YBIGTA 신입기수(20기) 교육세션 – 2022.01.25

Machine Learning (ML)

Science Team 19기 김상회

table of contents

01

ML 개요

1) 머신러닝 정의 AI/ML/DL
2) 머신러닝 유형 지도학습/비지도학습/강화학습
3) Validation Set
4) 지도학습 과정

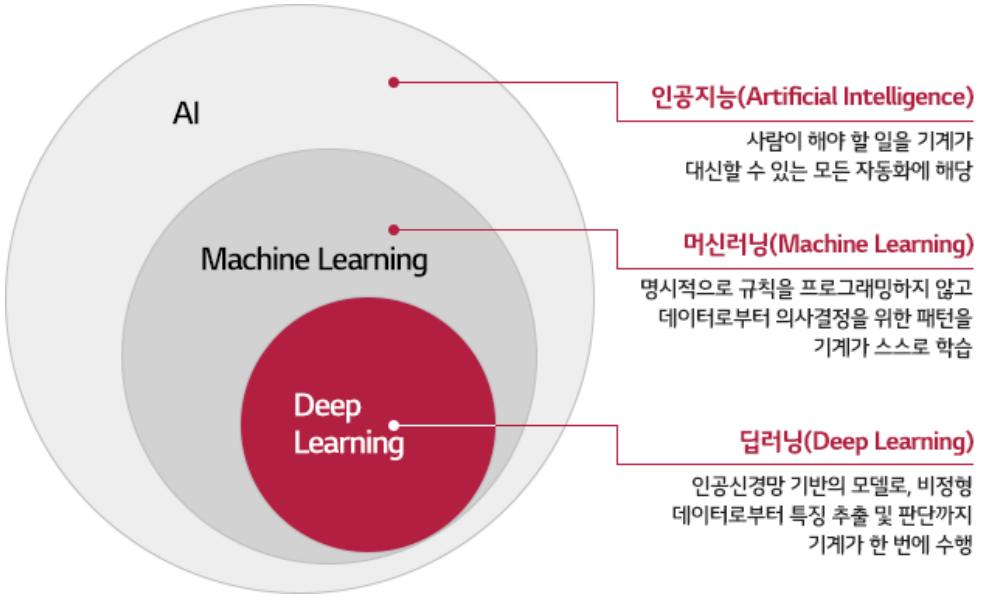
Machine Learning 개요

머신러닝 (Machine Learning)

: 인공지능의 연구 분야 중 하나로, 인간의 학습 능력과 같은 기능을 컴퓨터에서 실현하고자 하는 기술 및 기법 (출처: 두산백과)

- 머신러닝은 인공지능의 한 종류이고, 딥러닝은 머신러닝 중 인공신경망(Artificial Neural Network)의 한 종류
  (인공지능 ⊃ 머신러닝 ⊃ 인공신경망 ⊃ 딥러닝)

- 머신러닝은 **경험(E)**을 통해 특정한 **작업(T)**에 대한 **성능(P)**을 스스로 향상



[그림] AI, 머신러닝, 딥러닝 간의 관계를 나타내는 다이어그램 (출처: LG CNS)

일반 프로그램

다음 수식을 계산한 결과는?

4 x 2 = ?
4 x 3 = ?
5 x 8 = ?
7 x 6 = ?

머신러닝 프로그램

□와 △에 들어갈 정수는?

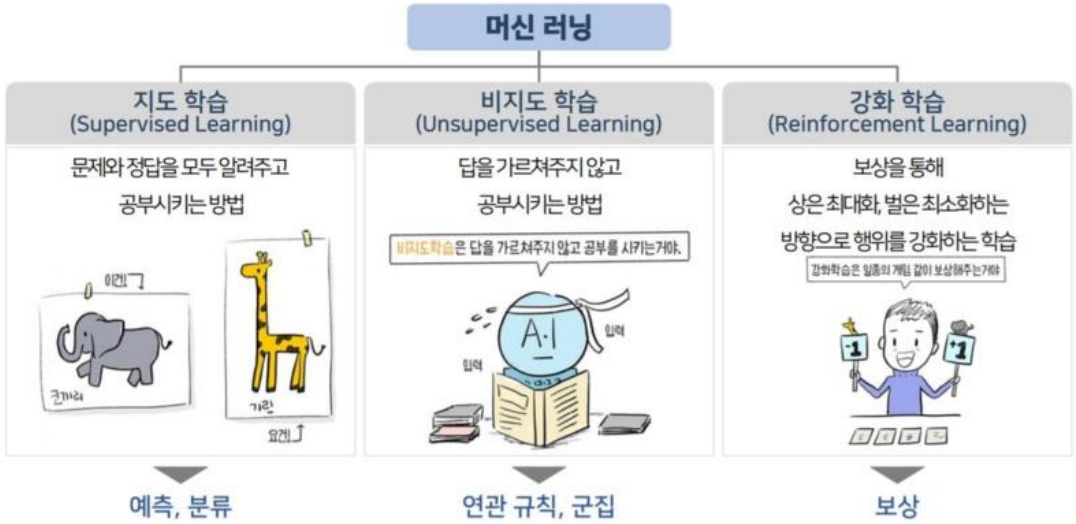
- 3 x □ + 2 x △ = 1
- 1 x □ + 4 x △ = -3
- 5 x □ + 5 x △ = 0
- 8 x □ + 3 x △ = 5

[**작업 T**] □와 △ 구하기
[**성능 P**] 수식이 정확할 확률
[**경험 E**] 입력값 (3, 2) (1, 4) (5, 5) (8, 3)를 입력,
출력값 1, -3, 0, 5를 도출하도록 학습
* **학습** : 경험 E를 통해 가중치(□=1, △=-1)를 찾는 것

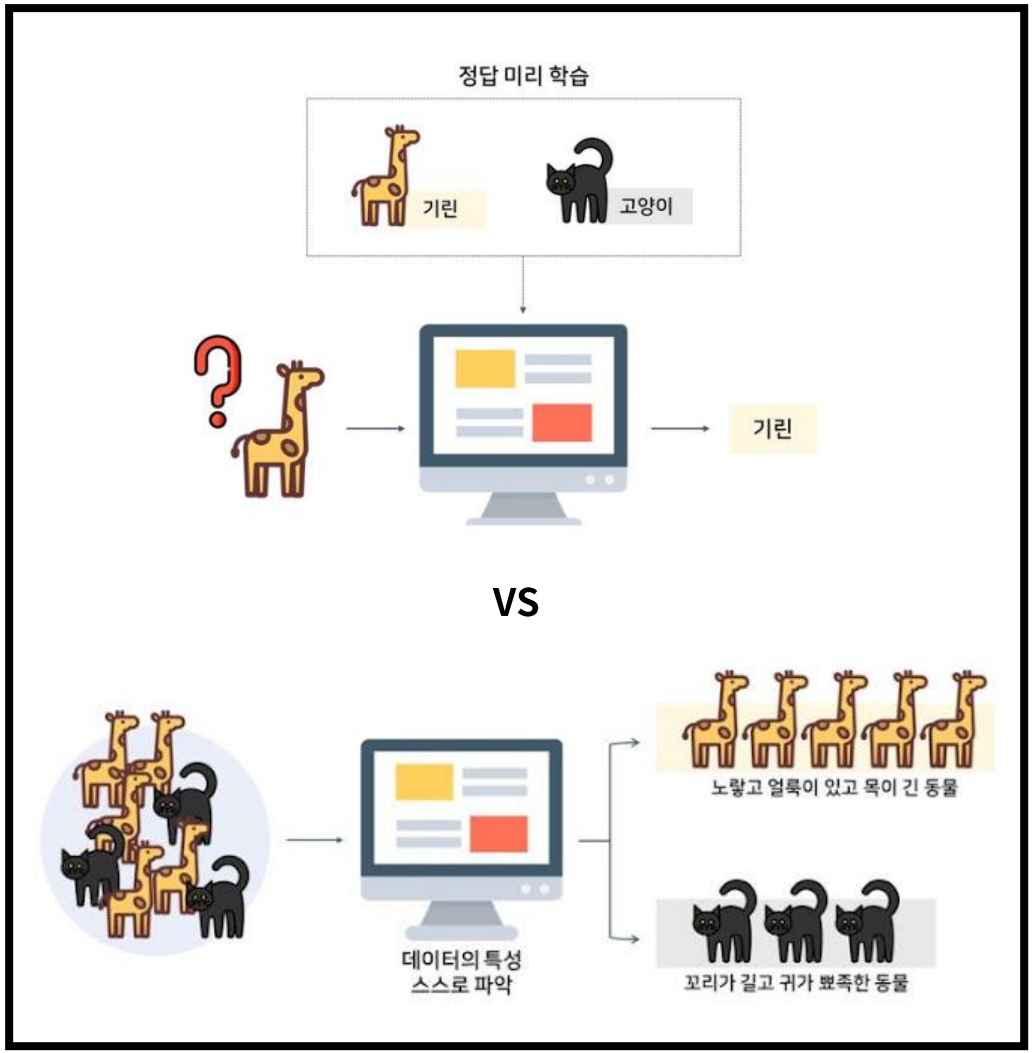
Machine Learning 개요

머신러닝 유형

- 지도 학습: **정답이 정해진 문제**에 대해 학습 ex) 분류, 예측
- 비지도 학습: **정답이 정해지지 않은 문제**에 대해 학습 ex) 군집화
- 강화 학습: 보상을 통해 스스로 문제 해결 방법을 학습 ex) 알파고 바둑



[그림] 머신러닝 구분



Machine Learning 개요

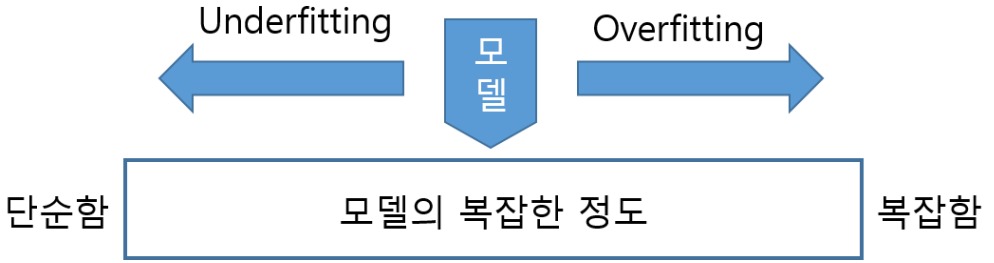
전통적 통계분석 방법

- 정해진 분포나 가정을 통해 실패 확률을 줄이는 것이 목적
- 모형의 복잡성보다 단순성을 추구하며 **신뢰도**를 중요하게 생각
- 파라미터의 **해석 가능성** 또한 중요하게 다룸
  ex) 키가 1cm 증가하였을 때 몸무게의 변화량 (선형회귀)

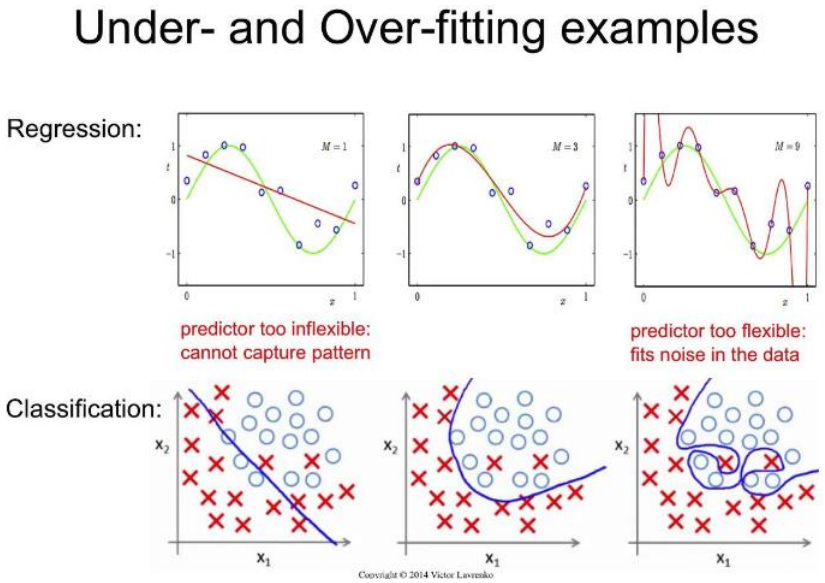
머신러닝

- 머신러닝은 **예측 성능을 높이는 것이 목적**
- 모델의 신뢰도나 정교한 가정(assumption)은 상대적으로 중요도가 낮아지며, 오버피팅(overfitting)은 어느 정도 감안하더라도 여러 인자를 사용해 예측 수행
  → 사용 가능한 인자를 모두 넣고 좋은 결과 뽑으면 장땡 but 오버피팅 번번히 발생
- 해당 인자가 왜 중요한지는 크게 중요하지 않음

→ Validation Set을 사용하여 Overfitting 방지

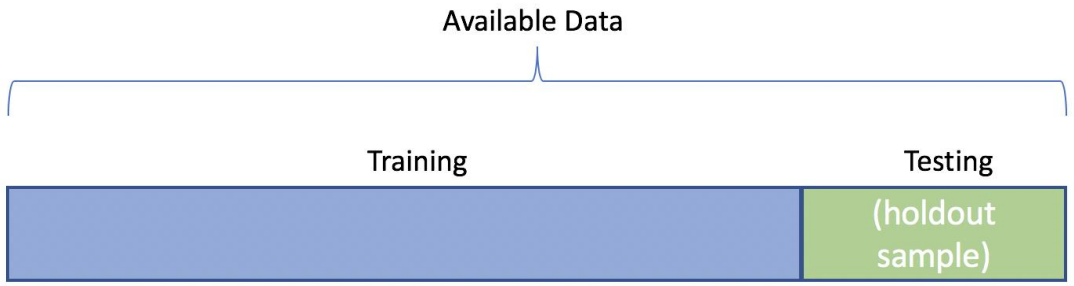


[그림] 모델의 복잡도와 피팅 정도 관계



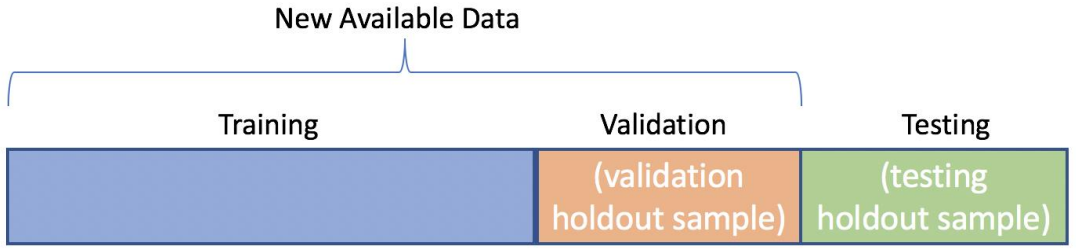
[그림] 오버피팅과 언더피팅

Machine Learning 개요



[그림] **Train / Test Split**

1) 전체 Dataset 을 Train Set / Test Set 으로 분할
2) Train Set으로 모델 학습
3) Test Set의 **실제 값**과 모델이 Test Set의 feature들로부터 **예측한 값**을 비교하여 모델의 성능 평가



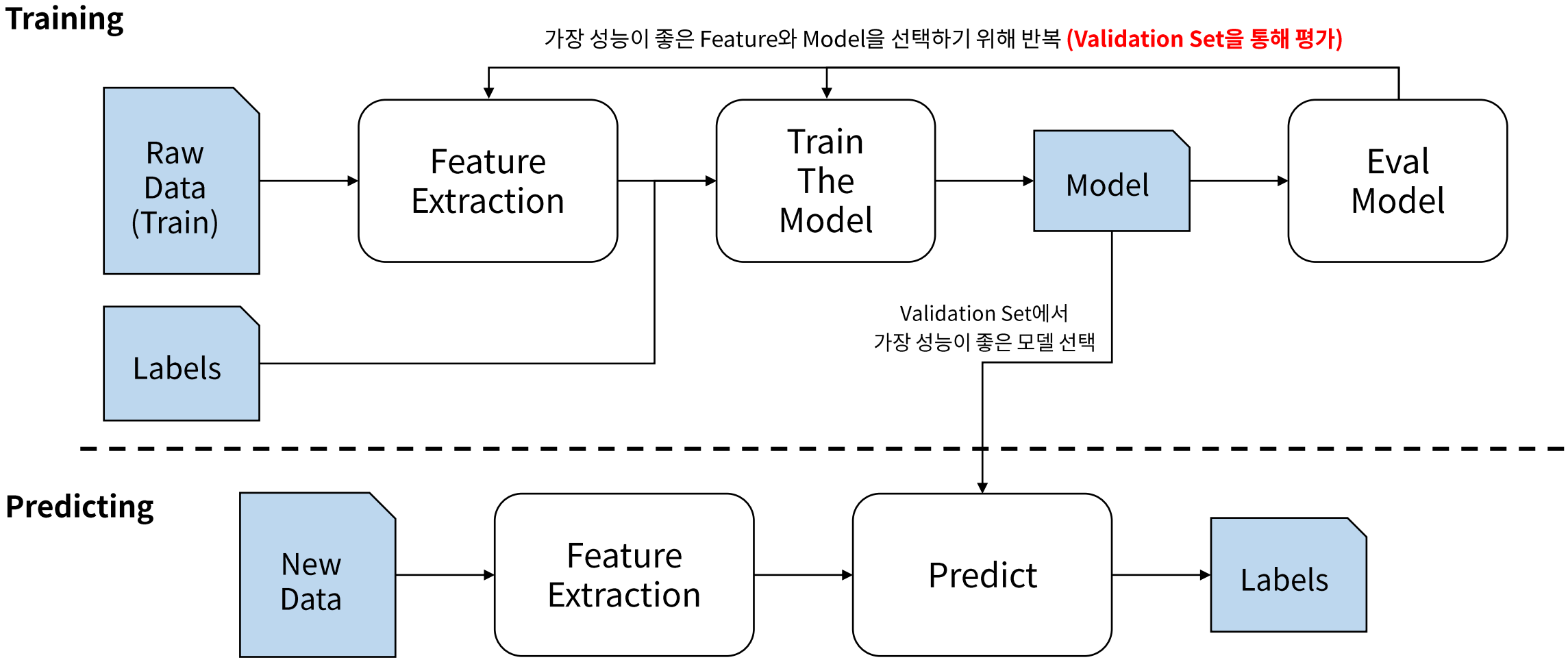
[그림] **Train / Valid / Test Split**

머신러닝 모델은 **예측 성능**을 높이기 위해 **복잡성**을 높일 경우 Train Set에 과적합되기 때문에 Test Set에는 낮은 성능이 나올 수 있음 → **Overfitting**

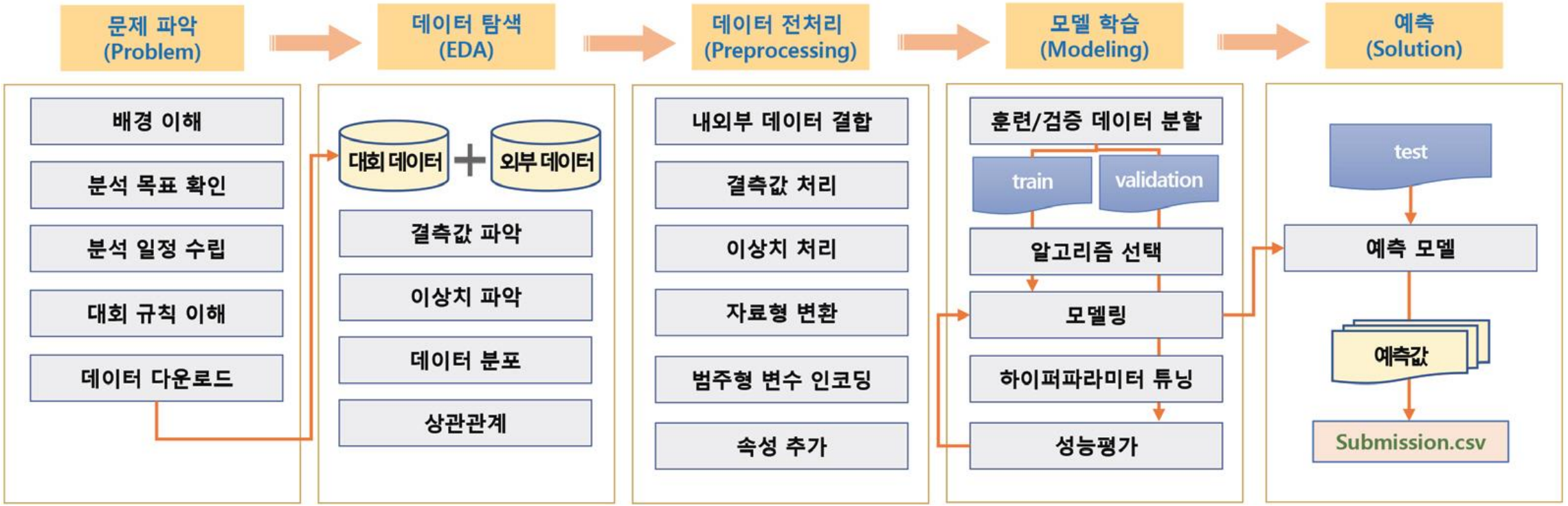
Overfitting을 막기위해 Train에서 일부분을 Validation으로 사용
ex) **문제집(Train)**으로 학습 / **모의고사(Valid)**로 학습결과 확인 / **수능(Test)**

지도학습 과정

Machine Learning 개요



Machine Learning 개요 (2)



[그림] 머신러닝 프로세스(경진 대회)

02

분류

1) 데이터 타입
2) 분류 성능 평가

분류(Classification)

데이터 타입

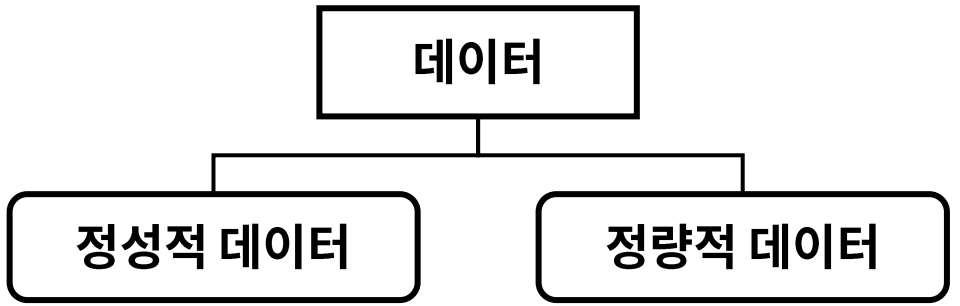
: Quantitative(정량적) vs Qualitative(정성적)

Quantitative(정량적): 숫자로 표시되는 연속적인 값
→가격(1,120원, 1,000원, 1,402원…), 성적(66점, 78점, 90점…)

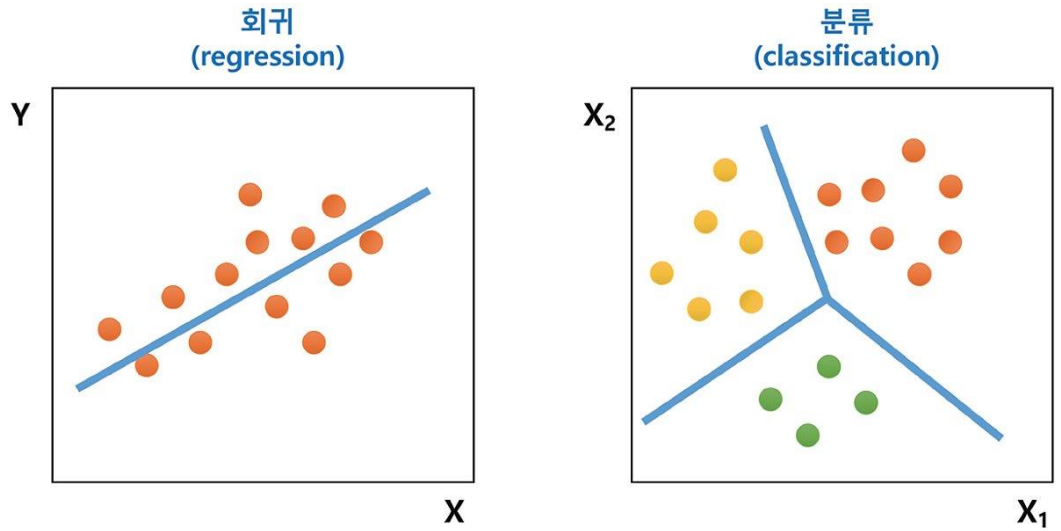
Qualitative(정성적): 카테고리(또는 Class)로 표시되는 값
→지역(서대문구, 송파구, 종로구…), 학과(정보산업공학과, 컴퓨터과학과…)

숫자로 되어 있다고 모두 정량적 변수가 아님. 정성적 데이터를 숫자로 표기할 수 있기 때문에 데이터 타입이 어떤 지 확인 필요
ex) 지역 번호 → 02 (서울), 032 (인천), 064 (제주도)…

구글링 할 때는 **continuous**와 **categorical** 사용



[그림] 데이터 종류



[그림] 회귀 vs 분류

분류 성능 평가

분류(Classification)

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

[그림] Confusion Matrix

True Positive(TP) : 실제 True → 예측 True (정답)
True Negative(TN) : 실제 False → 예측 False (정답)
False Positive(FP) : 실제 False → 예측 True (오답)
False Negative(FN) : 실제 True → 예측 False (오답)

$$(Accuracy) = \frac{TP + TN}{TP + FN + FP + TN}$$

분류 모델의 성능 지표를 볼 때 정확도(Accuracy)만 확인하면 안됨
→ **Accuracy Paradox**

예측값 / 실제값	Cancer (1)	Not Cancer (0)
Cancer (1)	0	0
Not Cancer (0)	5	95

암 환자의 비율은 매우 낮기 때문에 모델이 모두 암 환자가 아니라고 예측 할 경우 정확도는 95%로 매우 높게 나오지만 좋은 모델이 아님
→ 비대칭(imbalanced) 데이터의 경우 다른 지표들도 살펴봐야 함

$$(Precision) = \frac{TP}{TP + FP}$$

정밀도 : 모델이 True로 분류한 것 중 실제 True 비율

$$(Recall) = \frac{TP}{TP + FN}$$

재현율 : 실제 True인 것 중에서 모델이 True로 예측한 비율

분류 성능 평가

분류(Classification)

결정 임계값(Decision Threshold)에 따라, 정밀도와 재현율은 서로 **Trade-off** 관계

ex) 모든 메일을 스팸메일로 분류하면 실제 스팸메일 중 모두를 잡지만**(높은 재현율)**,
스팸메일로 예측 한 것 중 실제 스팸메일 비율은 낮아짐**(낮은 정밀도)**

$$(Recall) = \frac{TP}{TP + FN} \qquad (Precision) = \frac{TP}{TP + FP}$$

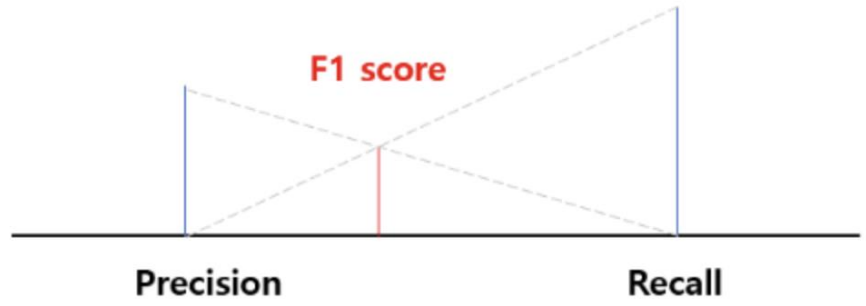
		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

해결하려는 **Task의 목적**과 **Dataset의 특성**에 따라서 **적절한 성능 평가 지표**를 사용해야 하지만

Recall과 Precision 모두를 고려하는 **F1-score**도 좋은 지표 (Recall과 Precision의 조화평균)

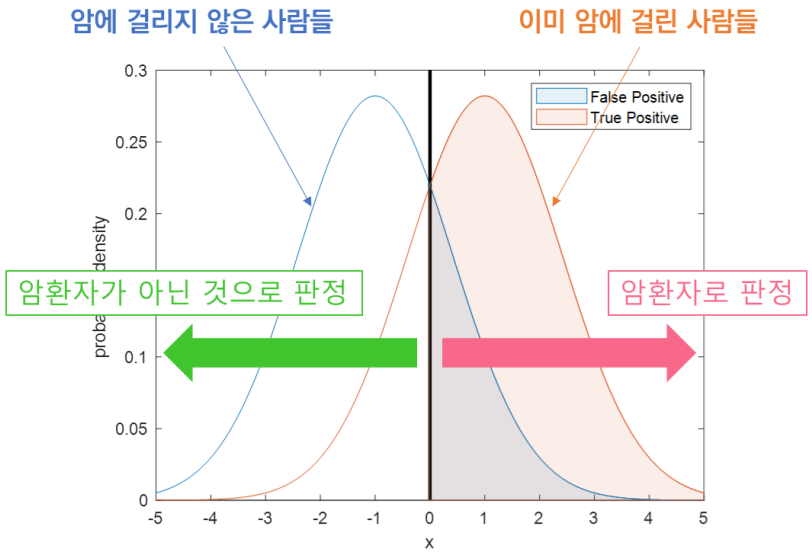
→ 산술평균 대신 조화평균을 쓰는 이유는 산술평균 보다 값이 **작은 값의 비중을 크게 잡기 때문**

$$(F1\text{-}score) = 2\times\frac{1}{\frac{1}{Precision}+\frac{1}{Recall}} = 2\times\frac{Precision \times Recall}{Precision + Recall}$$



[그림] 조화평균의 기하학적 의미

분류(Classification)



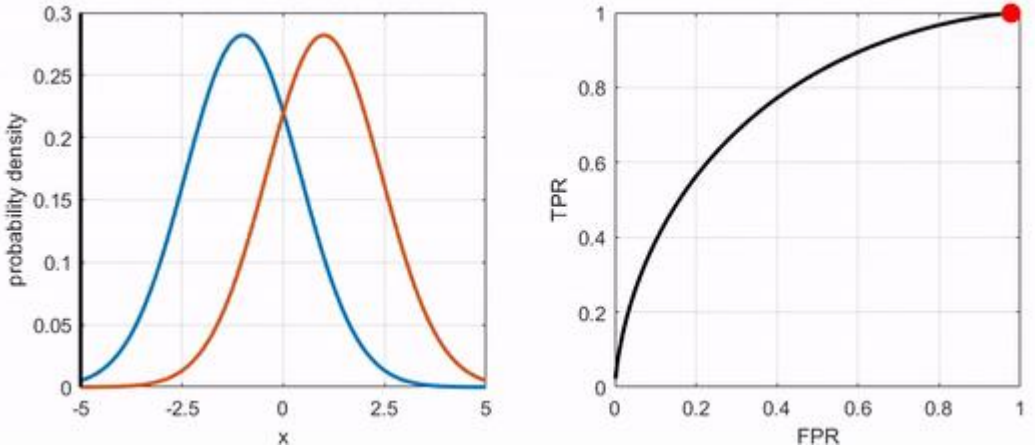
ROC curve

: **FPR**과 **TPR**을 각각 x, y 축으로 놓은 그래프

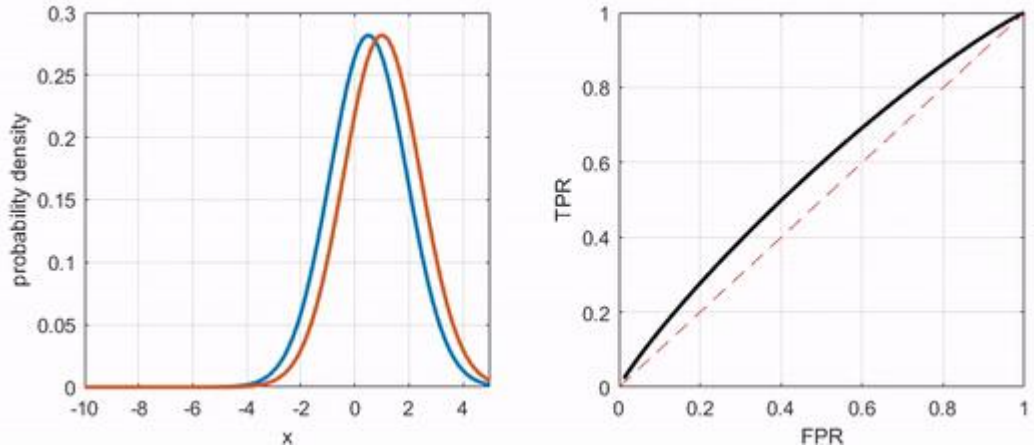
- TPR : True Positive Rate
  1인 케이스에 대해 1로 잘 예측한 비율. (암환자를 진찰해서 암이라고 진단)

- FPR : False Positive Rate
  0인 케이스에 대해 1로 잘못 예측한 비율. (암환자가 아닌데 암이라고 진단)

→ 같은 FPR일 때 높은 TPR일 수록 좋은 예측 모델



[그림] threshold 변화에 따른 ROC 커브 위의 점 위치 변화



[그림] 두 그룹을 더 잘 구별할 수 있을수록 ROC 커브는 좌상단에 붙게 된다.

출처 : 공돌이의 수학정리노트

03

지도학습 모델 종류

1) 선형 모델
2) Support Vector Machine (SVM)
3) Decision Tree(의사결정나무)

지도학습 모델 종류

지도학습 모델 종류

Support Vector Machine (SVM)

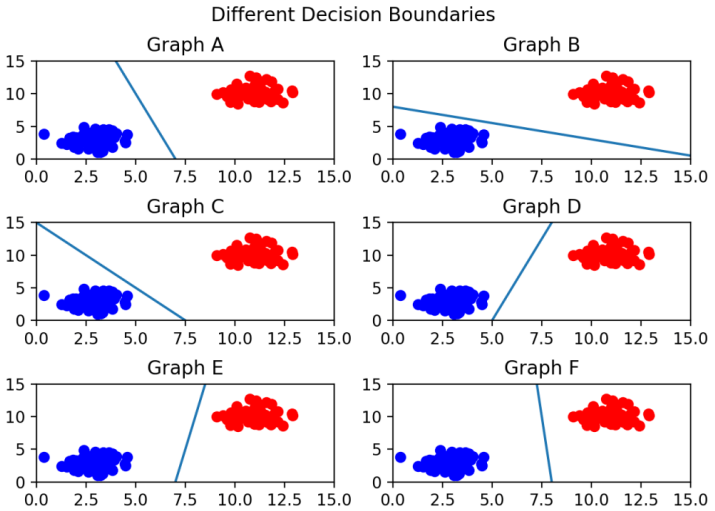
: **서포트 벡터 머신(SVM)**은 결정 경계(Decision Boundary), 즉 분류를 위한 기준 선을 정의하는 모델

→ 즉 데이터 집합을 가장 잘 분류하는 경계를 찾는 모델
(+회귀 문제에도 사용 가능)

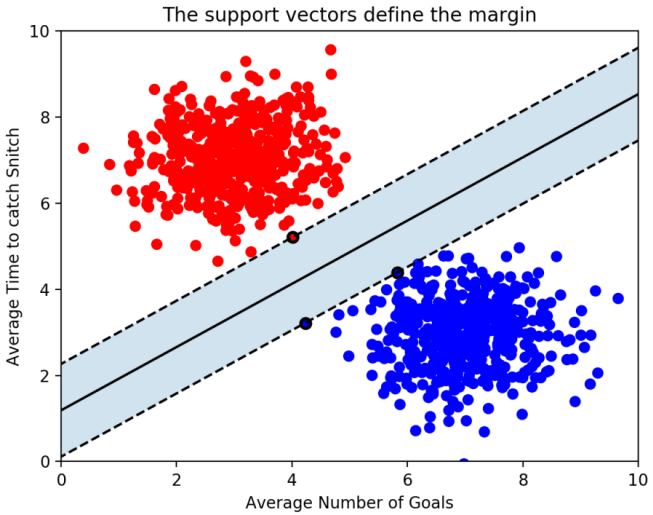
마진(margin)

: 결정 경계와 가장 가까운 훈련 샘플 사이의 거리이며, 이때 가장 가까운 훈련 샘플을 서포트 벡터(support vector)라 정의

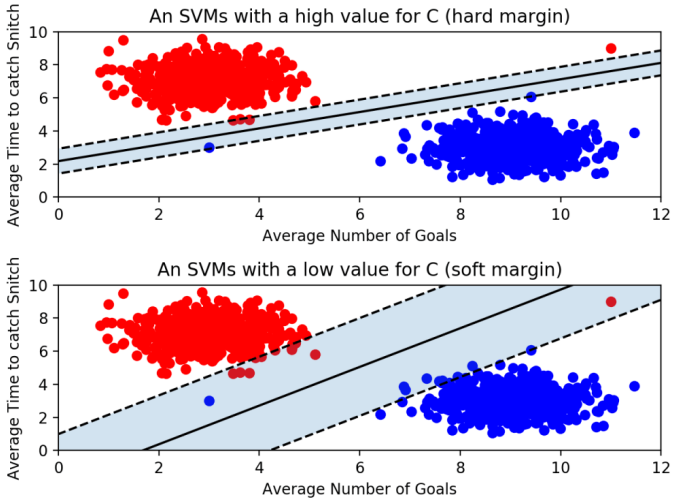
서포트 벡터 머신은 마진을 최대화 하는 결정 경계를 찾는 것



[그림] 모두 데이터를 잘 나눴지만 어떤 선이 좋은 선일까?



[그림] 마진(margin)이 가장 큰 선



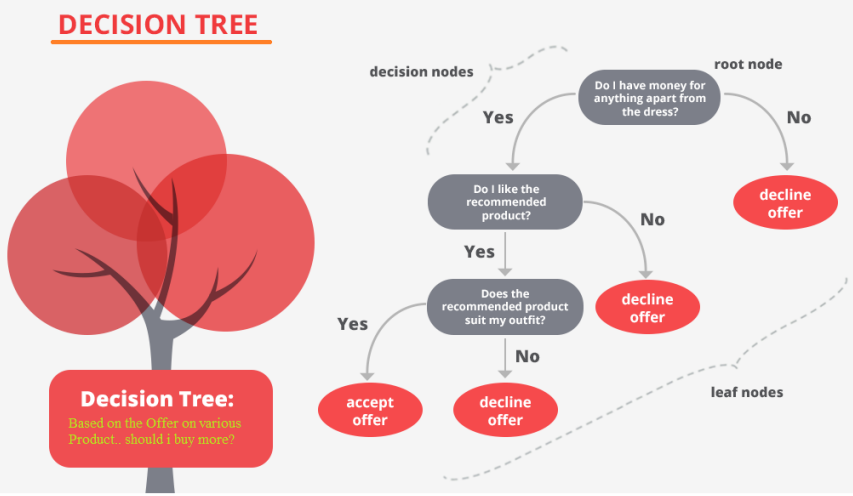
[그림] SVM에서 이상치 처리

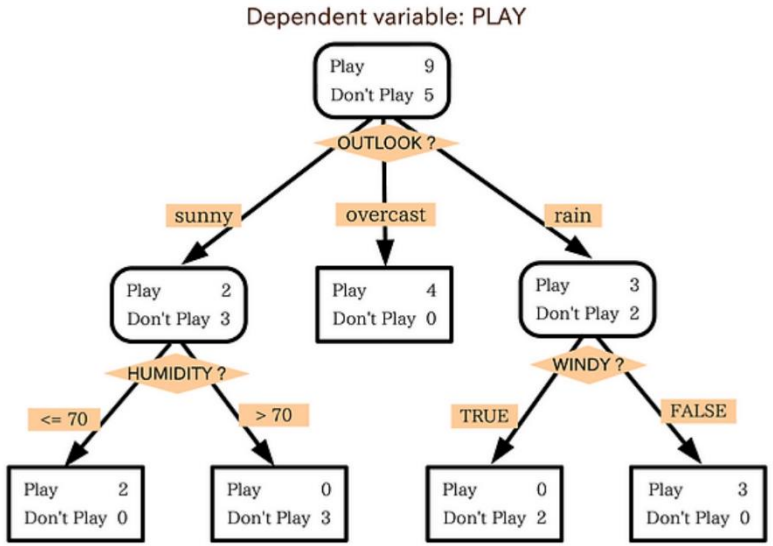
이상치(Outlier)를 얼마나 허용할 것인가?

왼쪽 그림에서 위의 SVM은 이상치를 허용하고 있지 않기 때문에 마진이 매우 작고 **오버피팅** 문제가 발생하기 쉽다. → Hard Margin

반면 아래 그림은 이상치를 어느정도 포함되도록 기준을 잡으니, 마진이 커졌지만 **언더피팅** 문제가 발생할 수 있다. → Soft Margin

지도학습 모델 종류





Decision Tree

- 여러가지 규칙을 순차적으로 적용하면서 독립 변수 공간을 분할하는 분류 모형
- 분류와 회귀 분석 모두 사용 가능한 지도학습 모델
- 특정 기준에 따라 데이터를 분리하며, 기준에 따라 분할된 박스를 노드라고 부름

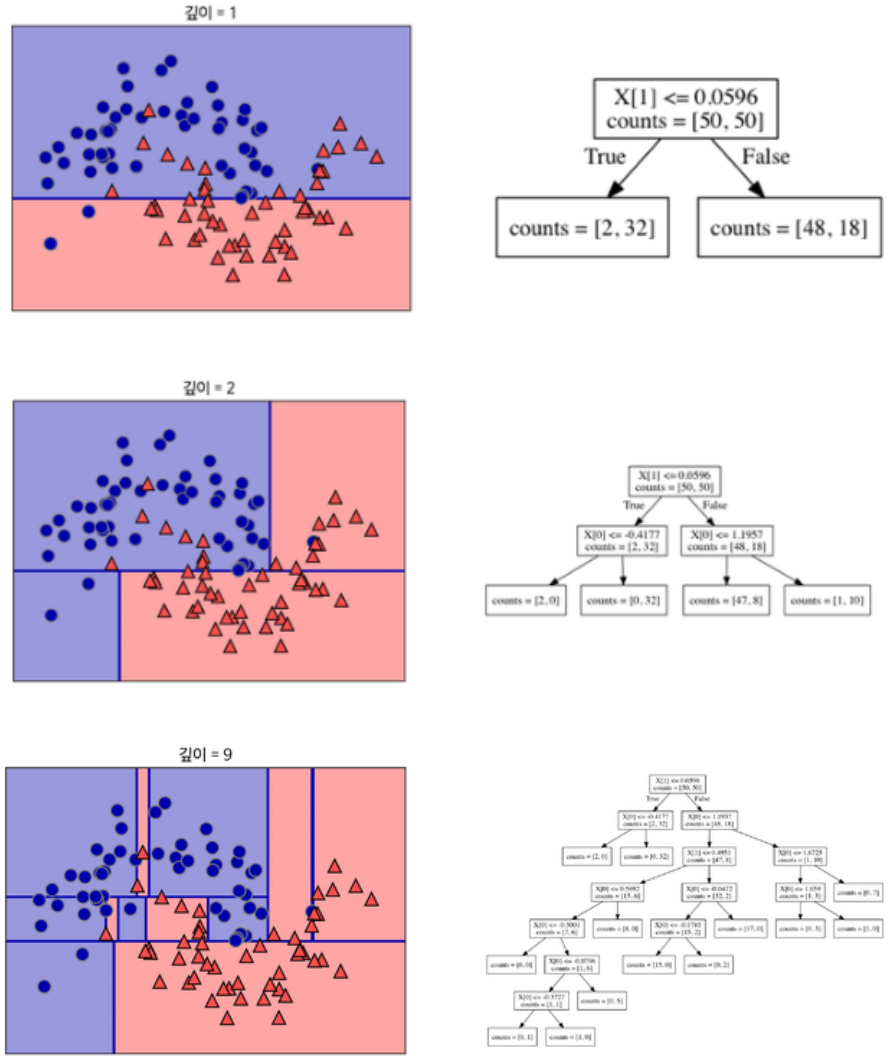
장점

1) 인간의 의사결정 과정과 닮아 인과관계 설명 가능
2) 범주형 자료를 input 변수로 사용 가능
3) 이상치에 강한 모델
4) 통계적을 크게 요구하지 않음

단점

1) 과적합에 취약 → 앙상블 & 가지치기 등으로 방지
2) Training Data가 조금만 바뀌어도 Tree 모양이 크게 바뀜

지도학습 모델 종류

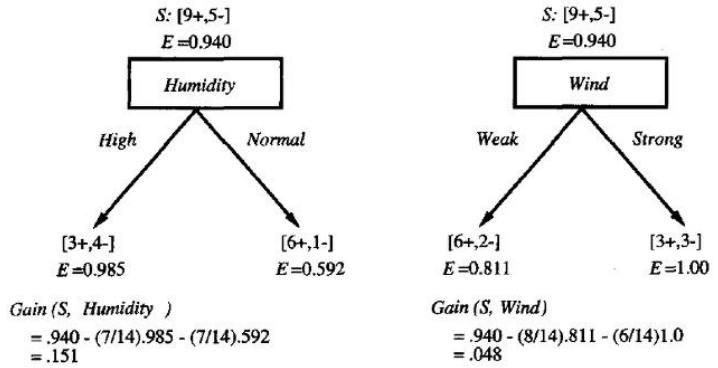


[그림] Process of Decion Tree

1) 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 분기

→ **Information Gain이 높은 방법으로 분기**

Information Gain = Entropy(parent) – [weighted average]Entropy(children)



[그림] 운동한 날을 습도와 바람으로 나눴을 때의 Gain

Entropy = 혼탁한 정도

$$Entropy(A) = -\sum_{k=1}^{m} p_k \log_2 (p_k)$$

2) 나뉜 각 범주에서 다시 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 분기

3) Terminal Node(마지막 분기)가 더 이상 분리되지 않을 때까지 않을 때까지 반복

But 분기를 지나치게 많이 하면 train data에 과적합 발생

→ Max Depth를 설정하여 과적합 방지

지도학습 모델 종류

dmlc XGBoost

LightGBM

여러 개의 Decision Tree를 합친 모델은 성능이 매우x100 좋다.

합치는 방법에 따라 Bagging, Boosting 등이 있으며,

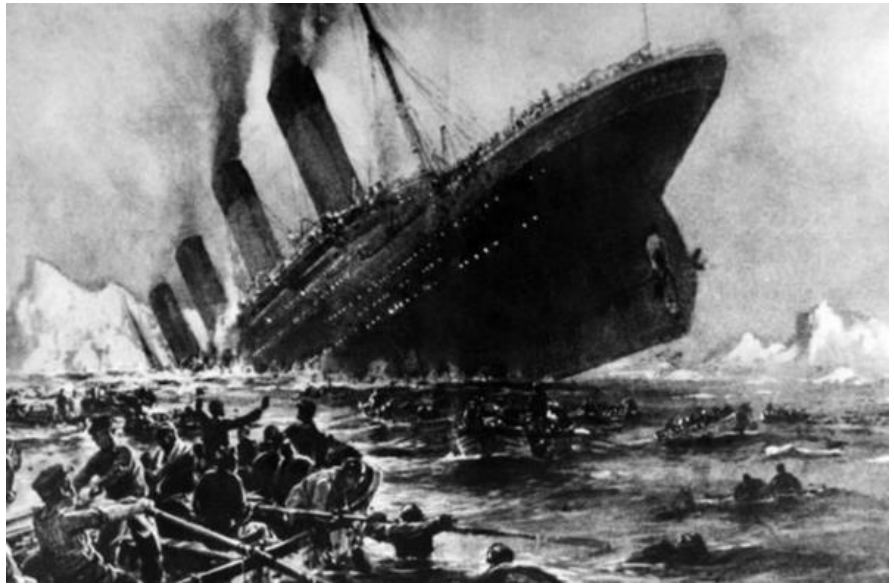
XGBoost, Light GBM이 대표적

+) 대부분의 문제에서 Deep Learning보다 쓰기도 쉽고 성능 좋음 **(*주관적인 견해*)**

04

실습&과제

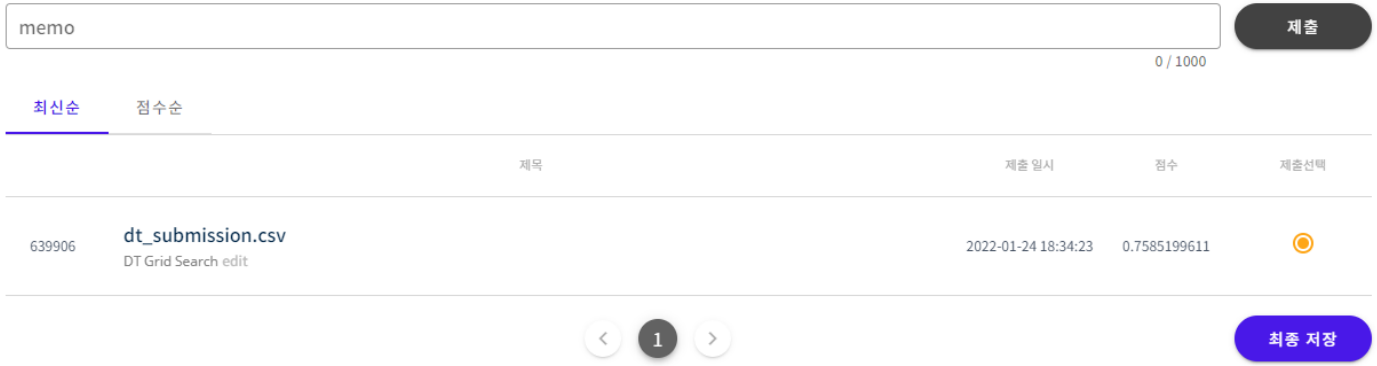

괜찮겠어?
난 멈추는 법을
모르는 담곰이인데.



과제는 대표적인 예측 문제인 타이타닉 생존자 예측하기 입니다.

Dacon에서 제공하는 “기초부터 연습하기”를 통해
데이터 분석 대회 참여 방식도 익혀 봅시다.
(https://dacon.io/competitions/open/235539/overview/description)

제공해드린 소스코드의 빈칸을 모두 채운 코드와
Dacon에 예측값을 제출하고 결과를 캡쳐해서
본인의 Git에 올리는 것 까지가 이번주 과제입니다~



[그림] 제출 캡처 예시

감사합니다