Machine Learning (1)

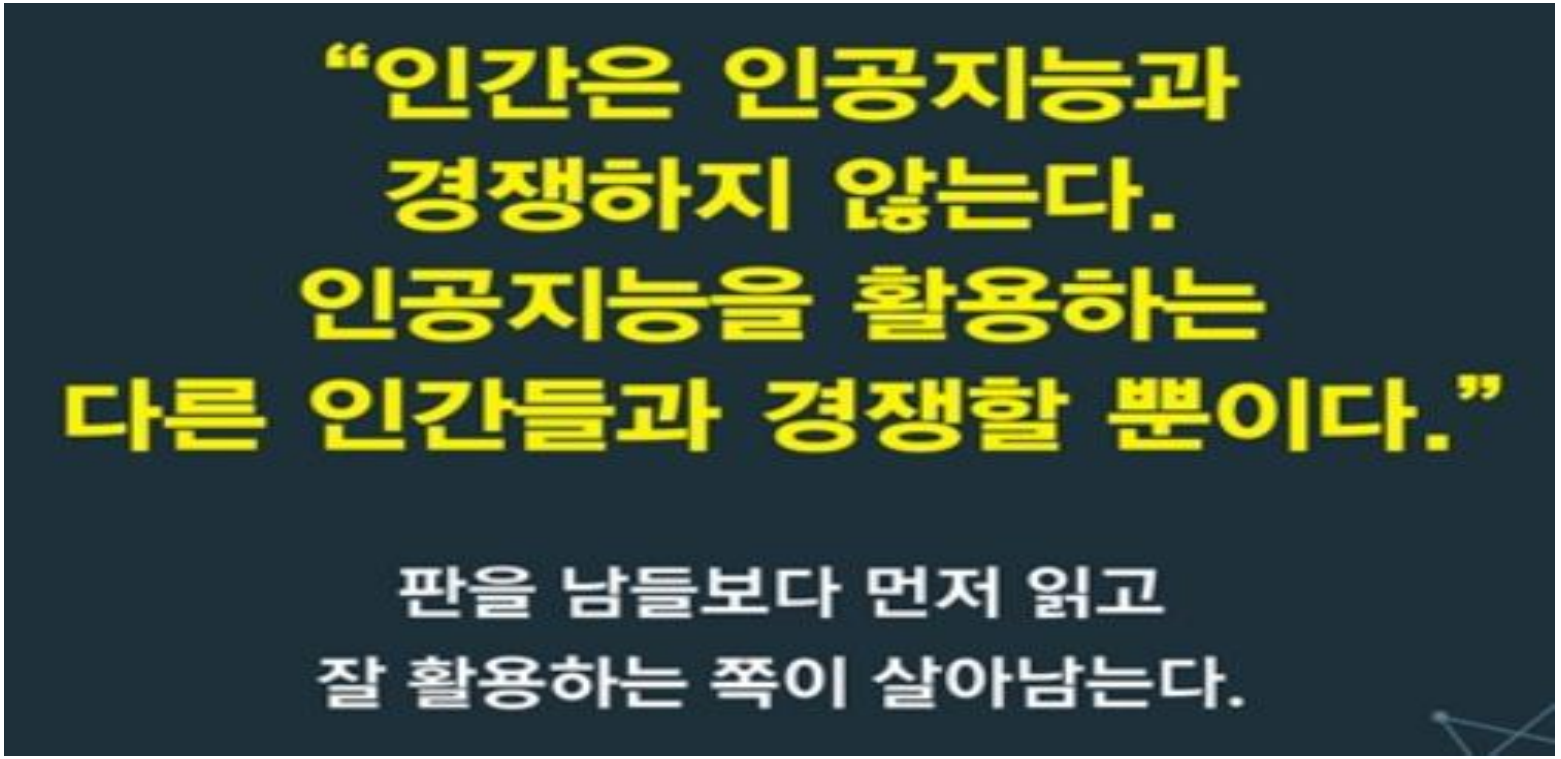
2019.10

References:

- **KPC, DSAC(Data Scientist Academy & Certificate) Manual, 2019**
- **many internet sites**

내용

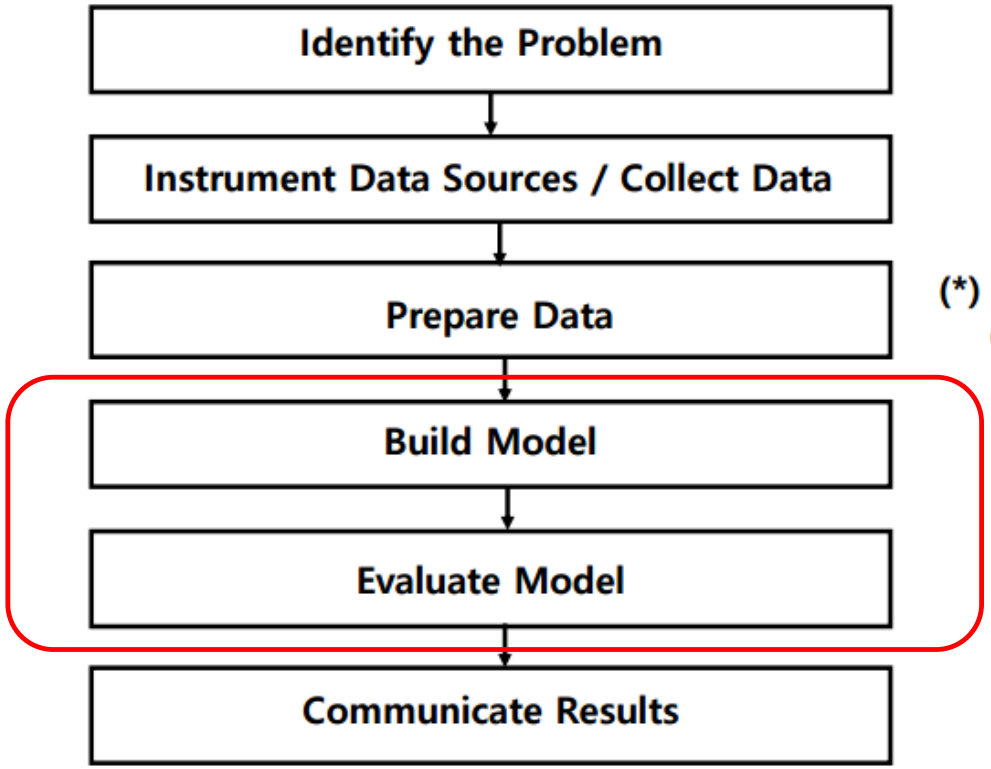
1. Machine Learning (기계학습)
2. Clustering
3.
1. 결정트리
2. 랜덤 포레스트
3. 서포트 벡터머신
4. 분류 성능
5. 특성공학
6. 모델 최적화
7. 이미지 분석
8. 텍스트 분석



출처: 인공지능시대의 비지니스전략

데이터 분석 모델

Data Analysis Model (Jeff Hammerbacher)



(*) integrate, transform,
clean, aggregate, alter

Machine Learning
(머신러닝 or 기계학습)

인공지능과 머신러닝

- 인공지능을 구현하는 방법은 다양
- 머신 러닝 기반의 AI가 2000년대 이후 급속히 발전
- 딥러닝: 신경망을 기반으로 하는 머신 러닝 기술
  - 마치 사람이 많은 정보에 접하면서 학습하듯이 컴퓨터도 데이터를 보고 학습하는 방법
  - 음성인식, 자동차 번호판 인식, 언어 번역, 채팅 대화, 글쓰기, 작곡, 논평, 소설, 예술 등 여러 분야에서 좋은 성과를 낸다.

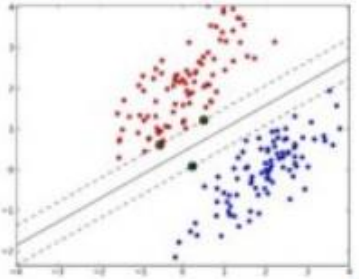
머신 러닝

인공지능(Artificial Intelligence)

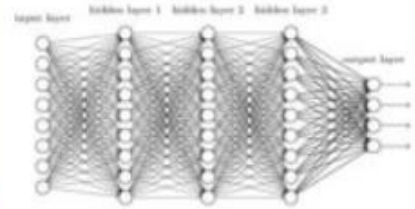
- Deep Blue

머신러닝(Machine Learning)

- Linear Regression
- Logistic Regression
- SVM



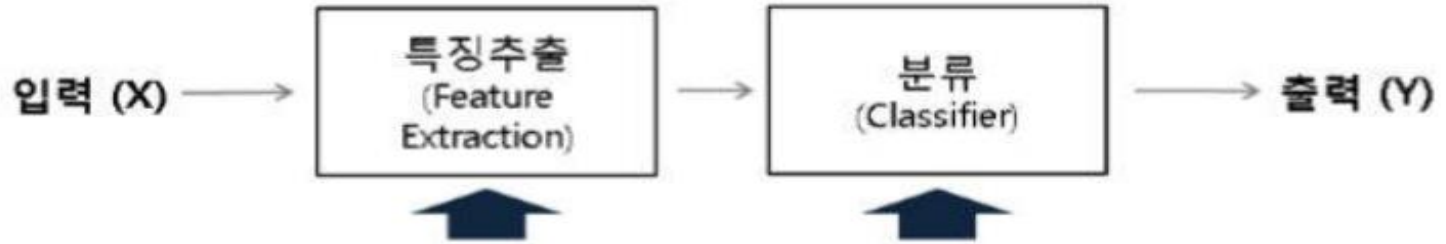
딥러닝(Deep Learning)



- CNN
- RNN
- LSTM

머신 러닝 vs 딥 러닝

< 머신러닝 (Machine Learning >



- 특징추출을 위한 알고리즘을 인간이 직접 제공
- 해당분야에 대한 지식 및 직관, 알고리즘 구축을 위한 상당한 노력 필요
- 컴퓨터 '학습'의 영역

< 딥러닝 (Deep Learning >

- 컴퓨터 '학습'의 영역

머신러닝 특징

- 예전에는 컴퓨터는 프로그래머가 코딩한 대로만 동작 (**알고리즘**)
  - 계산을 빨리 하든지,
  - 이미지를 처리하든지,
  - 정해진 알고리즘대로 빠르고 정확하게 동작하는 일
- 머신러닝 (**데이터**)
  - 컴퓨터가 데이터를 보고, 스스로 기능을 향상시키는 방법을 찾아내어서 점차 성능을 향상시킨다.

머신러닝 알고리즘과 머신러닝 모델

- 머신러닝 **알고리즘**
  - 머신러닝 '모델'을 생성하기 위해 데이터에서 실행되는 절차
  - (ex) Linear Regression, Logistic Regression, Decision Tree, Neural Networks, k-Nearest neighbors, k_Means
- 머신러닝 **모델**
  - 데이터에서 실행된 기계학습알고리즘의 출력 (알고리즘에서 학습된 내용)
  - Model data + Prediction algorithm
  - (ex) linear regression: linear equation

    decision tree: tree of if-then statements

    neural network: network with weight matrices
- Algorithm is used to find model, the model is the program that solves the problem.

모델의 예

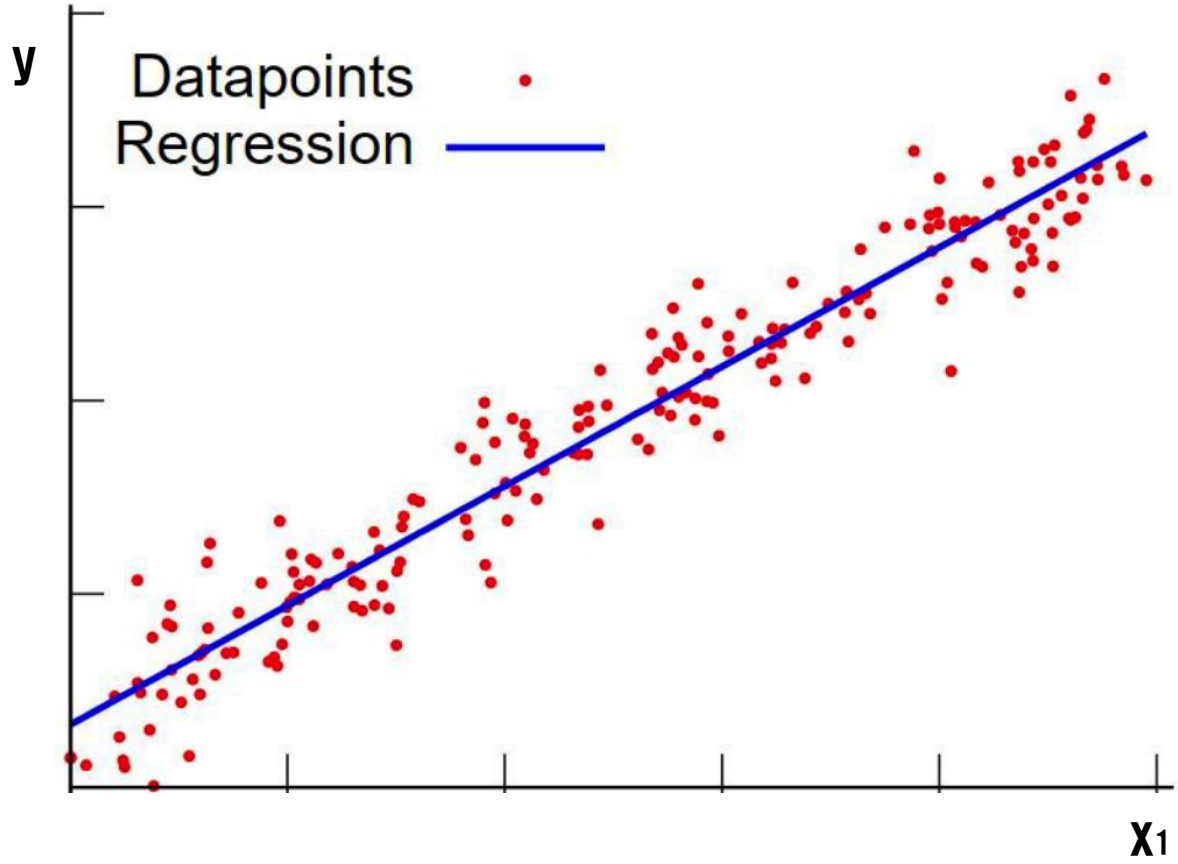
- 와인 품질 = 12.145 + (0.00117x겨울철 강수량)
  + (0.064x 재배철 평균기온) - (0.00386x 수확기 강수량)

머신러닝 모델

- 머신 러닝, AI 모델은 데이터 기반의 모델을 사용(학습)
- 현실 세계의 많은 현상
  - 수식으로 간단히 모델링하기 어렵고, 과학적으로 증명할 수도 없다.
  - 하지만 거의 정확히 예측할 수 있는 모델은 만들 수 있다.
    (단, 충분한 데이터 필요)
  - 머신 러닝 모델 예
    - 어느 고객이 불만이 많을 것인지
    - 어떤 영화가 관객을 많이 동원할지
    - 어떤 물건이 많이 팔릴지
    - 어떤 메일이 스팸일지
- 머신 러닝은 성능이 꽤 유용

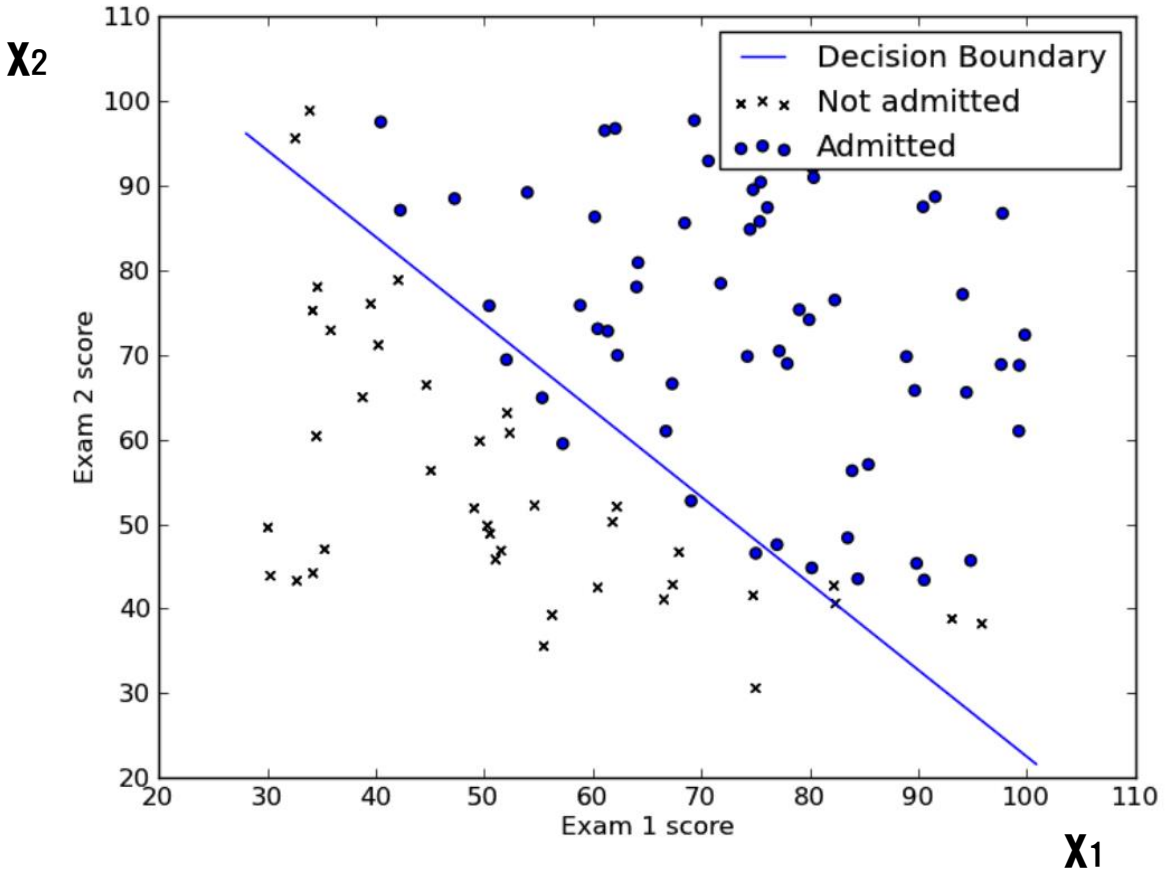
선형 회귀 모델

- 선형 회귀(regression) y = wX + b



선형 분류 모델

- 선형 분류(classification) ax1 + bx2 + c = 0



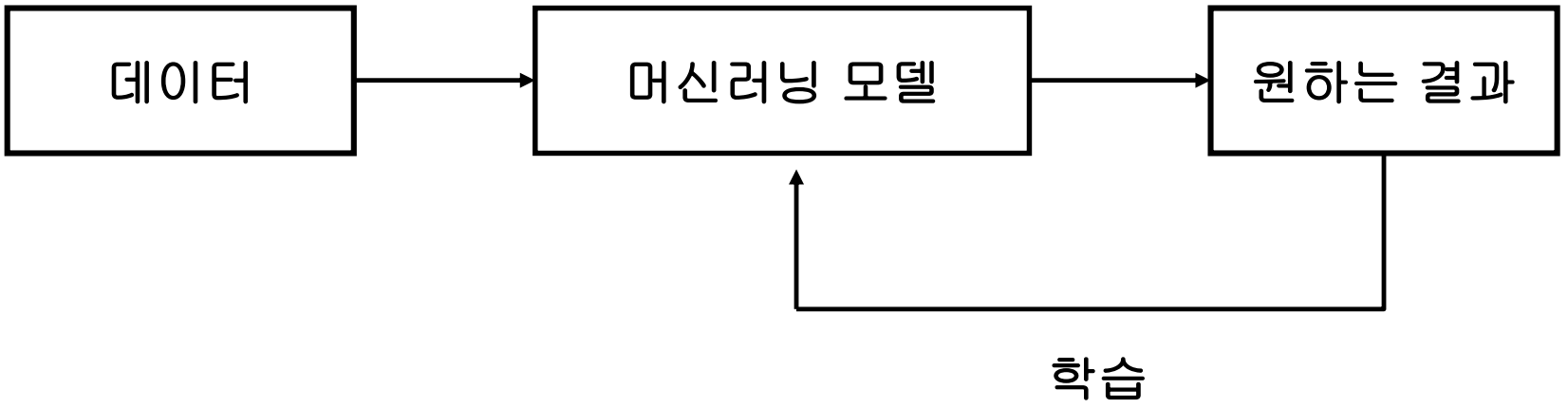
모델 파라미터

- 모델
  - 모델 구조 : 모델의 동작을 규정
  - 모델 파라미터 : 모델이 잘 동작하도록 정한 가중치 등 계수
- 특정 모델은 데이터 특성에 따라 예측 정확도가 달라질 수 있다
  - 적절한 모델 구조 : 프로그래머가 선택
  - 적절한 모델 파라미터 : 머신러닝 프로그램이 데이터 기반하여 학습

(*) Hyper-Parameter (하이퍼파라미터): 모델 외부에 있으며, 데이터에서 추정할 수 없는 파라미터. 주로 경험 있는 사람이 주며, heuristic이나 경험에 의해 결정됨.

머신 러닝의 기본 동작

- **머신러닝 목표**
  - **주어진 학습 데이터를 보고 원하는 동작을 잘 수행하는 모델을 만드는 것**
  - **즉, 회귀 또는 분류 작업을 정확하게 수행**
  - **좋은 모델을 만들기 위해서는 → 좋은 데이터가 필요**



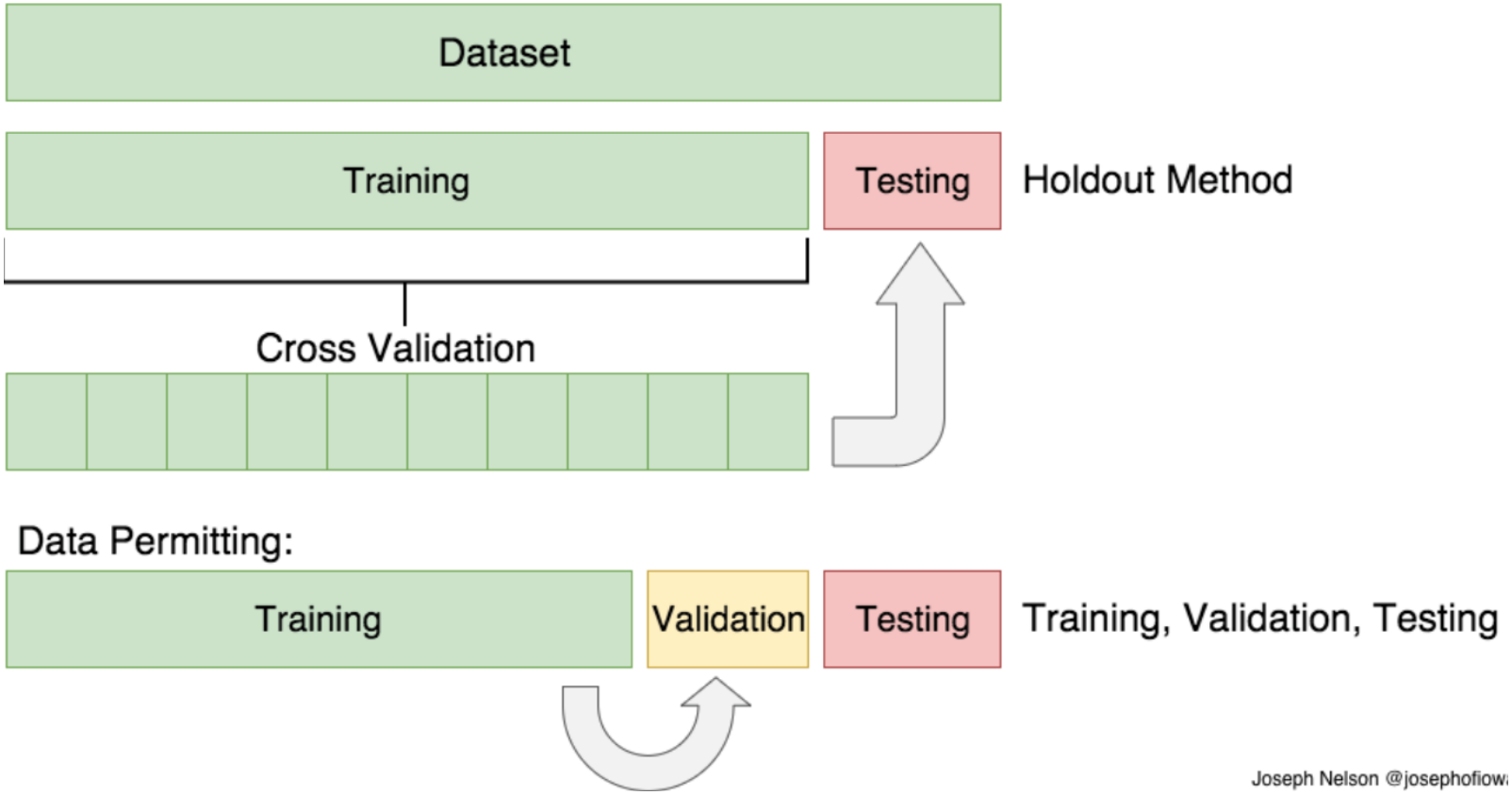
훈련과 검증

- **모델 훈련(Training)**
  - **모델이 데이터를 이용하여 모델 파라미터를 학습하는 과정**
  - **모델 파라미터 값 : 보통 랜덤한 값으로 초기화**
  - **학습**
    - **훈련 데이터에 기반하여 최적화 알고리즘에 의해서 모델 파라미터 값을 계속 갱신하여 모델의 예측 값이 실제 값에 수렴하도록 하는 훈련 과정**
- **모델 검증 (Validation)**
  - **모델을 학습시킨 후, 모델이 잘 동작하는지를 확인하는 과정**
  - **보통 검증 데이터를 따로 제공하지 않으므로 훈련에 사용할 데이터의 일부를 검증용으로 미리 확보해야 함.**

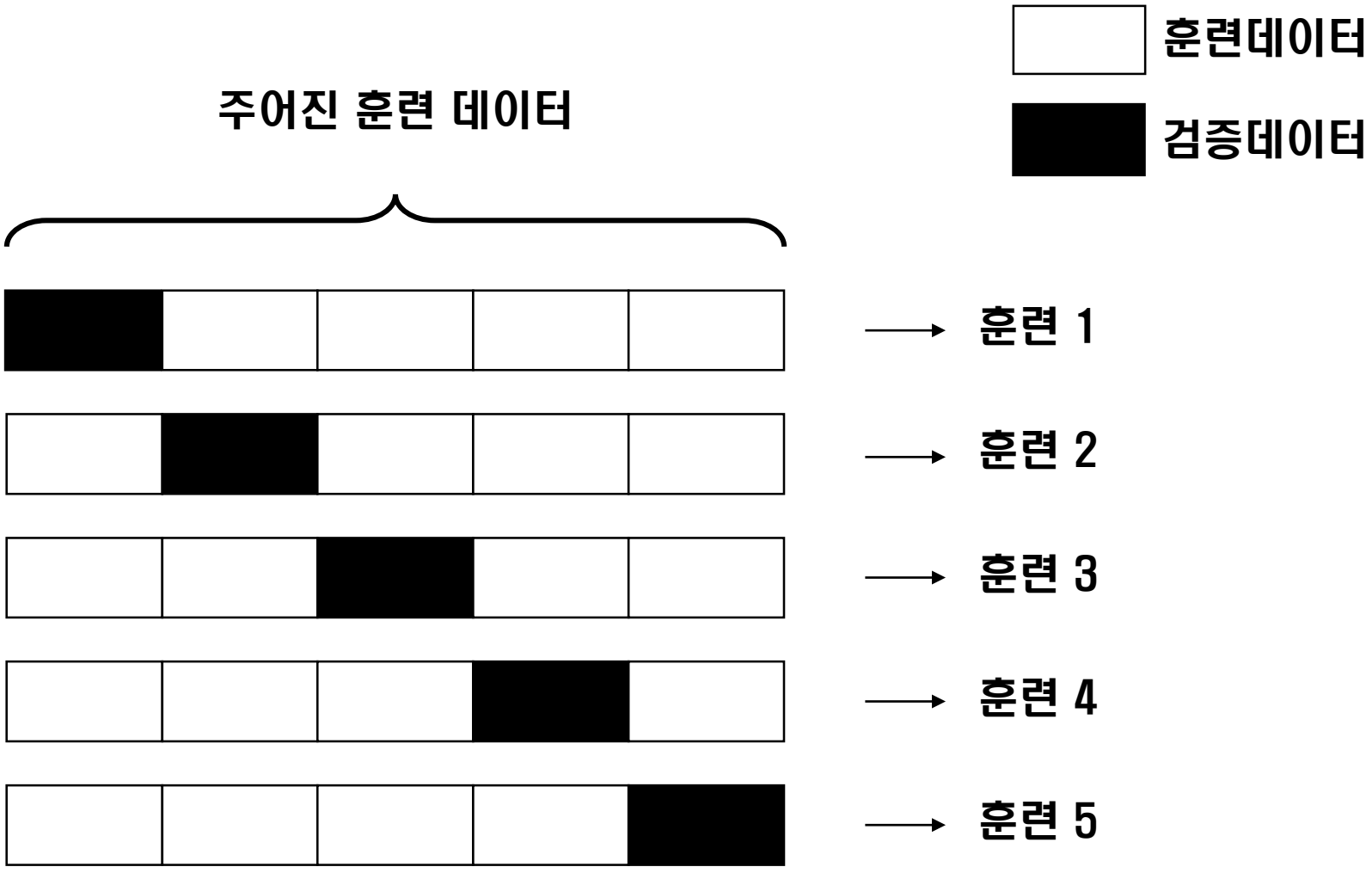
훈련, 검증, 테스트 데이터

- **훈련**(Training) 데이터
  - 모델 parameter를 학습시키는데 사용
- **검증**(Validation) 데이터
  - 모델의 학습 중에 과소적합, 과대적합을 검사하고 최적 모델 구조(hyper parameter 등)를 찾는데 사용
  - 훈련 데이터 중의 일부를 학습에 참여시키지 않고 남겨 둔 데이터
- **테스트**(Test) 데이터
  - 모델의 성능을 최종적으로 시험하는데 사용

훈련, 검증, 테스트 데이터


Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiow

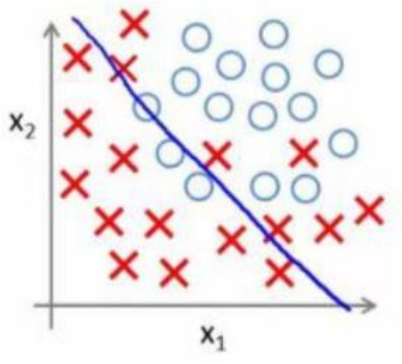
k-fold 교차 검증



k-fold 교차 검증

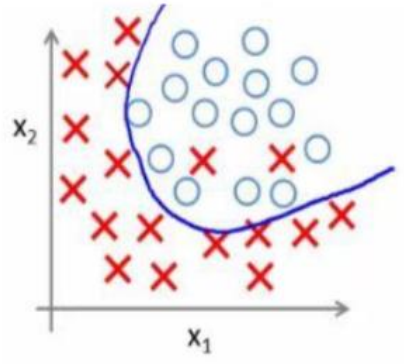
- fold : 검증 데이터
- 주어진 데이터 전체를 골고루 검증용으로 사용
  - 모델의 동작을 보다 정교하게 확인하기 위함
  - K 값은 보통 5~10 주로 사용
  - cross_val_score() : 교차 검증 자동 수행 & 성능 평가
- 교차 검증의 목적은 성능 검증
- K개의 점수가 골고루 나와야 안정적인 모델

과대 적합, 과소 적합

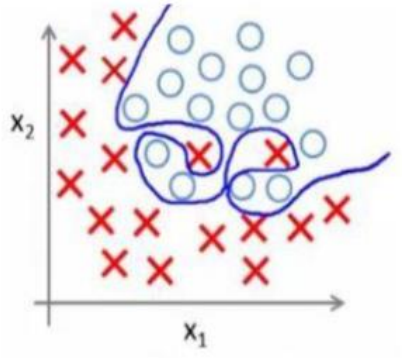


과소 적합

(underfitting)



Good fit

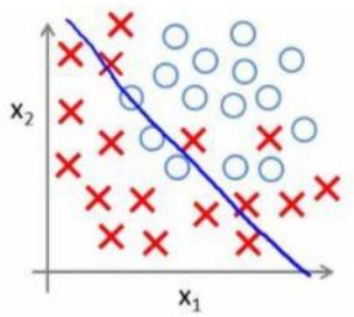


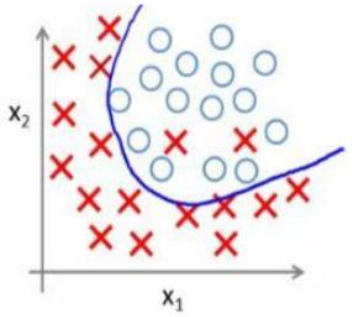
과대 적합

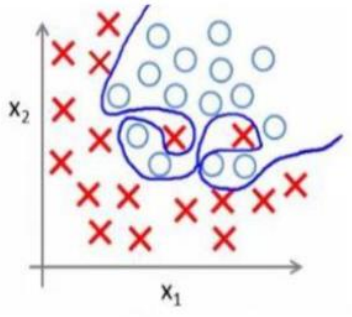
(overfitting)

과대 적합(Overfitting)

- 모델이 훈련 데이터에 대해서만 잘 동작하도록 훈련되어, 새로운 데이터에 대해서는 오히려 잘 동작하지 못하는 것
  - 주어진 훈련 데이터를 너무 세밀하게 학습에 반영하여 발생하는 현상
- 과대 적합된 모델은 훈련 데이터에 대해서는 매우 우수한 성능을 보이지만 **일반성**이 떨어진다







일반화(Generalization)

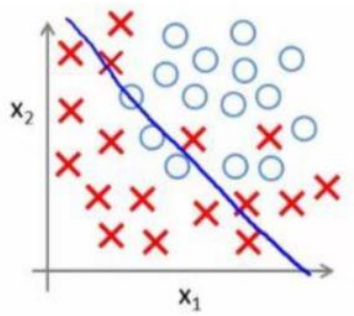
- 모델의 일반화(Generalization)
  - 머신러닝에서는 과대 적합을 피해서 일반적으로 잘 동작하게 모델을 만드는 것이 매우 중요
- 과대 적합의 원인 & 대책
  - 원인 : 훈련 데이터가 너무 적어서 학습을 충분히 할 수 없는 경우
    - 대책 : 다양한 경우를 고려한 훈련 데이터를 많이 확보
  - 원인 : 모델이 너무 복잡한 경우
    - 대책 : 모델을 좀 단순하게

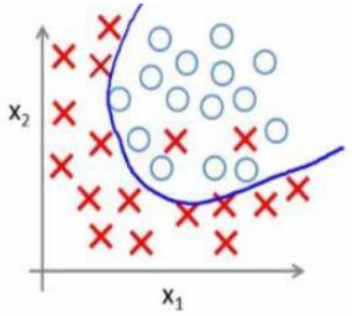
규제화(Regularization)

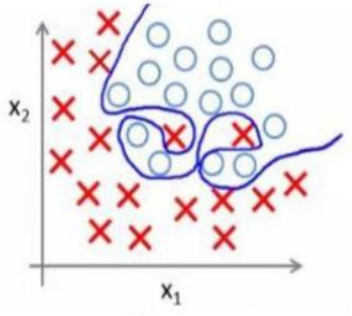
- 규제화
  - 모델이 일반화 능력을 갖도록 모델의 기능을 제한하는 것
  - 예) 만일 학습할 데이터가 부족하다면,
    - 모델 구조를 좀 단순하게 만들어서 주어진 데이터에 대한 과대 적합을 피해야 한다
- 머신러닝에서는 일반화 능력을 가진 모델을 만드는 것이 중요
  - 이를 위하여 모델에 적절한 제한을 가하는 기법을 사용

과소 적합(Underfitting)

- 문제의 복잡도에 비해 **모델이 너무 간단**하여 주어진 **훈련 데이터에서 조차도** 잘 동작하지 못하는 것
- 대책
  - 모델의 보다 복잡(상세)하게 구성
  - 제약을 줄여준다







데이터의 대표성

- 훈련 데이터 구성
  - 미래에 나타날 가능성이 있는 모든 데이터의 특징을 골고루 반영
  - 예) 투표 결과 예측
    - 실제 인구 구성에 비례하여 성별, 지역, 인종, 나이별, 소득별 등 균형성 유지
- 층화 샘플링 or 다단계 샘플링(stratified sampling)
  - 데이터의 대표성을 고려하여 데이터를 수집하는 방법
    - 예) 어느 학교의 남녀 학생 비율이 8:2 → 의견수렴 샘플도 8:2 유지
- 훈련, 검증, 테스트 샘플 데이터가 전체 데이터의 특징을 계속 유지할 수 있어야 함

모델 구축 과정

- 머신러닝 모델 선택
  - 해결할 문제에 최적의 모델 선택
    - 훈련 데이터, 원하는 목적(기능) 등 고려
  - 선형모델, 결정트리, 신경망, SVM, 랜덤포레스트 등
- 모델 학습 : 훈련 데이터 사용
  - fit() 함수
- 모델이 과대 적합 또는 과소 적합인지를 검증
  - 과대적합 → 모델을 더 일반화(모델 단순화 또는 규제화)
  - 과소적합 → 모델을 더 복잡(상세)하게 설계
- 성능 평가 : 실제 테스트 데이터를 적용
  - predict(), score(), predict_proba(), decision_function() 함수

머신 러닝의 문제 유형

- 머신러닝을 이용해 문제 해결 유형 예
  - 설명 (description)
  - 예측(prediction)
  - 추천 (recommendation)
  - 연관분석
  - 강화학습

머신 러닝의 유형별 대표적 알고리즘

	머신러닝 유형	알고리즘
지도학습 (supervised)	분류	kNN, 베이즈, 결정 트리, 랜덤 포레스트, 로지스틱 회귀, 그라디언트부스팅, 신경망
	회귀	선형 회귀, SVM, 신경망
비지도학습 (unsupervised)	군집화	k-means, hierarchical, DBSCAN
	데이터 변환	스케일링, 정규화, 로그변환
	차원축소	PCA, 시각화

지도학습

- 지도학습은 정답 (or 목적변수)을 예측하는데 사용된다.
- 정답은 목적(target) 변수, 레이블(label) 이라고도 한다
- 예측은 분류와 회귀로 나누어진다.
- 분류
  - 분류(classification)란 어떤 항목(item)이 어느 그룹에 속하는지를 판별하는 기능을 말한다.
  - 두 가지 카테고리를 나누는 작업을 이진 분류(binary classification)라고 하고 세 개 이상의 클래스를 나누는 작업을 다중 분류(multiclass classification)라고 한다.
- 회귀
  - 수치를 예측하는 것을 회귀(Regression) 라고 한다.

비지도학습

- 비지도 학습이란 정답이 없이 데이터로부터 중요한 의미를 찾아내는 머신러닝 기법이다.
  - **군집화**: 유사항 항목들을 같은 그룹으로 묶는다.
  - 데이터 변환: 데이터를 분석하기 좋게 다른 형태로 변환한다
  - **차원축소**: 데이터의 속성을 명확하게 시각화하기 위해서 고차원의 특성 값들을 2차원이나 3차원으로 차원을 축소하는 작업 (예: 주성분분석 (PCA))

강화학습

- 강화학습(reinforcement learning)은 머신러닝 모델이 어느 방향으로 만들어져야 하는지 방향성만 알려주는 학습 방법
  - 입력 샘플에 대한 정답이 있는 게 아니고 스스로 수행 (경험)하면서 답을 만들고 그 데이터를 가지고 학습.
- 강화학습에서는 일정 기간 동안의 행동(action)에 대해 보상(reward)을 해줌으로써 잘 하고 있는지, 잘못하고 있는지를 알려주며 학습을 시킨다.
  - 예를 들어 로봇이 혼자 그네를 타는 방법, 전자 게임을 하는 방법, 바둑을 두는 방법의 학습에 사용된다.
  - 2017년에 우리나라 이세돌을 이긴 알파고(Alpha Go) 바둑 프로그램

모델의 동작 성능

- 모델의 동작 속도
  - **학습 시간** : 모델을 만드는데 걸리는 시간
  - **동작 속도** : 모델을 적용하는데 걸리는 시간
- 일반적으로 모델이 정교하고 복잡할수록 성능은 좋아지지만 모델을 만들거나 적용하는데 시간이 오래 걸린다.

클러스터링

- 유사도 (Similarity)
- need "Scaling" as a preprocessing step
- K-Means
- 병합군집(agglomerative clustering)
- DBSCAN

유사도 (similarity)

- 항목간의 유사한 정도를 수치로 나타낸 것
- 분류나 예측에서 필요
- 메일이 스팸에 가까운지 아니면 정상 메일에 가까운지
- 추천에서 두 아이템 또는 사람이 서로 얼마나 가까운지

유사도 측정

- **A, B, C 중 누가 서로 가까울까?**
  - **상대적인 차이를 보통 사용 (z 변환(표준 스케일링))**

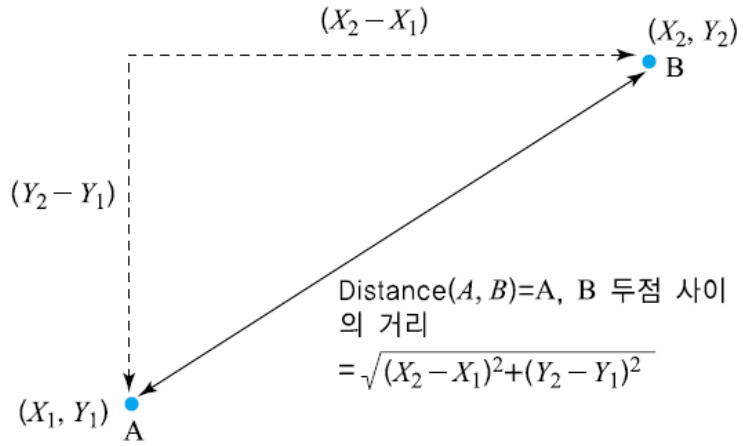
구분	키	몸무게	나이
A	174cm	70kg	21세
B	170cm	61kg	27세
C	162cm	73kg	29세

유사도와 거리

- 유사도 결과에 따라 데이터 분석 결과가 달라짐
- 분석 경험과 도메인에 대한 이해 필요함
- 최적의 분석 결과가 나오도록 유사도를 변경해 가면서 반복 수행 필요함
- 유사도 s(similarity)는 $0 \leq s \leq 1$ (1에 가까울수록 유사도 높음)
- 유사도의 상대 개념으로 거리(distance) 사용
  - ✓ 유사도와 거리의 관계: $d = 1 - s$

공간 거리

- 기하학의 공간(space) 상의 거리 – 유클리디언 (Euclidian) 거리

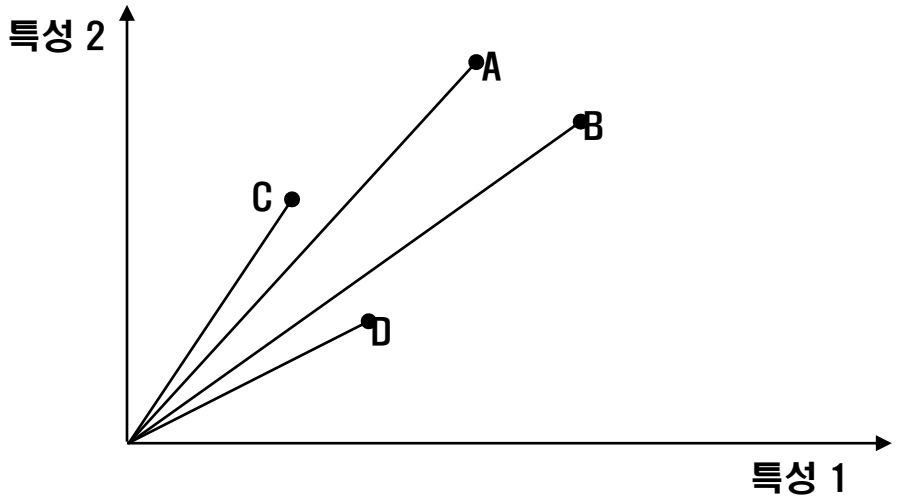


- n 차원 공간상의 두 점의 거리

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

코사인(cosine) 유사도 – 방향성, 취향

- 공간상의 두 점이 만드는 각도를 기준으로 유사도를 측정하는 방법 (-1 ~ +1 사이의 값을 갖는다)
- 공간상 거리가 멀어도 두 점이 가리키는 방향이 같으면 서로 비슷하다고 보는 것



$$s_{\cos}(x,y) = \frac{X \bullet Y}{|X||Y|}$$

- A와 C가 가깝고 B와 D가 가깝다고 정의

자카드(Jaccard) 유사도

- 비슷한 취향의 사람을 찾을 때 사용 – 영화, 도서, 음악 추천 등
- 영화 보는 취향에 따른 유사도 측정
- 지난 1년 동안 국내에 개봉된 영화가 500편
  - A와 B가 본 영화 중 겹치는 영화가 5편, 5/500 = 0.01
  - A와 C가 본 영화 중 겹치는 영화가 10편, 10/500 = 0.02
  - 즉, 0.01 < 0.02이므로 A와 C가 더 가깝다고 할 수 있음
  - 위와 같은 계산 방법이 적절한가?

자카드 유사도

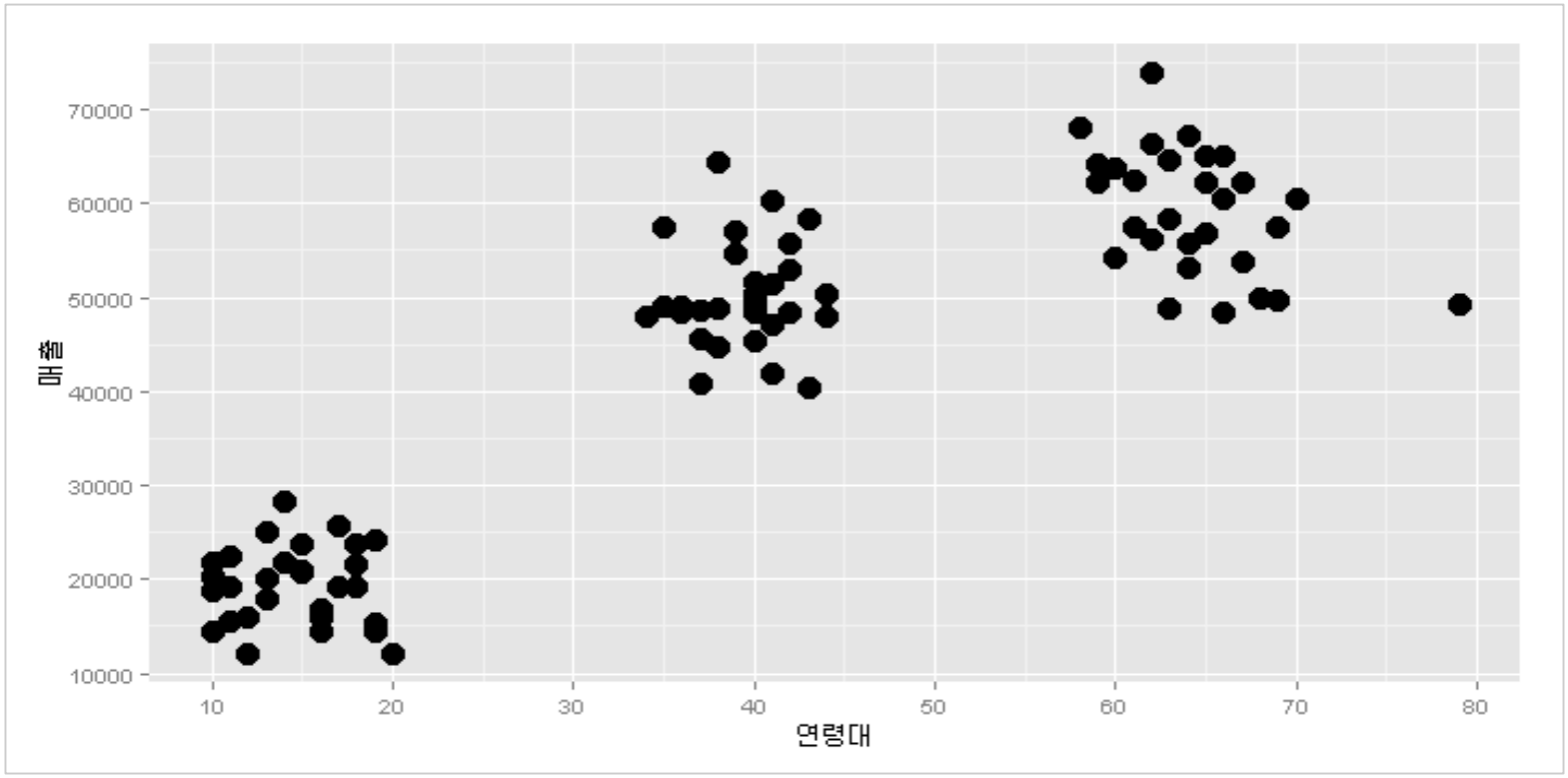
- 어떤 두 항목이 겹치는 부분의 절대량만을 보지 않고, 두 항목의 공통 부분이 얼마나 많은지를 고려하여 이에 대한 상대적인 값을 유사도로 사용해야 함

$$S_{Jaccard}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

  - A, B, C가 각각 지난해 본 영화의 총 개수가 20편, 50편, 200편
  - J(A,B) = 5 / (20+50-5) = 0.076
  - J(A,C) = 10 / (20+200-10) = 0.047
  - 즉, 0.076 > 0.0470이므로 A와 B가 더 가깝다고 할 수 있음

클러스터링 (Clustering)

- 성격이 비슷한 항목들을 그룹으로 묶는 작업

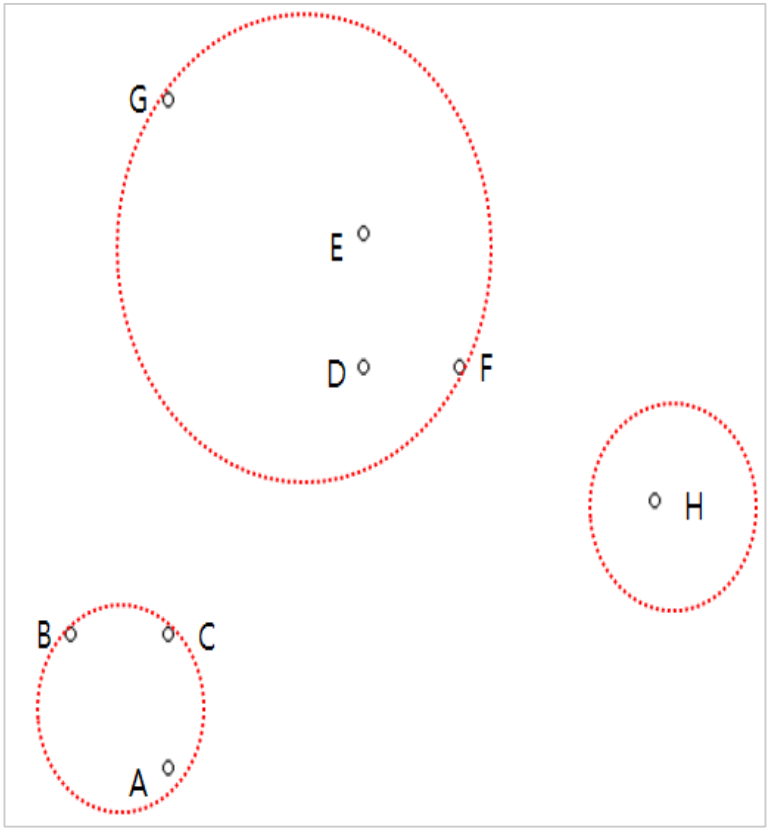


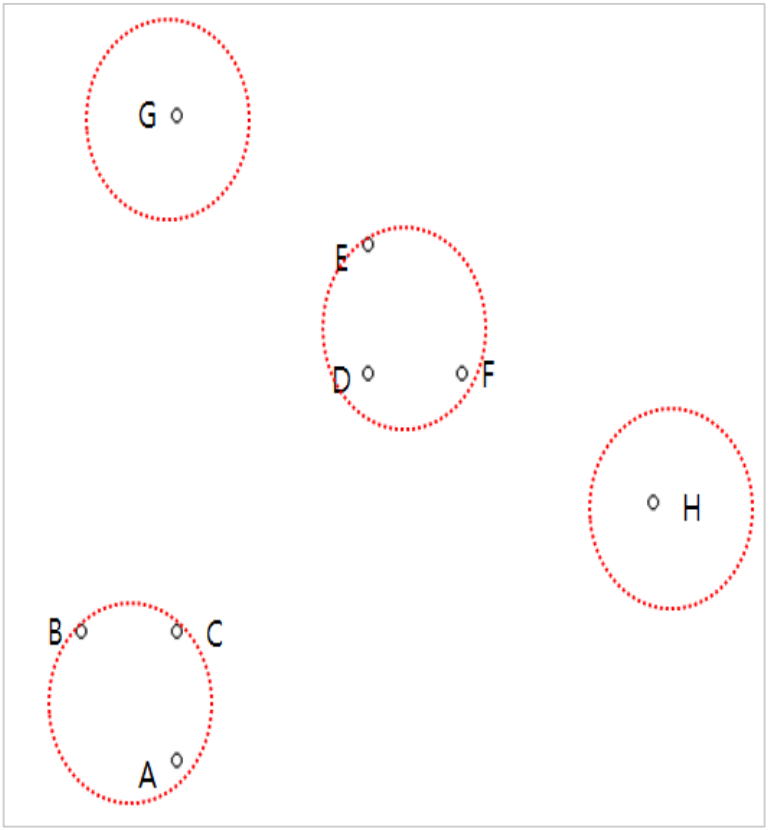
클러스터링 알고리즘

- 조건
  - 같은 그룹 내의 항목들은 서로 속성이 비슷함 (유사도가 큼)
  - 다른 그룹에 속한 항목과는 속성이 서로 다름 (유사도가 작음)
- 비정상 패턴 (이상치) 식별에도 사용된다
  - (ex) 컴퓨터 시스템에 침입한 해커의 행동

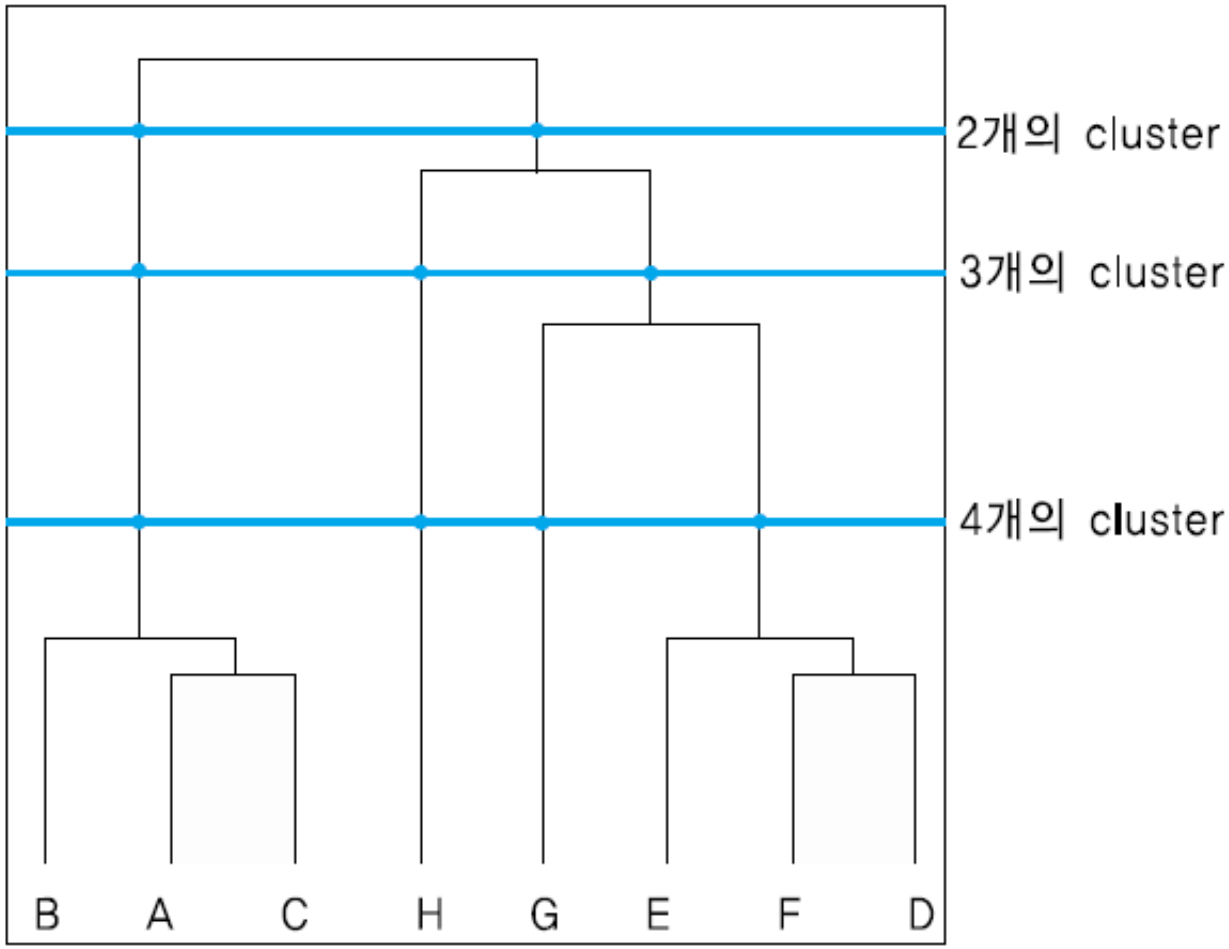
클러스터 수, k

- 적정한 군집의 수(k)를 먼저 찾아야 함





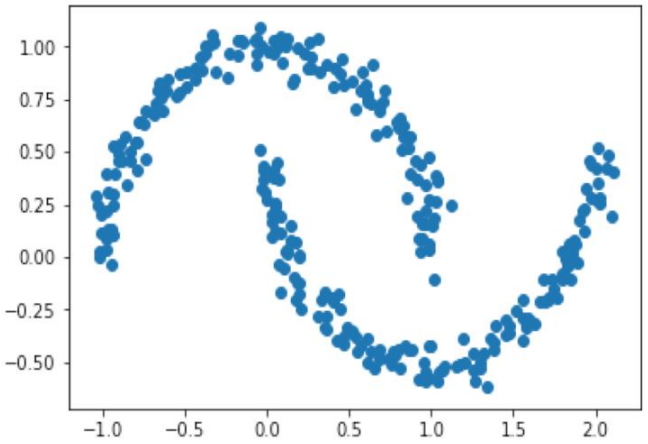
덴드로그램(Dendrogram)

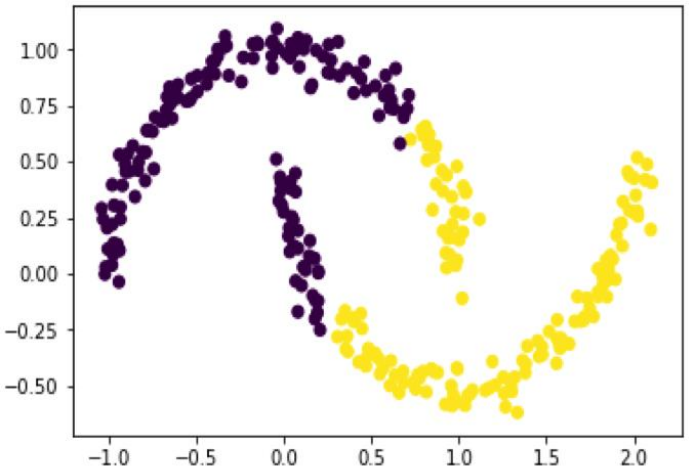


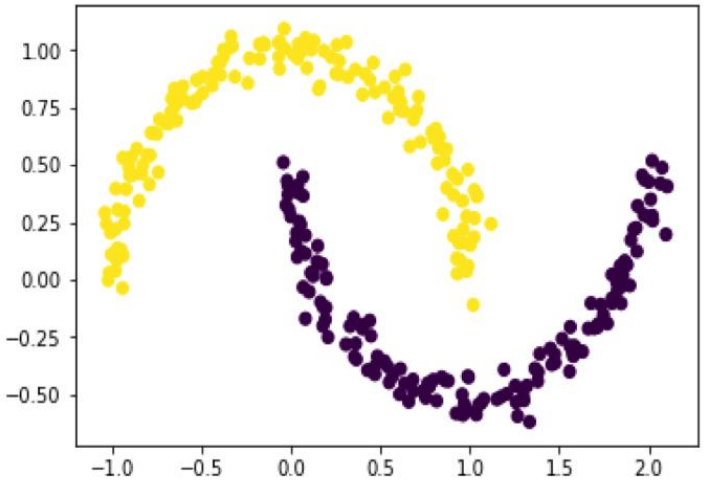
K-Means 알고리즘

- 공간상에 임의의 k 개의 임의의 초기 지점을 클러스터 중점으로 (cluster center) 정함
- 클러스터 중점을 중심으로 거리가 가까운 항목을 선택하여 클러스터 공간을 나눔
- 각 클러스터에 포함된 항목들의 평균 위치를 구해 이를 새로운 클러스터 중점(centroid)으로 변경
- 새로 설정된 **센트로이드**를 중심으로 경계를 다시 그림
  - 각 항목들이 소속된 클러스터가 바뀔 수 있음
- 변경된 항목들을 가지고 클러스터 중심을 다시 계산
- 더 이상 클러스터의 모양이 바뀌지 않을 때까지 반복 수행함
  - KMeans() **사용**

Two Moons 데이터





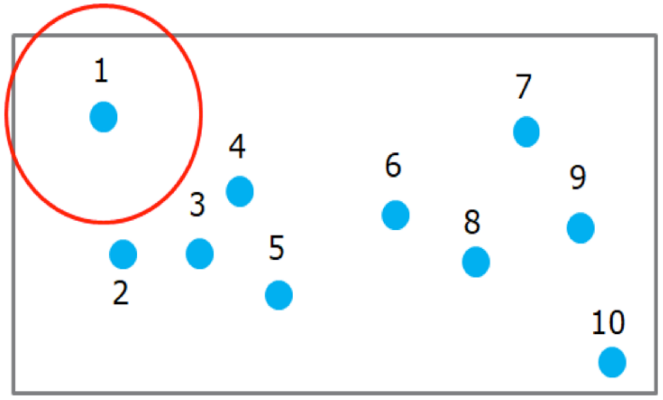


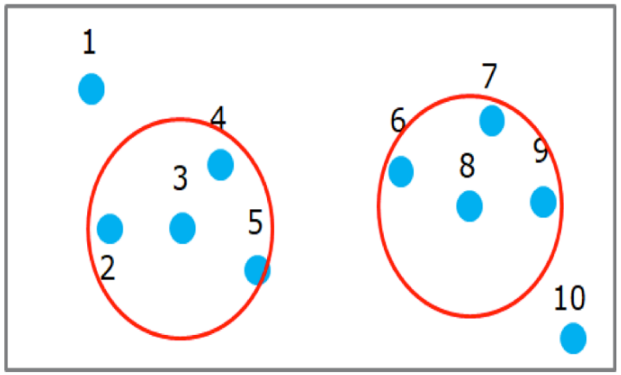
밀도기반 알고리즘(DBSCAN)

- **Density based Spatial Clustering of Applications with Noise (one of the most common clustering algorithms)**
- **밀도 기반 클러스터링 알고리즘이다.**
- **k-means처럼 단순히 거리만을 기준으로 군집화를 하는 것이 아니라 "가까이 있는 샘플들은 같은 군집에 속한다" 는 원칙으로 군집을 차례로 넓혀가는 방식이다.**
- **샘플들의 몰려 있는 정도 즉, 밀도가 높은 부분을 중심으로 인접한 샘플들을 포함시켜 나간다.**
- **한 점을 기준점으로 반경 r내에 점이 n 개 이상 있으면 하나의 군집으로 인식하는 방식이다.**

밀도기반 알고리즘(DBSCAN)

- 1번 데이터를 중심으로 보면 반지름 r 인 원 안에 군집이 되기 위한 최소기준인 (예를 들어 n=4라면) 샘플이 없다.
- 이 데이터는 노이즈 데이터(noise point)가 되며 클러스터에서 제외한다.
- 3번과 8번 데이터를 중심으로 보면 원 안에 4개의 점이 있으며 이러한 데이터를 코어 데이터(core point)라고 한다.
  - 코어 데이터들은 스스로 클러스터를 형성할 수 있다.





밀도기반 알고리즘(DBSCAN)

- 2번 데이터는 최소 기준인 4개의 데이터를 포함하지는 못하지만 코어 데이터인 3번을 포함한다. 이런 데이터를 **경계 데이터**(border point) 라고 하며 인접한 군집에 포함시킨다.
- 정해진 반지름 r 인 원을 이용해 코어 데이터, 경계 데이터, 노이즈 데이터들을 분류하면 아래와 같다.
- 두 개의 클러스터와 두 개의 노이즈를 구분했다.
- 코어데이타가 다른 코어의 일부가 되면 하나의 군집으로 연결.