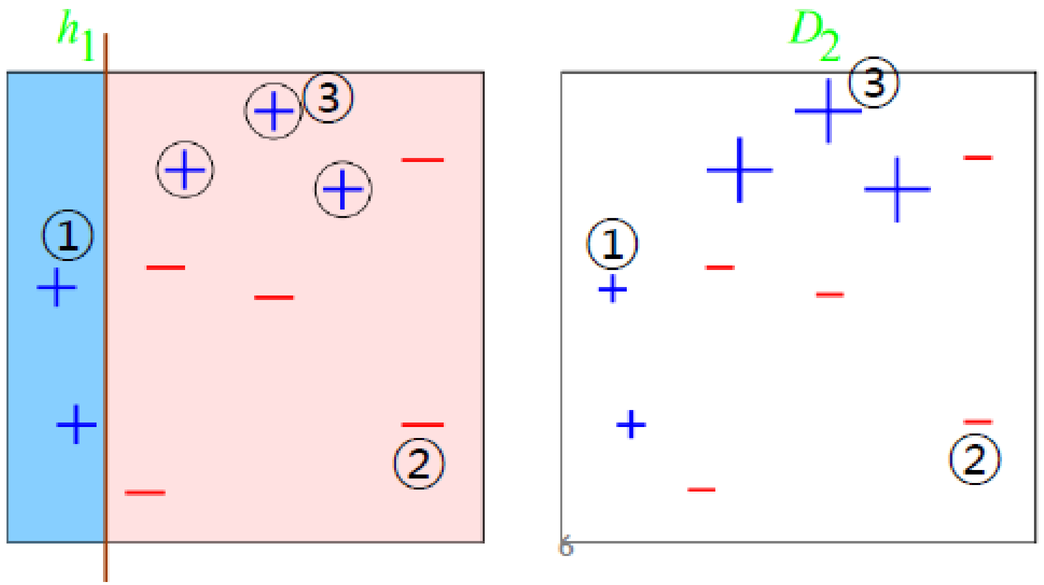
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

✓ Case 1: $y_i = 1, h_t(x_i) = 1 \rightarrow y_i h_t(x_i) = 1 \rightarrow -\alpha_t y_i h_t(x_i) < 0$ → decrease p

✓ Case 2: $y_i = -1, h_t(x_i) = -1 \rightarrow y_i h_t(x_i) = 1 \rightarrow -\alpha_t y_i h_t(x_i) < 0$ → decrease p

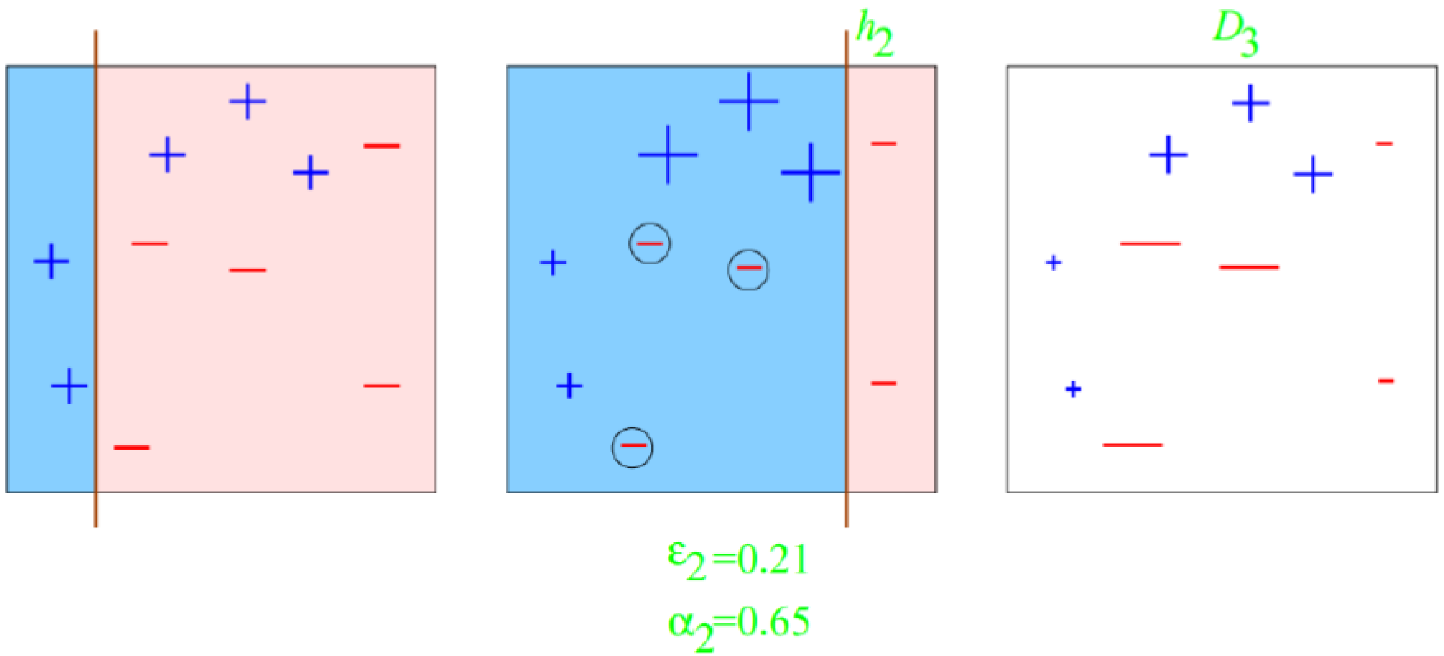
✓ Case 3: $y_i = 1, h_t(x_i) = -1 \rightarrow y_i h_t(x_i) = -1 \rightarrow -\alpha_t y_i h_t(x_i) > 0$ → increase p

✓ α_t is the confidence of the current model that controls the magnitude of change



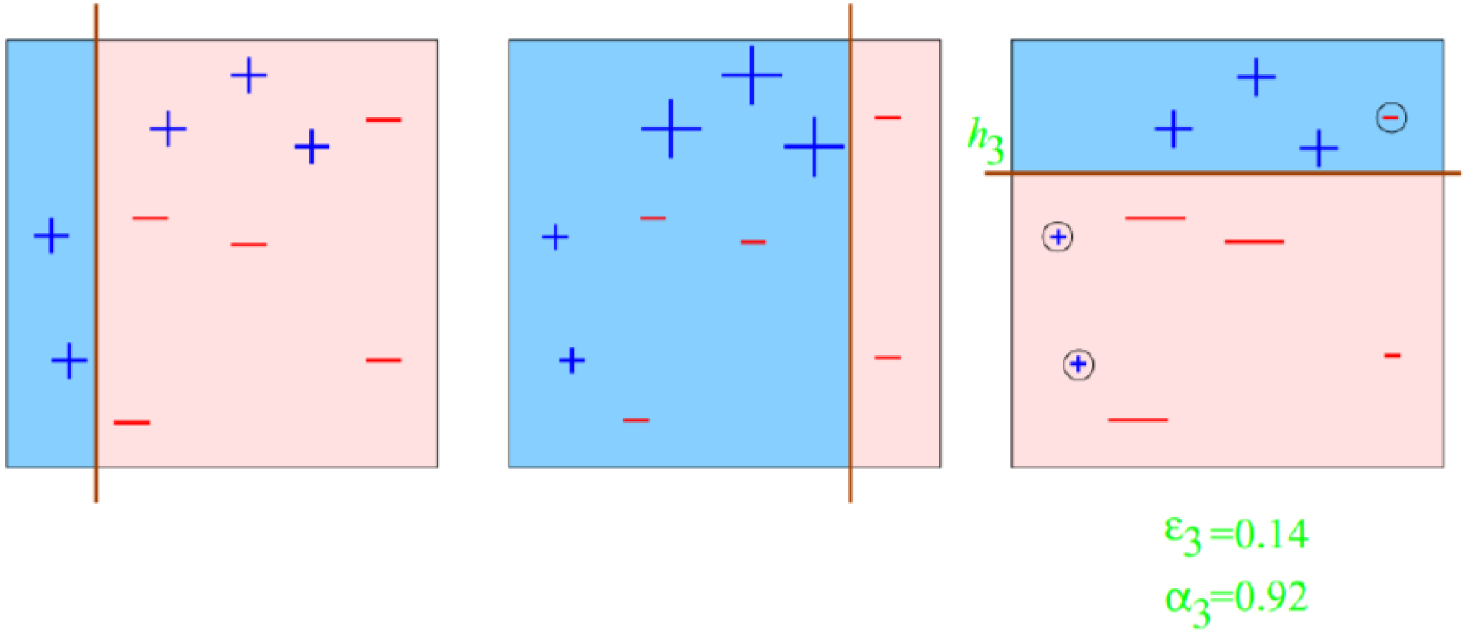
AdaBoost

- AdaBoost = Adaptive + Boosting
  - Round 2



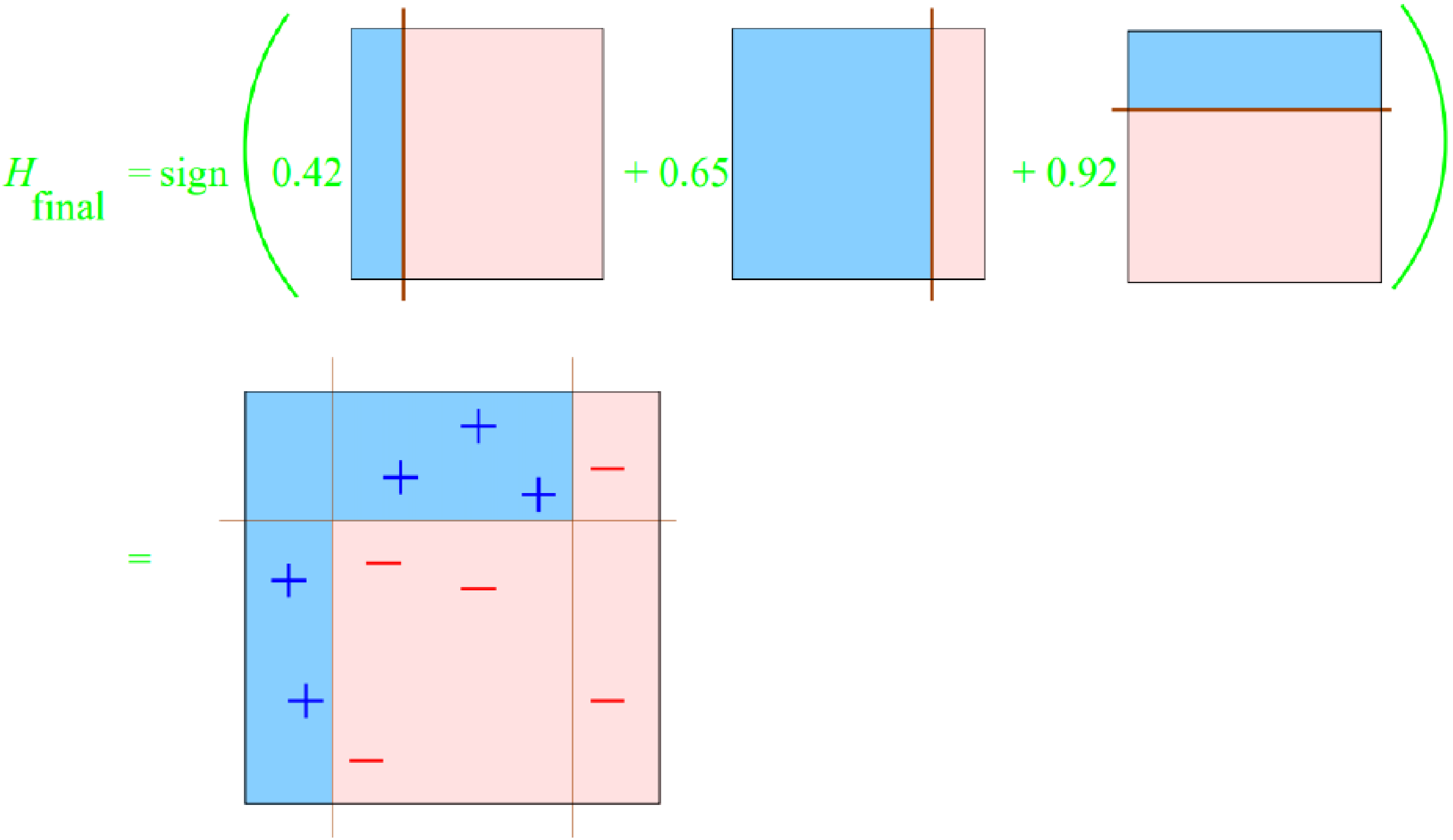
AdaBoost

- AdaBoost = Adaptive + Boosting
  - Round 3



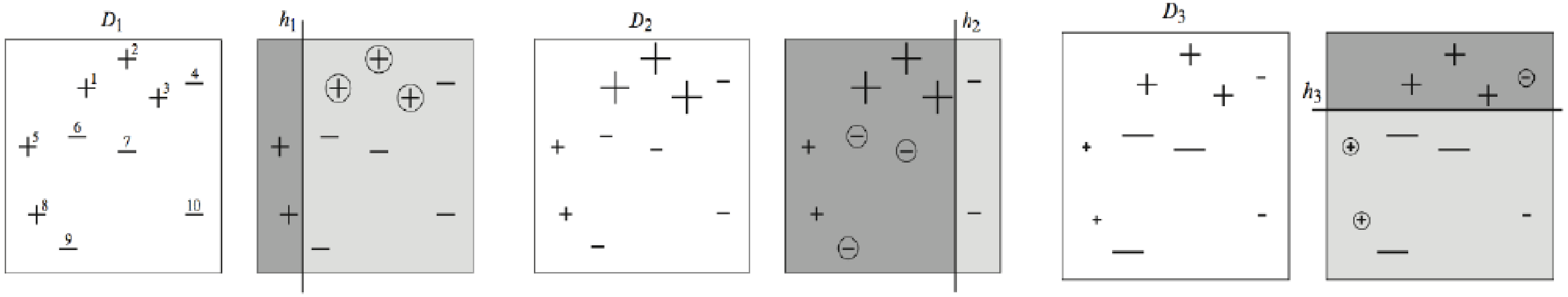
AdaBoost

- AdaBoost = Adaptive + Boosting
  - Final Classifier = Strong Classifier

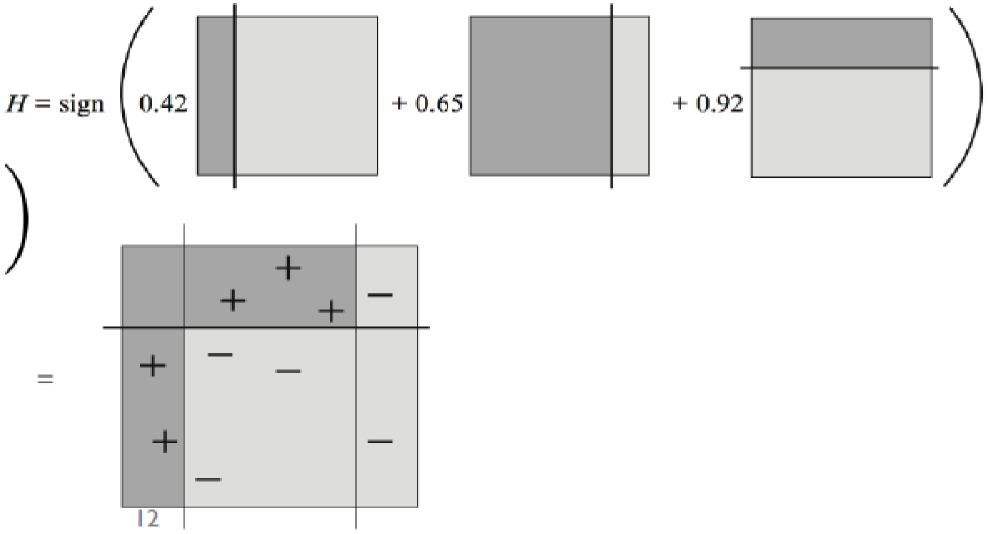


AdaBoost

- AdaBoost = Adaptive + Boosting
  - Round 1

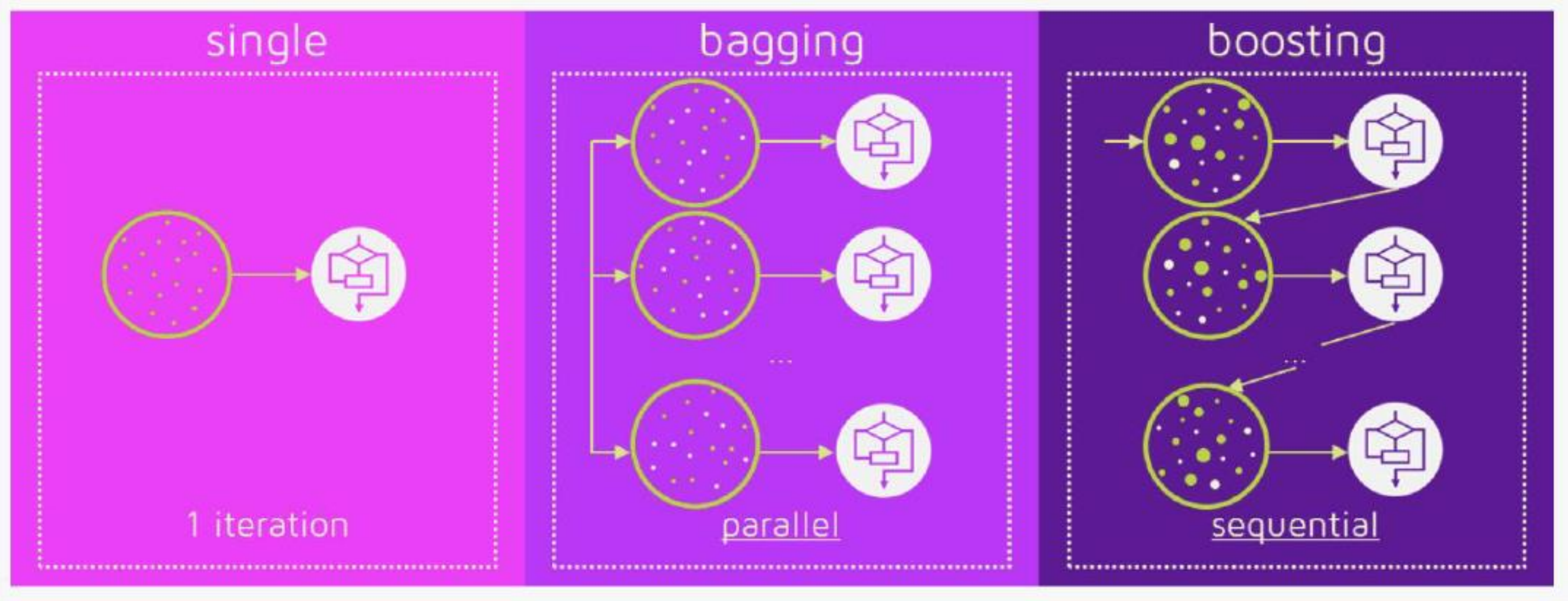


$$H(x') = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x')\Big)$$



Summary

- Single vs Bagging vs Boosting



https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Q & A