

교재 05-3장

---

# 트리의 앙상블

---

2022.11.16 (수)

2조 강유정 이두진 이주석 임홍주

# 목차

01. 데이터 종류

02. 앙상블 학습

03. Bagging - 랜덤 포레스트

04. Bagging - 엑스트라 트리



# 01. 데이터 종류

# 정형 데이터

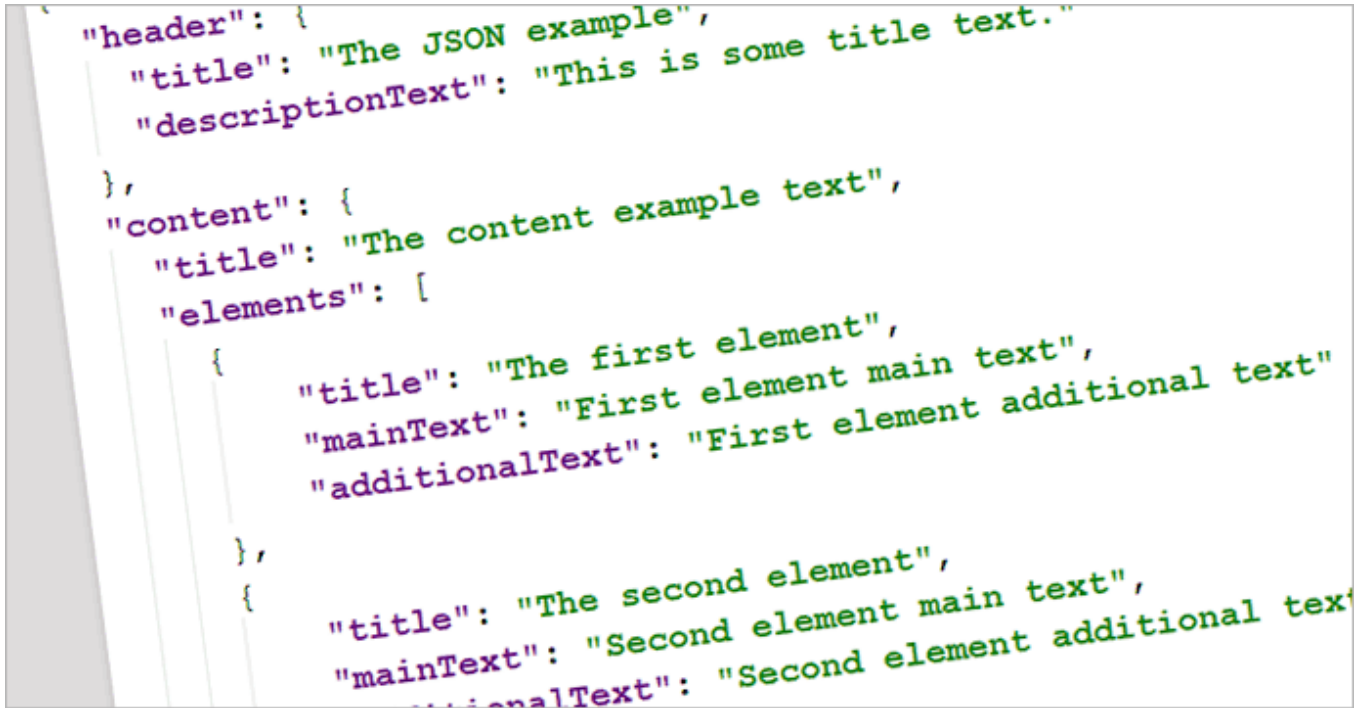
	A	B	C	D	E	F	G	H
1	일자	요일	시간대	업종	시도	시군구	읍면동	통화건수
2	20180601	금	0	음식점-죽	서울특별시	강남구	논현동	5
3	20180601	금	0	음식점-죽	서울특별시	강동구	길동	5
4	20180601	금	0	음식점-죽	서울특별시	강서구	내발산동	5
5	20180601	금	0	음식점-죽	서울특별시	동대문구	제기동	5
6	20180601	금	0	음식점-죽	서울특별시	서대문구	창천동	7
7	20180601	금	0	음식점-죽	서울특별시	서초구	양재동	5
8	20180601	금	0	음식점-죽	서울특별시	성동구	성수동2가	5
9	20180601	금	0	음식점-죽	서울특별시	성북구	동선동2가	5
10	20180601	금	0	음식점-죽	서울특별시	송파구	송파동	5
11	20180601	금	0	음식점-죽	서울특별시	영등포구	문래동3가	5

- 구조화된 데이터, 즉 미리 정해진 구조에 따라 저장된 데이터
- 수치 만으로도 의미 파악이 쉬움

ex) 엑셀의 스프레드시트, 관계 데이터베이스의 테이블



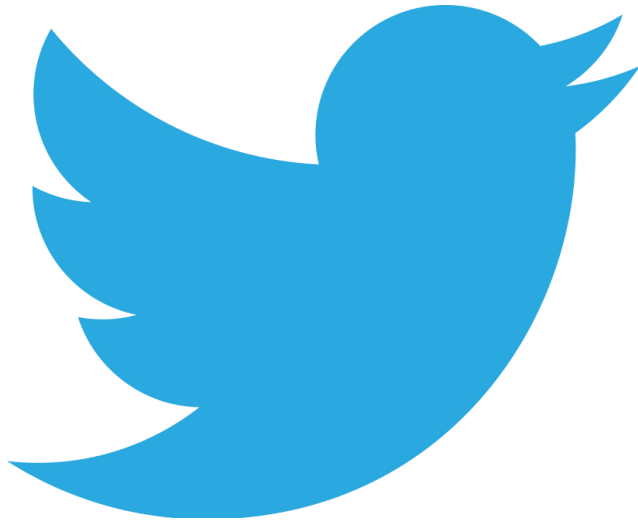
# 반정형 데이터



```
"header": {  
  "title": "The JSON example",  
  "descriptionText": "This is some title text."  
},  
"content": {  
  "title": "The content example text",  
  "elements": [  
    {  
      "title": "The first element",  
      "mainText": "First element main text",  
      "additionalText": "First element additional text"  
    },  
    {  
      "title": "The second element",  
      "mainText": "Second element main text",  
      "additionalText": "Second element additional text"  
    }  
  ]  
}
```

- 데이터의 구조 정보를 데이터와 함께 제공하는 파일 형식의 데이터
  - 데이터의 형식과 구조가 변경될 수 있음
- ex) XML, HTML, JSON

## 비정형 데이터

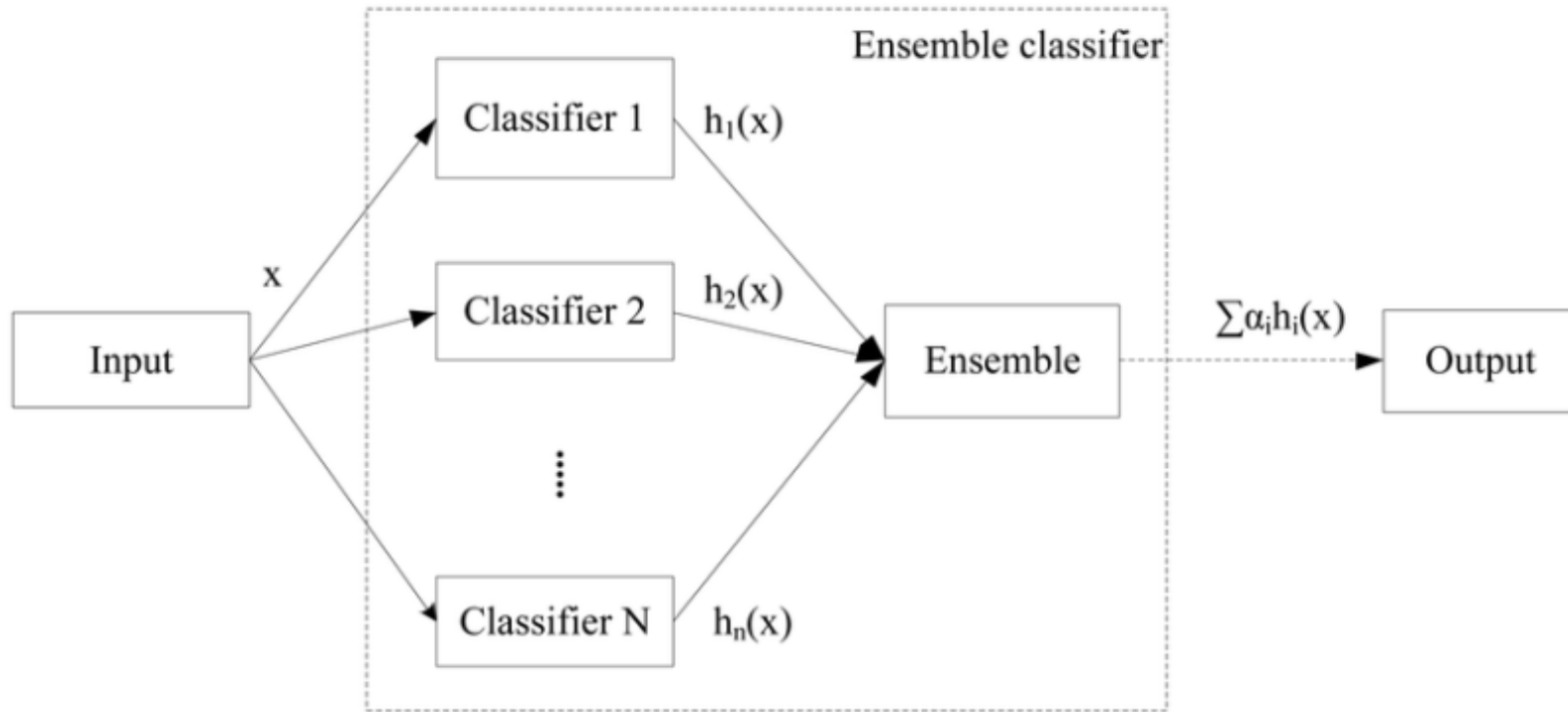


- 정해진 구조가 없이 저장된 데이터
  - 형태가 없어 연산이 불가
- ex) 소셜 데이터의 텍스트, 이미지, 영상



## 02. **양상블** 학습

# 앙상블 학습 의미



- 여러 개의 weak learner들이 모여 투표 (voting)를 통해 더욱더 강력한 strong learner를 구성
- 일반화된(generalized) 모델 생성 가능

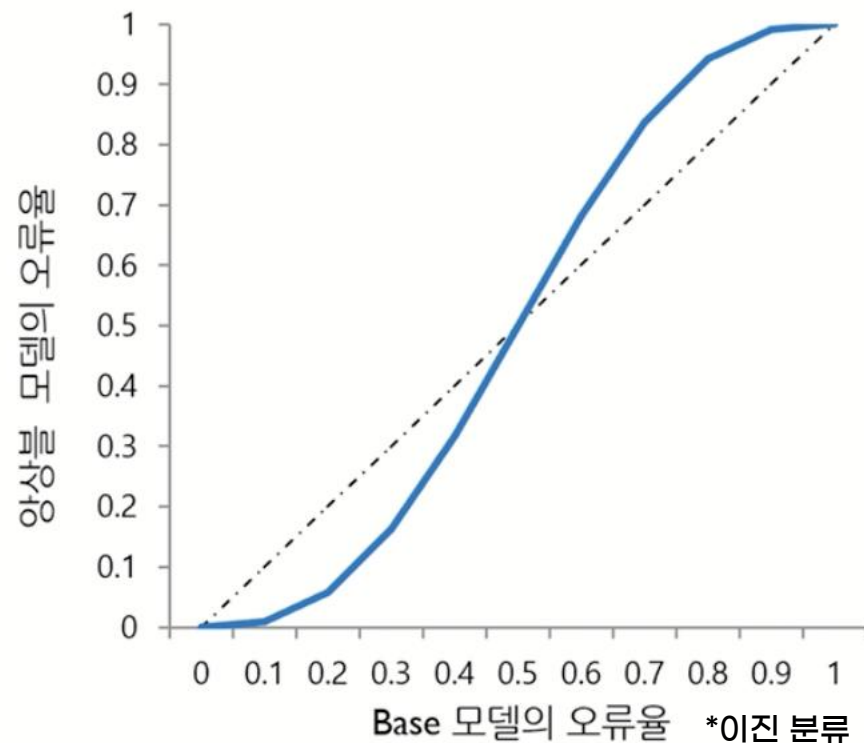


## 앙상블 학습 의미 (2)

- 여러 Base 모델을 통합하여 예측 정확성을 향상시킴

다음 조건을 만족할 때 앙상블 모델이 Base 모델보다 우수한 성능을 보임

- Base 모델이 서로 독립적
- Base 모델이 무작위 예측을 수행하는 모델보다 성능이 좋을 때  
이진 분류일 경우 오류율 0.5



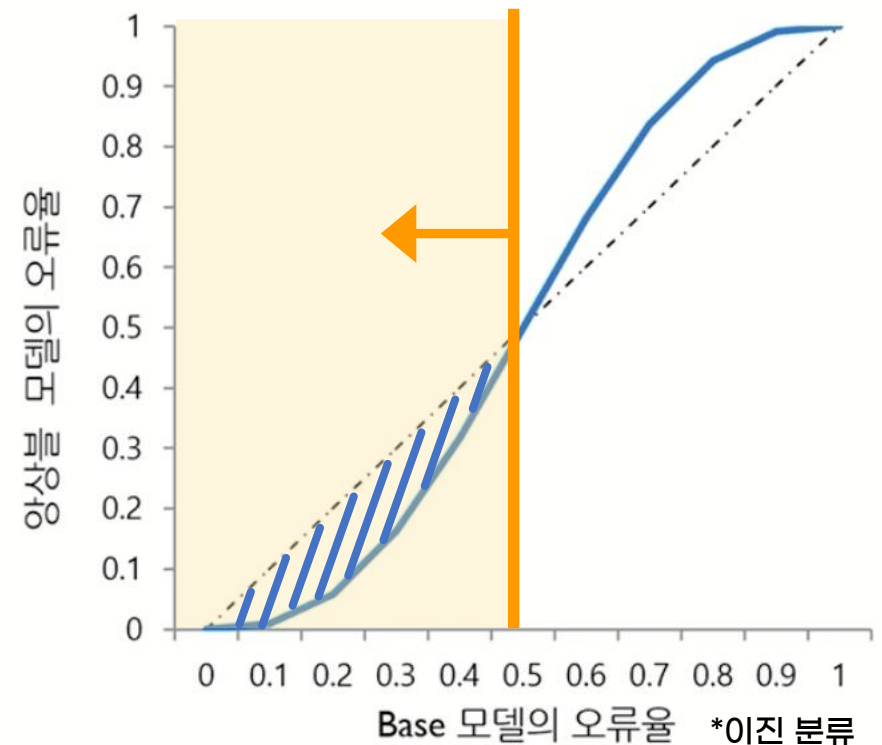
## 앙상블 학습 의미 (2)

- 여러 Base 모델을 통합하여 예측 정확성을 향상시킴

다음 조건을 만족할 때 앙상블 모델이 Base 모델보다 우수한 성능을 보임

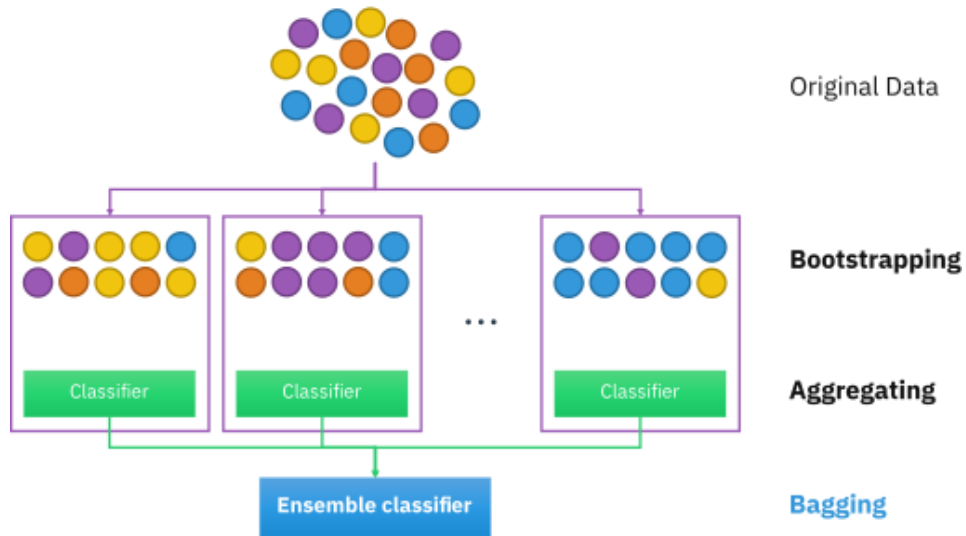
- Base 모델이 서로 독립적
- Base 모델이 무작위 예측을 수행하는 모델보다 성능이 좋을 때

이진 분류일 경우 오류율 0.5  
색칠한 영역만큼 오류율을 낮춤



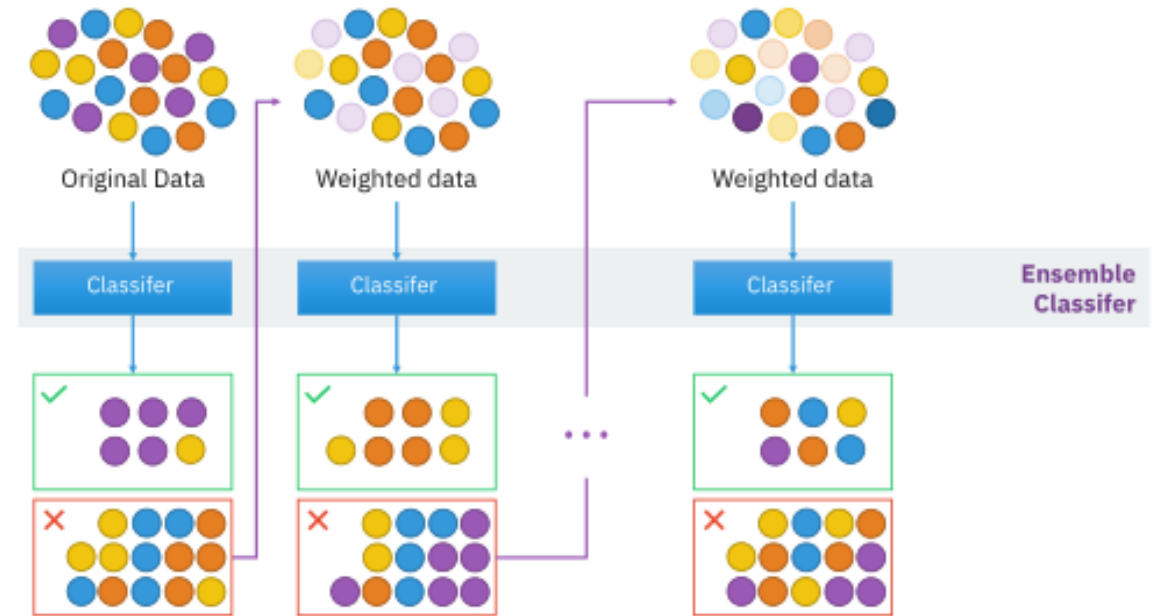
# 앙상블 학습 기법 (1)

## Bagging vs Boosting



## Bagging

: 병렬, 복원 추출

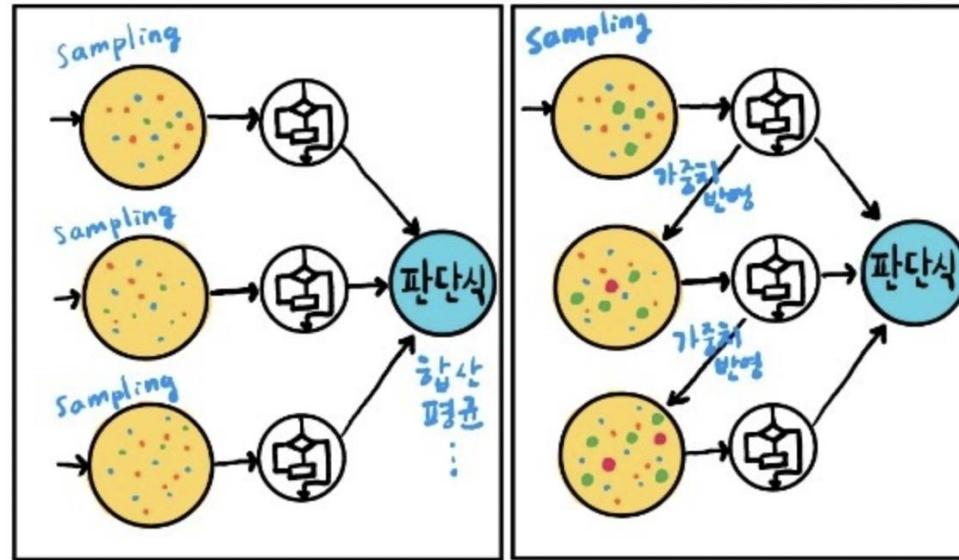


## Boosting

: 순차적, 복원 추출

# 앙상블 학습 기법 (1)

## Bagging vs Boosting



< Bagging >

모집단에서  
Sampling 후 알고리즘  
적용한 결과를 모은다.

< Boosting >

모집단에서  
Sampling 후 알고리즘  
적용한 결과에서  
Sampling 후 알고리즘  
적용한 결과를 모은다.





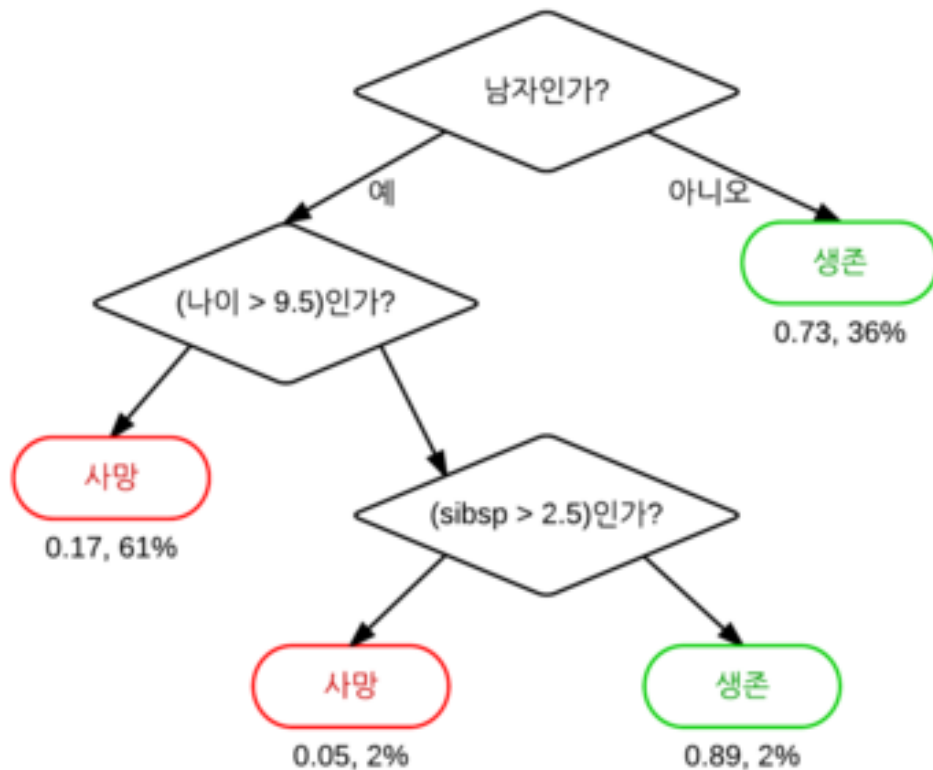
## 03. 랜덤 포레스트





## 03-1. 랜덤 포레스트 개요

# Decision Tree



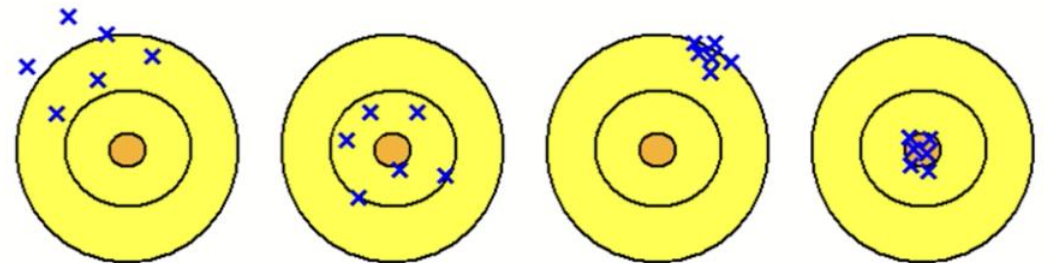
- 입력 값에 대한 예측 값을 나무 형태로 보여주는 모형
- **장점**
  - ☒ 해석이 쉬움
  - ☒ nominal variable을 numerical하게 변환할 필요 없음

# 개별 트리 모델의 단점

- 계층적 구조로 인해 중간에 에러 발생 시 ☒ 다음 단계로 에러 전파 (에러 수정 불가)
- 특정 데이터에만 잘 작동 → 학습 데이터의 미세한 변동에도 최종 결과가 크게 영향받음
- 적은 개수의 노이즈에도 크게 영향받음
- 나무의 최종 노드 개수를 늘리면 (full-grown tree)
  - ☒ training error 최소화하나, 과적합 위험 (low bias, large variance)
  - ☒ 즉, 새로운 데이터에 대한 예측 성능 떨어짐



## 랜덤 포레스트

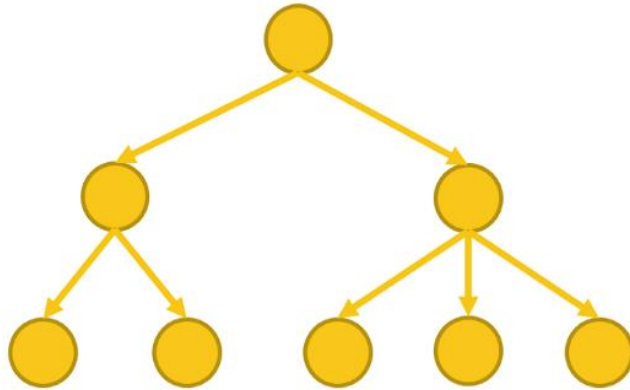


Bias	High	Low	High	Low
Variance	High	High	Low	Low

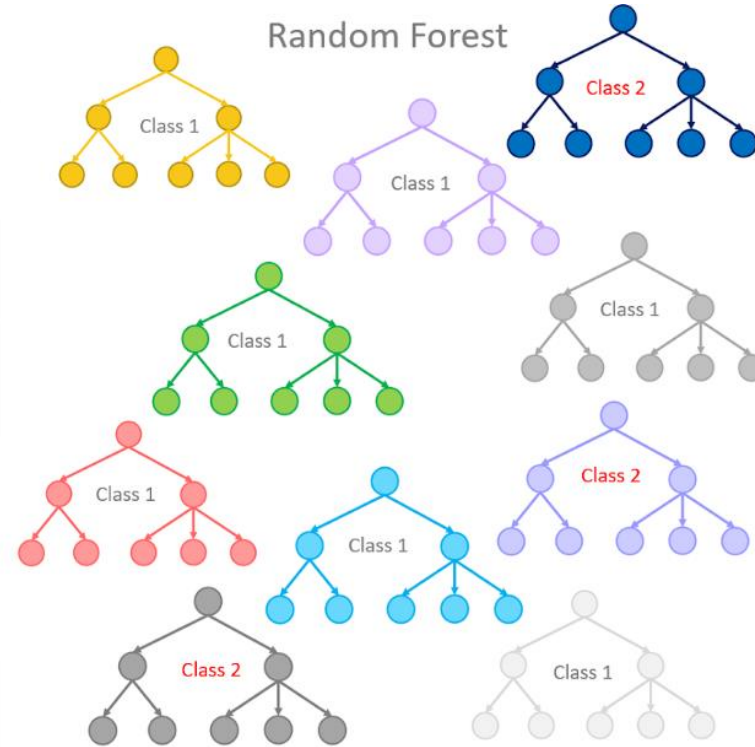


# 랜덤 포레스트

Single Decision Tree

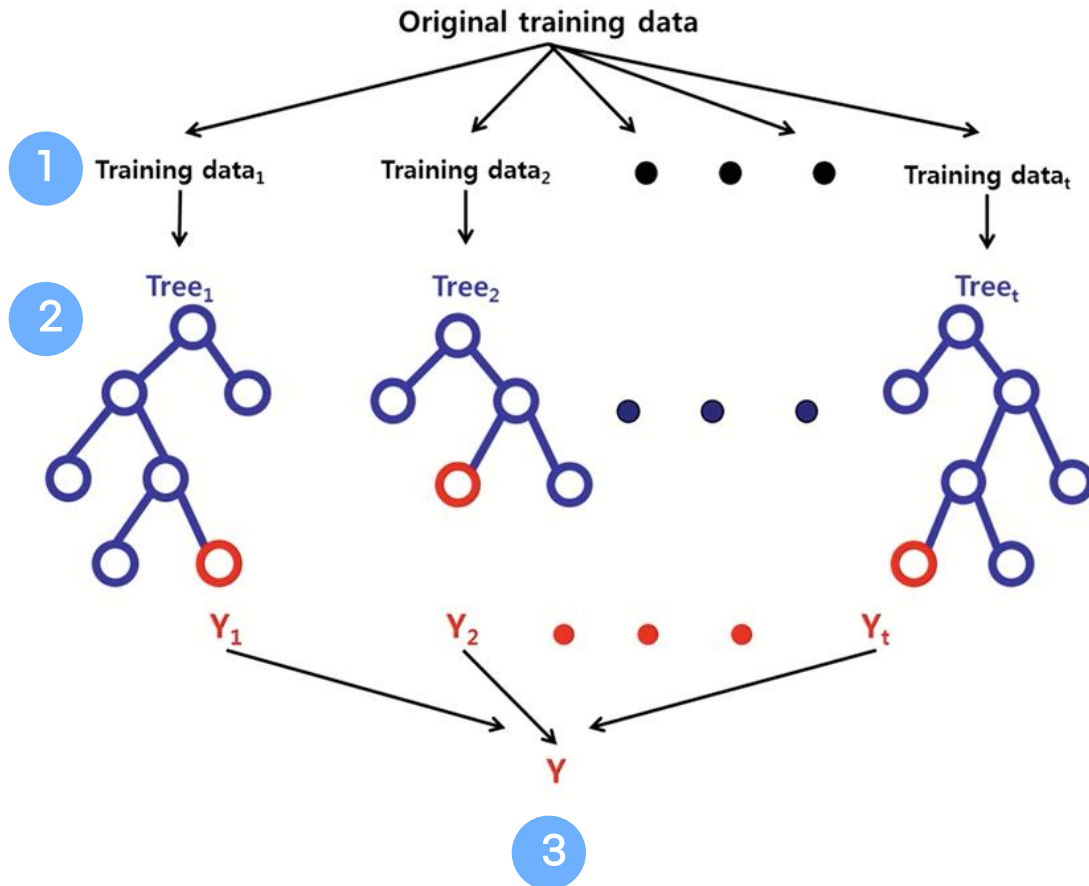


Random Forest



- Decision Tree의 특정 데이터에만 잘 작동할 가능성이 크다는 단점을 극복하기 위해 만들어진 알고리즘
- 여러 개의 Decision Tree가 전체 데이터에서 Bagging 방식으로 각자의 데이터를 샘플링 하여 개별적으로 학습을 수행한 후 최종적으로 모든 분류기가 voting이나 평균을 통해 예측 결정

# 랜덤 포레스트 개요



## 과정

1. Bootstrap 샘플 생성
2. Bootstrap 샘플로 decision tree 구축
3. 예측 종합

## 핵심 아이디어

2-1. 각 샘플마다 개별 decision tree 구축

☒ Bagging (다양성)

2-2. 전체 변수(특성) 중 일부 변수를 무작위(랜덤) 선택

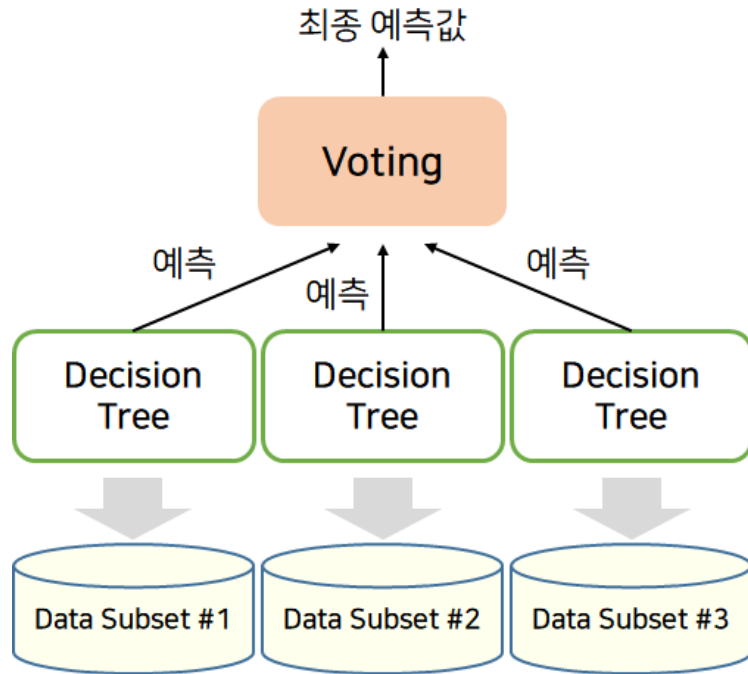
☒ Random Subspace



## 03-2. Bagging



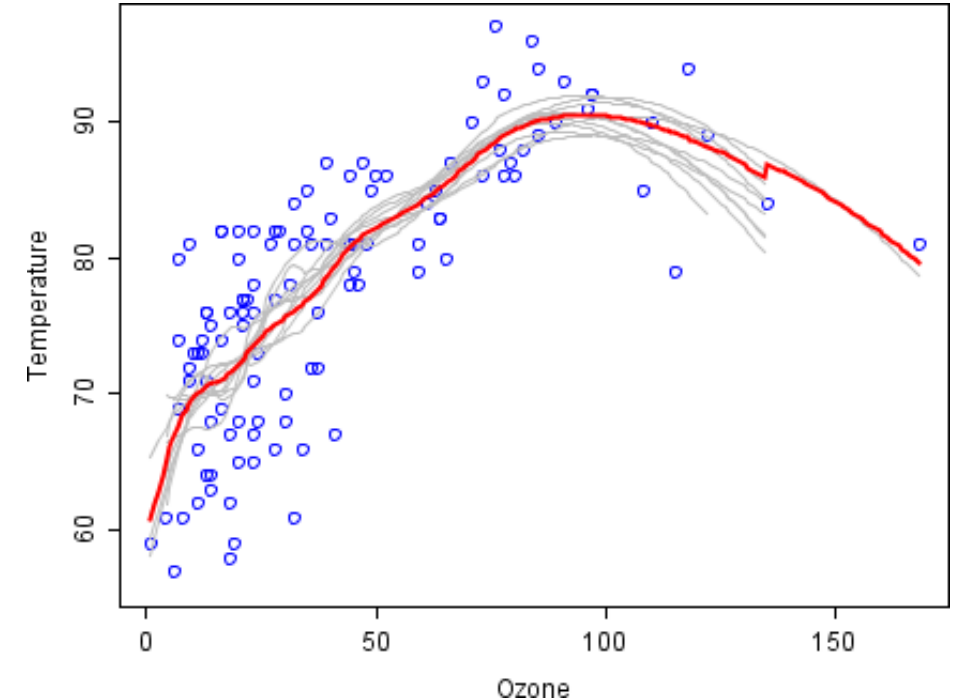
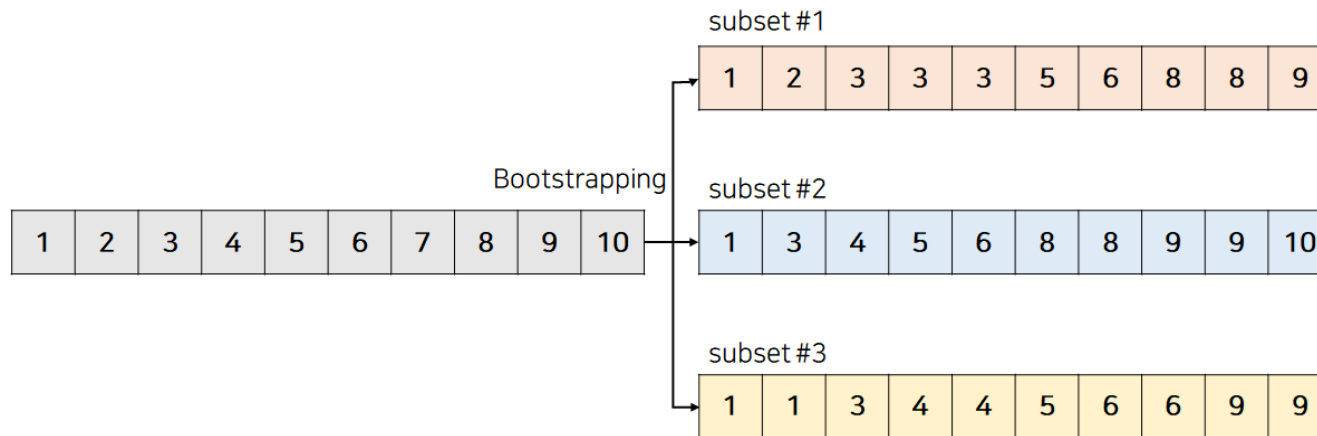
# Bagging



- Bootstrap Aggregating의 약자
- 단일 알고리즘 기반의 모델을 다른 데이터를 기반으로 학습시켜 결과를 결합
- 각각의 분류기에 Bootstrapping 분할 방식으로 생성된 샘플데이터를 학습 최종적으로 각각의 결과를 voting이나 평균을 통해 결정



# Bagging 1단계: Bootstrapping 분할 방식



- 단순 복원 임의추출법을 통해 raw data로부터 같은 크기의 표본 자료들을 생성하는 것
- 데이터양을 임의적으로 늘리고, 데이터 셋의 분포가 고르지 않을 때 고르게 만드는 효과
- high variance 모델의 variance를 낮추는 데에 효과적

# Bagging 1단계: Bootstrapping 분할 방식

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

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

- 이론적으로 한 개체가 하나의 Bootstrap에 한 번도 선택되지 않을 확률

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

# Bagging 2단계: Aggregating

- Bootstrap 후에 aggregate 하는 방법
  - 연속형 예측 문제 ☒ 평균
  - 범주형 분류 문제 ☒ voting

1

## Majority voting

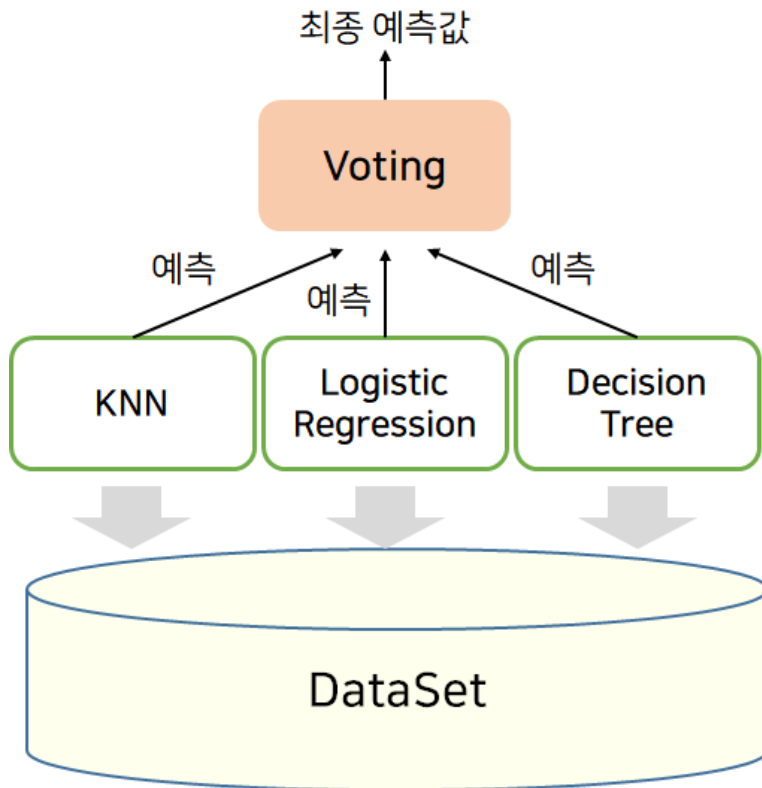
- 다수결
- Hard voting

2

## Weighted voting

- 가중평균
- 2-1. Weight = Training accuracy
- 2-2. Weight =  $P(Y=1)$  예측 확률

# Voting



- 서로 다른 알고리즘을 가진 모델을 병렬로 결합
- Output이 continuous value 이면(회귀)  
각 모델의 예측 값을 더해 평균을 냄으로써 앙상블 모델의 출력을 얻음
- 최종 output이 class label 이면(분류)  
Hard voting, Soft voting 중에 선택함



# Majority Voting

- Majority voting

$$Ensemble(\hat{y}) = \underset{i}{\operatorname{argmax}} \left( \sum_{j=1}^n I(\hat{y}_j = i), i \in \{0, 1\} \right)$$

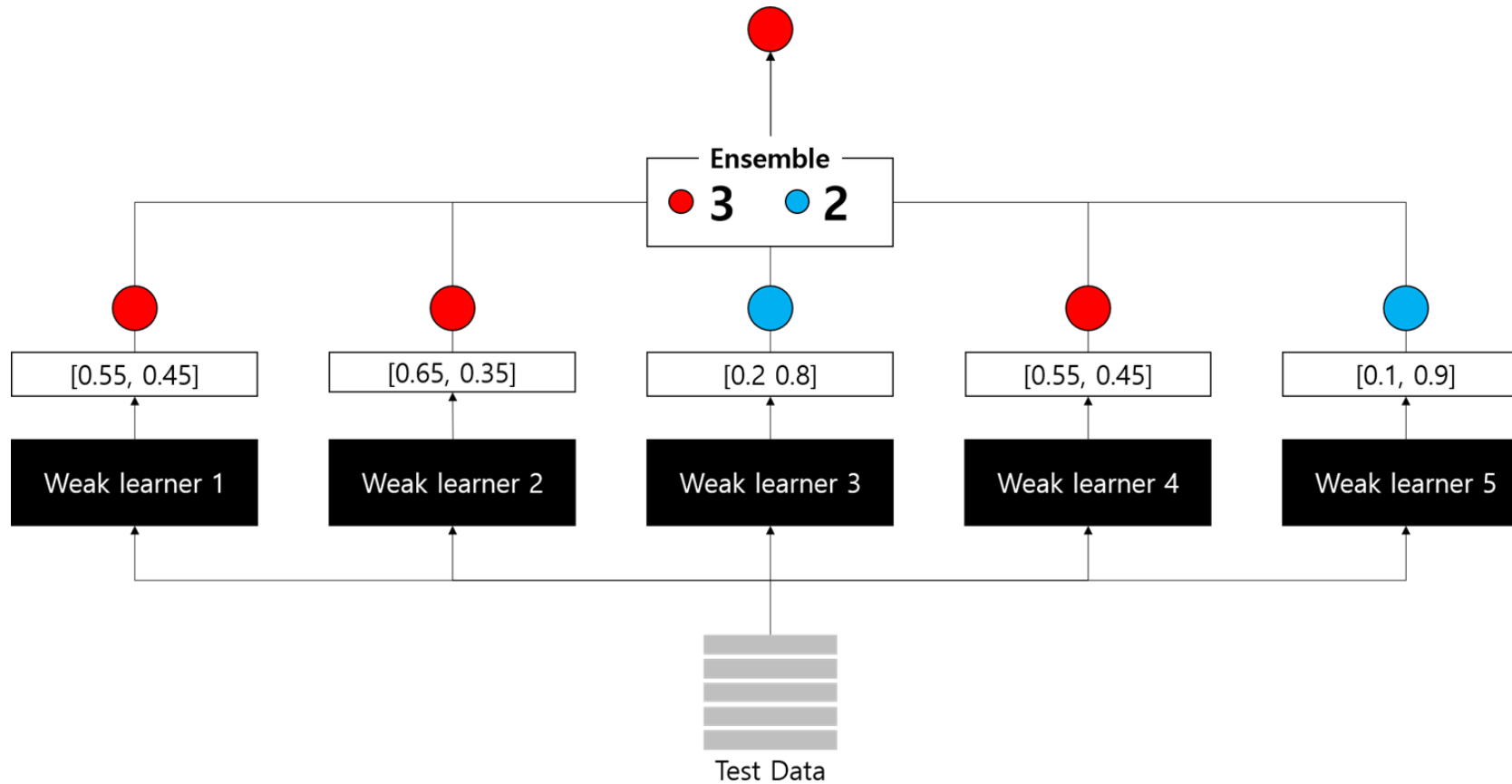
Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

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

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

$$Ensemble(\hat{y}) = 1$$

# Majority Voting



- 각 weak learner들의 예측 결과값을 바탕으로 다수결 투표하는 방식

# Weighted Voting

- Weighted voting (weight = training accuracy of individual models)

$$Ensemble(\hat{y}) = \operatorname{argmax}_i \left( \frac{\sum_{j=1}^n (TrainAcc_j) \cdot I(\hat{y}_j = i)}{\sum_{j=1}^n (TrainAcc_j)}, i \in \{0, 1\} \right)$$

Training Accuracy
0.80
0.75
0.88
0.91
0.77
0.65
0.95
0.82
0.78
0.83

Ensemble population
Model 1
Model 2
Model 3
Model 4
Model 5
Model 6
Model 7
Model 8
Model 9
Model 10

P(y=1) for a test instance
0.90
0.92
0.87
0.34
0.41
0.84
0.14
0.32
0.98
0.57

Predicted class label
1
1
1
0
0
1
0
0
1
1

$$\frac{0.91 + 0.77 + 0.95 + 0.82}{0.90 + 0.75 + \dots + 0.83} = 0.424$$

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

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

$$Ensemble(\hat{y}) = 1$$

# Weighted Voting

- Weighted voting (weight = predicted probability for each class)

$$Ensemble(\hat{y}) = \operatorname{argmax}_i \left( \frac{1}{n} \sum_{j=1}^n P(y = i), i \in \{0, 1\} \right)$$

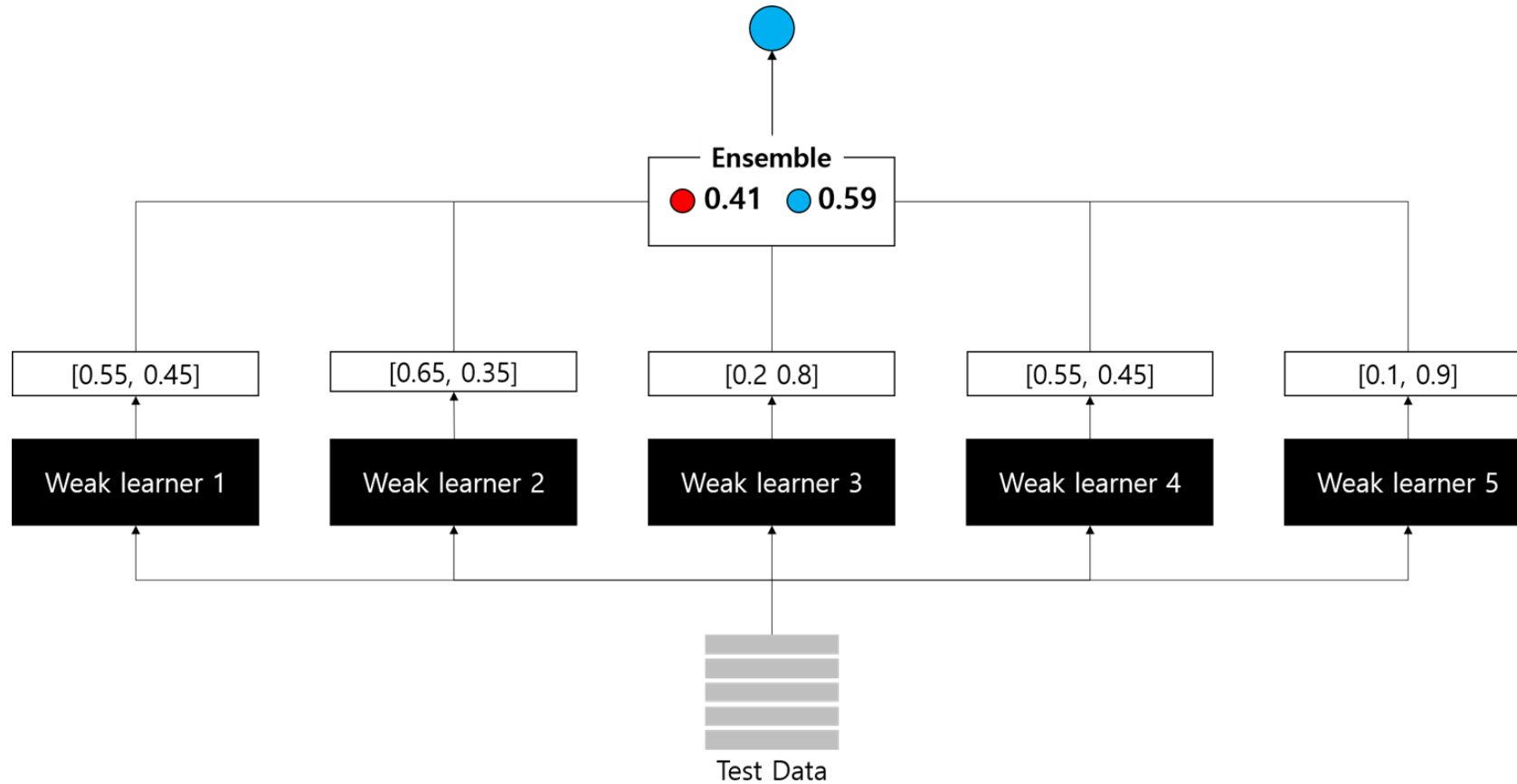
Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

$$\frac{1}{10} \sum_{j=1}^{10} P(y = 0) = 0.371$$

$$\frac{1}{10} \sum_{j=1}^{10} P(y = 1) = 0.629$$

$$Ensemble(\hat{y}) = 1$$

# Weighted Voting



- 예측 확률 값을 단순 평균 내어 확률이 더 높은 클래스를 최종 예측 값으로 결정

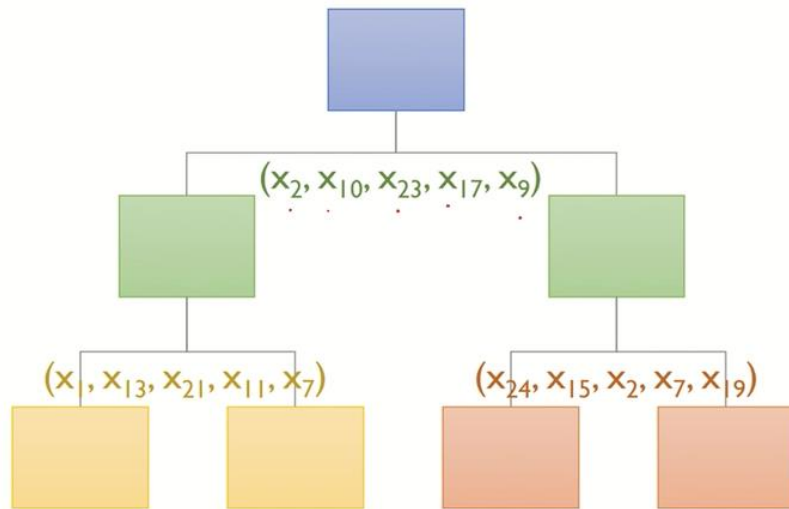




# 03-3. Random Subspace

# Random Subspace의 의미

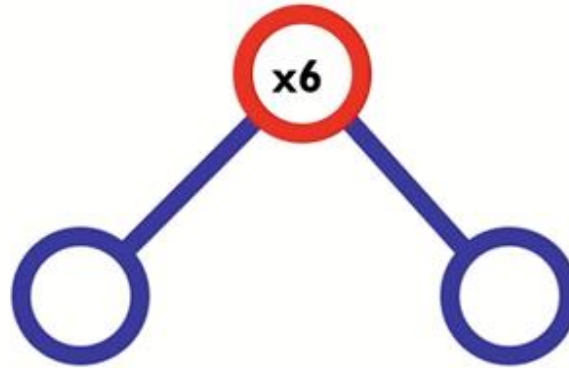
- 의사결정 나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들만을 고려 대상으로 함



- 1 번째 Bootstrap 샘플에 변수 25개가 있다고 가정
- 각 노드를 분할할 때마다 변수 5개 무작위 선택 → 최선의 분할인 1개를 찾음

# Random Subspace의 의미

Tree



원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

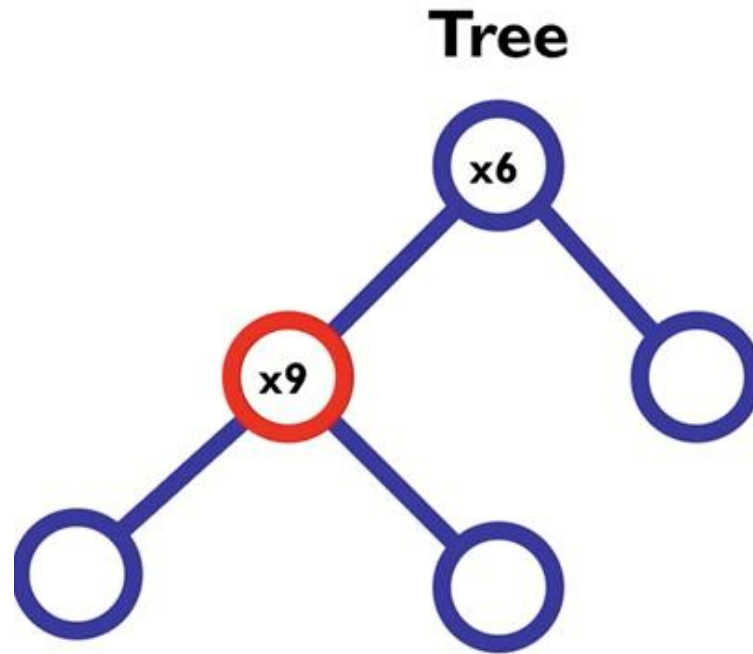
1. 원래 변수들 중에서 모델 구축에 쓰일 **입력 변수**를 무작위로 선택

원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

2. 선택된 입력 변수 중에 분할될 변수를 선택



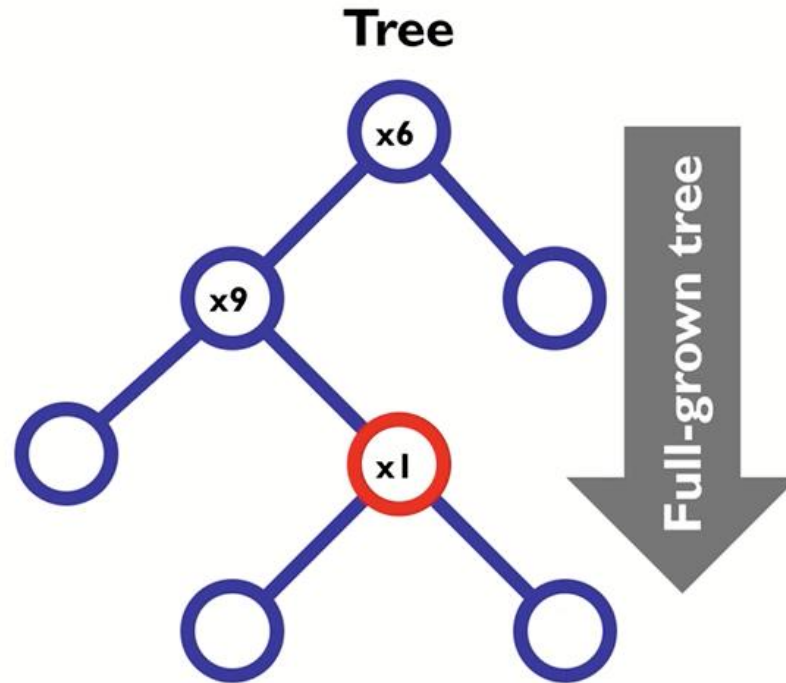
# Random Subspace의 의미



Original Features	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
Input Features	x1				x5				x9	x10						

3. 이러한 과정을 **full-grown tree**가 될 때까지 반복

# Random Subspace의 의미



원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수	x1						x7						x13	x14		

3. 이러한 과정을 **full-grown tree**가 될 때까지 반복





## 03-4. 랜덤 포레스트 특성

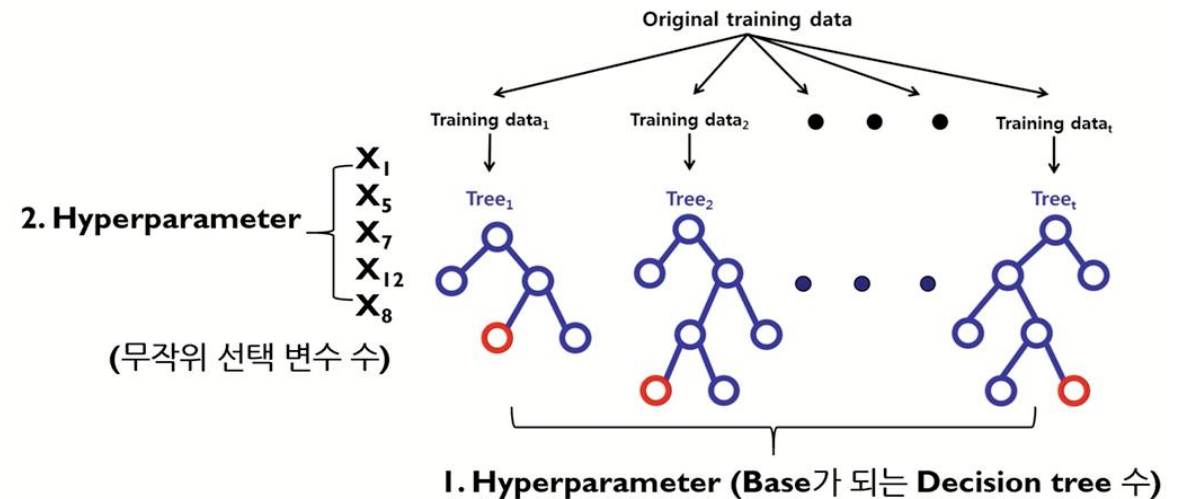
# 랜덤 포레스트-하이퍼 파라미터

## (1) n\_estimators

- Strong Law of Large Numbers 만족시키기 위해 2,000개 이상의 트리 필요 (가이드라인)
- Default: 100

## (2) max\_features

- 노드 분할 시 무작위로 선택되는 변수의 수
- 일반적으로 변수의 수에 따라 다음과 같이 추천됨
  - Classification:  $\sqrt{\text{변수의 수}}$
  - Regression: 변수의 수 / 3



## 랜덤 포레스트-장단점

### 장점

- Classification 및 Regression 문제에 모두 사용 가능
- 대용량 데이터 처리에 효과적
- 과대적합 문제 최소화하여 모델의 정확도 향상

### 단점

- 랜덤 포레스트 특성상 데이터 크기에 비례해서 수백 개에서 수천 개의 트리를 형성하기에 예측에 오랜 프로세스 시간이 걸림
- 랜덤 포레스트 모델 특성상 생성하는 모든 트리 모델을 다 확인하기 어렵기에 해석 가능성이 떨어짐

## 중요 변수 선택

### 변수의 중요도

- 랜덤 포레스트는 선형 회귀 모델/로지스틱 회귀 모델과는 달리 개별 변수가 통계적으로 얼마나 유의한지에 대한 정보를 제공하지 않음
- 대신 랜덤 포레스트는 다음과 같은 간접적인 방식으로 변수의 중요도를 결정
  - 1단계: 원래 데이터 집합에 대해서 OOB Error를 구함
  - 2단계: 특정 변수의 값을 임의로 뒤섞은 데이터 집합에 대해서 OOB Error를 구함
  - 3단계: 개별 변수의 중요도는 2단계와 1단계 OOB Error 차이의 평균과 분산을 고려하여 결정



# 변수 중요도

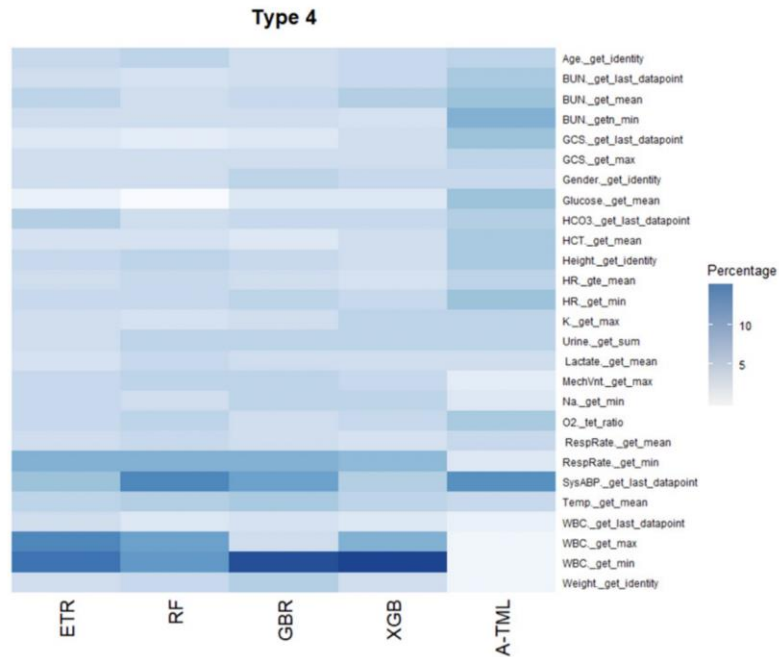


Figure 7. Heatmap of variable importance ratio for surgical intensive care unit.

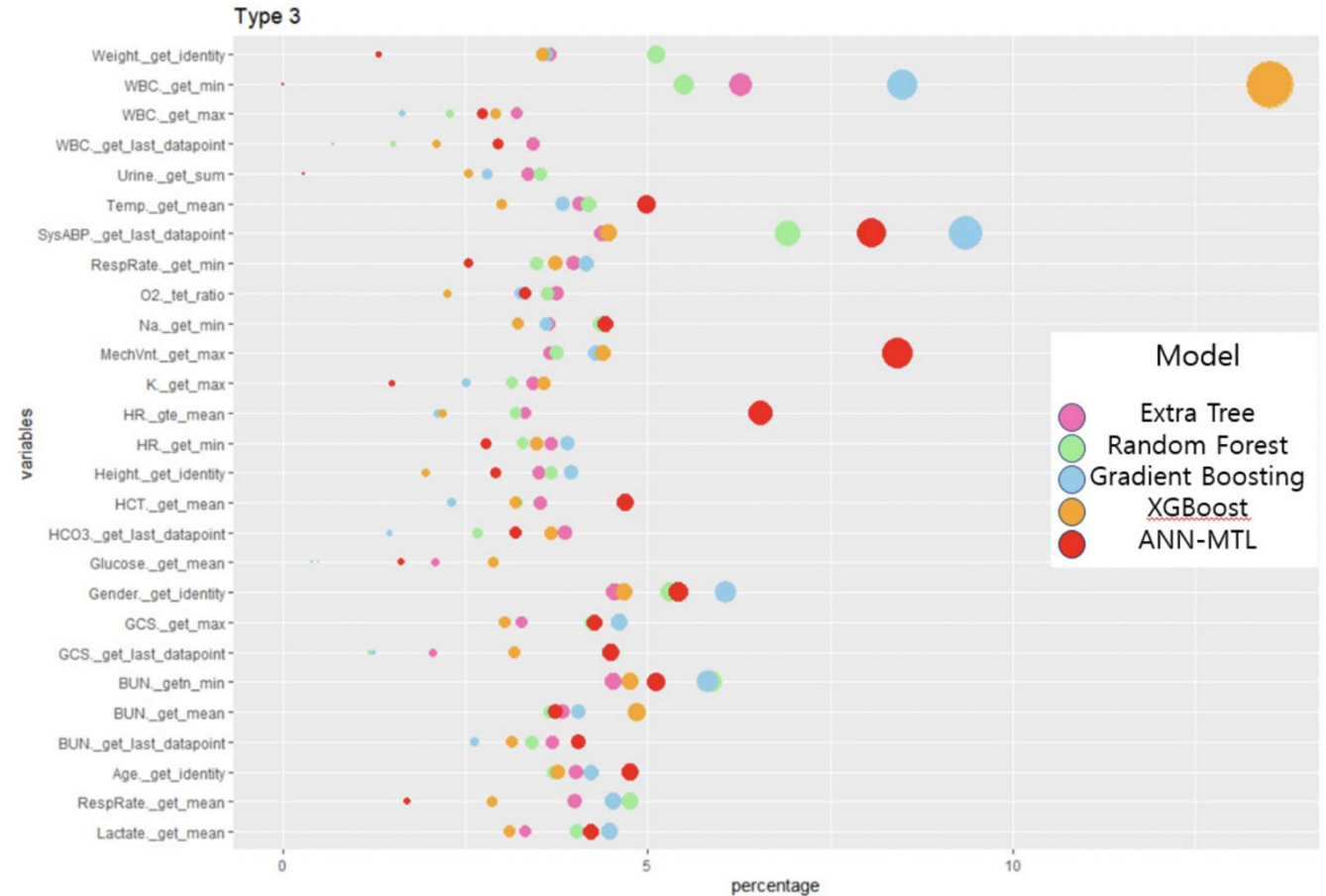


Figure 6. Bubble chart of variable importance ratio for internal endoplasmic reticulum intensive care unit.

- 어떤 변수가 중환자실 사망 예측에 영향을 미칠지에 대한 변수 중요도 추출
- 기법별 변수 중요도 시각화한 결과, 사망 예측에 중요한 변수는 백혈구 수치 (가장 큰 변수 중요도를 갖는다고 자주 선별됨)

# OOB Score

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Weak	Yes
Windy	Cold	Low	Weak	Yes

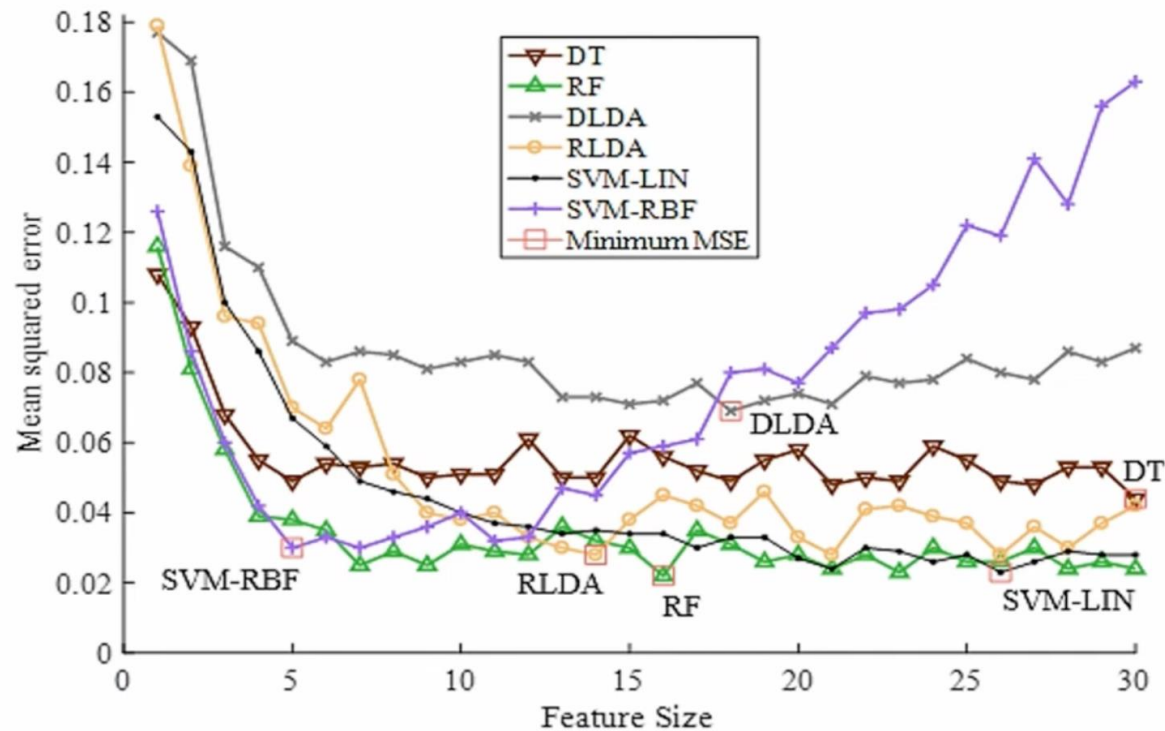
Bootstrap  
sample

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Weak	Yes
Windy	Cold	Low	Weak	Yes

Out of Bag  
sample

- OOB(Out Of Bag) 샘플은 부트스트랩 샘플에 포함되지 않고 남은 샘플을 의미
- 남은 샘플을 사용하여 부트스트랩 샘플로 훈련한 결정 트리를 평가, 검증 세트 같은 역할
- OOB 점수를 사용하면 교차 검증을 대신할 수 있어, 훈련 세트에 더 많은 샘플을 사용 할 수 있음

## 랜덤 포레스트 vs 결정 트리



- X축: 변수 개수
- Y축: 오류율
- 랜덤 포레스트(초록색)의 성능이 결정 트리(갈색) 보다 우수함





## 04. **엑스트라 트리**



## 엑스트라 트리

- 랜덤 포레스트와 매우 비슷하게 동작
- 부트스트랩 샘플을 사용하지 않고, 전체 훈련 세트를 사용한다는 차이점
- 노드를 무작위하게 분할 (→ 편향 증가, 분산 감소)

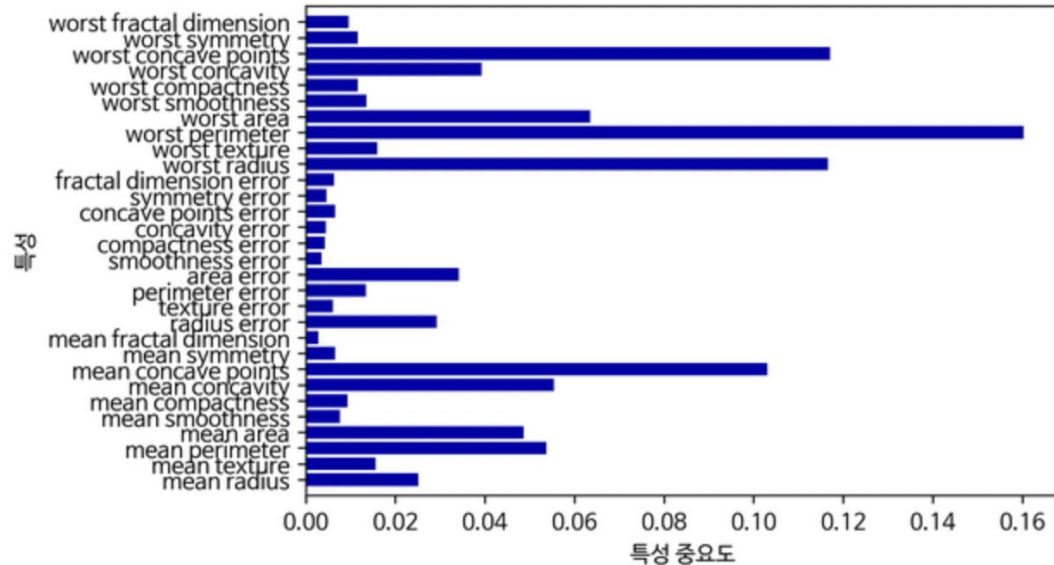
### 장점

- 많은 트리를 앙상블 하기 때문에, 과대 적합은 막고 검증 세트의 점수를 높이는 효과
- 랜덤하게 노드를 분할하기 때문에, 속도가 빠름

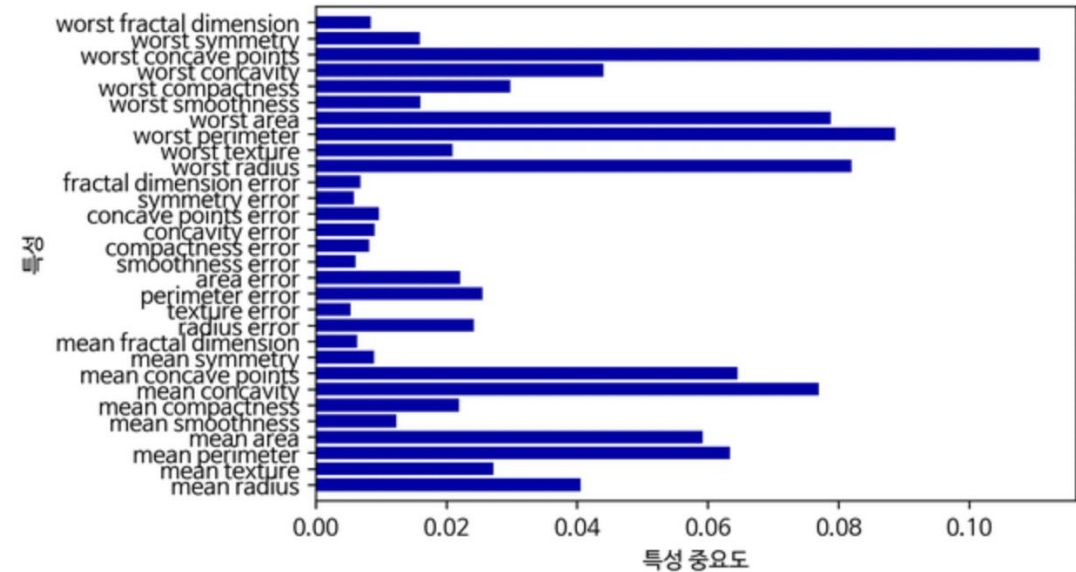
### 단점

- 성능이 낮아짐
- 일반화 성능을 높이기 위해선 '많은 트리'를 만들어야 함

## 랜덤 포레스트 vs 엑스트라 트리



랜덤 포레스트

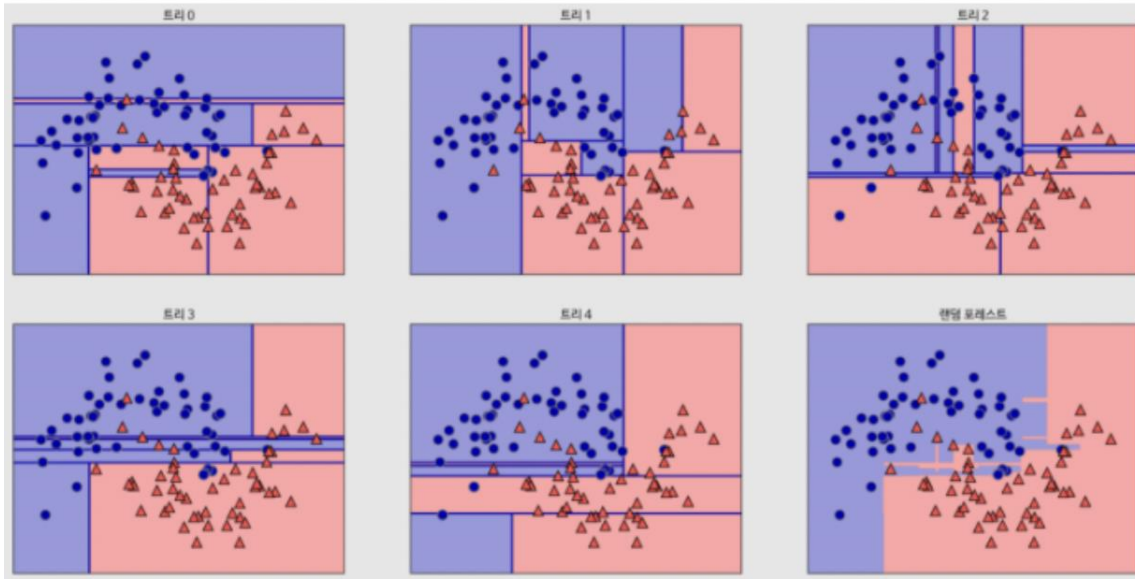


엑스트라 트리

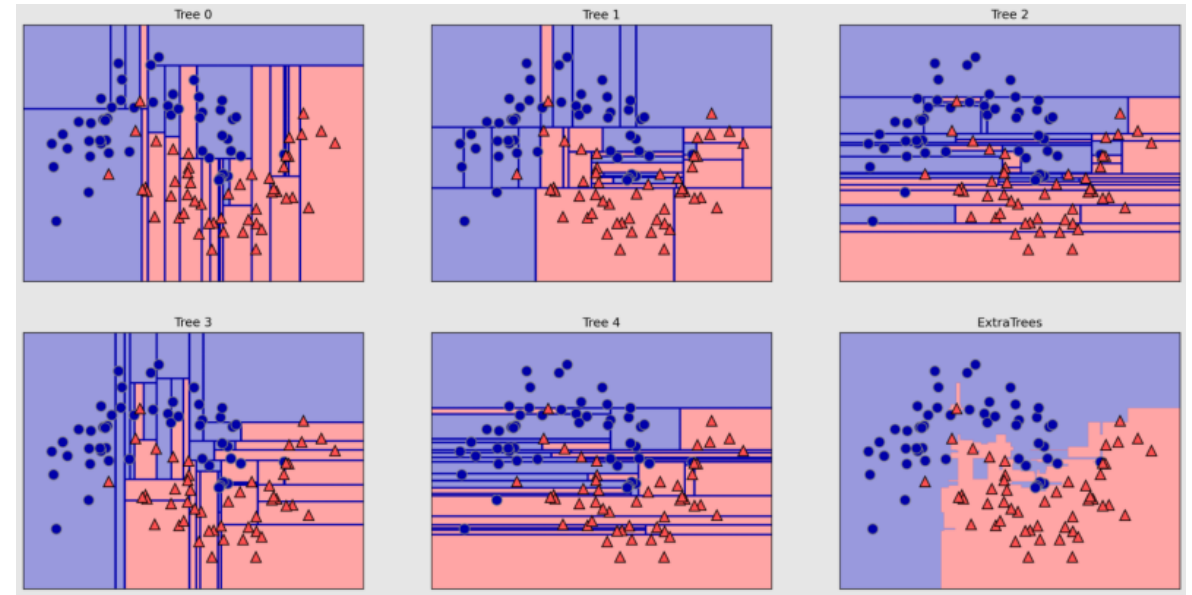
- 무작위성이 강한 엑스트라 트리에서 특성 중요도가 전반적으로 상승
- 엑스트라 트리의 특성 중요도가 비교적 고르다

# 랜덤 포레스트 vs 엑스트라 트리

\*앞 5개는 개별 트리, 우측 하단은 앙상블한 것



랜덤 포레스트



엑스트라 트리

- 엑스트라 트리의 개별 트리: 노드를 랜덤 분할한 뒤 최선의 분할을 찾기 때문에 결정 경계가 복잡함
- 하지만 앙상블한 엑스트라 트리의 결정 경계는 안정적