기계학습(Machine Learning)

발표자 : 하상천

2020.08.03

팀 프로젝트

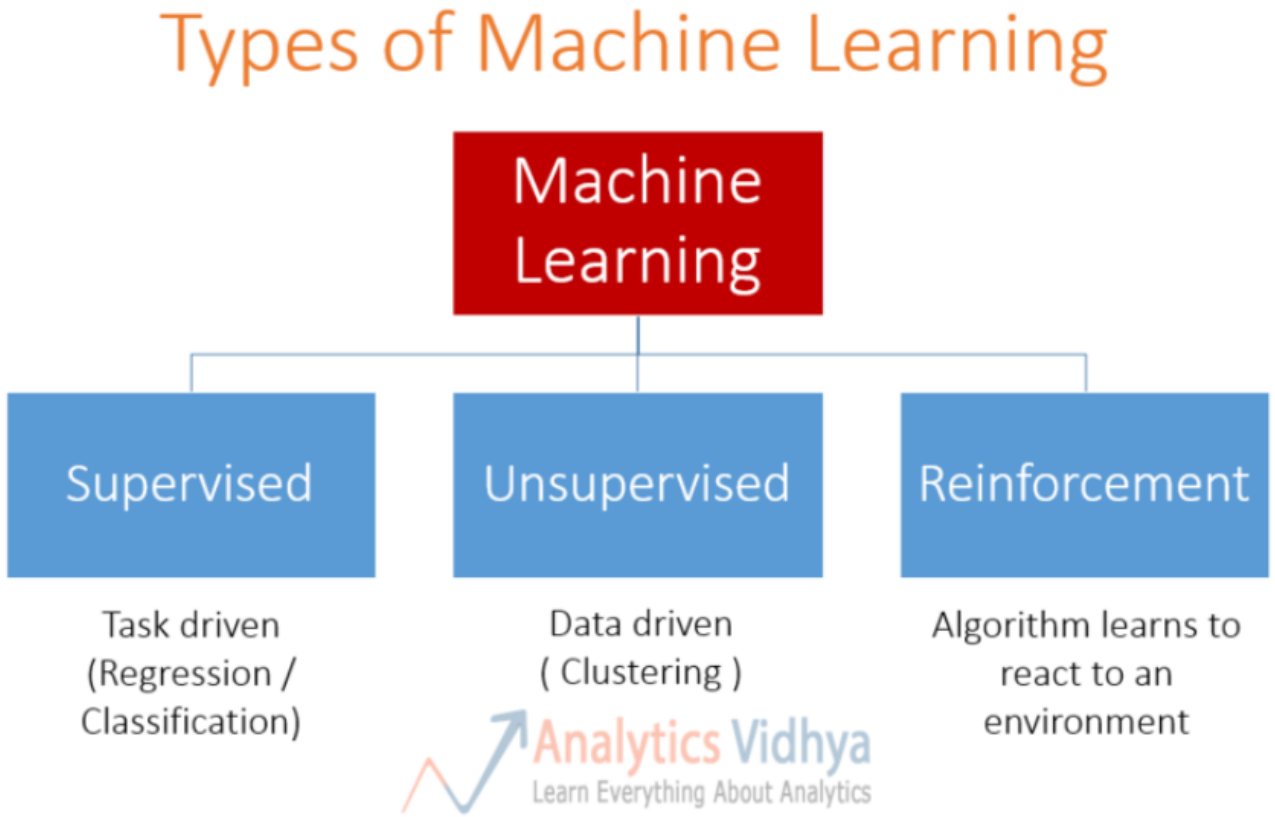
궁금했던 꽃 사진 등록

꽃의 이름과 꽃말, 개화시기 등을 알려준다.

Machine Learning

- **머신러닝의 학습 방법**

  - 지도 학습
  - 비지도 학습
  - 강화 학습



지도 학습

- 정답이 있는 데이터를 활용해 데이터를 학습 시키는 것

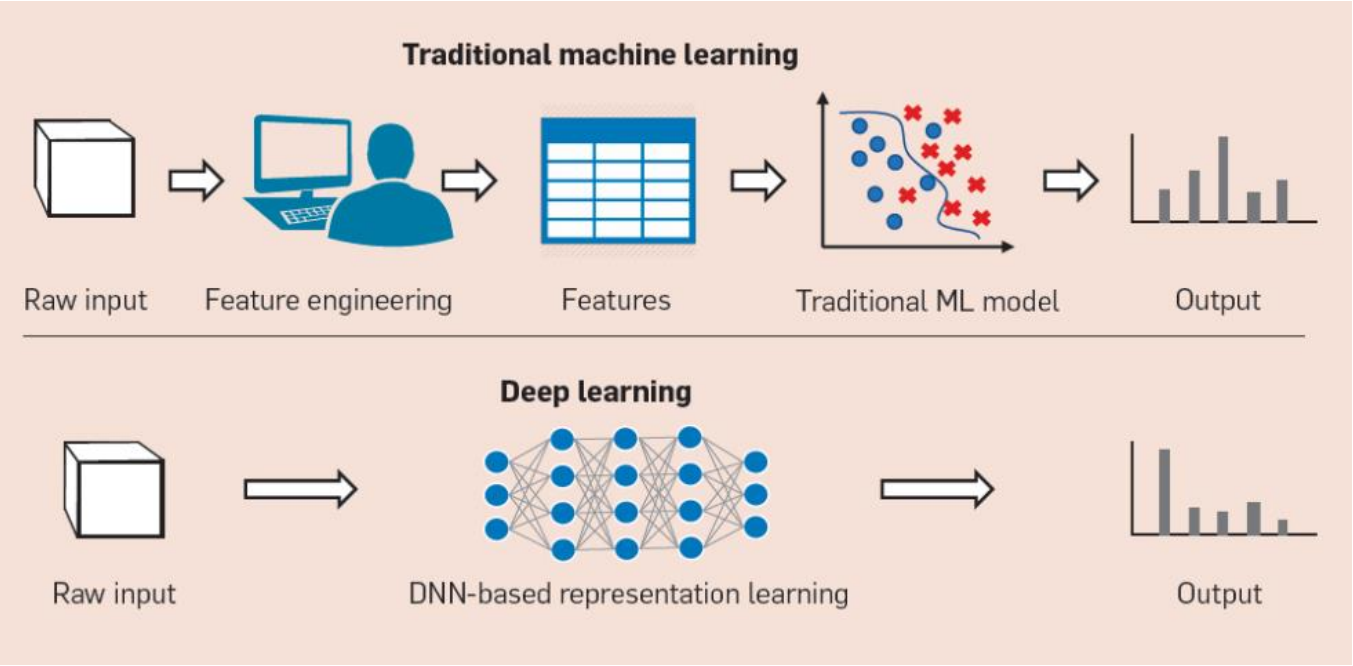
- 입력 값(X data)와 입력 값에 대한 label(Y data)를 주어 학습 시키는 것

1. 분류
주어진 데이터를 정해진 카테고리에 따라 분류하는 문제
이진 분류 문제 , 다중 분류 문제
Ex. 스팸메일 분류

2. 회귀
어떤 데이터들의 feature을 기준으로 연속된 값을 예측하는 문제
분류 처럼 답이 1, 0 이렇게 딱 떨어지는 것이 아니고 어떤 수나 실수로 예측됨.(그래프 형태)
Ex. 보스턴 주택 가격 예측

Feature

- 데이터의 값을 잘 예측하기 위한 데이터의 특징
- 지도, 비지도, 강화학습 모두 적절한 feature을 잘 정의하는 것이 핵심
- Label, class, target, response, dependent variable 으로도 불림



비지도 학습

- 정답 라벨이 없는 데이터를 비슷한 특징끼리 군집화 하여 새로운 데이터에 대한 결과를 예측하는 방법
- 라벨링 되어있지 않은 데이터로부터 패턴이나 형태를 찾아야 하기 때문에 지도학습보다 더 난이도 있다.
- 지도 학습에서 적절한 feature을 찾기 위한 전처리 방법으로 비지도 학습을 이용하기도 한다.
- 대표적으로 클러스터링

Ex. 여러 과일의 사진이 있을 때 이 사진이 어떤 과일인지 정답이 없는 데이터에 대해
색깔이 무엇인지, 모양이 어떠한지 등에 대한 feature를 토대로 바나나다, 사과다 등으로 군집화 하는 것

- 지도/비지도 학습 모델을 섞어서 사용할 수도 있다.

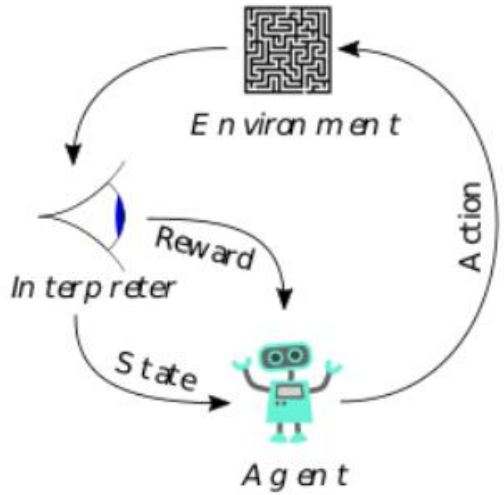
강화 학습

- 행동 심리학에서 나온 이론으로
  분류할 수 있는 데이터가 존재 하는 것도 아니고, 데이터가 있어도 정답이 따로 정해져 있지 않아
  자신이 한 행동에 대한 보상을 받으며 학습하는 것

- 환경이 있고 에이전트가 그 환경 속에서 어떤 액션을 취하고
  그 액션에 따라 어떤 보상을 얻게 되면서 학습이 진행 됨

- 보상을 최대화 하도록 하면서 학습이 진행됨

- 알파고 학습 방법



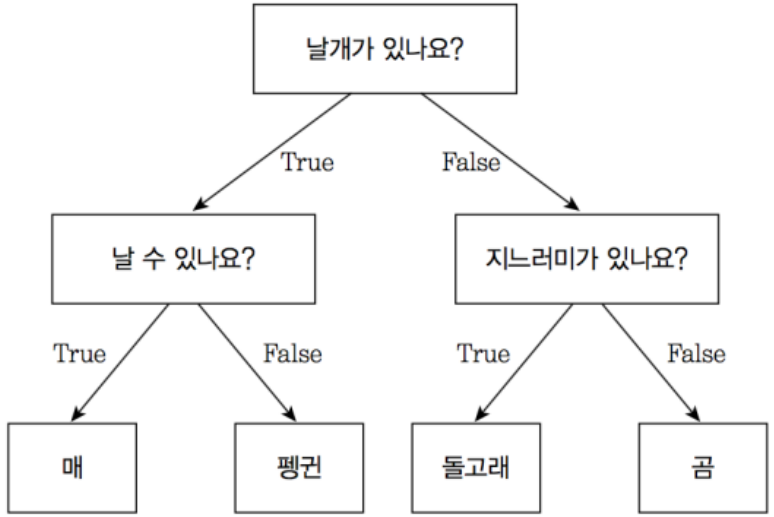
Machine Learning

- 결정 트리 (Decision Tree)
- 앙상블 분류기 (Ensemble Classifier)
- K-근접이웃 알고리즘 (K - Nearest Neighbor Algorithms, K-NN Algorithm)
- 군집화 알고리즘
- 나이브 베이즈 분류기
- 서포트 벡터 머신
- 신경망

결정 트리

▪ 결정 트리 (Decision Tree)

- 전체적인 모양이 나무를 뒤집어 놓은 것과 같아서 이름이 Decision Tree
- 분류와 회귀 모두 가능한 지도 학습 모델
- Root node, Intermediate node, terminal node
- 데이터를 가장 잘 구분할 수 있는 질문으로 나눠 줌.
- 이를 지나치게 많이 하면 오버피팅이 됨.

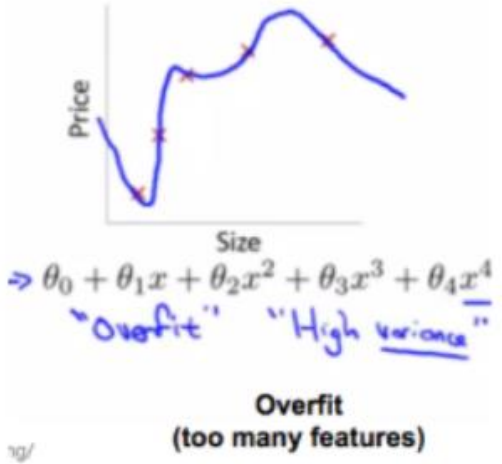


결정 트리

▪ 오버피팅

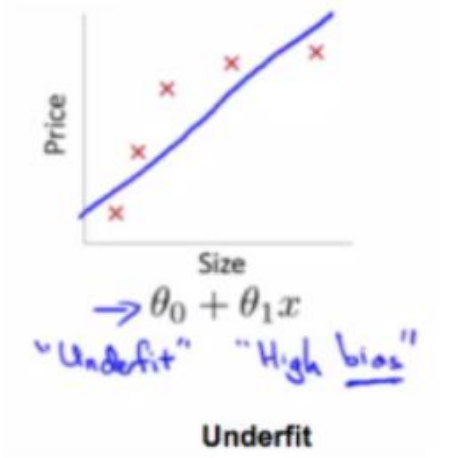
- 샘플 데이터에 너무 정확하게 학습 된 경우
- 샘플 데이터를 가지고 판단 하면 100%에 가까운 정확도를 보이지만 다른 데이터를 넣으면 정확도가 급격하게 떨어진다.

해결책 : feature 수를 줄이거나, 충분히 많은 학습 데이터를 이용



▪ 언더피팅

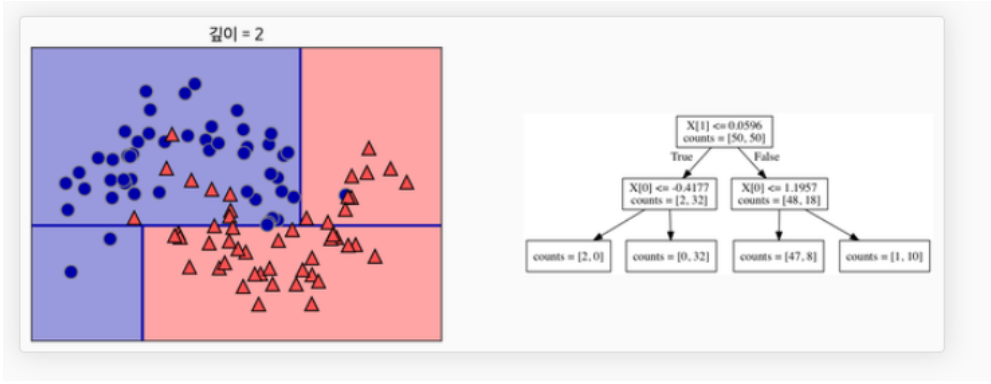
- 학습 데이터가 모자라거나 학습이 제대로 되지 않아서 그래프가 학습 데이터와 많이 떨어진 경우

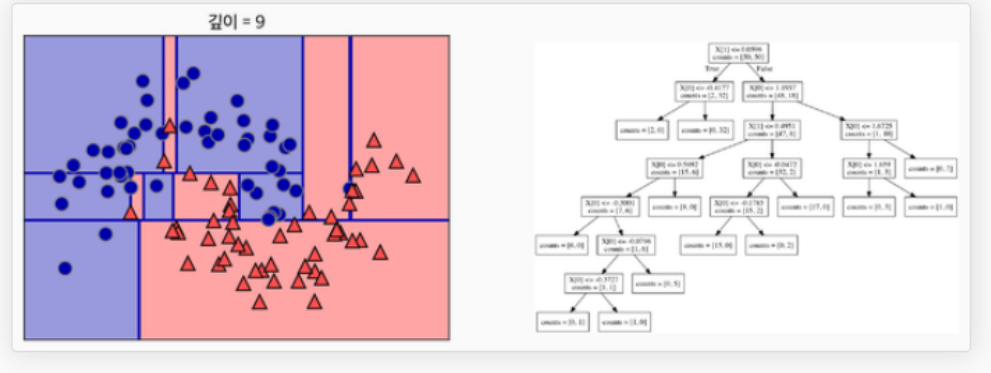


결정 트리

▪ 가지치기

- 오버 피팅을 막기 위한 전략
- Min_sample_split 파라미터를 조정하여 한 노드에 들어있는 최소 데이터 수 정함.
- Max_depth 파라미터를 통해서 최대 깊이를 정함.





결정 트리

- 불순도 : 해당 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지를 뜻함.

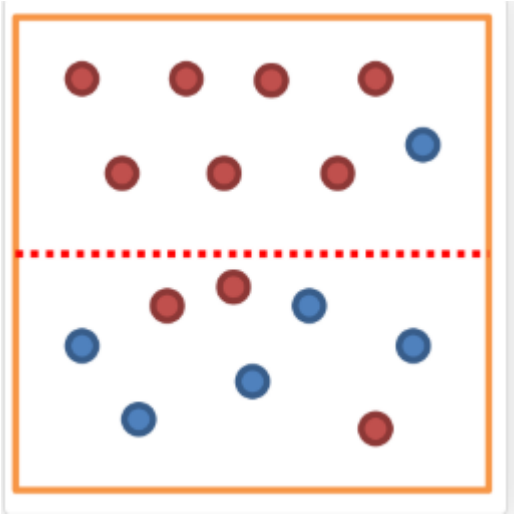
- 엔트로피 : 불순도를 수치적으로 나타낸 척도
  엔트로피 ↑ -> 불순도 ↑

  엔트로피가 1이면 불순도 최대
  엔트로피가 0이면 불순도 최소 -> 한 범주 안에 하나의 데이터만 있는 것

- 정보획득 : 엔트로피가 1인 상태에서 0.7인 상태로 바뀌었다면 정보획득은 0.3 이다.

  Information gain = entropy(parent) - [weighted average]entropy(children)

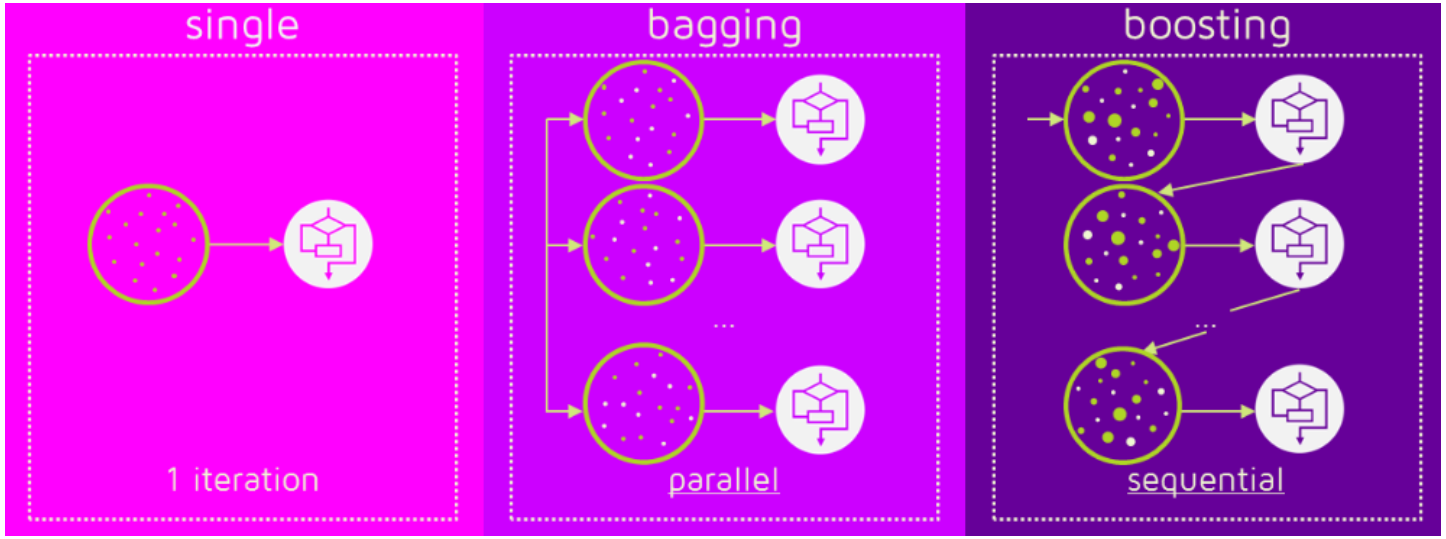
- 결정 트리 알고리즘은 정보 획득을 최대화하는 방향으로 학습이 진행됩니다.



$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$

앙상블 분류기

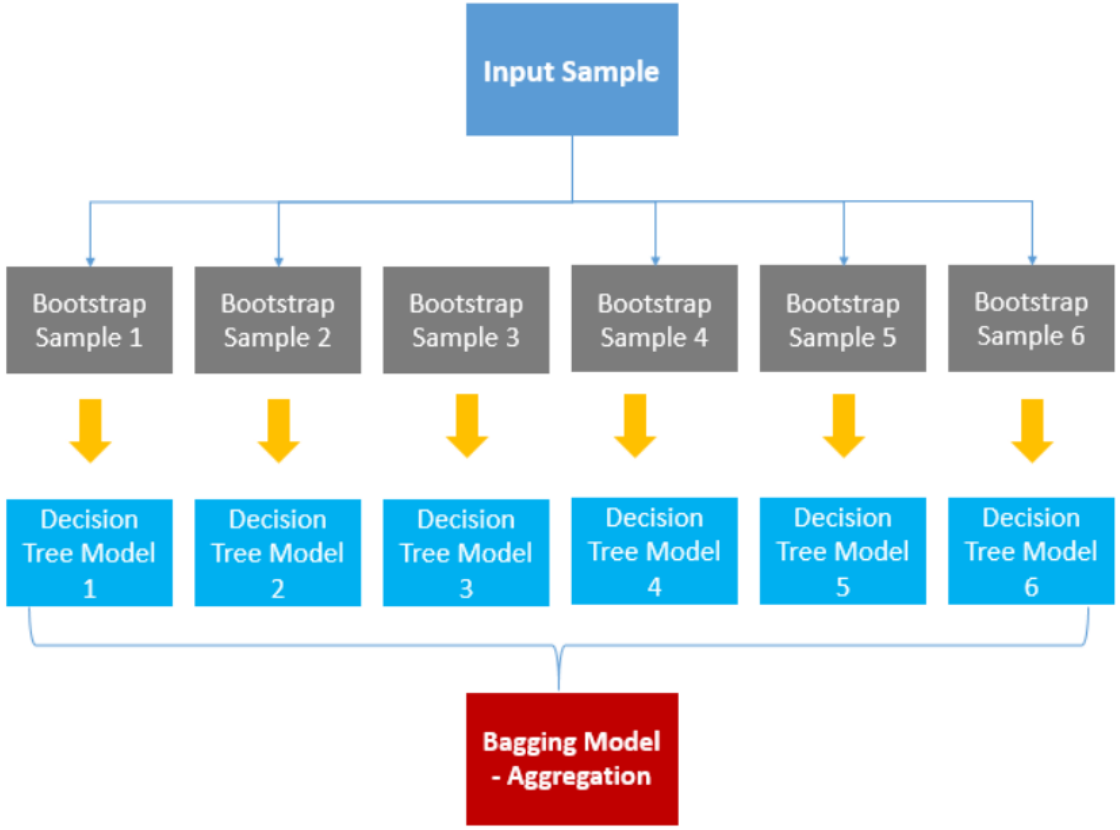
- 여러 개의 결정 트리를 결합하여 하나의 결정 트리보다 더 좋은 성능을 내는 머신러닝의 기법
- 여러 개의 약 분류기를 결합하여 강 분류기를 만드는 것
- 배깅(Bagging) 부스팅(Boosting)



앙상블 분류기

- 배깅(Bagging)

- 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과물을 집계하는 방법
- 전체 모델에서 예측한 값 중 가장 많은 값을 최종 예측 값으로 선정
- 병렬로 학습
- Decision Tree가 서로 독립적으로 결과 예측
- 배깅 기법을 활용한 모델이 랜덤 포레스트



앙상블 분류기

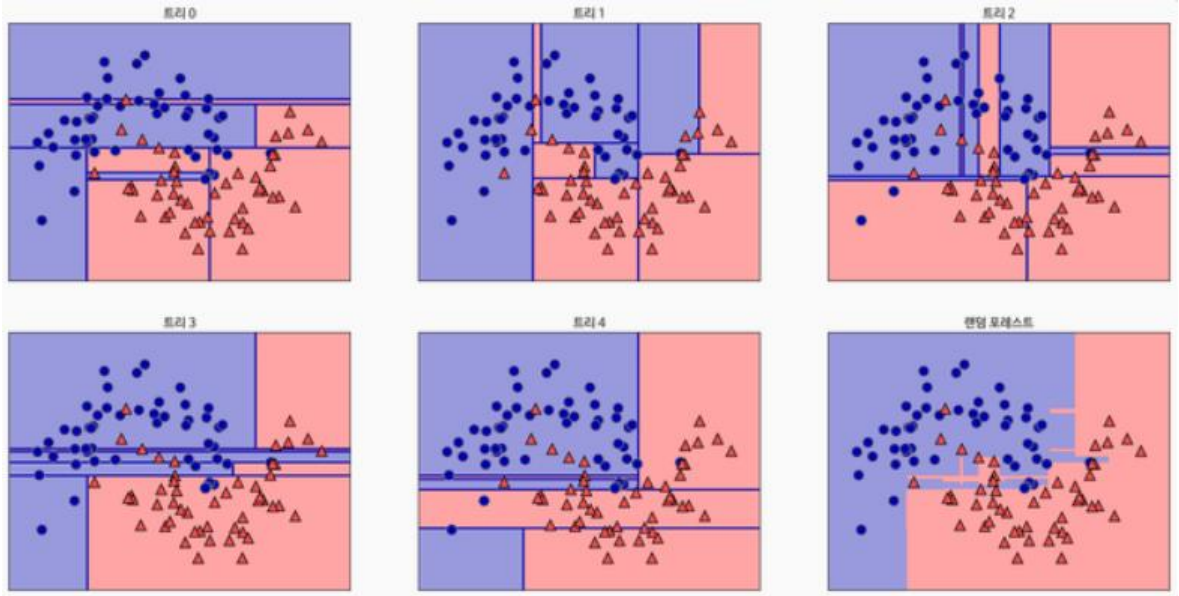
- 랜덤 포레스트

- 결정 트리가 모여 랜덤 포레스트(숲)를 구성.

- 결정 트리의 단점은 훈련 데이터에 오버 피팅이 되는 경향이 있다.
  -> 여러 개의 결정 트리를 통해 랜덤 포레스트를 만들면 오버 피팅 되는 단점을 해결할 수 있다.

- 트리를 만들 때 사용 될 feature들을 제한하여
  각 나무들에게 다양성을 줘야 한다.

- Feature의 수는 보통 전체 feature 수의 제곱근만큼 선택



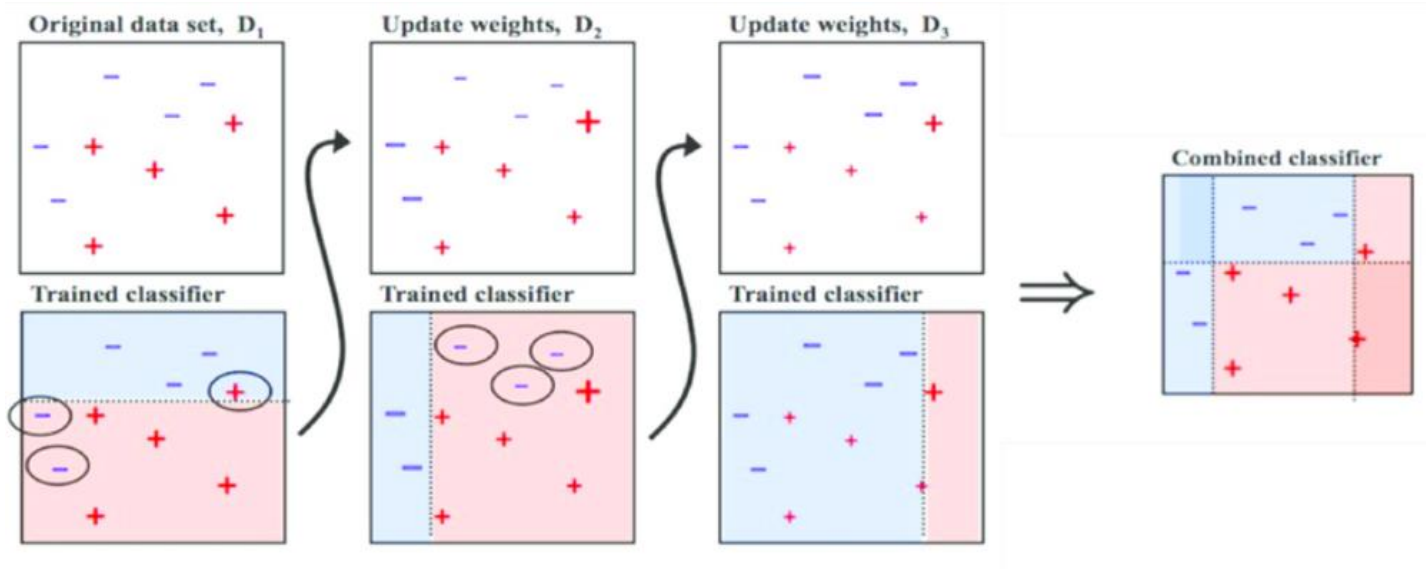
파라미터

- n_estimators : 랜덤 포레스트 안의 결정 트리 개수
  클수록 좋다. 더 깔끔한 Decision Boundary
  but. 메모리와 훈련시간 증가

앙상블 분류기

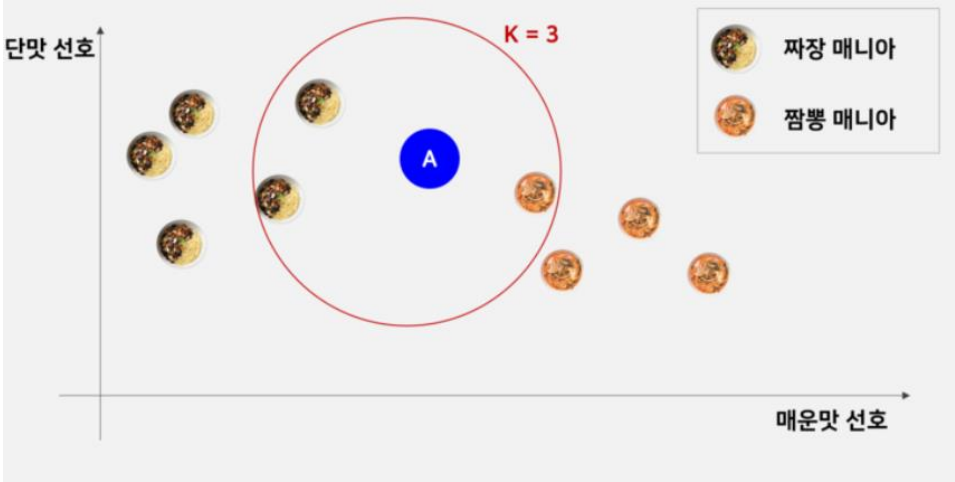
- 부스팅(Boosting)

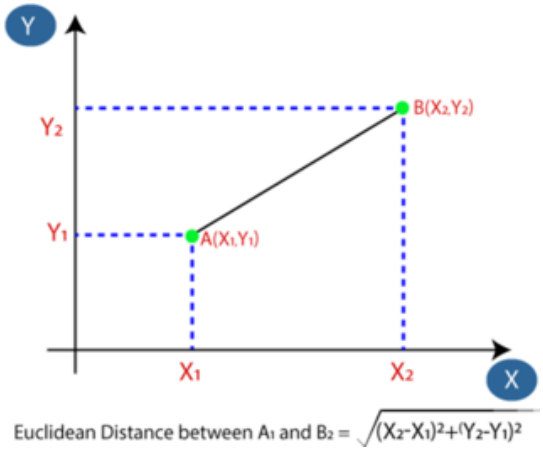
- 가중치를 활용하여 약 분류기를 강 분류기로 만드는 방법
- 예측 결과에 따라 데이터에 가중치가 부여되고, 부여된 가중치가 다음 모델에 영향을 줌.
- 오답에 대해서는 높은 가중치를 부여, 정답에 대해서는 낮은 가중치를 부여.
  → 오답을 정답으로 맞추기 위해 오답에 더 집중할 수 있게 됨.
- 배깅에 비해 error가 적다.
- 속도가 느리고 오버 피팅 될 가능성이 있다.



K-NN 알고리즘

- K-NN 알고리즘은 데이터로부터 거리가 가까운 'k'개의 다른 데이터의 레이블을 참조하여 분류하는 알고리즘
- 거리를 측정할 때 '유클리드 거리' 계산법을 사용
- 인접한 K개의 데이터를 찾아, 해당 데이터의 라벨이 다수인 범주로 데이터를 분류
- 최선의 K값을 찾는 것이 중요하지만, 일반적으로 총데이터의 제곱근 값 사용





K-NN 알고리즘

- 장점

1. 알고리즘이 간단하여 구현하기가 쉽다.
2. 훈련을 통해 모델을 생성하지 않고, 훈련 데이터를 그대로 가지고 있어서 훈련 단계가 매우 빠르다.

- 단점

1. 모델을 생성하지 않아 특징과 클래스간 관계를 이해하는데 제한적이다.
2. 적절한 k의 선택이 필요하다.
3. 데이터가 많아지면 분류 단계가 느리다.

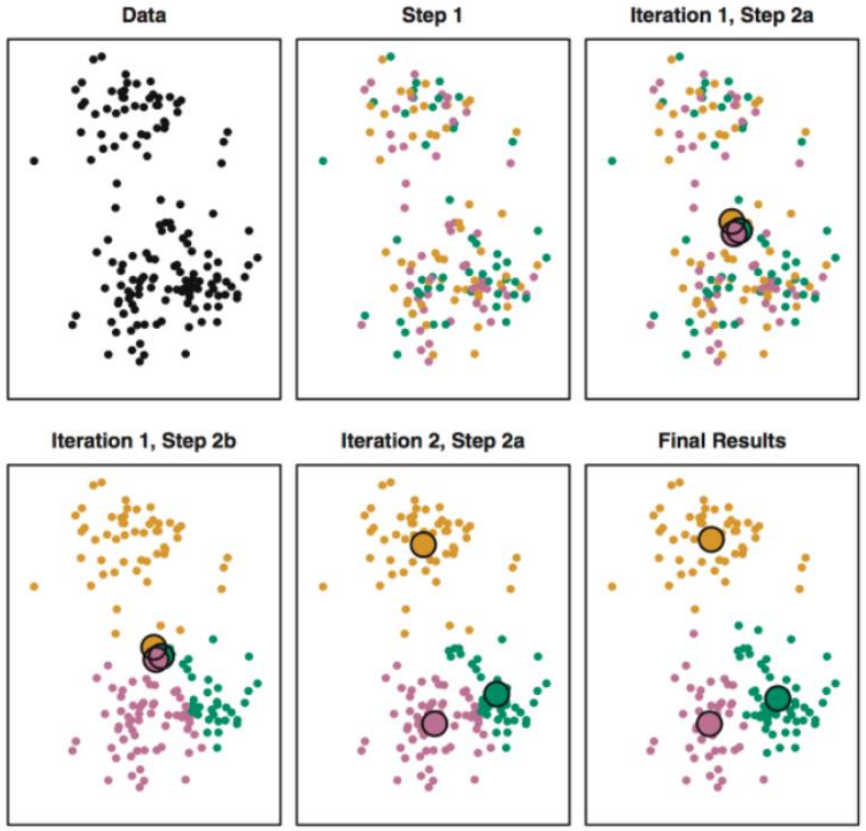
군집화 알고리즘

- 클러스터링(Clustering;군집화) : 비슷한 개체끼리 한 그룹으로, 다른 개체는 다른 그룹으로 묶는 것
- 정답이 없는 비지도학습
- But, 분류는 정답이 있는 지도학습
- 대표적으로 K-means clustering

군집화 알고리즘

- K-means clustering

- 원하는 클러스터의 개수 k를 고르고, label이 없는 데이터를 입력 받아 각 데이터에 label을 할당함으로써 군집화 수행
- 데이터가 속한 cluster의 중심과 데이터 간의 거리가 최소가 되도록 데이터들을 K개의 cluster에 할당한다.
- 모든 데이터 포인트는 k개 클러스터 중에 하나에 속한다.
- 어떤 데이터 포인트도 두 개 이상의 클러스터에 속할 수 없다.
- 클러스터 사이의 순서는 없고, 랜덤성이 있다.
- 매우 간단하고 기본적인 clustering 알고리즘 -> 실행 속도가 빠르다.



나이브 베이즈 분류기

- 베이즈 정리

- 기하학에 피타고라스 정리가 있다면 확률론에는 베이즈 정리가 있다.
- 무작위로 뽑은 표본에서 얻어지는 정보로 모수의 성능을 평가하는 것 보다
  표본 정보와 사전 정보를 함께 사용하여 모수의 성능을 평가하는 것이 더 바람직하다.

$$p(\theta|X) = \frac{p(\theta, X)}{p(X)} = \frac{p(X|\theta)p(\theta)}{p(X)}$$

나이브 베이즈 분류기

- 베이즈 정리에 기반한 통계적 분류 기법
- 가장 단순한 지도 학습 중 하나
- 빠르고 정확하며 믿을 만한 알고리즘
- 정확도도 높고 대용량 데이터에 대해 속도도 빠르다.
- But. feature끼리 서로 독립이라는 조건 필요
- Ex. 스팸 메일 분류에서 광고성 단어의 개수와 비속어의 개수가 서로 연관이 있으면 안된다.

나이브 베이즈 분류기

- 문제 1
  날씨가 overcast일 때 경기를 할 확률은 ?

P(yes I overcast) = P(overcast I yes) P(yes) / P(overcast) ← 베이즈 정리

1. 사전 확률
P(overcast) = 4/14 = 0.29
P(yes) = 9/14 = 0.64

2. 사후 확률
P(overcast I yes) = 4/9 = 0.44

3. 베이즈 정리 공식에 대입
P(yes I overcast) = P(overcast I yes) P(yes) / P(overcast)
= 0.44 * 0.64 / 0.29 = 0.98

즉. 날씨가 overcast일 때 축구를 할 확률이 0.98

Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

나이브 베이즈 분류기

- 문제 2
  날씨가 overcast일 때 경기를 하지 않을 확률은 ?

P(no I overcast) = P(overcast I no) P(no) / P(overcast) ← 베이즈 정리

1. 사전 확률
P(overcast) = 4/14 = 0.29
P(no) = 5/14 = 0.36

2. 사후 확률
P(overcast I no) = 0/5 = 0

3. 베이즈 정리 공식에 대입
P(no I overcast) = P(overcast I no) P(no) / P(overcast)
= 0 * 0.36 / 0.29 = 0

즉, 날씨가 overcast일 때 축구를 하지 않을 확률이 0

Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

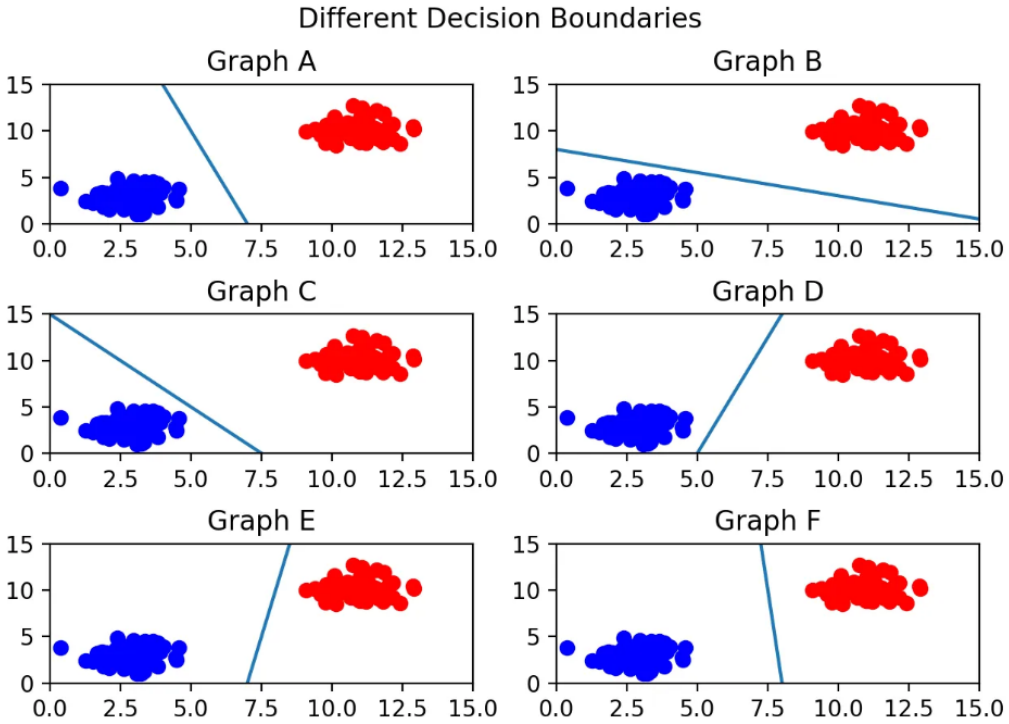
Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

서포트 벡터 머신

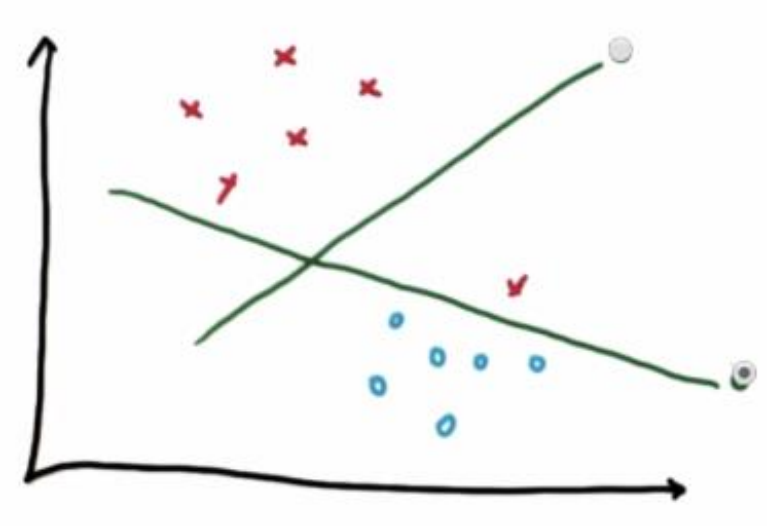
- SVM은 분류에 사용되는 지도학습 머신러닝 모델
- SVM은 서포트 벡터를 사용해서 결정 경계(Decision Boundary)를 정의
- 서포트 벡터는 결정 경계(Decision Boundary)에 가장 가까운 각 클래스의 점들이다.
- 서포트 벡터와 결정 경계(Decision Boundary)의 거리를 마진(margin) 이라고 한다.
- SVM은 허용 가능한 오류 범위 내에서 가능한 최대 마진을 만들려고 한다.

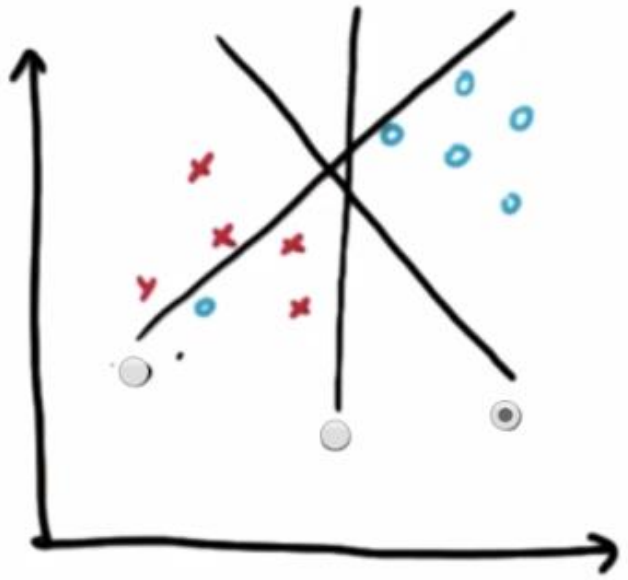


서포트 벡터 머신

- 무작정 마진을 크게 하는 결정 경계를 택하는 것이 아니라.
  데이터를 정확하게 분류하는 범위를 먼저 찾고 그 범위 안에서 마진을 최대화하는 결정 경계 선택

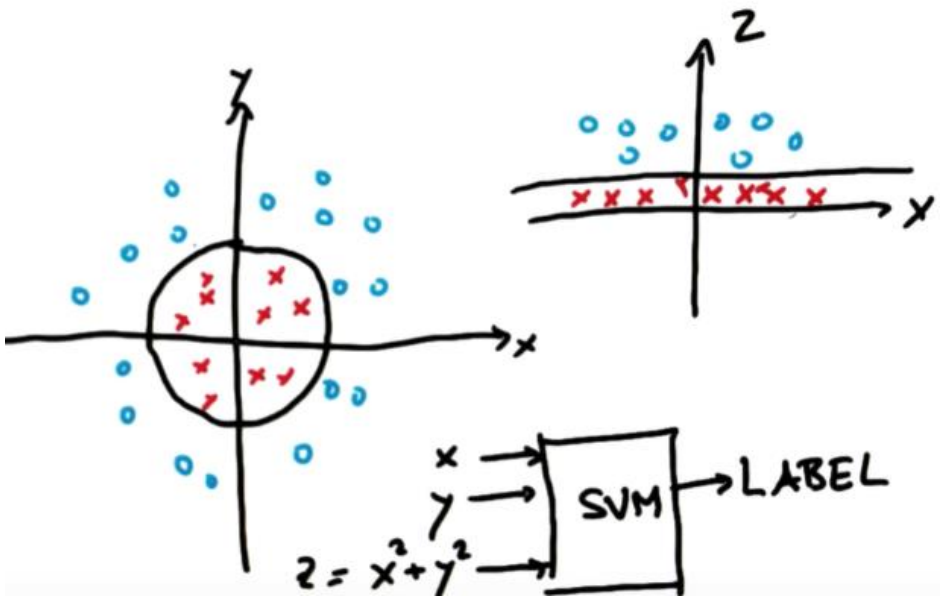
- 두 데이터를 정확하게 구분하는 직선이 없기 때문에
  어느 정도 outlier를 무시하고 마진을 최대화하는 결정 경계 선택





서포트 벡터 머신

- 오른쪽 그래프에서 linear하게 그린 결정 경계는 왼쪽에서 원형으로 된 결정 경계와 동일.
- SVM에서는 선형으로 분리할 수 없는 점들을 분류하기 위해 커널을 사용한다.
- 저 차원 공간을 고 차원 공간으로 매핑해주는 작업을 커널 트릭 이라고 한다.



감사합니다

2020.08.03