

decision tree  $\times$  0.7171212121212121 = random forest  
base model ensemble

## Inclass 16: Decision Tree and Random Forest

[SCS4049] Machine Learning and Data Science

---

Seongsik Park (s.park@dgu.edu)

School of AI Convergence & Department of Artificial Intelligence, Dongguk University

# Decision Tree

# Decision tree

특징  $\text{input space} \in \mathbb{R}^D \rightarrow \text{여러 영역 분할}$

- $\text{input}$  Feature 공간을 여러 개의 단순한 영역으로 분할
- Feature 공간을 분할하는 일련의 규칙들은 tree의 형태로 표현

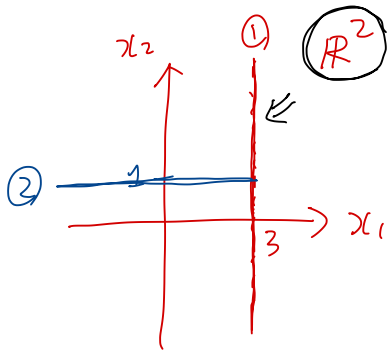
## 장점

- 데이터나 모델에 대한 전제, 가정이 없음
- 간단하기 때문에 이해나 해석이 용이함
- Classification과 regression에 모두 사용 가능
- Numeric feature와 categorical feature 모두 처리 가능
- 스케일링, 중앙화 등 같은 데이터에 대한 전처리 거의 필요 없음
- 탐색적 데이터 분석(exploratory data analysis)에서 유용

## 단점

- 데이터의 회전이나 작은 변화에 매우 민감
- 예측 정확도 측면에서 다른 알고리즘보다 떨어지는 경우가 많음  
= Bagging, boosting, random forests 등을 사용해 정확도 향상

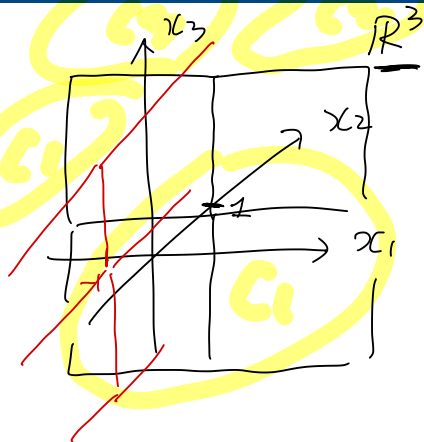
threshold,  $\geq 1$   $\frac{3}{4}$   $\frac{1}{4}$ .



①  $\underline{x_1} \geq 3$

②  $\underline{x_2} \geq 1$

$x_1 = 3$   
 $x_2 = 1$



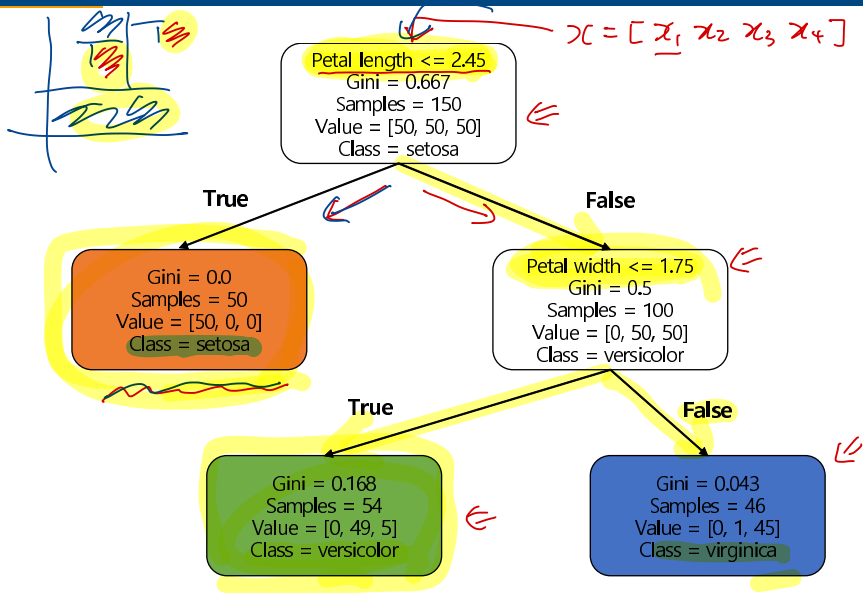
①  $x_2 \geq 1.$

②  $x_1 \geq -1.$

# Iris dataset

- 3개의 클래스: iris setosa, iris versicolor, iris virginica
  - 4개의 feature: sepal length/width, petal length/width
  - 각 클래스별로 50개의 샘플
- 50      50      50      = 150

# Decision tree for iris dataset

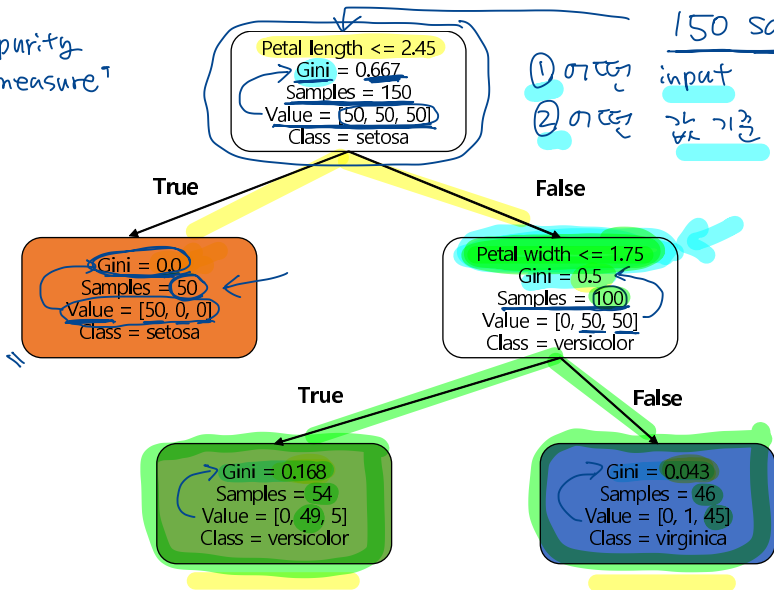


# Decision tree for iris dataset

6  
impurity  
measure<sup>7</sup>

150 sample

①의 경우 input  
②의 경우 값이 1인  
혹은 2인





# Gini impurity measure

## Gini impurity measure

$$\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right] \uparrow$$

- 임의의 node  $i$ 에서 impurity  $G_i$ 는 다음과 같이 정의

$$G_i = 1 - \sum_k p_{i,k}^2 \quad (1)$$

$G_i \rightarrow 0$

- $p_{i,k}$ : node  $i$ 에서 클래스  $k$ 에 속하는 instance의 비율
- e.g.,

$$G = 1 - \left(\frac{49}{54}\right)^2 - \left(\frac{5}{54}\right)^2 \approx 0.168 \quad (2)$$

gini = 0.168  
samples = 54  
value = [0, 49, 5]  
class = versicolor

$$\frac{0}{54}, \frac{49}{54}, \frac{5}{54}$$

# Information gain measure

## Information gain (entropy) measure

- 한 노드에서의 엔트로피

$$H \downarrow p = [1, 0, 0]$$

$$H \uparrow p = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$$

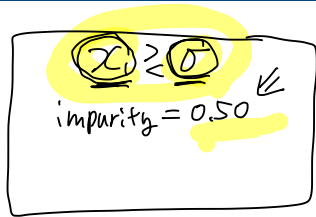
$$H = - \sum_k p_{i,k} \log p_{i,k} \quad (3)$$

- $p_{i,k}$ : node  $i$ 에서 클래스  $k$ 에 속하는 instance의 비율
- e.g.,

$$H = - \frac{49}{54} \log \frac{49}{54} - \frac{5}{54} \log \frac{5}{54} \approx 0.31 \quad (4)$$

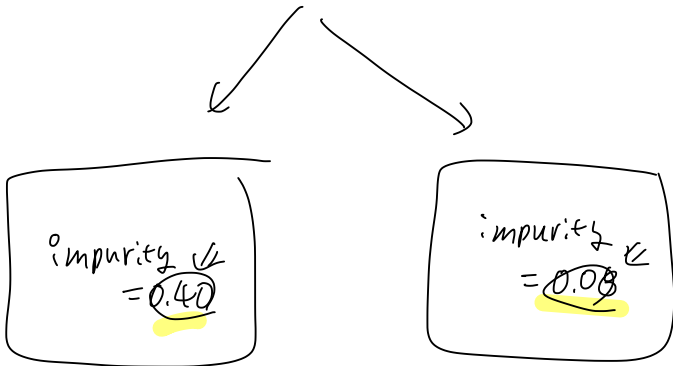
$$\left( \begin{array}{ccc} 0 & 49 & 5 \\ \hline 54 & 54 & 54 \end{array} \right)$$

gini = 0.168  
samples = 54  
value = [0, 49, 5]  
class = versicolor



Splitting  
criterion

CART  
⋮



# Regularization hyperparameters

Decision tree와 과적합(overfitting)

- Decision tree는 데이터에 대한 가정이 별로 없음
- Tree growing에 제약을 주지 않으면 과적합되기 쉬움

과적합 방지를 위한 정규화(regularization)

- Tree growing에 있어 자유도에 제한을 가하는 것
- Regularization 방법은 알고리즘에 따라 다름

작은 실데이터



완벽하게

필요이상으로

+

모델을

복잡하게.

Regularization hyperparameters



Hyperparameter	Default	Description
<u>max_depth</u>	none	트리의 최대 깊이
<u>min_samples_split</u>	2	내부 node를 split하기 위한 최소 샘플 수
<u>min_samples_leaf</u>	1	leaf node가 되기 위한 node 내 샘플 수
<u>min_weight_fraction_leaf</u>	0	<u>min_samples_leaf</u> 와 같으나 샘플 수 비율로 표시
<u>max_leaf_nodes</u>	none	leaf node 최대 수
<u>max_features</u>	none	각 node에서 split을 위해 계산하는 최대 feature 수

# Instability of decision tree

Decision boundary가 항상 좌표축에 수직  $\Rightarrow$  데이터셋의 회전에 민감

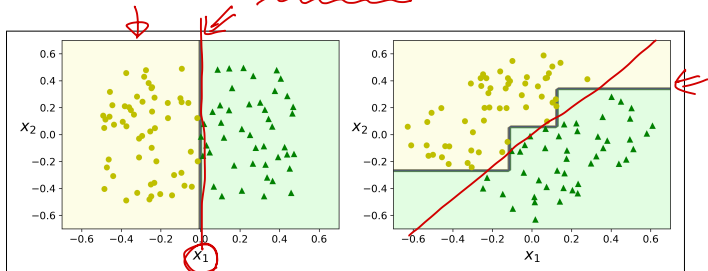
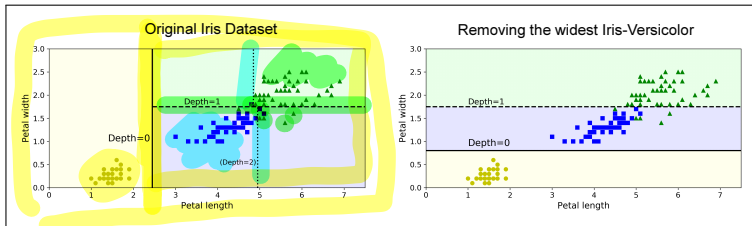


Figure 6-7. Sensitivity to training set rotation

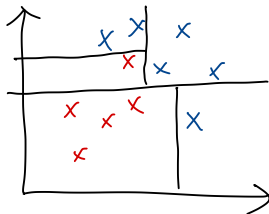
# Instability of decision tree

데이터의 작은 변화에도 매우 민감



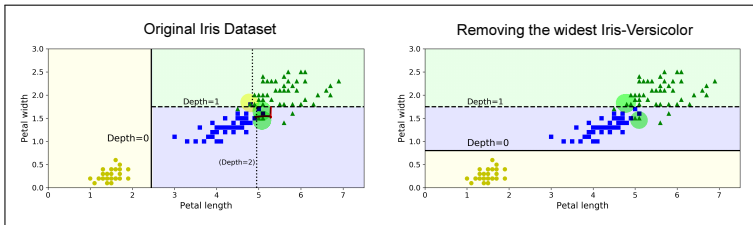
$\mathbb{R}^2$  입력

3 class 구분



# Instability of decision tree

데이터의 작은 변화에도 매우 민감



base model     $\times$  'ensemble' = random forest.  
decision tree                    + 여러

## Random Forest

---



# Ensemble learning

## Ensemble learning

- 한 전문가의 의견 vs. 여러 사람의 의견을 종합
- 하나의 좋은 예측기 vs. 보통 예측기 집단의 예측

## Ensemble learning의 목표

- 여러 다양한 의견을 고려
- 논리적 과정을 통해 결합
- 결정에 대한 신뢰성을 높임

# Ensemble learning

## Ensemble을 통한 변동성 축소

- 분류기들의 오류는 각 샘플에 대해 서로 다른 오류를 발생시키지만, 옳은 분류에 대해서는 일반적으로 일치
- 여러 분류기 출력을 averaging 하는 것은 오류 요소들이 averaging out 되게 만들어 전체 ensemble 모델의 오류를 감소

⇒ Decision tree와 ensemble learning = random forest

- Decision tree는 base model로써의 활용도가 높음
- Low computational complexity: 데이터의 크기가 방대해도 빠르게 구축
- Nonparametric: 데이터의 분포에 대한 가정이 필요 없음

# Ensemble learning: diversity

다음 조건을 만족할 때, ensemble model이 base model보다 우수

- Base model이 서로 독립적이고
- Base model이 무작위로 예측할 때보다 성능이 뛰어난 경우

Ensemble model의 성능을 확보하기 위한 핵심: 다양성과 무작위성

- 훈련 데이터의 서로 다른 부분집합을 사용 bootstrap
- 사용 가능한 feature의 서로 다른 부분집합을 사용: random subspace

## random forest

= base model - decision tree

= boot strap

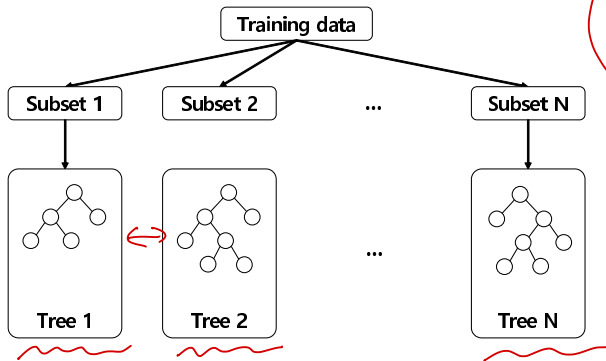
= aggregation

} bagging

= random subspace

## Bagging = bootstrap aggregating

- **Bootstrap** 기법으로 다수의 학습 데이터 생성
- 생성된 데이터로 모델 구축
- 주어진 새 입력에 대해 예측을 종합

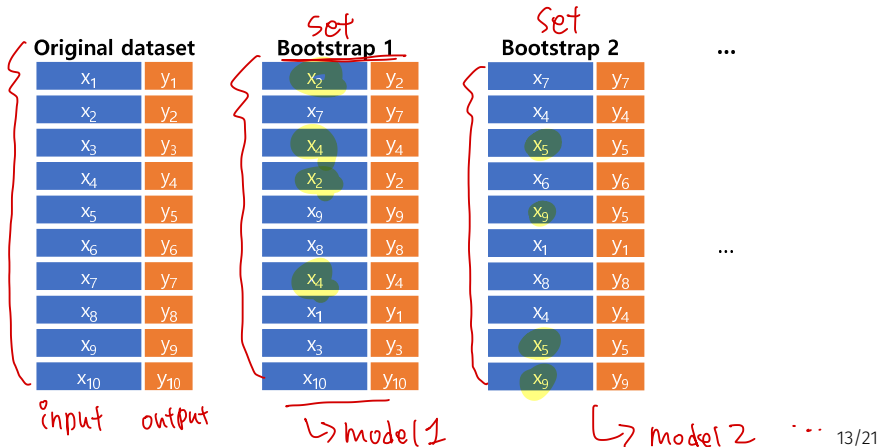


Hand-drawn diagram illustrating a sequence of operations or data flow:

- A box containing  $921$  and  $2100/2$  has an arrow pointing to a row of four boxes labeled  $\#1$ ,  $\#2$ ,  $\#3$ , and  $\#4$ .
- The boxes are arranged in two rows of two, with wavy lines under the bottom row.
- A vertical ellipsis is at the bottom.

# Bootstrapping

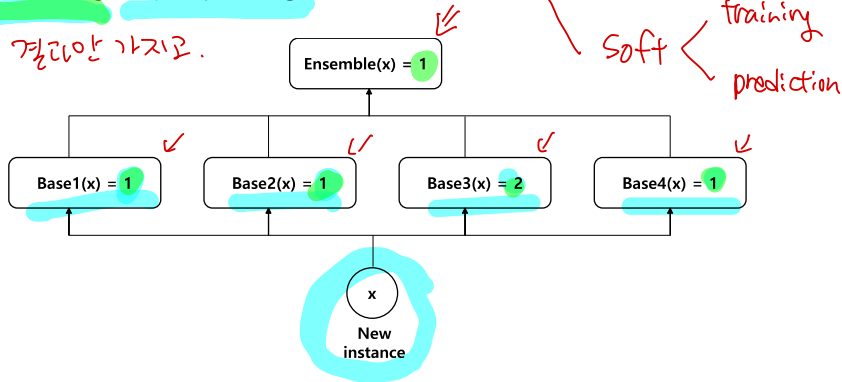
- 각 모델은 서로 다른 학습 데이터셋을 이용
- 각 데이터셋은 복원 추출을 통해 원래 데이터 수만큼의 크기를 갖도록 샘플링
- 개별 데이터셋을 bootstrap set이라 부름



# Result aggregating: hard voting

Hard voting, majority voting

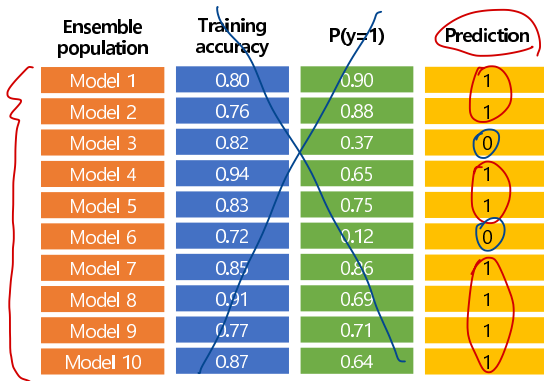
결정값만 가지고.



# Result aggregating: hard voting

Hard voting, majority voting

0,1



Ensemble population	Training accuracy	P(y=1)	Prediction
Model 1	0.80	0.90	1
Model 2	0.76	0.88	1
Model 3	0.82	0.37	0
Model 4	0.94	0.65	1
Model 5	0.83	0.75	1
Model 6	0.72	0.12	0
Model 7	0.85	0.86	1
Model 8	0.91	0.69	1
Model 9	0.77	0.71	1
Model 10	0.87	0.64	1

Hard voting = 1



# Result aggregating: soft voting

Soft voting, weighted voting: training accuracy 이용

Ensemble population	Training accuracy	$P(y=1)$	Prediction
Model 1	0.80	0.90	1
Model 2	0.76	0.88	1
Model 3	<del>0.82</del>	0.37	0
Model 4	0.94	0.65	1
Model 5	0.83	0.75	1
Model 6	<del>0.72</del>	0.12	0
Model 7	0.85	0.86	1
Model 8	0.91	0.69	1
Model 9	0.77	0.71	1
Model 10	0.87	0.64	1

$$P(\text{Ensemble} = 0) = 0.186$$

$$P(\text{Ensemble} = 1) = 0.814$$

$$\text{Weighted voting} = 1$$

$$P(\text{ensemble} = 1) = \frac{0.80 + 0.76 + 0.94 + 0.83 + \dots}{0.80 + 0.76 + 0.82 + \dots + 0.87}$$

# Result aggregating: soft voting

Soft voting, weighted voting: prediction probability 이용

Ensemble population	Training accuracy	<u>P(y=1)</u>	Prediction
Model 1	0.80	0.90	1
Model 2	0.76	0.88	1
Model 3	0.82	<del>0.88</del> 0.4	0
Model 4	0.94	0.65	1
Model 5	0.83	0.75	1
Model 6	0.72	<del>0.88</del> 0.3	0
Model 7	0.85	0.86	1
Model 8	0.91	0.69	1
Model 9	0.77	0.71	1
Model 10	0.87	0.64	1

새이력영향  
↓

새이력영향  
↓

$$\begin{aligned}
 & 0.60 + 0.70 \\
 & \frac{0.70 + 0.88 + 0.60 + 0.65}{+ 0.75 + 0.70 + \dots}
 \end{aligned}$$

0.60

$$P(\text{Ensemble} = 0) = 0.199$$

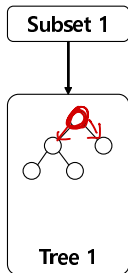
$$P(\text{Ensemble} = 1) = 0.801$$

Weighted voting = 1

0.70

# Random subspace

원래 변수들 중에서 모델 구축에 쓰일 입력 변수를 무작위로 선택



$\lambda_1, \lambda_3, \lambda_4, \lambda_5$



분기할때,

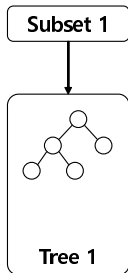
어떤 입력을  
고를까?

$$x \in \mathbb{R}^8$$

원래 변수	<u>x1</u>	<u>x2</u>	<u>x3</u>	x4	x5	x6	x7	<u>x8</u>
입력 변수	x1		x3	x4			x7	

# Random subspace

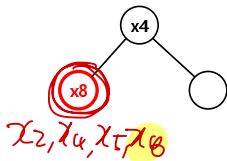
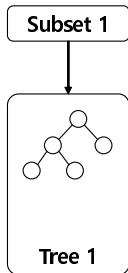
선택된 입력 변수 중에 분할될 변수를 선택



원래 변수	<b>x1</b>	x2	<b>x3</b>	<b>x4</b>	x5	x6	<b>x7</b>	x8
입력 변수	x1		x3	<b>x4</b>			x7	

# Random subspace

분할된 노드에서 동일한 과정을 반복



원래 변수	x1	x2	x3	x4	x5	x6	x7	x8
입력 변수		x2		x4	x5			x8

# Generalization error

- 각각의 decision tree는 과적합 되더라도
- Random forest는 그 수가 충분히 많다면 error가 바운드됨

$$e \leq \frac{\bar{\rho}(1-s^2)}{s^2} \quad (5)$$

- $\bar{\rho}$ : decision tree 사이의 평균 상관관계 다양성, ↓ 좋음.
- $s$ : 올바로 예측한 tree의 수와 잘못 예측한 tree의 수 차이의 평균 ↑ 좋음.
- 개별 tree의 정확도가 높을수록  $s$ 가 증가함
- Bagging과 random subspace로 모델 사이의 상관관계 감소  $\bar{\rho}$  ↓
- 개별 tree의 정확도가 높고, 각각의 독립성이 높을수록 전체 성능 증가