Supervised learning 3

성균관대학교 모용철

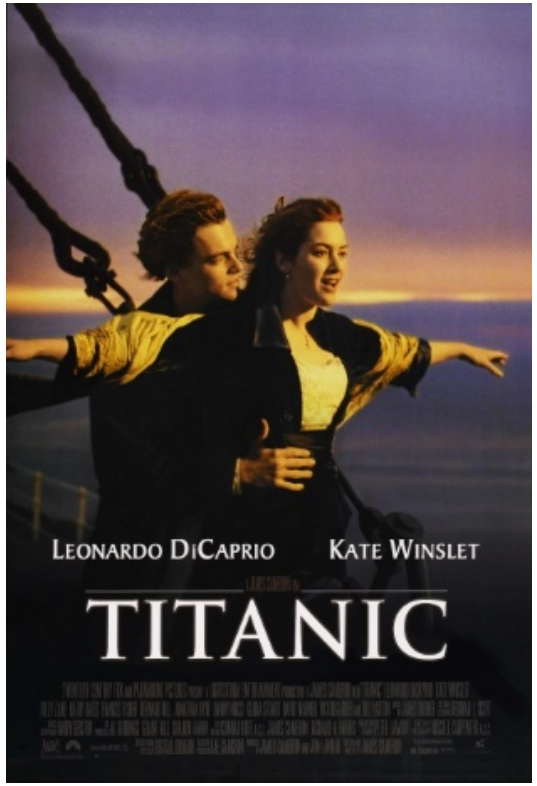
목차

- Decision Tree(DT)
- Random Forest(RF)
- Support Vector Machine(SVM)

Decision Tree

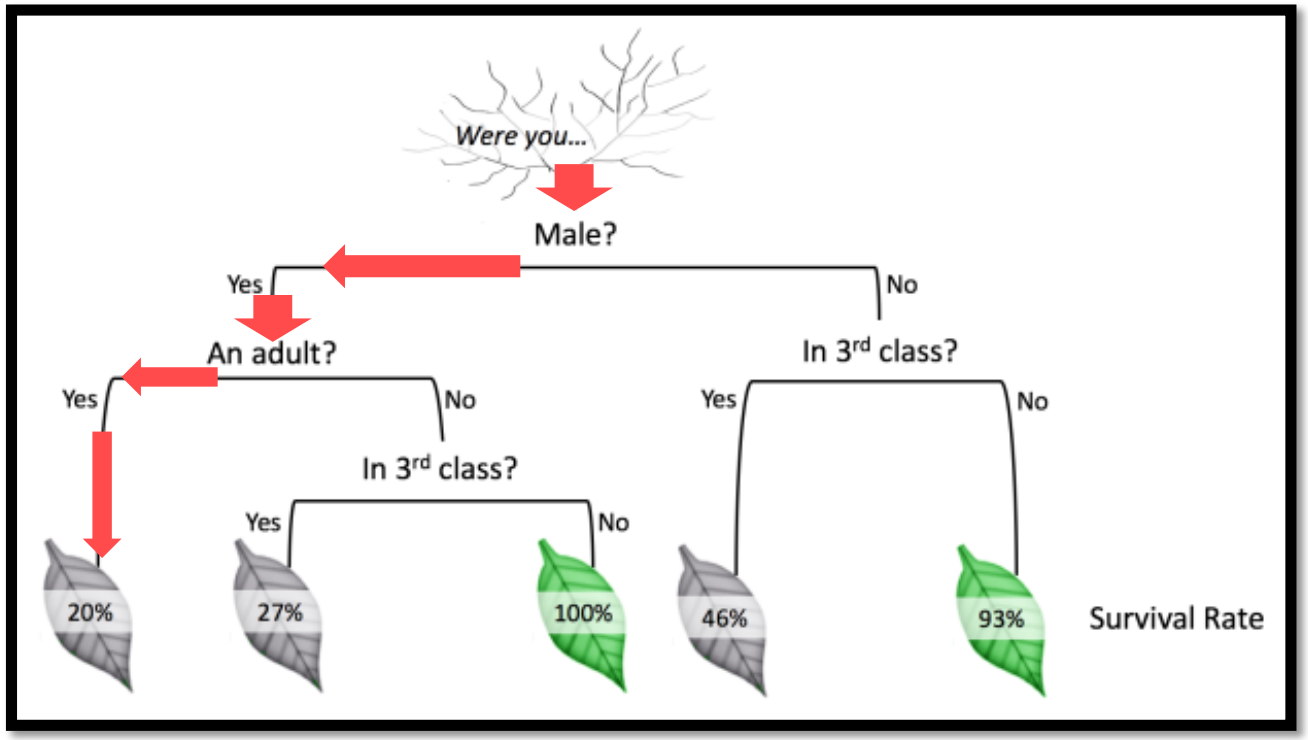
Decision Tree

- Example : 타이타닉 호 생존율 예측



Decision Tree

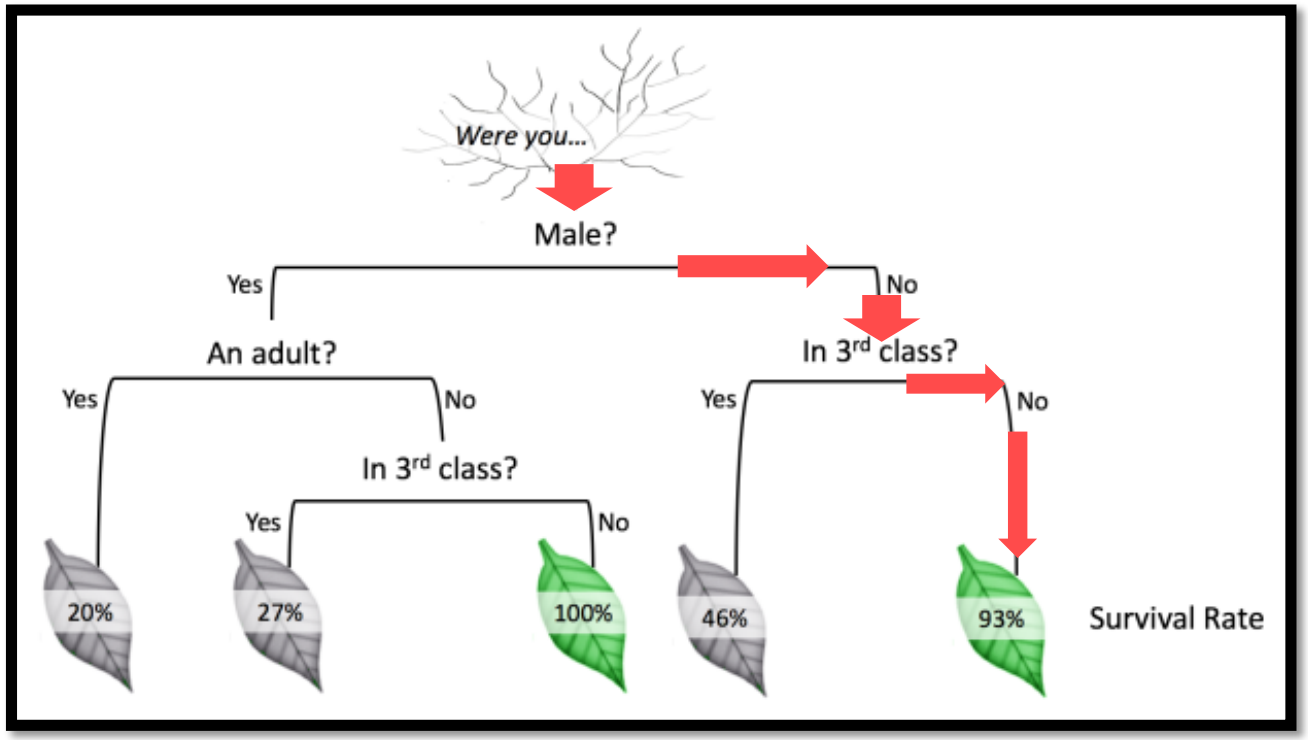
- Example : 타이타닉 호 생존율 예측
  - 탑승자 데이터를 기반으로 생존율 예측하는 decision tree모델을 만들었다.



  - Q : 이승헌 조교가 타이타닉 2등석에 탑승했다. 타이타닉 침몰 후에 생존했을 확률은? **20%**

Decision Tree

- Example : 타이타닉 호 생존율 예측
  - 탑승자 데이터를 기반으로 생존율 예측하는 decision tree모델을 만들었다.



  - Q : 아이유가 타이타닉 1등석에 탑승했다. 타이타닉 침몰 후에 생존했을 확률은? **93%**

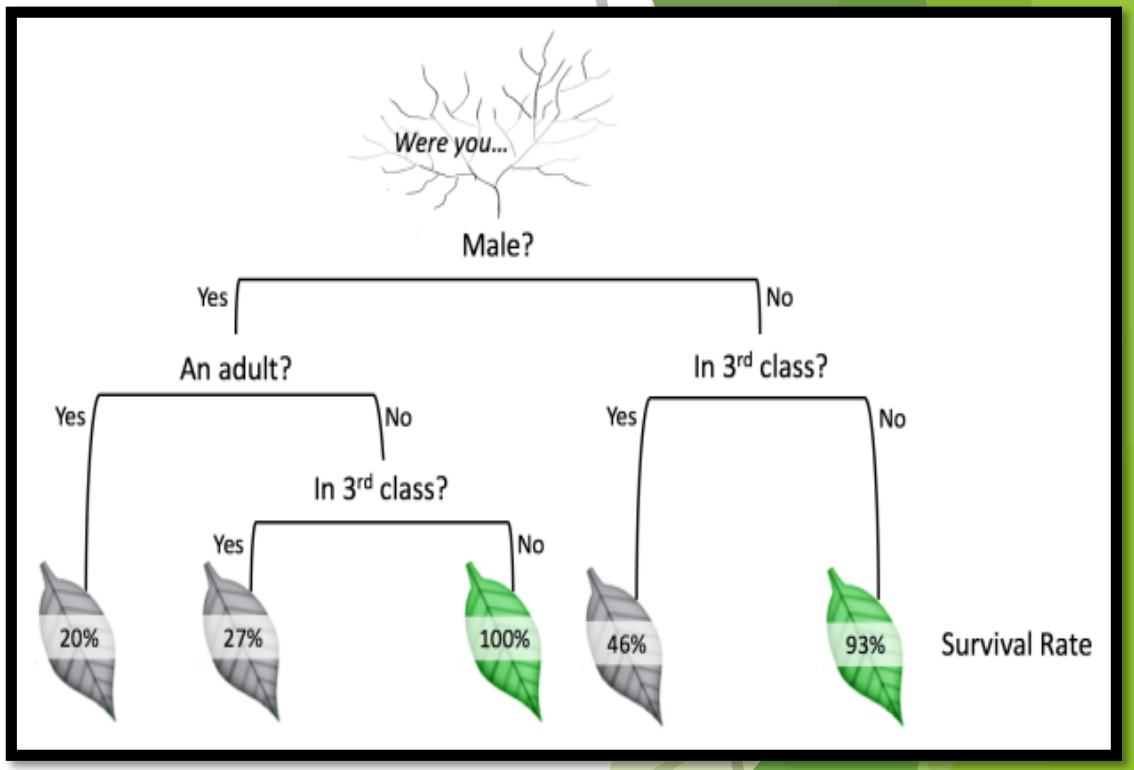
Decision Tree

- 지도학습(supervised learning) 알고리즘
- Training:

  각 node에서 target 값이 잘 분류되도록 feature를 기준으로 데이터를 나눔.
- Predict:

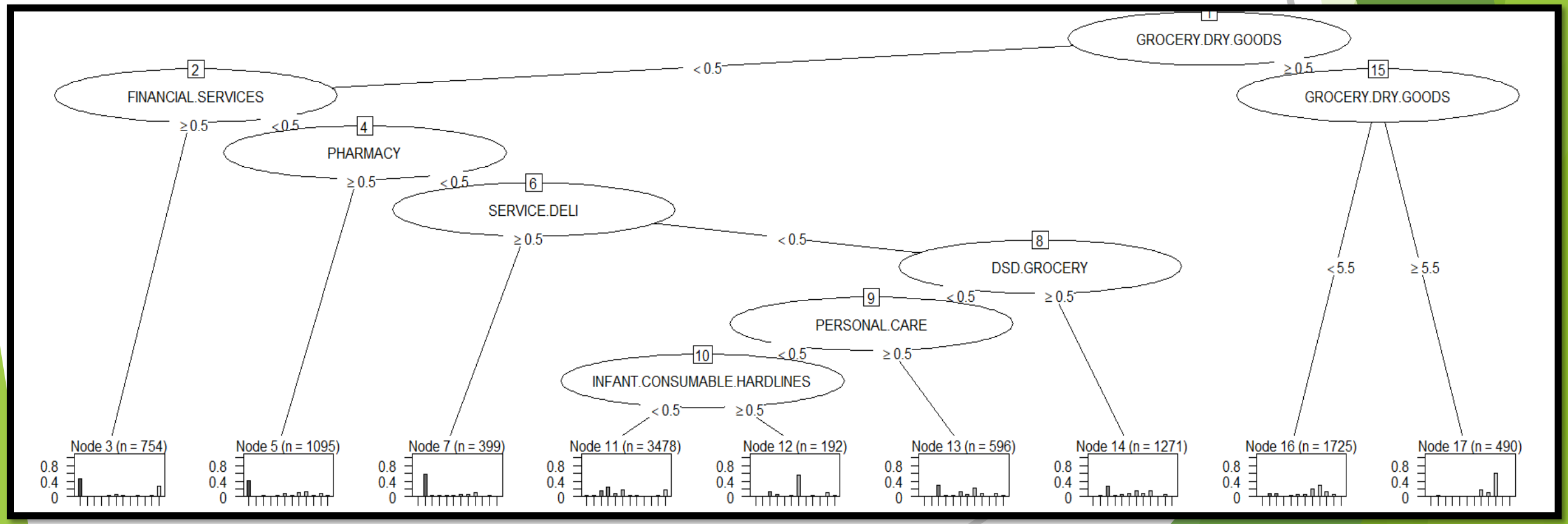
  맨 위에서 시작하여 데이터의 input 값을 이용하여 node의 분류를 따라 내려감.

  예측 결과 = 마지막 node에 모인 데이터들의 target 값으로 결정



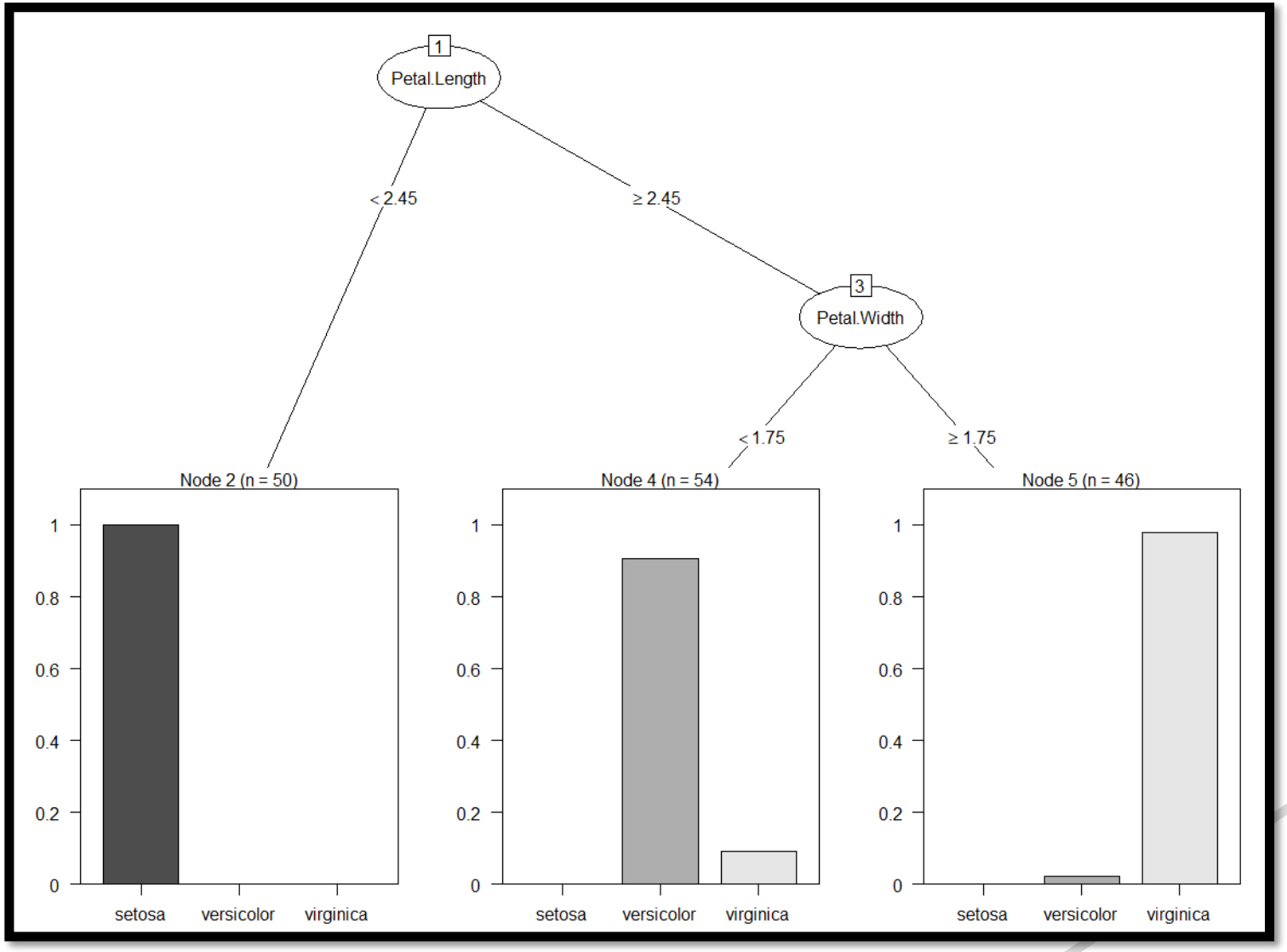
Decision Tree

- 쓰임새
  - Classification, regression 모델
  - 데이터 셋 구조에 대한 통찰(insight) 제공
  - 중요한 feature 확인



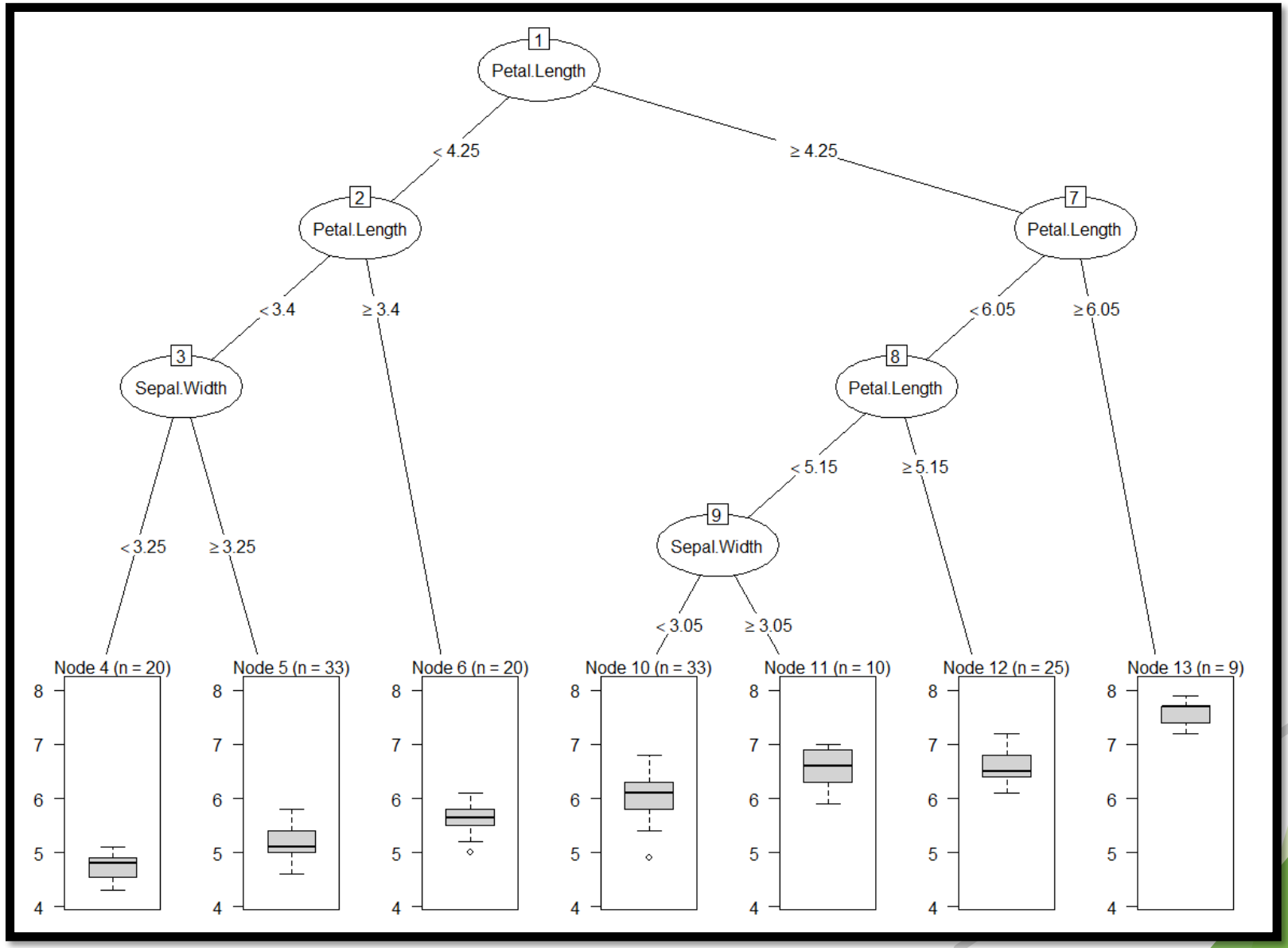
Decision Tree

- Q : class가 여러 개인 경우도 가능? YES



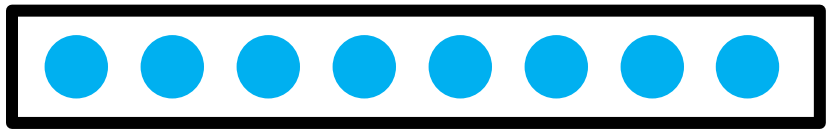
Decision Tree

- Q : regression도 가능? YES

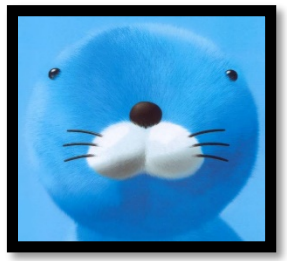


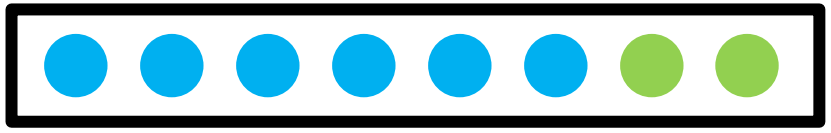
Decision Tree

- Decision Tree 만드는 방법
  - 집합(Set)의 불순도(impurity)
    - 집합이 얼마나 순수(pure)하지 않은가에 대한 척도(여러 척도가 존재함)
    - Ex) 정보 엔트로피: $\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$

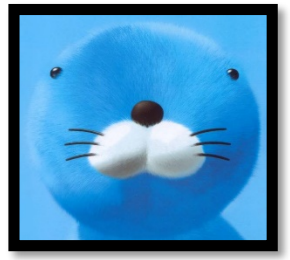


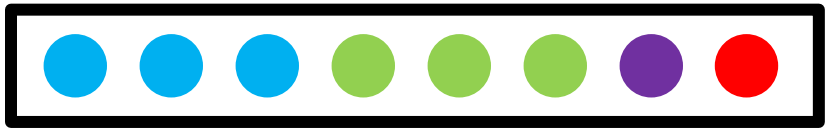
이것은
하나도 안
impure 하다



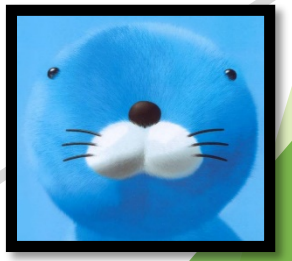


이것은
조금
impure 하다



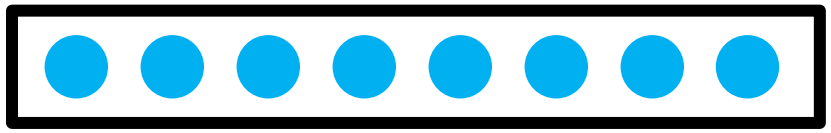


이것은
많이
impure 하다

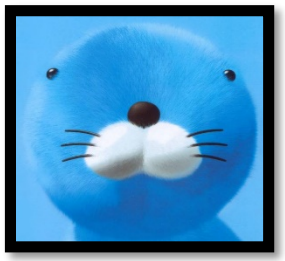


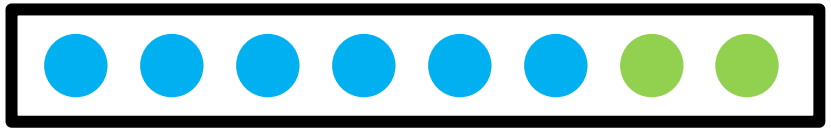
Decision Tree

- Decision Tree 만드는 방법
  - 집합(Set)의 불순도(impurity)
    - 집합이 얼마나 순수(pure)하지 않은가에 대한 척도 (여러 척도가 존재함)
    - Ex) 정보 엔트로피: $\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$

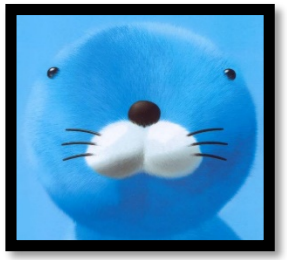


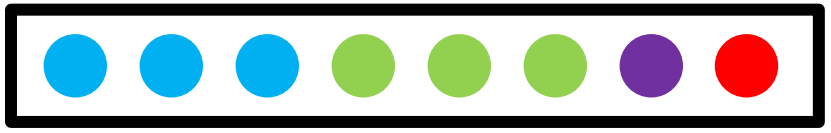
이것은
0만큼
Impure 하다



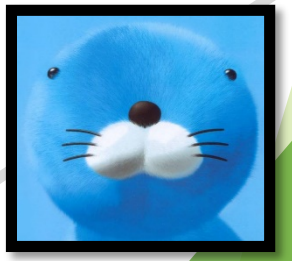


이것은
0.588만큼
impure 하다





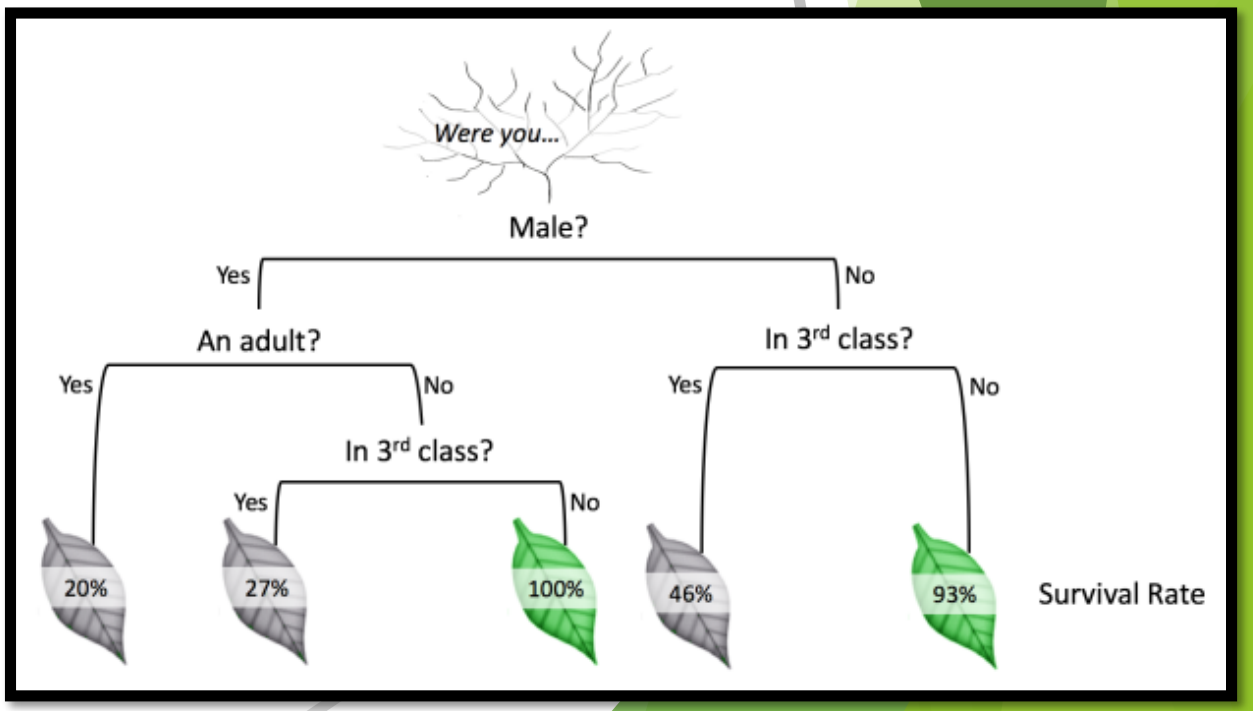
이것은
1.256만큼
impure 하다



Decision Tree

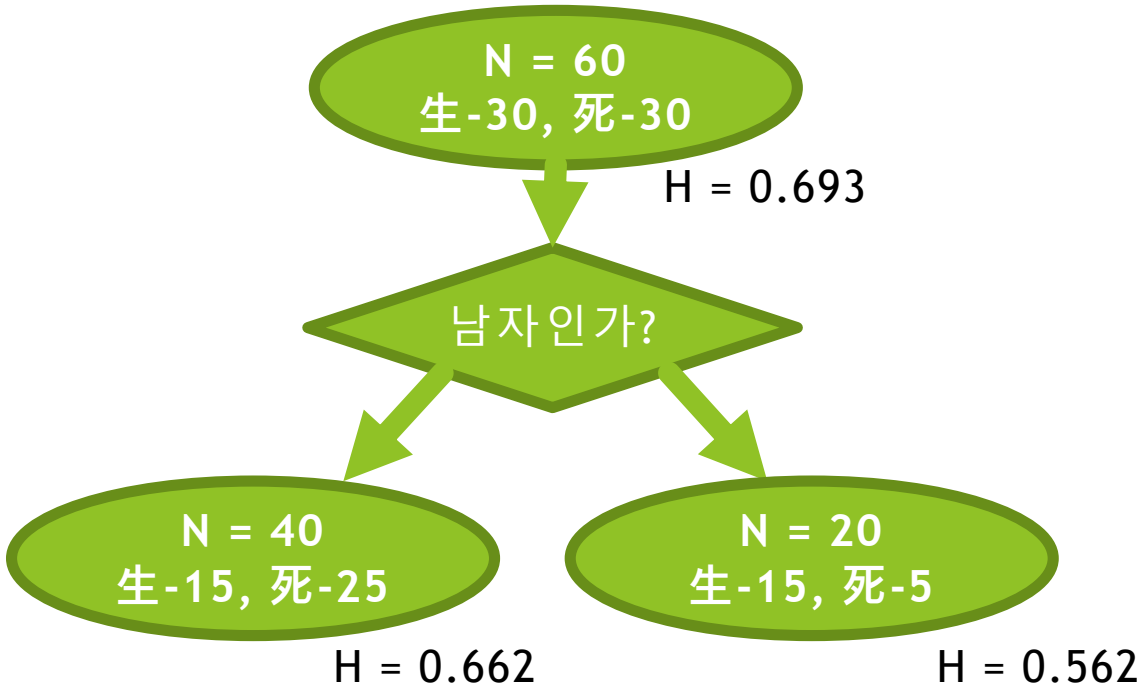
- Example : 타이타닉 호 생존율 예측
  - 탑승자 데이터

이름	성별	성인	객실	사망
디카프리오	남자	성인	3등석	사망
윈슬렛	여자	성인	1등석	생존
손흥민	남자	아동	2등석	생존
⋮	⋮	⋮	⋮	⋮



Decision Tree

- Decision Tree 만드는 방법

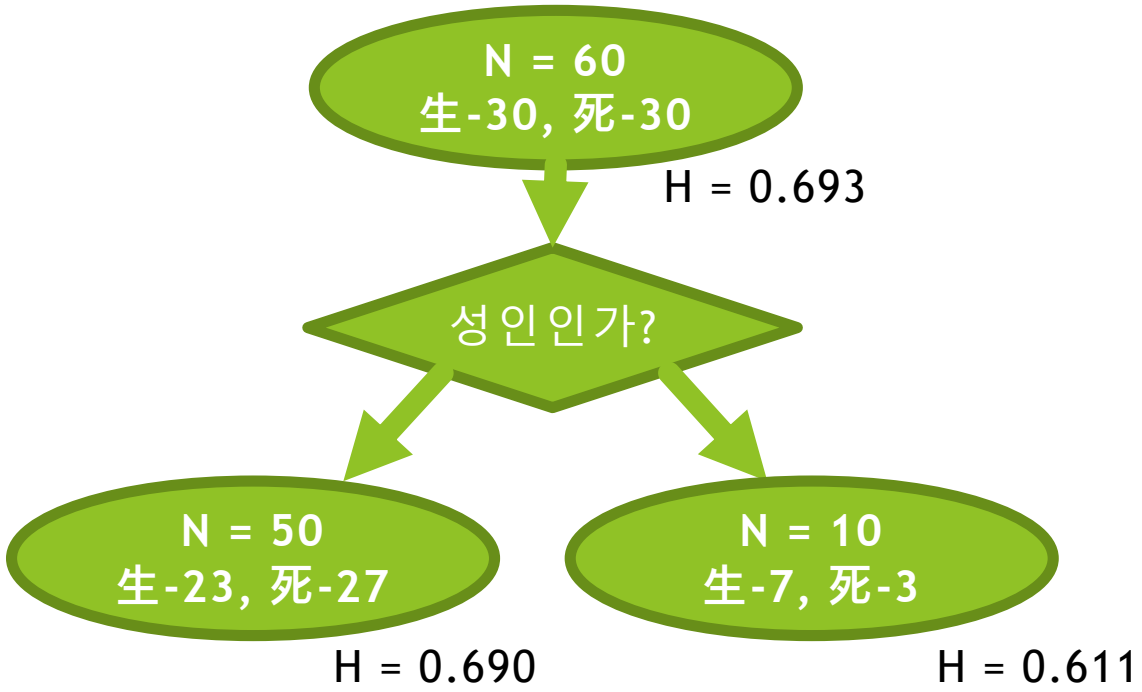


성별을 기준으로 나누었을 때
감소한 impurity(여기선 엔트로피) :

$$\frac{1}{60}[(60 * 0.693) - (40 * 0.662 + 20 * 0.562)]$$
$$=0.064$$

Decision Tree

- Decision Tree 만드는 방법

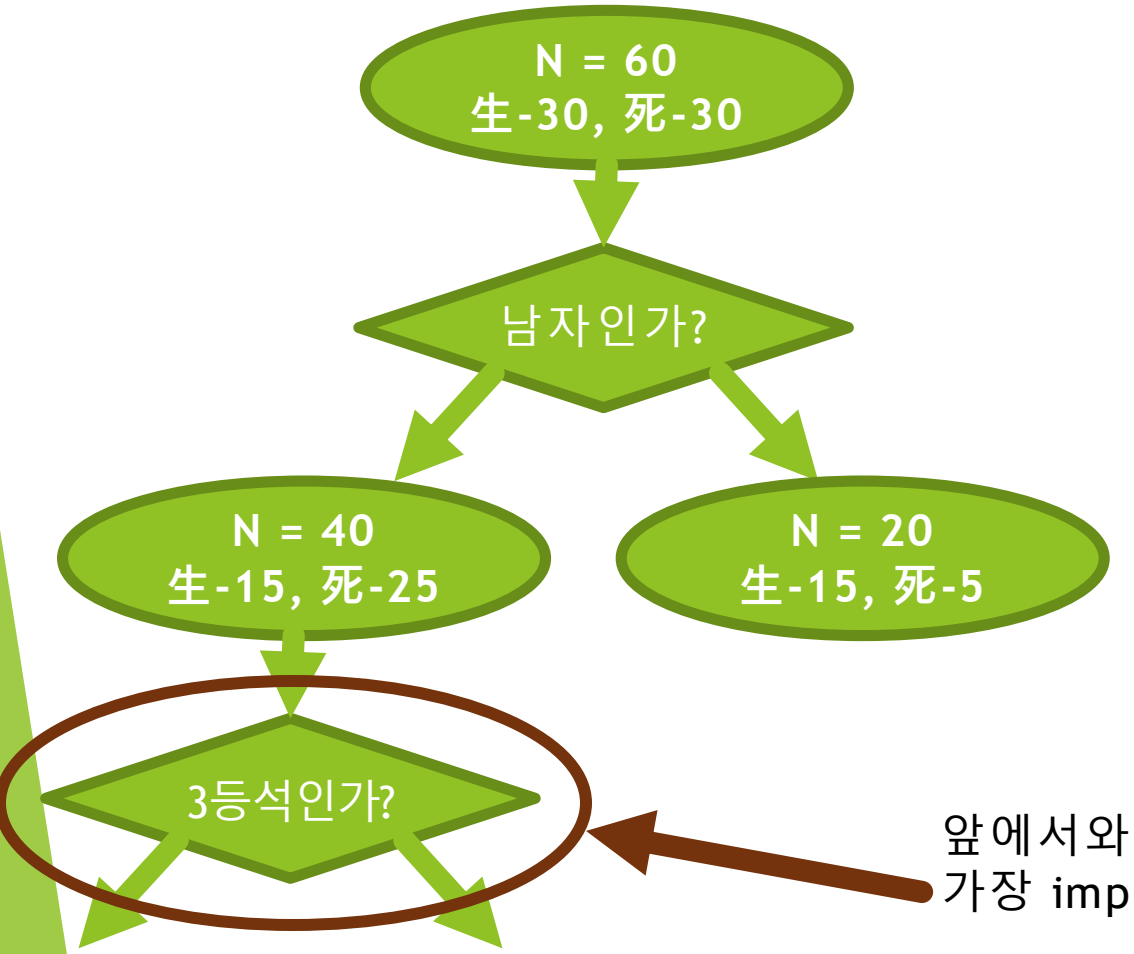


- 성별을 기준으로 나누었을 때
  감소한 impurity : 0.064
- 성인 여부를 기준으로 나누었을 때
  감소한 impurity : 0.016
- 같은 방식으로 모든 feature에 대해서
  impurity 감소 조사

→가장 impurity를 많이 감소시킨 분류 기준 선택

Decision Tree

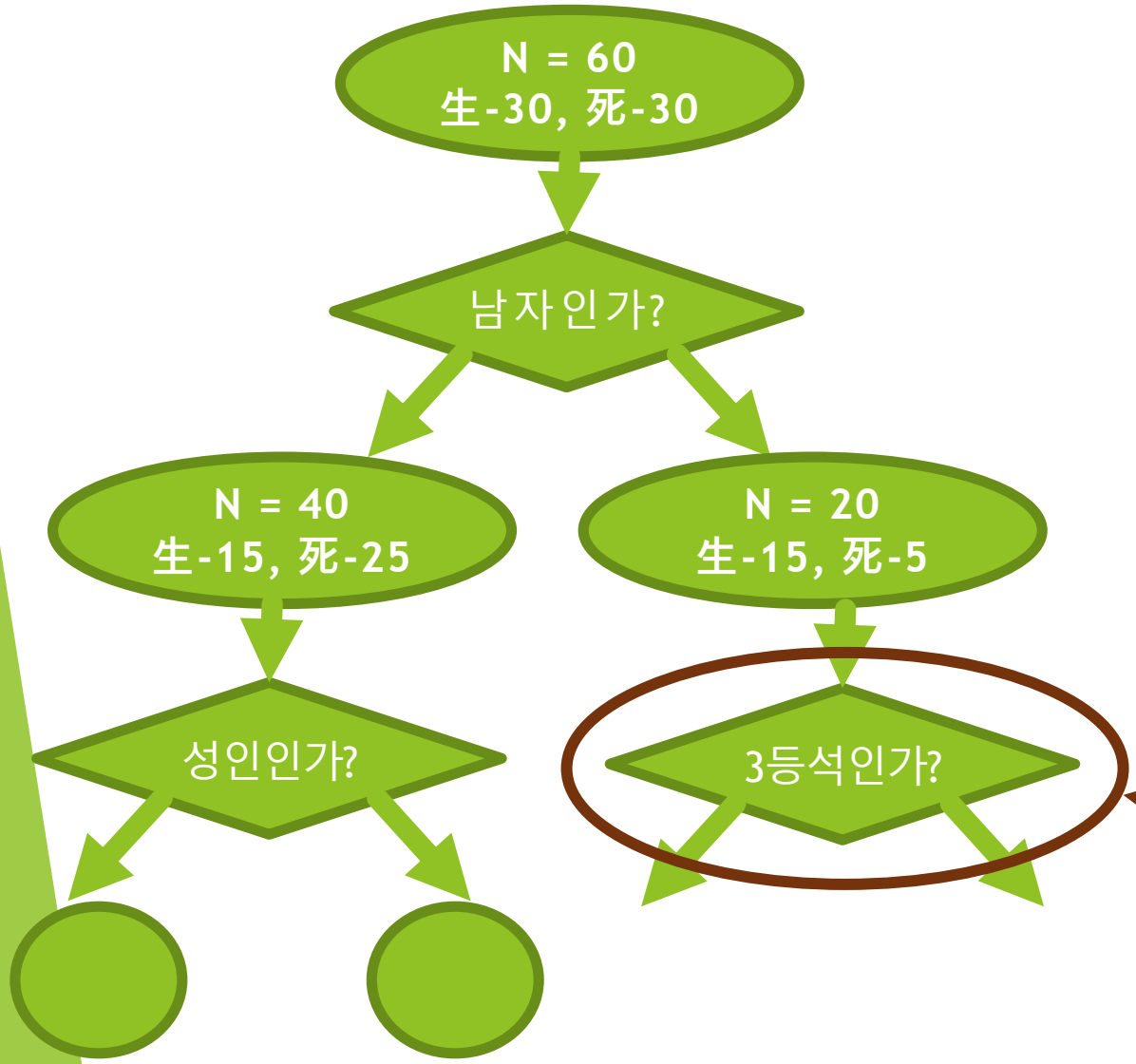
- Decision Tree 만드는 방법



앞에서와 마찬가지로
가장 impurity를 많이 감소시키는 feature 조사

Decision Tree

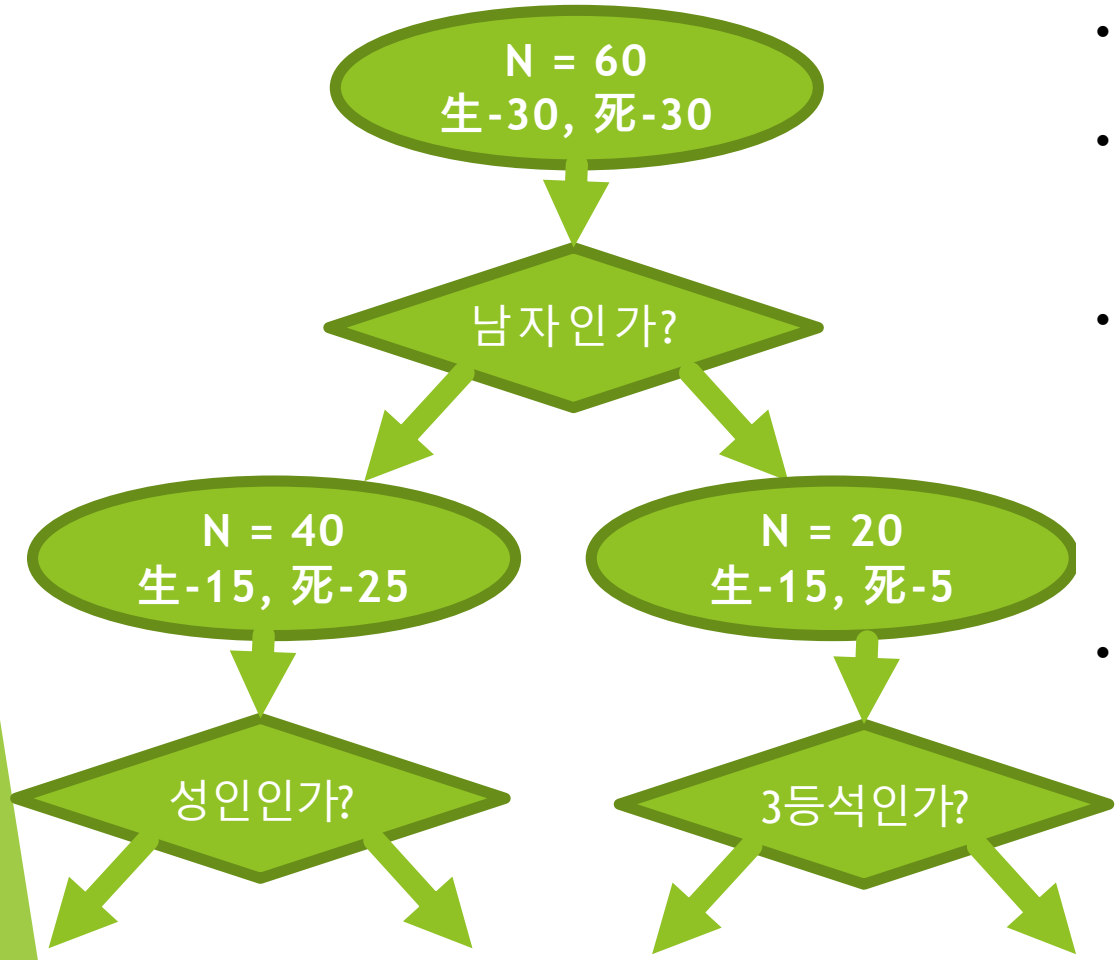
- Decision Tree 만드는 방법



가장 impurity를 많이 감소시키는 feature 조사

Decision Tree

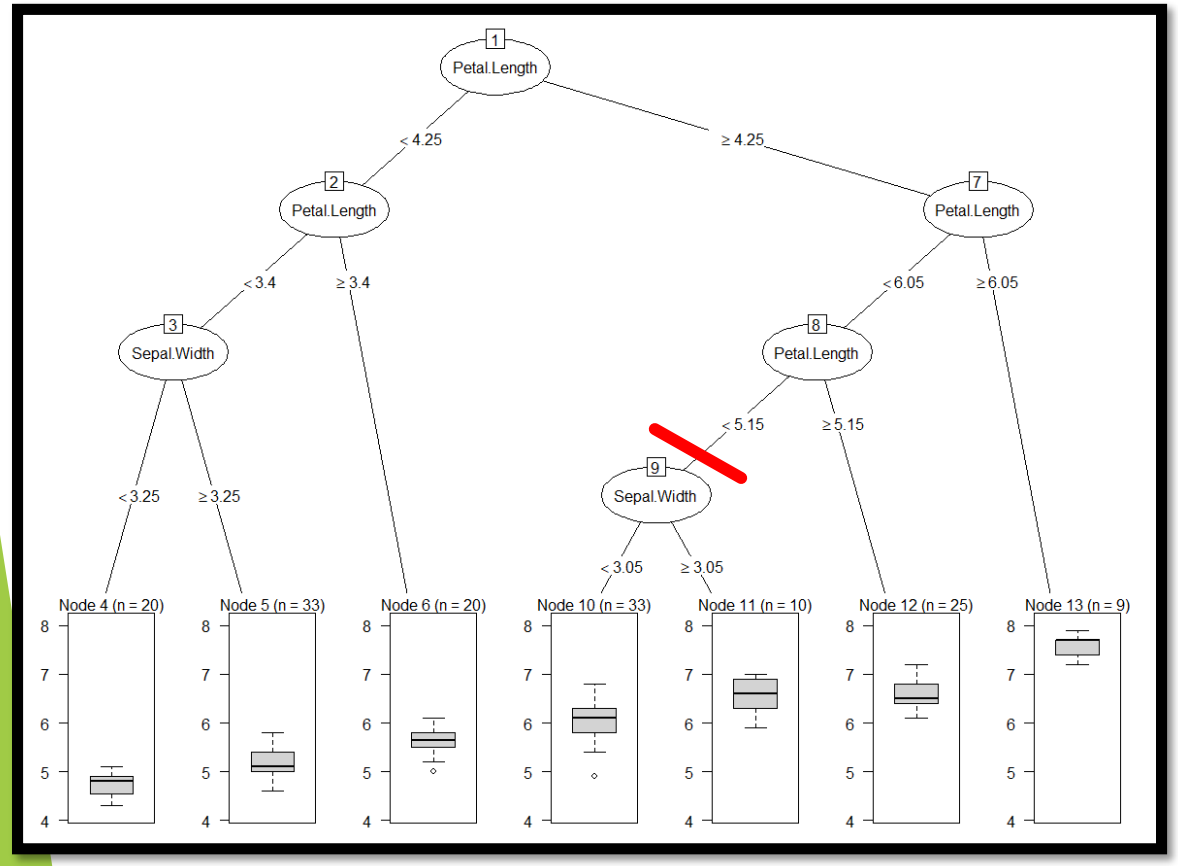
- Decision Tree 만드는 방법

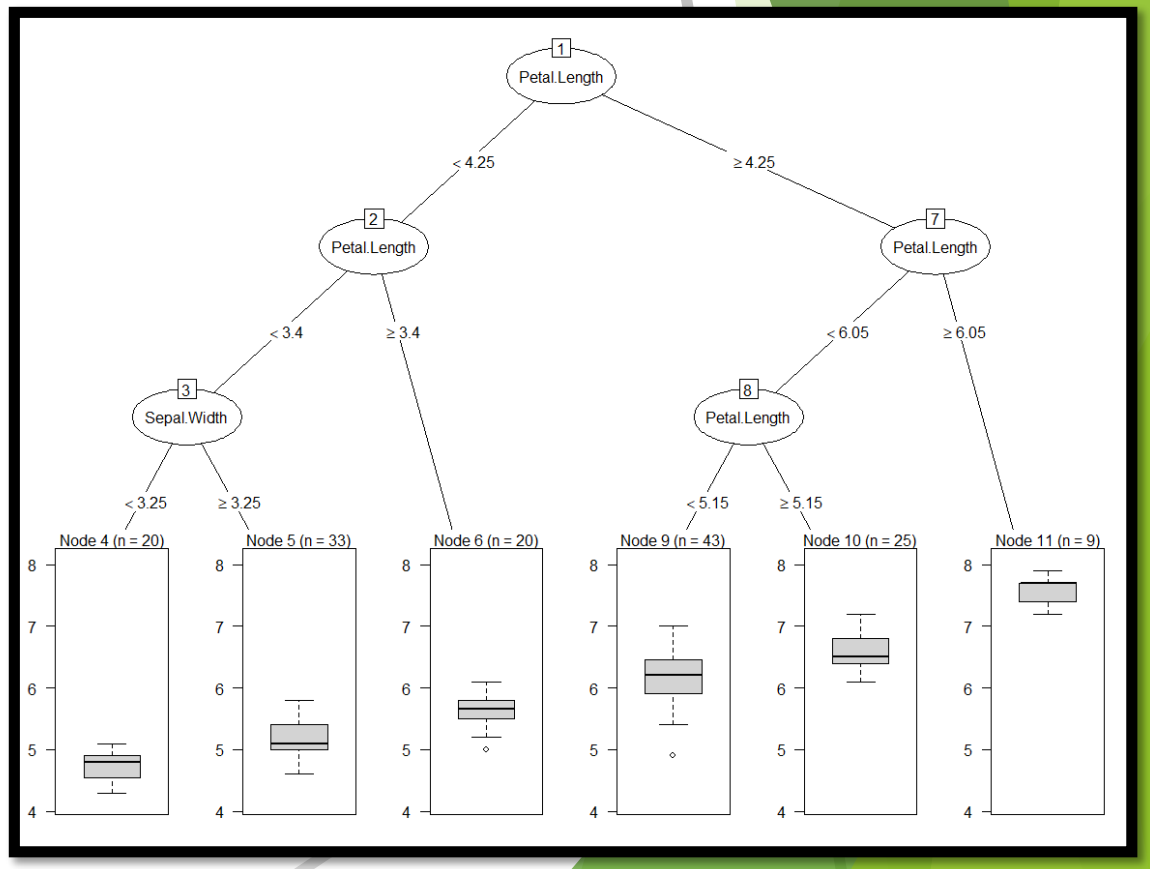


- 이런 식으로 계속 가지를 쳐 나간다.
- 언제까지? 모든 가지에서
  종료조건을 만족할 때까지
- 종료 조건
  - 원하는 깊이에 도달
  - 나눌 데이터의 숫자가 너무 작음
  - Impurity 감소가 너무 작음
  - 등등
- 계산 노가다이지만,
  계산은 누가 한다? 내가 아니라 컴퓨터가 ㅎ

Decision Tree

- Pruning(가지치기)
  - 만들어 놓은 decision tree의 가지를 제거하기
    - Overfitting을 막아 예측 정확도 상승

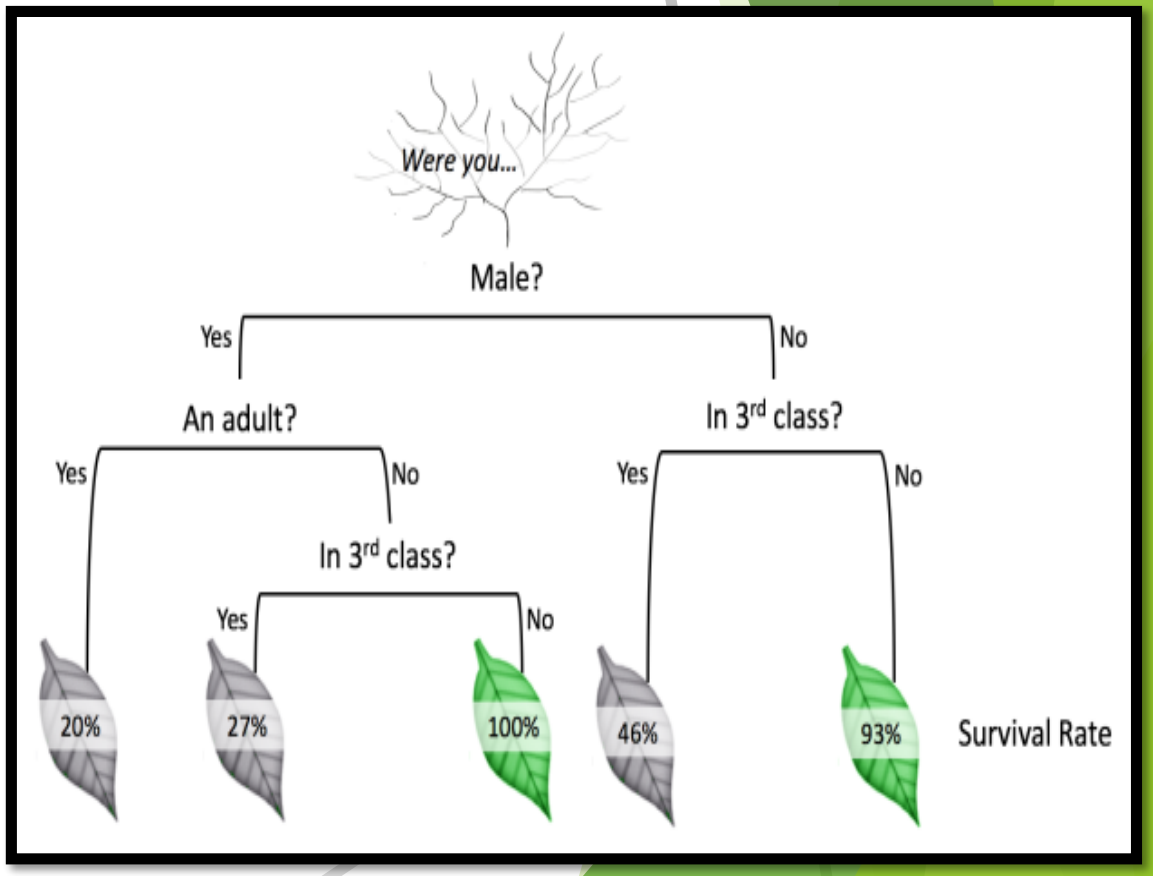




Decision Tree

- 장점
  - **White box algorithm** : 모델이 왜 이런 결과를 내었는지 쉽게 해석이 가능하다.
  - 데이터 셋 구조에 대한 통찰(insight) 제공
  - 중요한 feature 확인
  - simple and fast : 다른 알고리즘에 응용됨

- 단점
  - 한번에 하나의 feature에 대해서만 분류
    - Decision boundary가 축에 수직하다
  - Feature의 수가 많아지면 계산량이 급격히 증가
  - 단독으로 쓰이면 다른 복잡한 모델에 비해 낮은 예측정확도

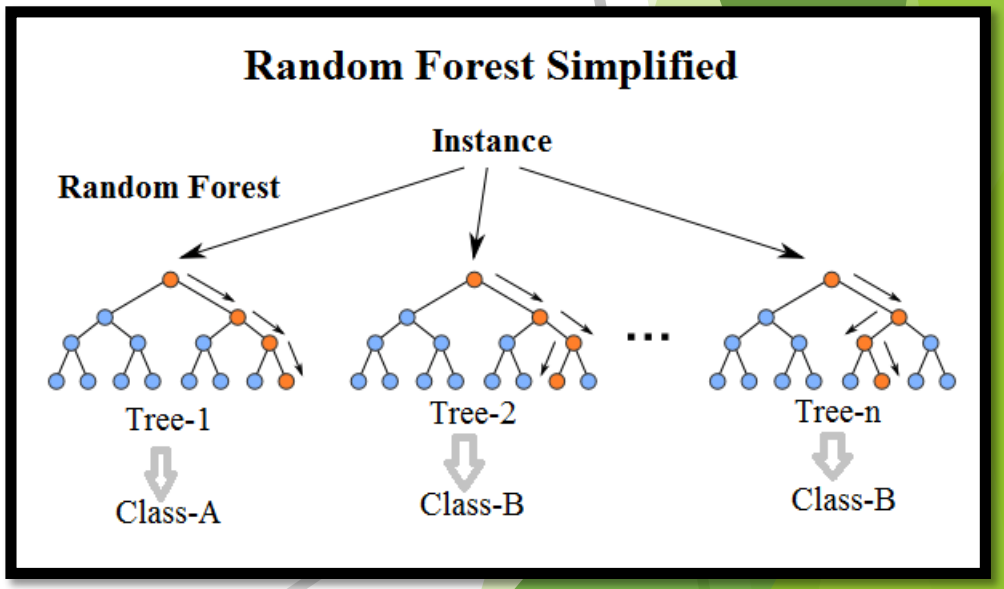


Random Forest

Random Forest

- Theorem
  같은 알고리즘을 사용하는 서로 독립인 예측 모델들을
  평균/다수결로 합치면 정확도가 상승한다.
  - 증명 : One can easily show that…
  - 더 깊이 알고 싶다면
    bias-variance trade off 검색 혹은 가까운 조교에게 문의

- Random Forest
  - 여러 개의 조금씩 다른 decision tree를 만든 뒤
    다수결 투표 결과로 데이터 분류
  - Forest = 숲 = 많은 나무 = 많은 decision tree

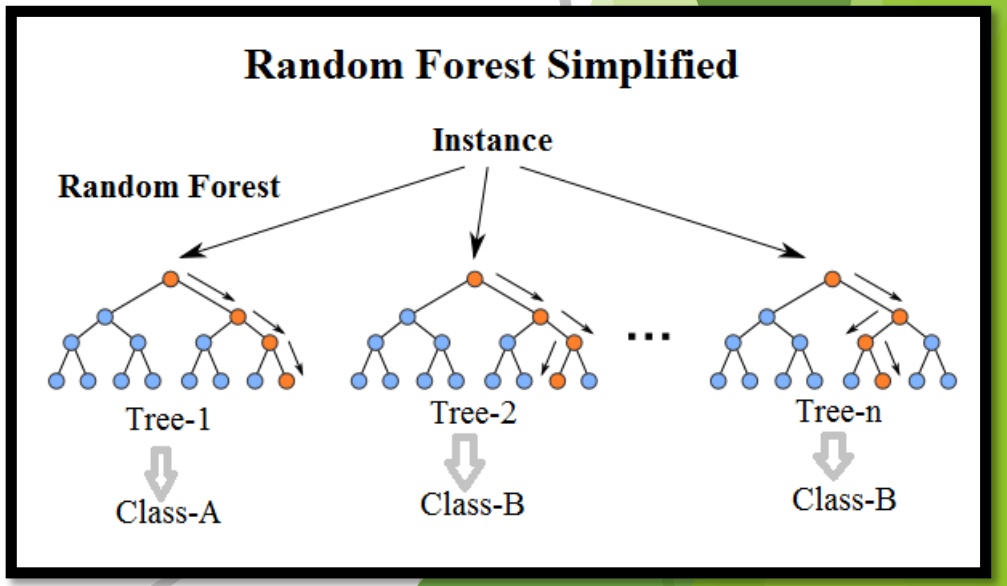


Random Forest

- Random Forest
  - 여러 개의 조금씩 다른 decision tree를 만든 뒤
    다수결 투표 결과로 데이터 분류

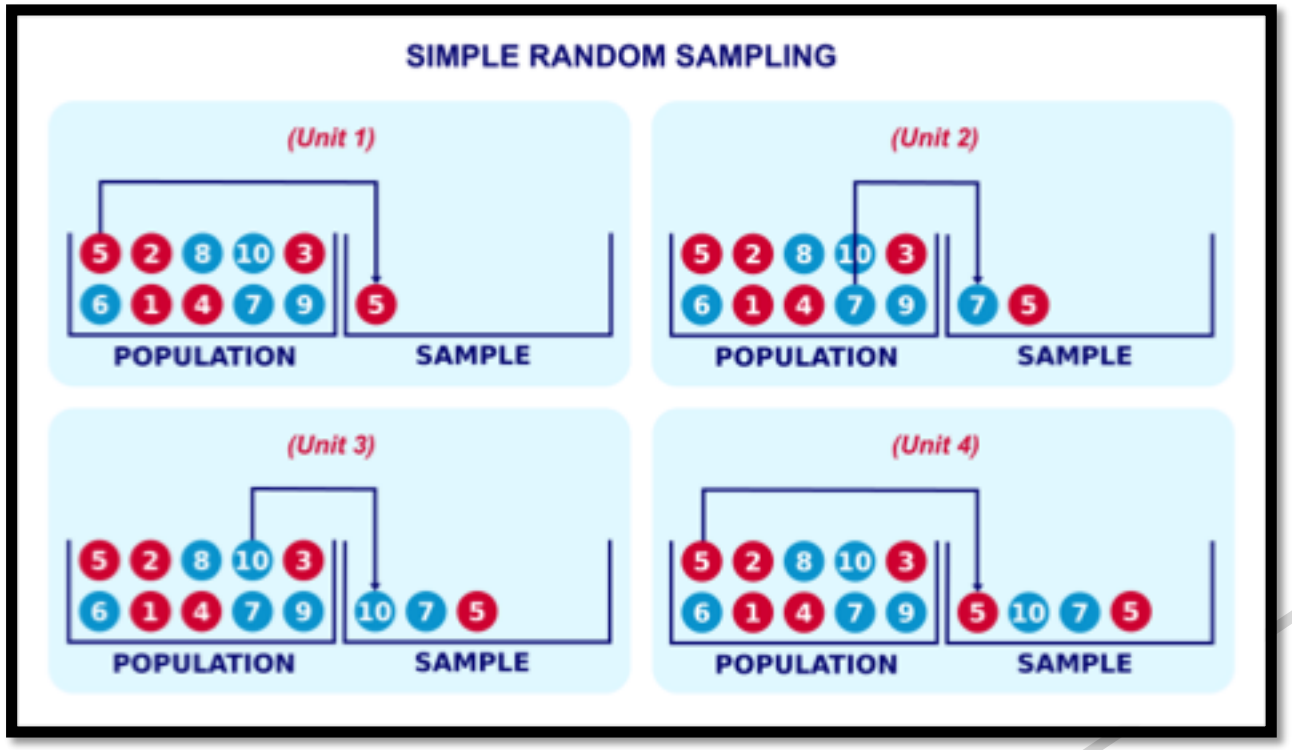
  - Fact 1 : decision tree는 input data가 같으면 항상 같은 모델이 만들어짐.
  - Fact 2 : Theorem에서, 다수결에 참여하는 모델들 사이에
    correlation이 낮을 수록(output이 서로 독립에 가까울수록)
    상승하는 정확도가 커짐.

  - 따라서 각 decision tree가
    가능한 최대한 다르도록 2가지 테크닉을 사용함.
    1. Bagging
    2. Random subspace sampling



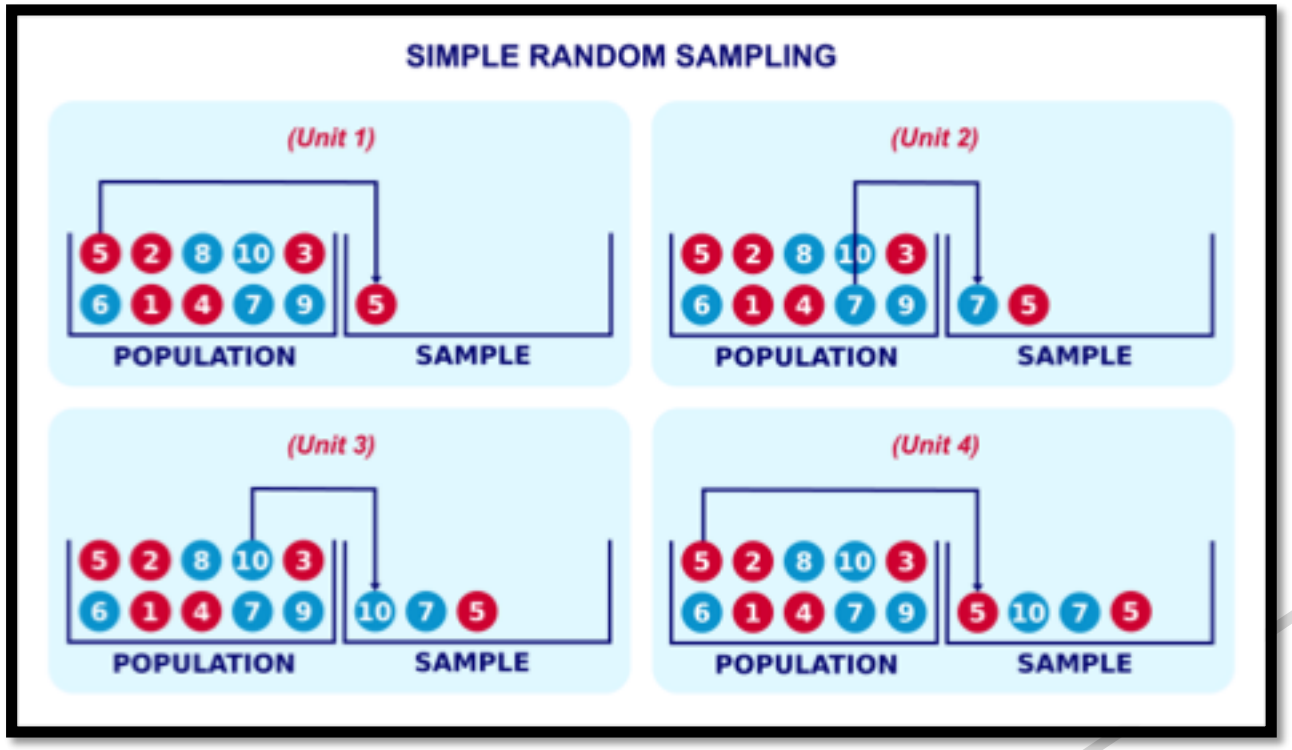
Random Forest

- Bagging
  - Bootstrap aggregation의 약자
  - Training dataset과 비슷하지만 다른 여러 데이터 셋을 만드는 테크닉
  - Training dataset(data n개)에서 n번 복원추출하여 bootstrap set을 만든다.



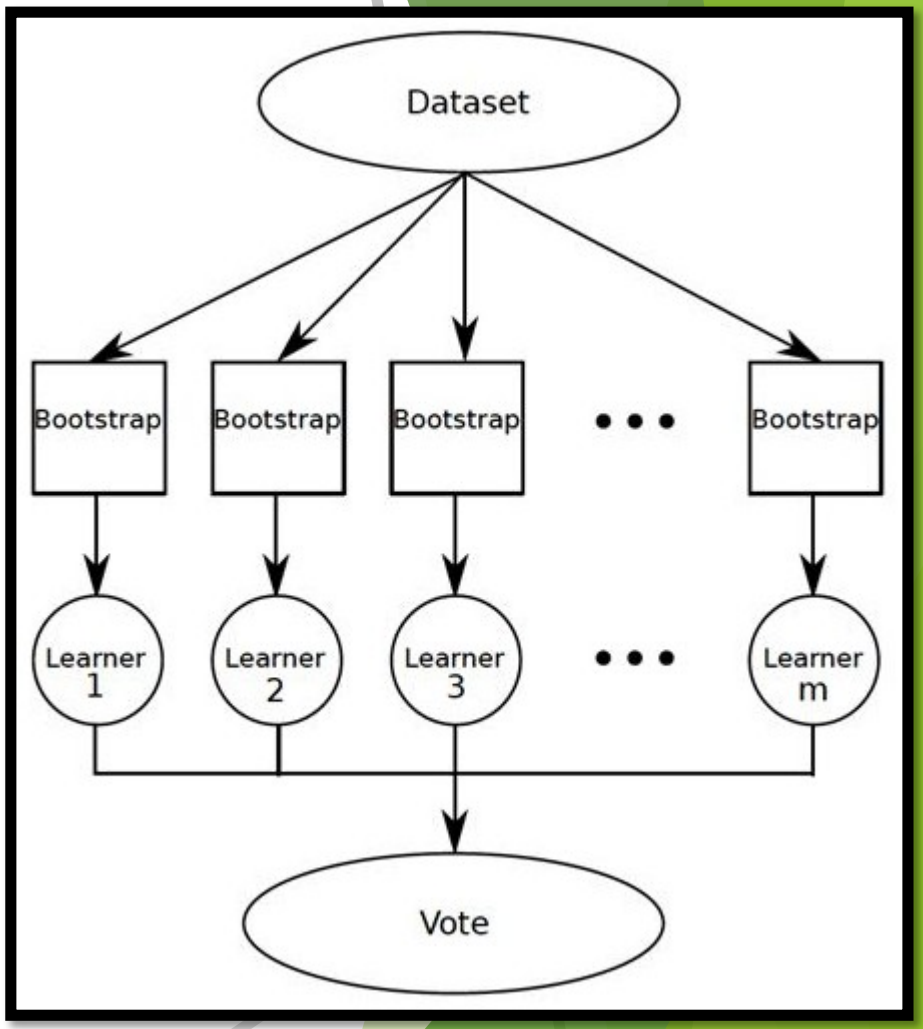
Random Forest

- Bagging
  - Bootstrap aggregation의 약자
  - Training dataset과 비슷하지만 다른 여러 데이터 셋을 만드는 테크닉
  - Training dataset(data n개)에서 n번 복원추출하여 bootstrap set을 만든다.



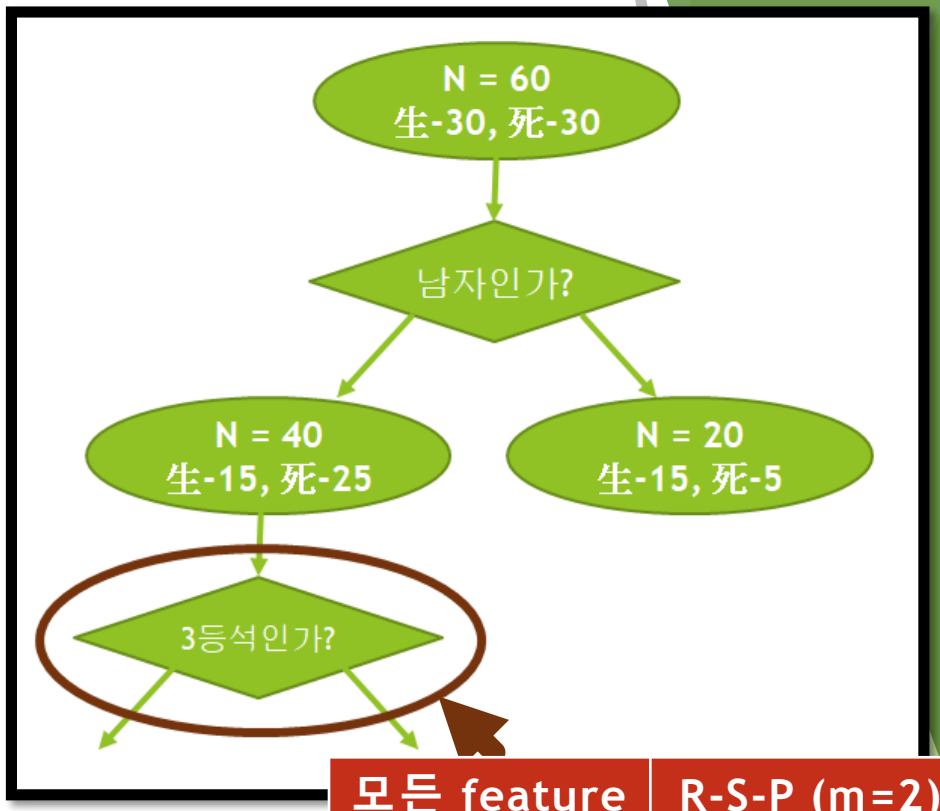
Random Forest

- Bagging
  - Training dataset(data n개)에서 n번 복원추출하여 bootstrap set을 만든다.
  - bootstrap set을 여러 개를 만든다.
  - 각 bootstrap set으로 예측모델을 training한다.
  - 모델의 output = 각 모델의 결과들을 다수결



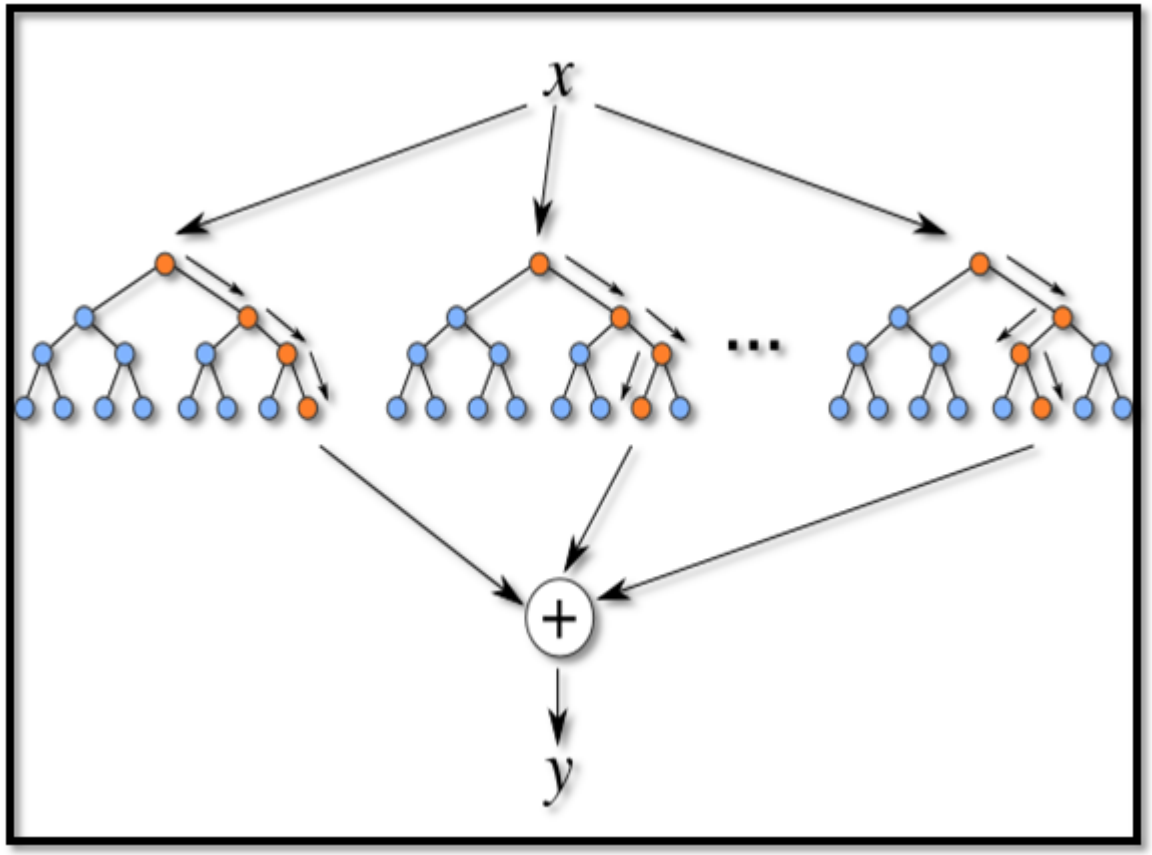
Random Forest

- Random subspace sampling
  - Decision Tree의 node를 만들 때
    - 모든 feature 중에서 impurity를 많이 감소시키는 feature를 선택한다.
    - 근데 random subspace sampling은,
      모든 feature중에서 찾는 게 아닌
      모든 feature 중 랜덤으로 m개를 뽑아서
      그 중에서 impurity 감소를 조사한다.

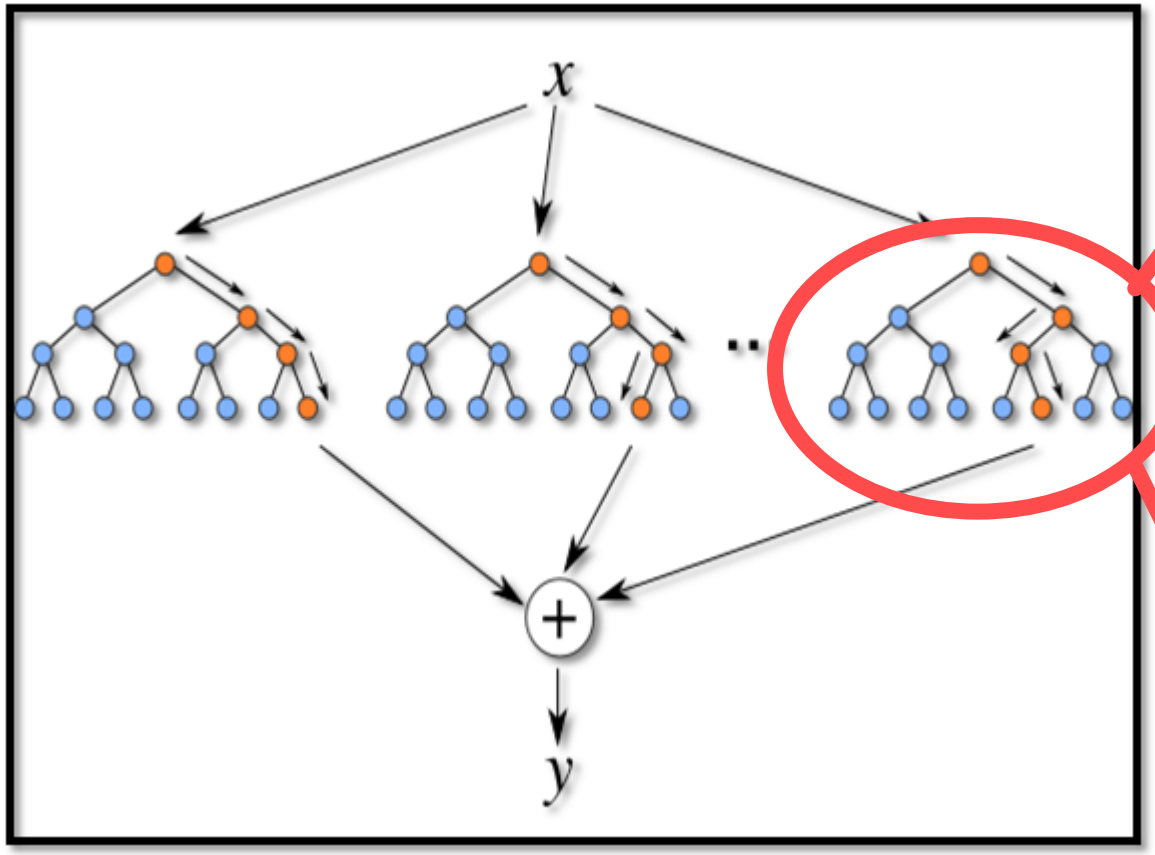


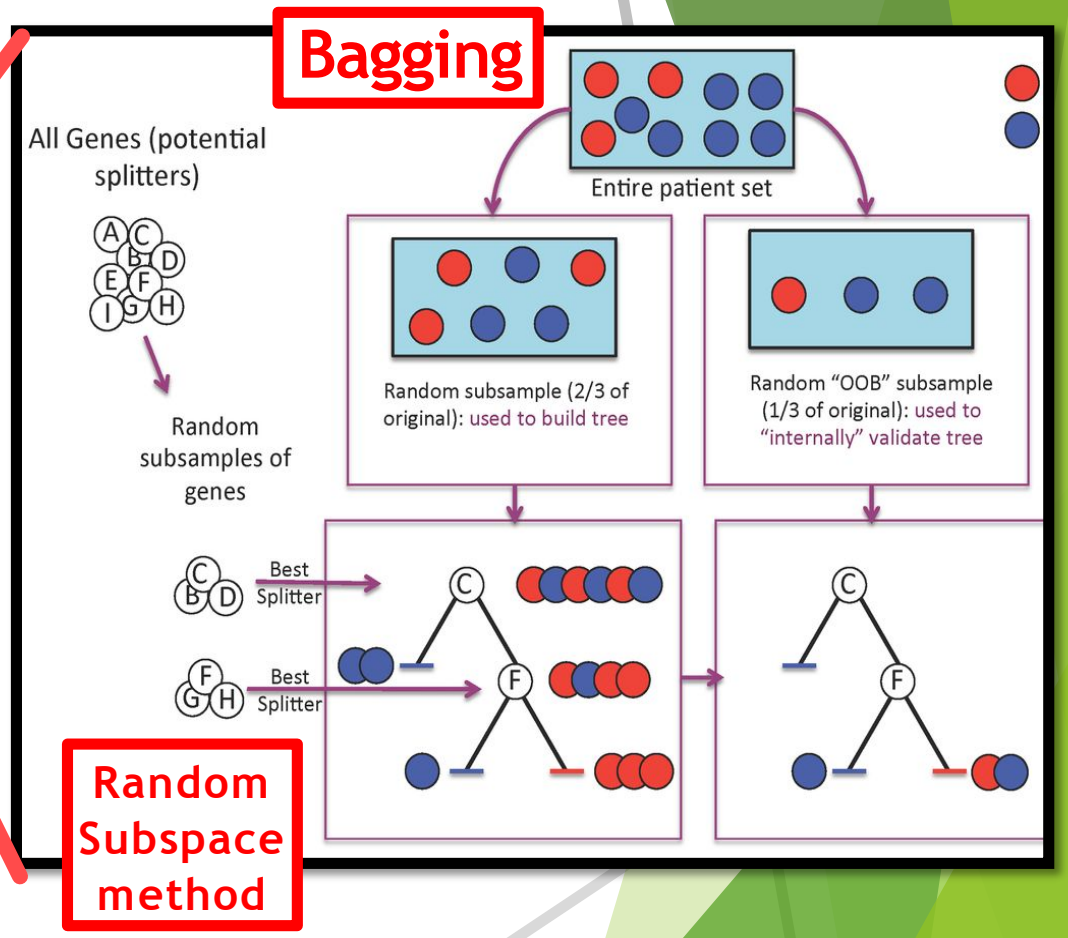
모든 feature	R-S-P (m=2)
성별	성별
성인	객석
객석	

Random Forest



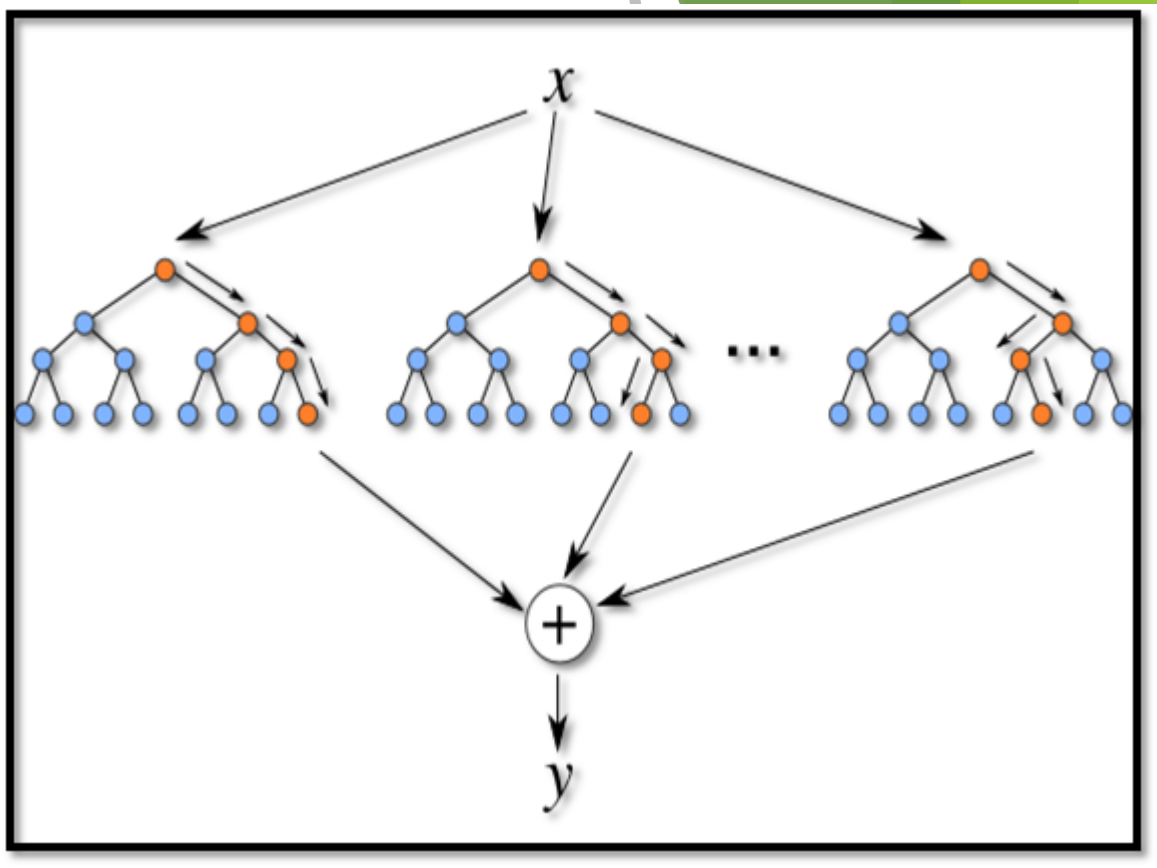
Random Forest





Decision Tree

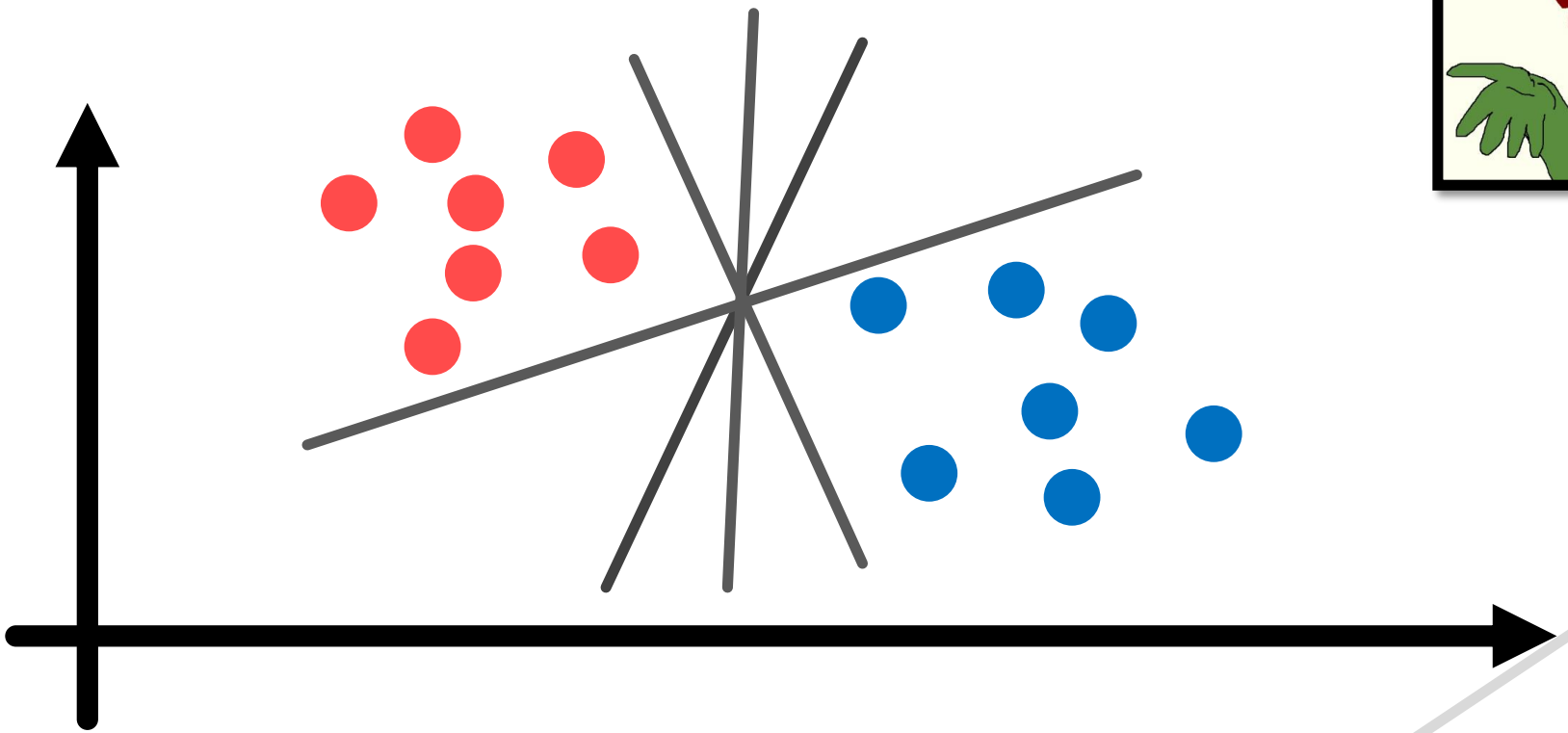
- 장점
  - **많이 정확하다.**
  - 좋은 '부산물'
    - 변수중요도 제공
    - 자체 error rate 제공
- 단점
  - 한번에 하나의 feature에 대해서만 분류
    - Decision boundary가 축에 수직하다
  - Feature의 수가 많아지면 계산량이 급격히 증가



Support Vector Machine

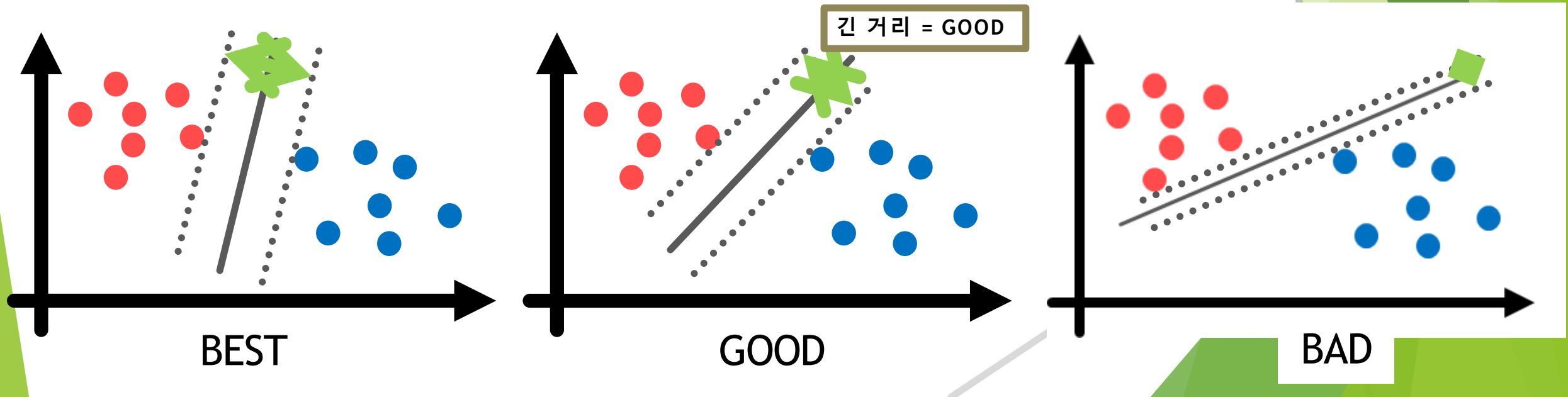
Support Vector Machine(SVM)

- 주어진 데이터에서
  red와 blue를 분류하는 직선(혹은 hyperplane)을 찾는 문제
- Red와 blue를 나누는 직선은 무수히 많다.
- 그 중에 어떤 직선이 제일 좋을까?



Support Vector Machine(SVM)

- 이런 기준은 어떨까?
  - (직선-가장 가까운 red 거리) + (직선-가장 가까운 blue 거리)가 크다 = GOOD

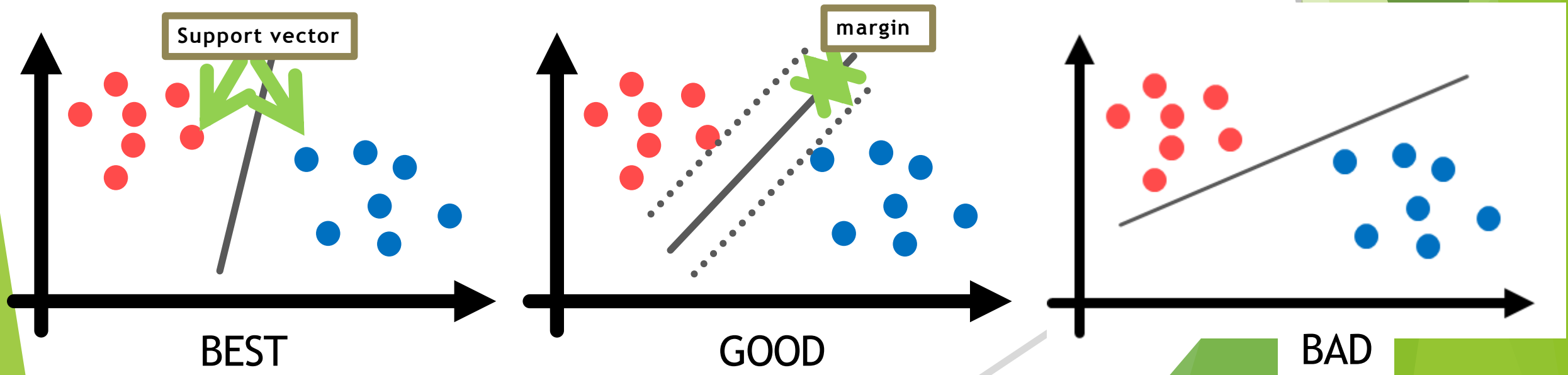


Support Vector Machine(SVM)

- 이런 기준은 어떨까?
  - (직선-가장 가까운 red 거리) + (직선-가장 가까운 blue 거리)가 크다 = GOOD

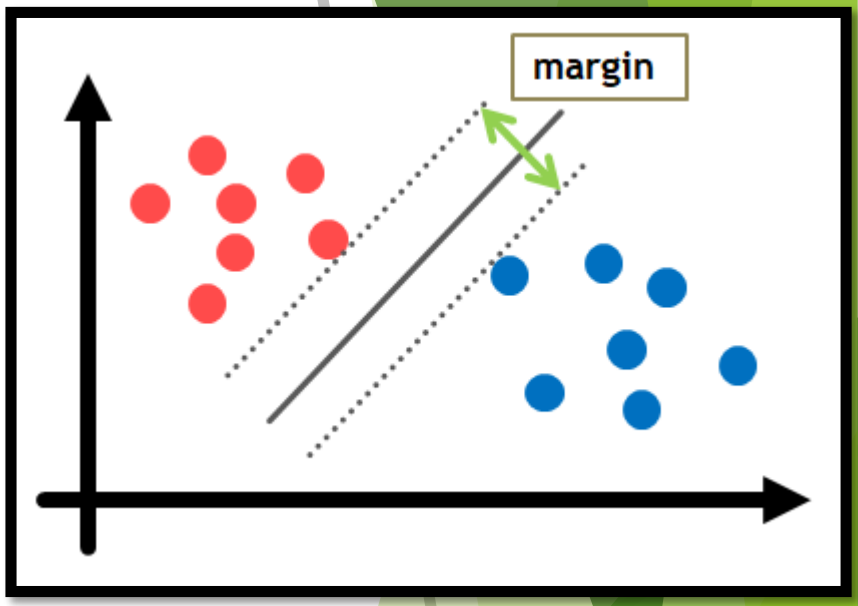
≡ margin
가장 가까운 red, blue : support vector

  - margin이 클수록 분류가 안정적이다.
  - 따라서 margin이 최대가 되는 직선을 찾자!
- 이것이 SVM의 Key Idea



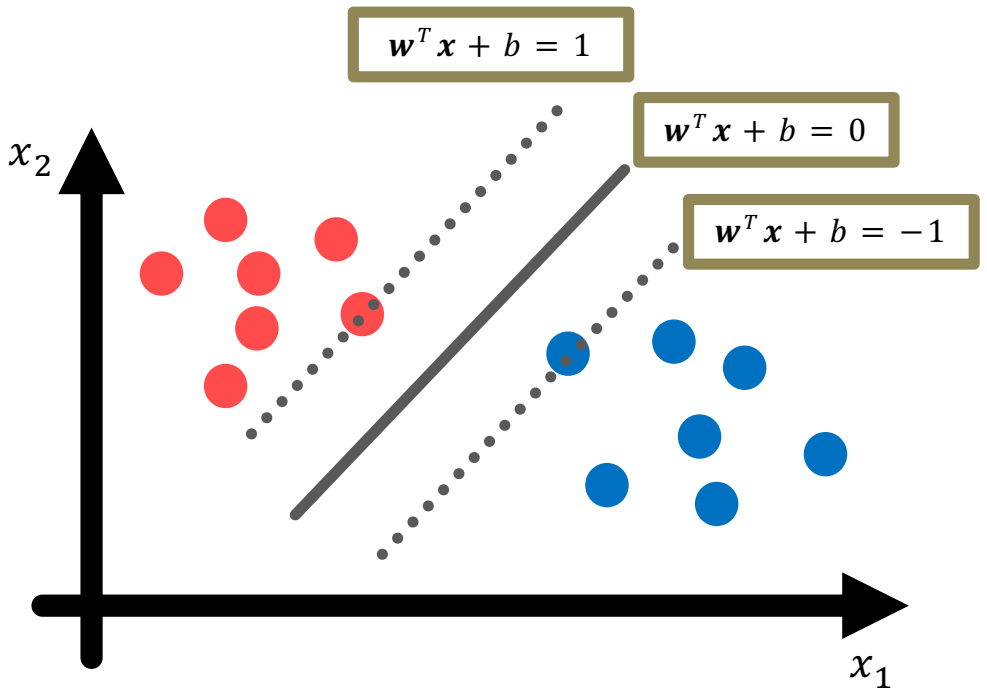
Support Vector Machine(SVM)

- 두 집단을 분류하는 직선(혹은 hyperplane)을 찾는 문제
- Margin이 최대가 되는 직선을 찾으면 된다!



- **Q : 근데 그 좋은걸 어떻게 찾나요?**

Support Vector Machine(SVM)



- 직선(hyperplane)의 방정식 : $w^T x + b = 0$
- 직선 $w^T x + b = 0$과 support vector 사이의 거리 : $\frac{1}{||w||_2}$
- Margin = $\frac{1}{||w||_2} + \frac{1}{||w||_2} = \frac{2}{||w||_2}$ 를

  최대화 하는 w와 b를 찾자!

Red는 직선의 왼쪽, Blue는 직선의 오른쪽 :
x_1가 Red에 속하는 점이면 $w^T x + b > 1$
x_2가 Blue에 속하는 점이면 $w^T x + b < 1$

Support Vector Machine(SVM)

- 풀어야 하는 수학문제 :

- Find w, b such that

  maximize $\frac{2}{||w||_2}$ = minimize $\frac{1}{2}||w||_2^2$

  - x_1가 Red에 속하는 점이면 $w^T x + b > 1$
  - x_2가 Blue에 속하는 점이면 $w^T x + b < 1$

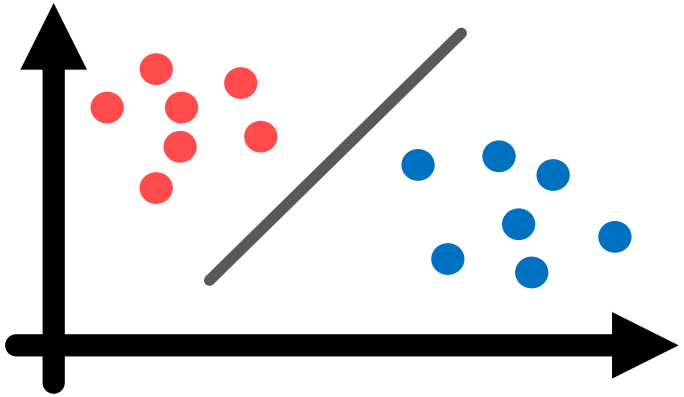
- 이런 문제를 푸는 수학 분야 :

Optimization

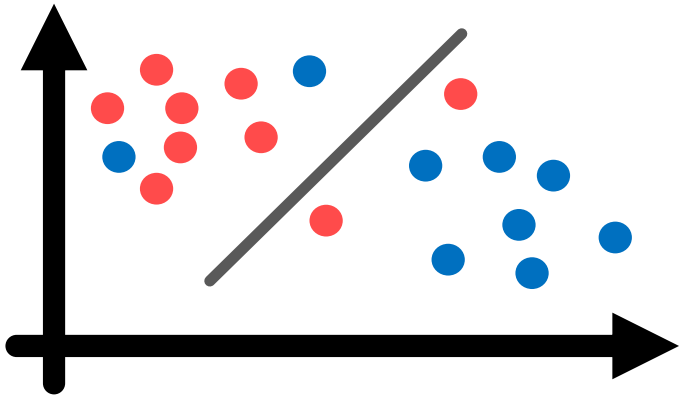
Optimization이론을 이용해서 풀면 된다!
어려워서 생략
더 깊이 알고 싶다면
SVM 검색 혹은 가까운 조교에게 문의

Support Vector Machine(SVM)

- Q : 데이터가 이렇게 이쁘게 나뉘어 있지 않고



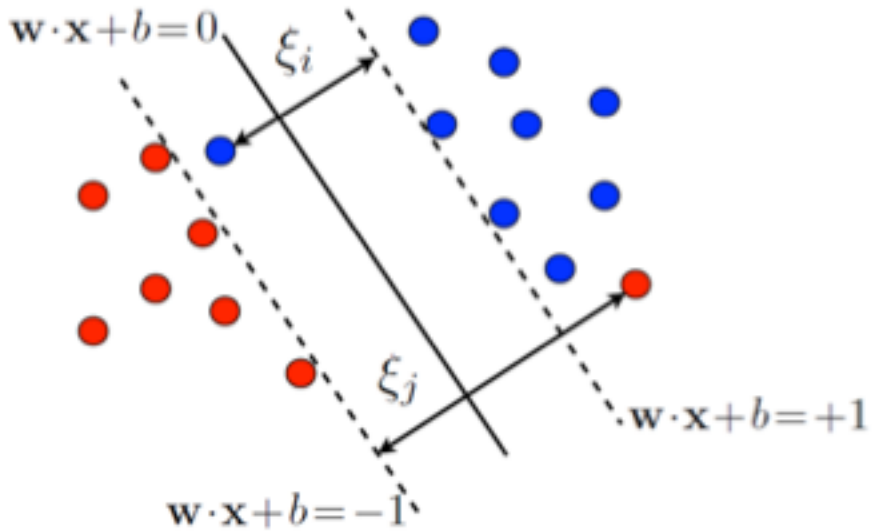
이렇게 섞여있으면 못쓰지 않나요?



- **A:** 그럴까봐 만들었습니다. **C-SVM**

Support Vector Machine(SVM)

- C-SVM
  - 분류 선을 넘긴 데이터를 허용, but 넘은 만큼 페널티



Find w, b such tha

maximize $\frac{2}{\|w\|_2}$ =

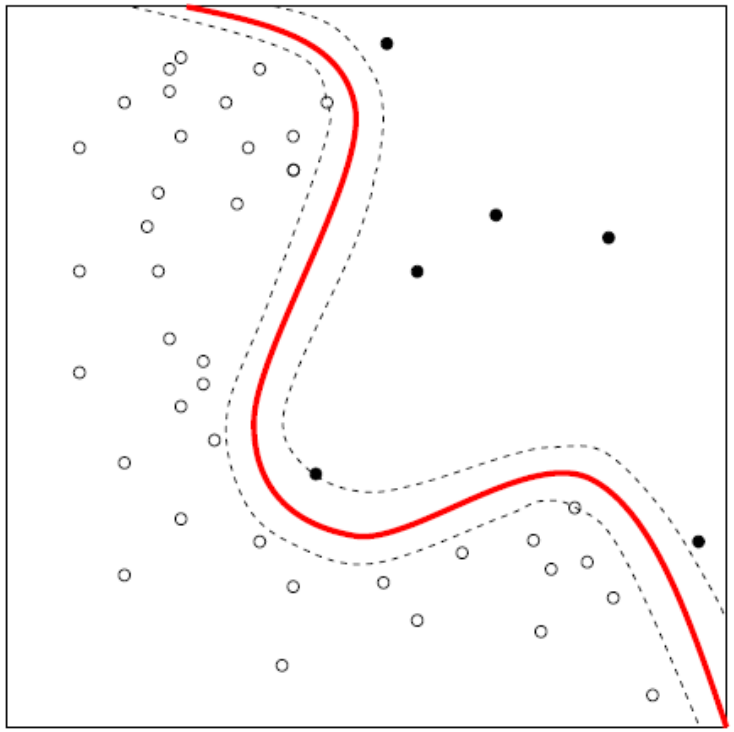
$$\min \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

- x_1가 Red에 속하는 점이면 $w^T x + b > 1 - \xi_i, \quad \xi_i \geq 0$
- x_2가 Blue에 속하는 점이면 $w^T x + b < 1 - \xi_i, \quad \xi_i \geq 0$

→ Optimization

Support Vector Machine(SVM)

- Q : 데이터가 이렇게 직선으로 못나뉘면 못쓰잖아요.



- **A:** 그럴까봐 만들었습니다. **Kernel Method**

Support Vector Machine(SVM)

- Kernel Method
  - 데이터를 변형시켜서 모델에 넣는 방법
  - Ex)
    - 2차원 데이터 : (키,몸무게)
      → 3차원 데이터 (키,몸무게,BMI = $\frac{\text{키}^2}{\text{몸무게}}$)
  - $(f_1, f_2, \cdots, f_n) \overset{\phi}{\rightarrow} \phi\big((f_1, f_2, \cdots, f_n)\big) = (g_1, g_2, \cdots, g_n)$
  - 어떤 알고리즘들은 ϕ를 직접 정의하지 않고 $\phi(x_i) \cdot \phi(x_j)$만 정의해도 됨
    - 예를 들어 SVM

Support Vector Machine(SVM)

- Kernel Method
  - 어떤 알고리즘들은 ϕ를 직접 정의하지 않고

    $\phi(x_i) \cdot \phi(x_j)$만 정의해도 됨

    - Polynomial kernel : $K(x, y) = (x^\mathsf{T} y + c)^d$
    - RBF kernel : $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$
    - 등등


uck this s it. I'm out.
끝
someecards
user card