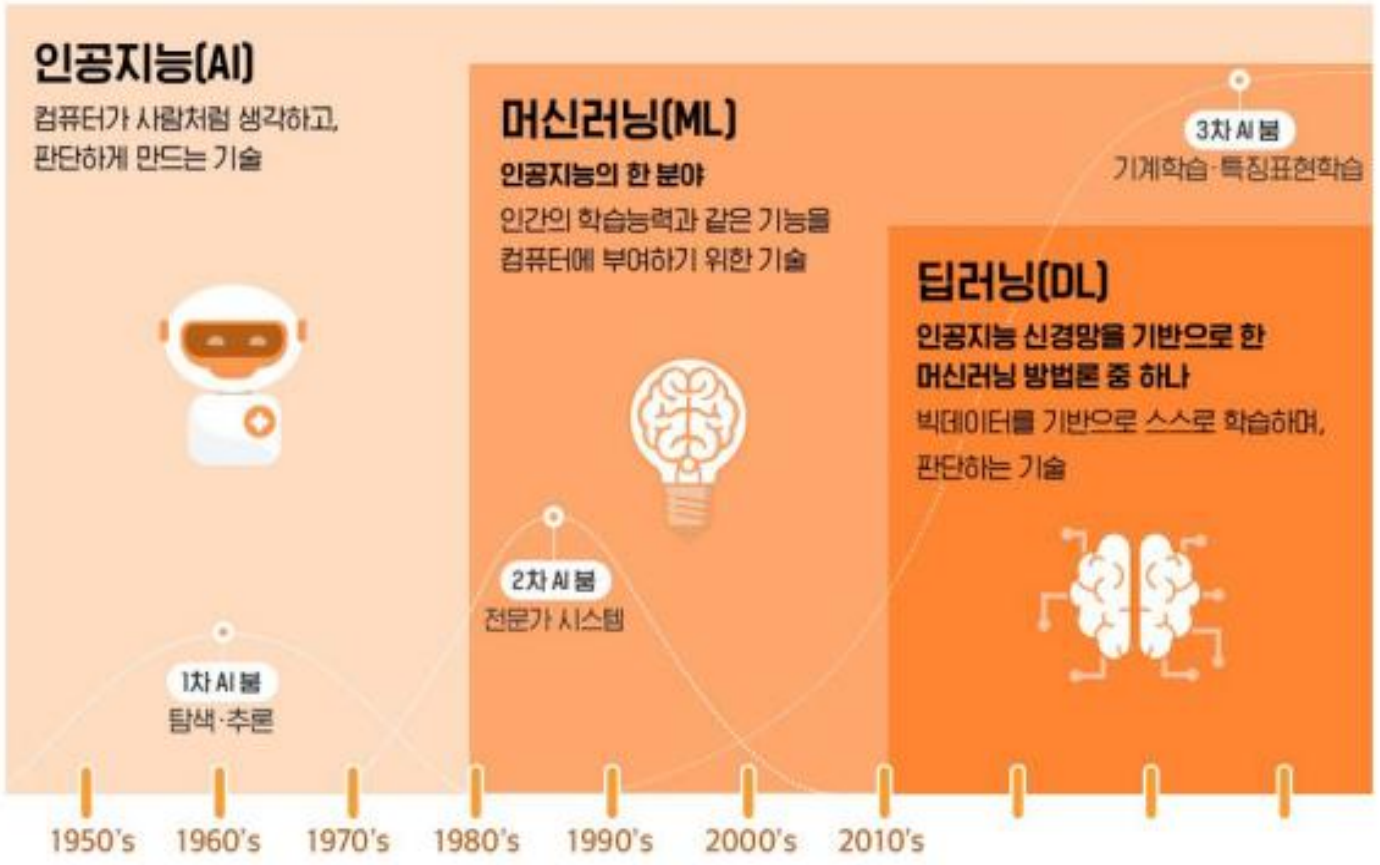
머신 러닝(Machine Learning)

1. 머신러닝

1959년, 아서 사무엘은 기계 학습을 "기계가 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구 분야"라고 정의하였다. 기계 학습의 핵심은 표현(representation)과 일반화(generalization)에 있다. 표현이란 데이터의 평가이며, 일반화란 아직 알 수 없는 데이터에 대한 처리이다.



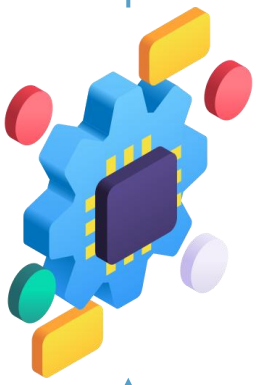
2. 지도학습과 비지도학습

	지도학습 (Supervised Learning)	비지도학습 (Unsupervised Learning)	강화학습 (Reinforcement Learning)
훈련 방식	정답이 있는 데이터로 훈련	정답이 없는 데이터로 훈련	자신의 행동에 대한 보상을 받으며 목표를 달성하는 방향으로 학습
주요 알고리즘	분류, 회귀	군집화, 차원축소	로보틱스, 시뮬레이션
예시	• 강아지와 고양이 사진 분류하기(분류) • 집 값 예측하기(회귀)	• 라벨이 없는 데이터를 n개의 집단으로 구분(군집화) • 변수의 여러가지 특징을 보다 작은 수로 축소(차원 축소)	• 자율주행 차 • 알파고 등

2. 지도학습과 비지도학습

라벨(Label), 종속변수

집값(Y)
?



특징(feature), 독립변수

위치(X1)	크기(X2)
강남	32평

라벨의 유무

지도학습
(Supervised Learning)

정답이 있는 데이터로 훈련

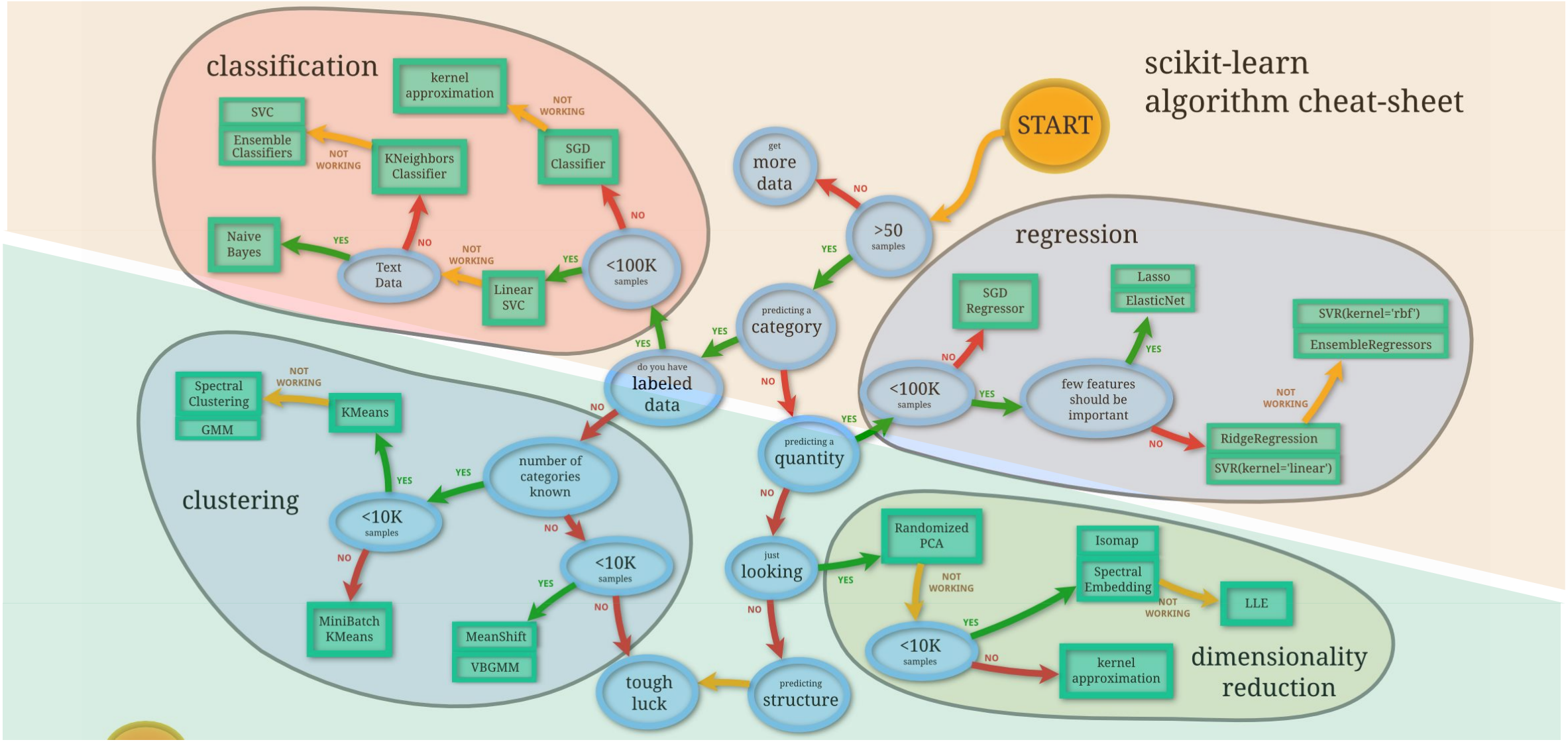
분류, 회귀

비지도학습
(Unsupervised Learning)

정답이 없는 데이터로 훈련

군집화, 차원축소

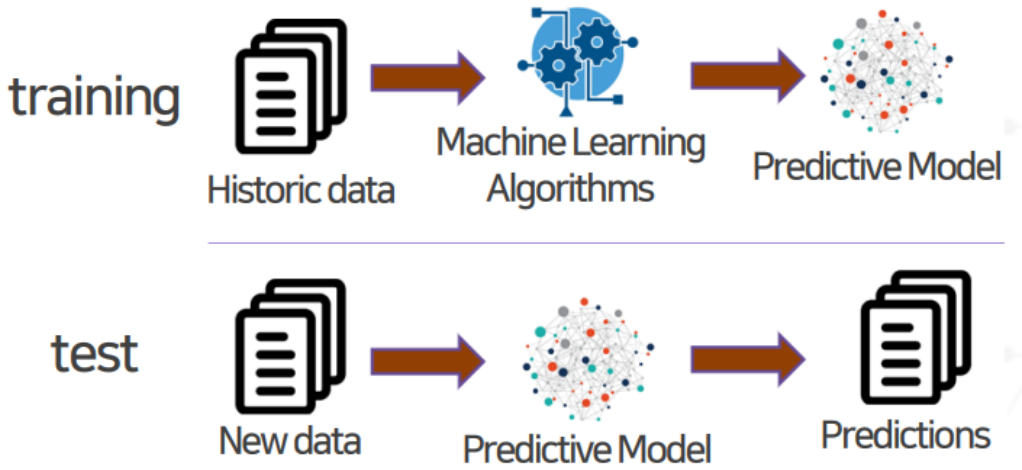
2. 지도학습과 비지도학습



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

2. 지도학습과 비지도학습 – 지도학습의 경우

- 본 적이 없는(학습데이터에 없었던) test data의 output을 예측(prediction)
- 어떤 input이 output에 어떻게 영향을 미쳤는지 이해하고 분석(inference)
- 모델을 평가하고, 다시 훈련하는 반복과정을 거쳐 성능을 향상시킴



데이터 준비 (데이터-라벨 쌍) → 데이터 정제 → 데이터 훈련 → 성능 평가 → 최종 결론

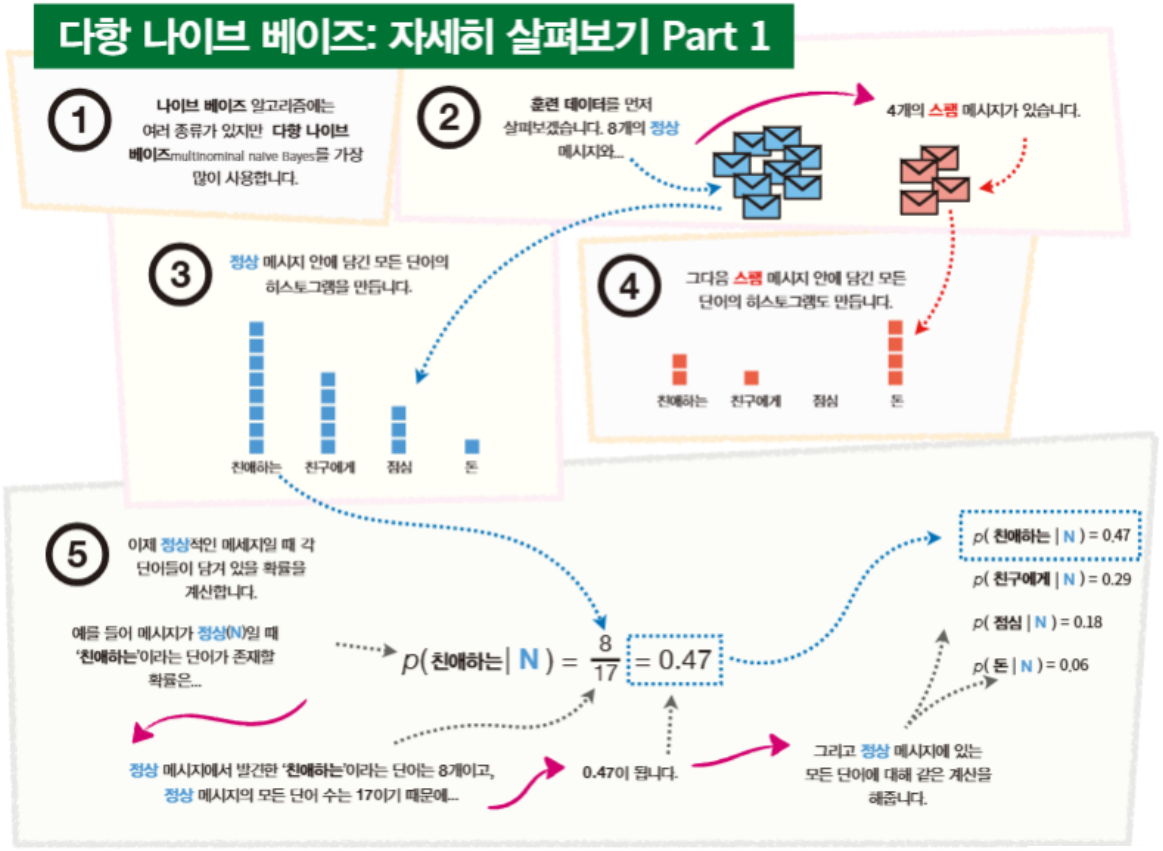
https://singaporebusinessintelligence.blogspot.com/2018/10/what-is-automated-machine-learning.html

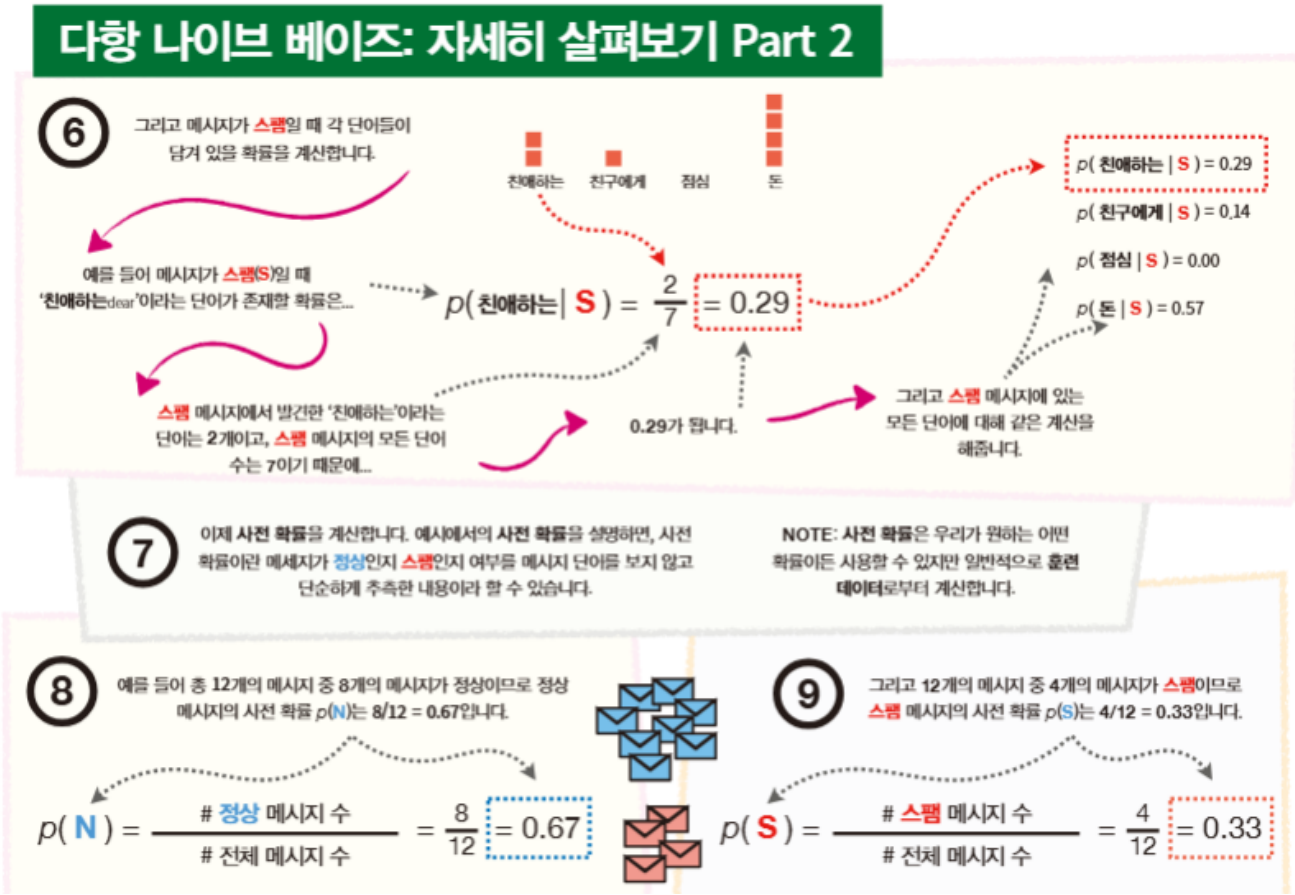
지도학습 알고리즘

	알고리즘 명	분류(classification)	회귀(regression)
1	나이브 베이즈 분류(Naïve Bayes Classification)	O	X
2	로지스틱 회귀(Logistic Regression)	O	X
3	서포트 벡터 머신(SVM)	O	O
4	K-최근접 이웃(K-nearest neighbors)	O	O
5	결정 트리(Decision Tree)	O	O
6	선형 회귀(Linear Regression)	X	O
7	Lidge, Lasso	X	O
8	ElasticNet	X	O

3. 지도학습 – 분류

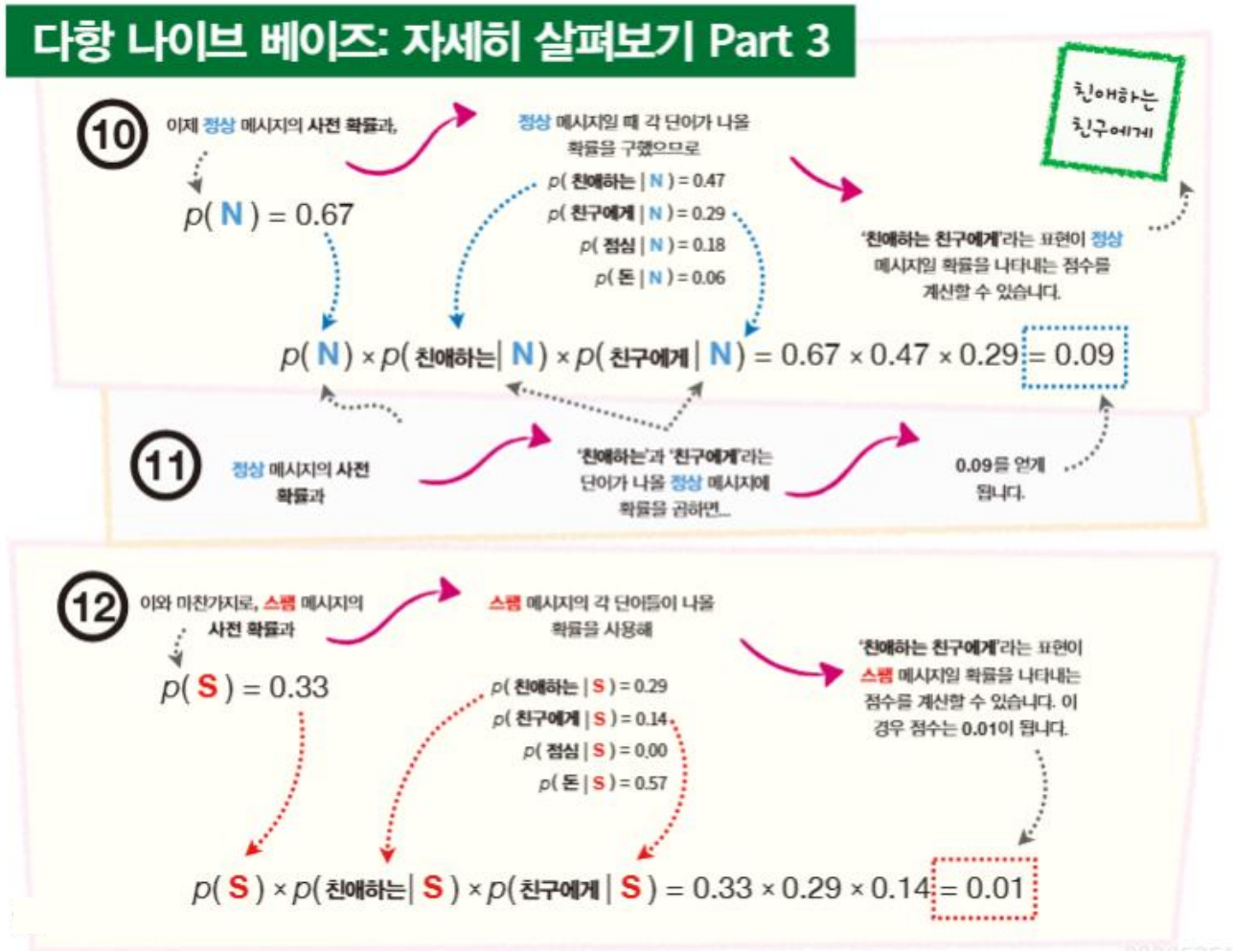
나이브 베이즈 분류기





3. 지도학습 – 분류

나이브 베이즈 분류기



다항 나이브 베이즈: 자세히 살펴보기 Part 4

13

다시 복습해봅시다. 우리의 목표는 '**친애하는 친구에게**'라는 메시지가 **정상 메시지**인지 **스팸**인지 여부를 구별하는 것입니다.

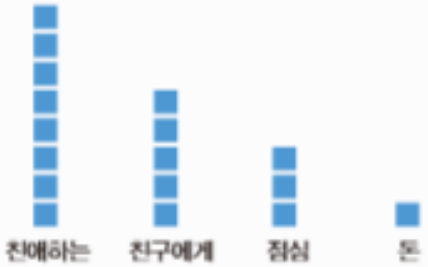
8개의 **정상** 메시지와 4개의 **스팸** 메시지가 있는 **훈련 데이터**를 사용했습니다.

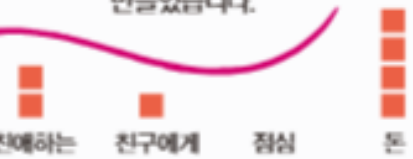
$$p(\mathbf{N}) = \frac{\#\text{정상 메시지 수}}{\#\text{ 전체 메시지 수}} = 0.67$$

$$p(\mathbf{S}) = \frac{\#\text{스팸 메시지 수}}{\#\text{ 전체 메시지 수}} = 0.33$$

먼저 **정상** 혹은 **스팸**일 수 있는 메시지에 담긴 내용을 보지 않고 추측한 확률인 **사전 확률**을 계산했습니다.

그다음 메시지에 담긴 각 단어의 히스토그램을 만들었습니다.





그리고 히스토그램을 사용해 확률을 계산했습니다.

$p(\text{친애하는} \mid \mathbf{N}) = 0.47$
$p(\text{친구에게} \mid \mathbf{N}) = 0.29$
$p(\text{점심} \mid \mathbf{N}) = 0.18$
$p(\text{돈} \mid \mathbf{N}) = 0.06$

$p(\text{친애하는} \mid \mathbf{S}) = 0.29$
$p(\text{친구에게} \mid \mathbf{S}) = 0.14$
$p(\text{점심} \mid \mathbf{S}) = 0.00$
$p(\text{돈} \mid \mathbf{S}) = 0.57$

그다음 메시지가 **정상** 혹은 **스팸**일 조건에 따라 **사전 확률**과 각 단어가 나올 확률로 '**친애하는 친구에게**'라는 메시지의 점수를 계산합니다.

$$p(\mathbf{N}) \times p(\text{친애하는} \mid \mathbf{N}) \times p(\text{친구에게} \mid \mathbf{N}) = 0.67 \times 0.47 \times 0.29 = 0.09$$

$$p(\mathbf{S}) \times p(\text{친애하는} \mid \mathbf{S}) \times p(\text{친구에게} \mid \mathbf{S}) = 0.33 \times 0.29 \times 0.14 = 0.01$$

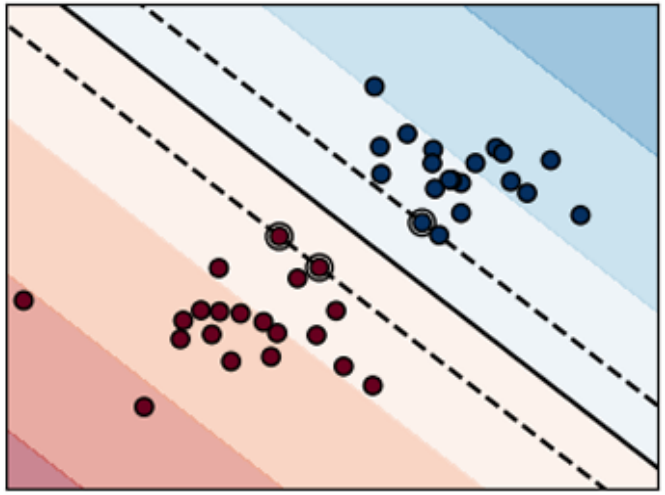
이제 '**친애하는 친구에게**'라는 메시지를 분류할 수 있게 되었습니다. 왜냐하면 **정상** 메시지일 확률, 즉 점수는 0.09로 해당 메시지가 **스팸**일 점수(0.01)보다 크기 때문에 우리는 이 메시지를 **정상** 메시지로 구분합니다.

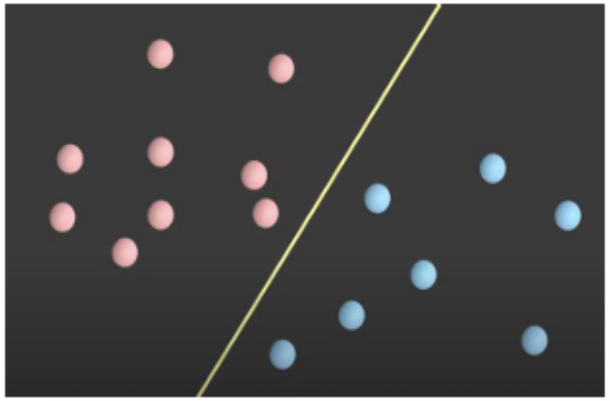
친애하는 친구에게 

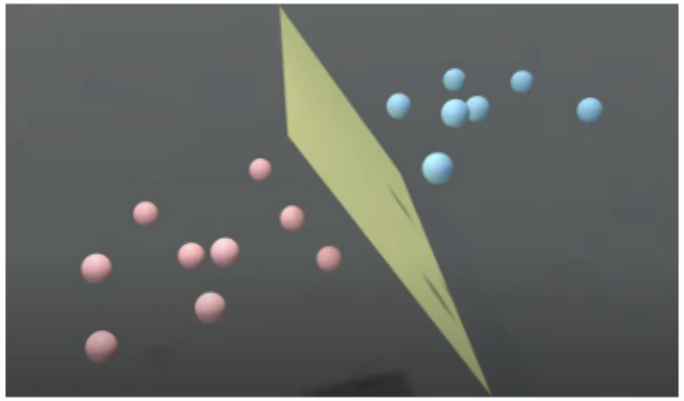
BAM!!!

3. 지도학습 – 분류

서포트 벡터(Support Vector)머신







데이터의 특징들(x, features) 중 가능한 한 먼 결정 경계를 찾아내는 것

마진(결정 경계에 가장 가까운 점과 결정 경계 사이의 거리) 의 최대화

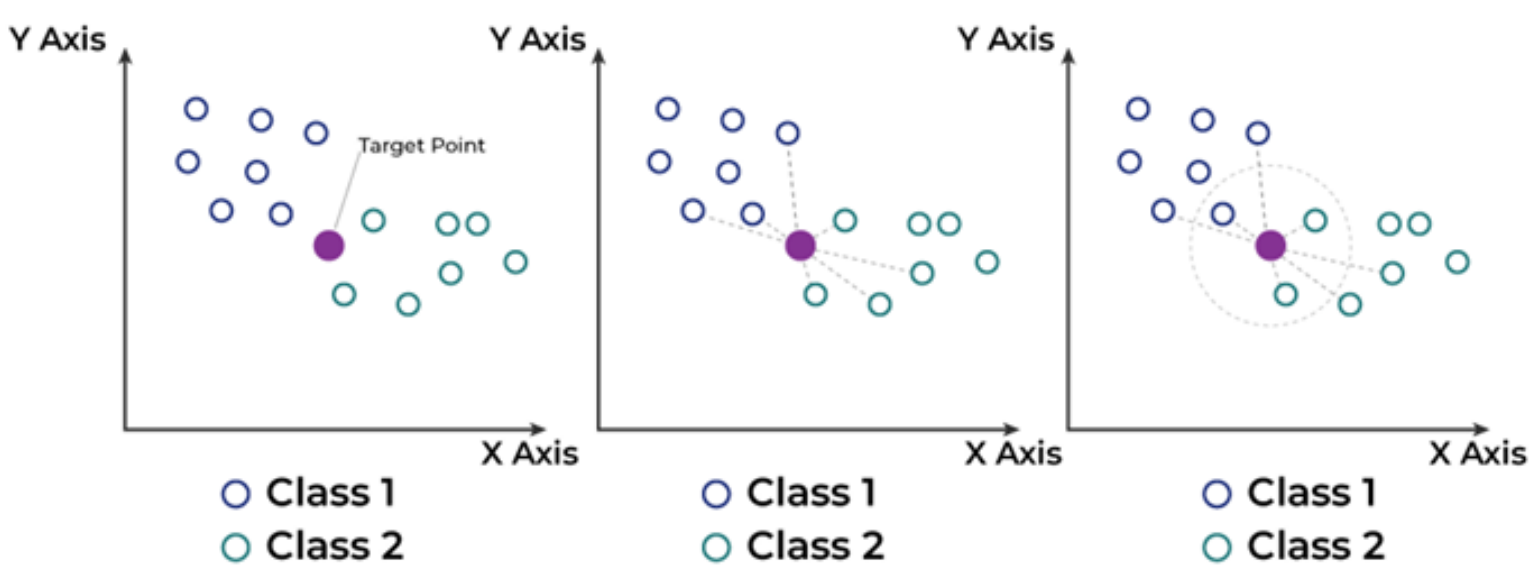
- 최적의 초평면 찾기 -> 초평면에 가장 가까운 데이터를 '서포트 벡터'로 식별 -> 서포트벡터를 기준으로 결정경계 구축
- 장점 : 일반화 능력(다양한 케이스에 적용 가능)과 높은 성능을 보여줌
- 단점 : 파라미터를 어떻게 세팅하느냐에 따라 결과의 변동성이 큼, 모델 해석이 어려움
- 주의할 점 : 소프트마진(데이터를 완전히 분류하기 어려울 때 일부 데이터가 마진 안에 포함되는 경우) 가 발생할 수 있음

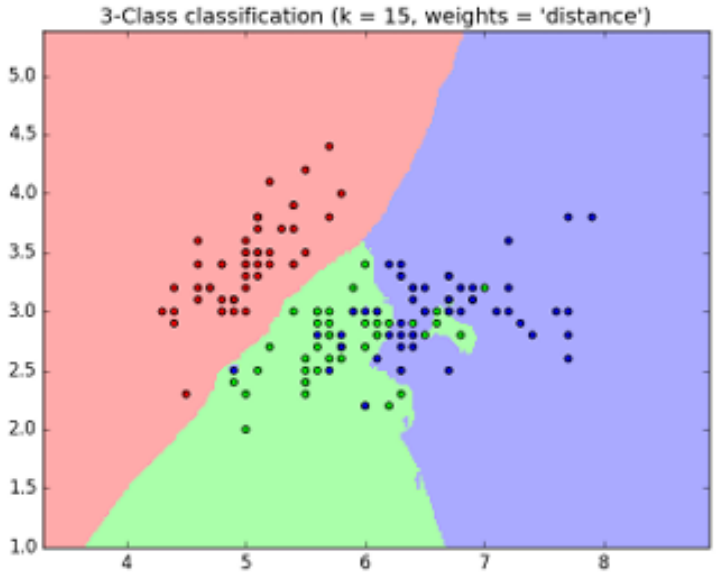
https://www.youtube.com/watch?v=ny1iZ5A8iIA

https://www.youtube.com/watch?v=_YPScrckx28

3. 지도학습 – 분류

K-최근접 이웃





주변 K개 데이터 포인트와의 거리를 계산하여, 다수결로 더 가까운 데이터로 분류/회귀

주어진 데이터에 대해 주변 k개 데이터와의 유클리드 거리를 계산

- 장점 : 구현이 쉽고, 데이터 포인트가 비교적 섞여 있는 경우에도 안정적으로 분류 가능
- 단점 : 리소스의 소모가 크고 과적합에 취약

3. 지도학습 – 분류

결정 트리

조건 분기에 따라 학습 데이터를 나누어 분류

불순도를 최소화하도록 데이터를 나누며 분류 수행

- 장점 : 직관적인 형태로 모델 이해 가능, 스케일링 불필요, 비선형 데이터에 대해 강건
- 단점 : 과적합 가능성이 높음, 데이터 변화에 민감
- 분류는 지니 계수를 기준으로, 회귀의 경우 MSE를 기준으로 노드를 나누어 분류/회귀 모두 사용
- 불순도 지표 : 지니 계수

3. 지도학습 – 분류

로지스틱 회귀

*머신러닝 도감, 아키바 신야

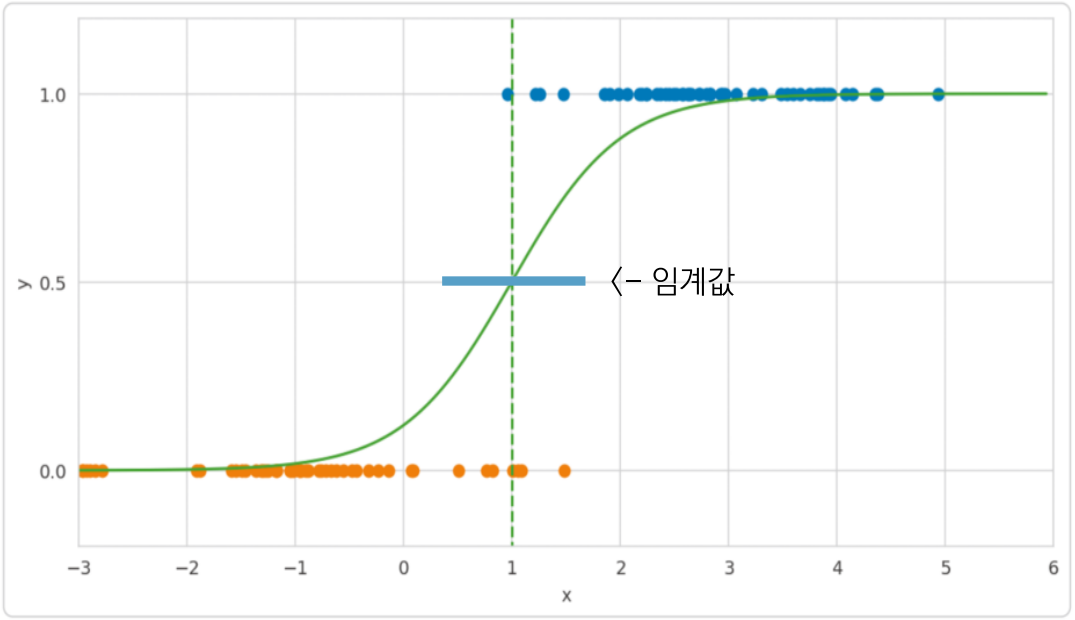


그림 2-13 로지스틱 회귀

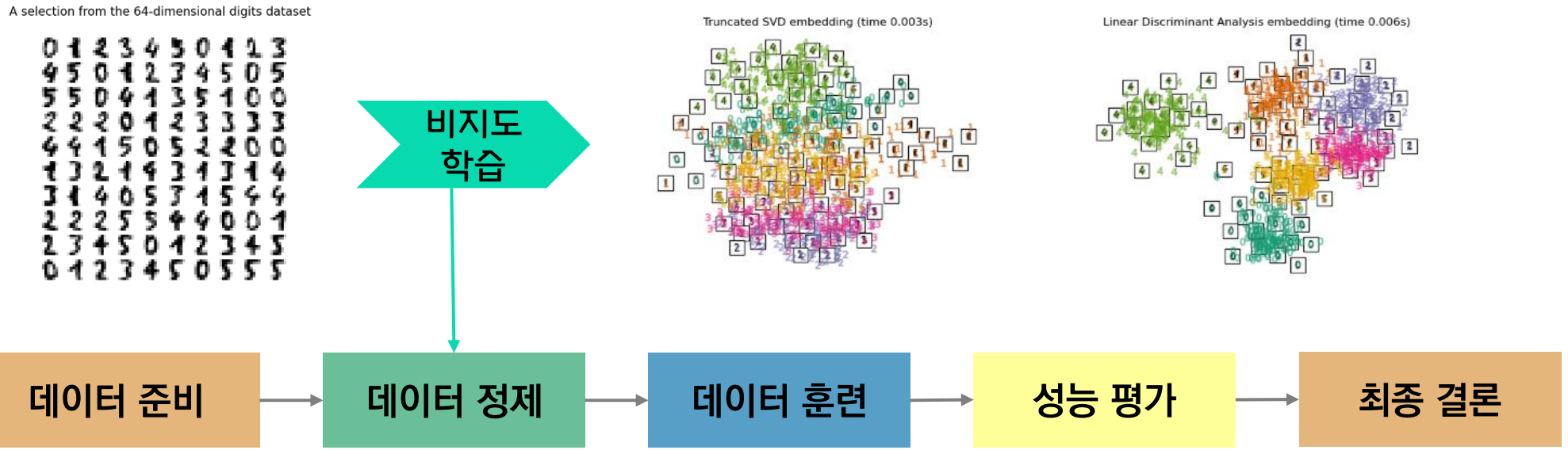
- HOW ? : y=ax+b의 그래프를 만든 뒤, 결과값인 y에 시그모이드 함수를 적용하여 확률값(0~1사이)로 변환한다.
- 특정한 임계값(보통 0.5)를 기준으로, 클래스 0에 속할지 1에 속할지를 계산하게 한다.
- *이를 응용하여 다중 분류를 수행하는 경우, 각 클래스별로 y=ax+b의 값을 소프트맥스 함수로 감싸 나온 확률 중 가장 높은 값을 반환하게 한다.
- 장점 : 결과값을 해석하기 쉬우며, 구현과 학습이 간단함. 임계값을 조정하여 결과를 수정할 수 있음
- 단점 : 비선형 관계인 데이터에는 잘 작동하지 않고, 이상치에 민감함. 고차원 데이터에서는 성능에 한계가 있음

2. 지도학습과 비지도학습 – 일반적인 머신 러닝의 순서

1. 데이터 불러오기
2. 데이터 확인하기(통계적 특징, 데이터의 크기 등)
3. 데이터 전처리(결측치 및 이상값 정리, 스케일링 등)
4. train_test_split 및 x, y 데이터의 정의
5. 머신러닝 모델 정의 및 훈련, 검증

2. 지도학습과 비지도학습 – 비지도학습의 경우

- 데이터의 특징을 활용하여 군집화/차원축소
- 분석가는 군집화/차원축소의 결과물을 활용하여 데이터를 분석하거나, 지도학습을 수행
- 목표와 일치하지 않는 군집화/차원축소인 경우, 다른 모델을 쓰거나 재군집화/차원축소



https://singaporebusinessintelligence.blogspot.com/2018/10/what-is-automated-machine-learning.html

4. 지도학습 – 회귀

선형회귀(Linear Regression)

SSE	$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2$
MSE	$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
RMSE	$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \check{y}_j)^2}$
MAE	$MAE = \frac{\sum_{i=1}^{n} \lvert y_{pred,i} - y_i \rvert}{n}$

*머신러닝 도감, 아키바 신야

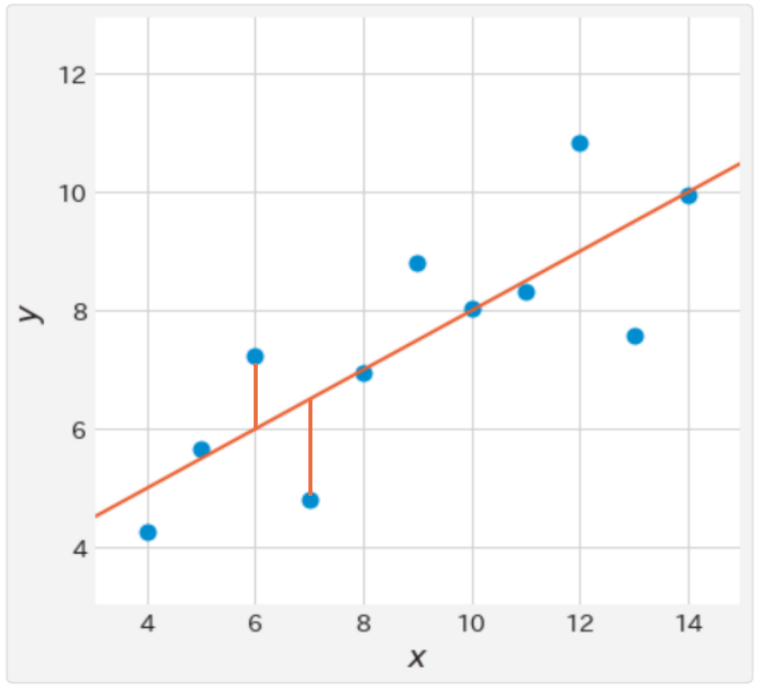
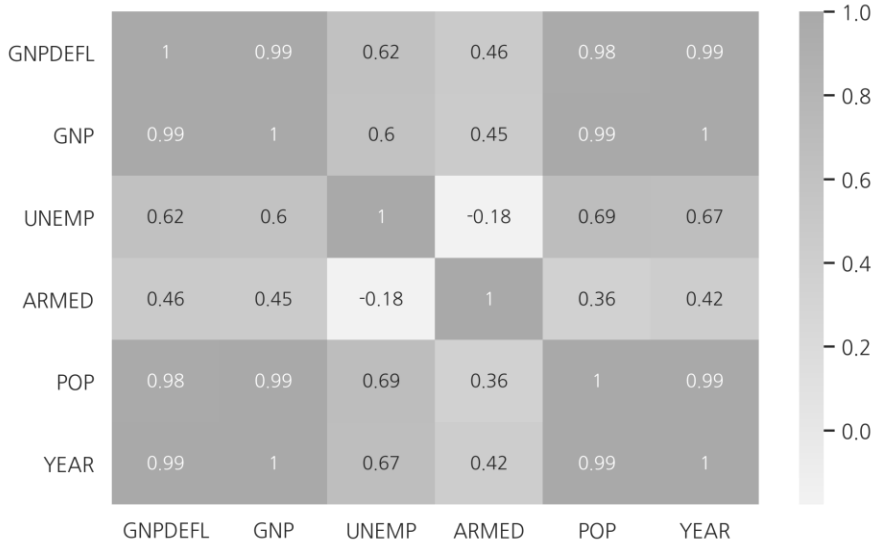


그림 2-1 선형회귀 예

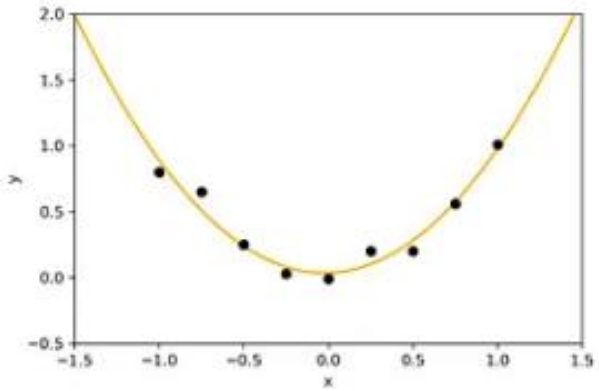


0 ~ 0.3	약한 상관관계
0.3 ~ 0.7	중간 상관관계
0.7 ~ 1	강한 상관관계

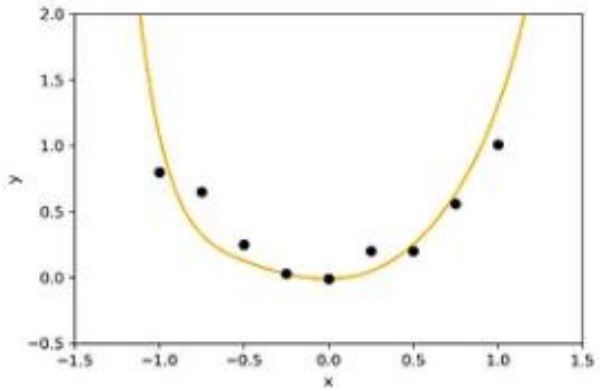
- HOW ? : 데이터(x)를 기반으로 y=ax+b 형태의 직선을 만들어 예측을 수행한다.
- 장점 : 결과값을 해석하기 쉬우며, 구현과 학습이 간단함. 데이터의 수가 적을 때에도 잘 작동한다.
- 단점 : 비선형 관계인 데이터에는 잘 작동하지 않고, 이상치에 민감함. 고차원 데이터에서는 성능에 한계가 있으며
- x 사이에 관계가 있을 때에는 '다중 공선성' 문제가 발생할 수 있다.

참고 : https://shorturl.at/prNPY

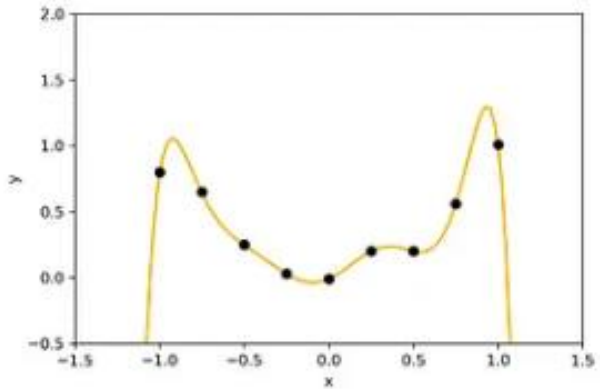
정규화



(a) 2nd-degree polynomial



(b) 8th-degree polynomial



(c) Another 8th-degree polynomial

Figure a: $\hat{y} = 0.04 + 0.04x + 0.9x^2$
Figure b: $\hat{y} = -0.01 + 0.01x + 0.8x^2 + 0.5x^3 - 0.1x^4 - 0.1x^5 + 0.3x^6 - 0.3x^7 + 0.2x^8$
Figure c: $\hat{y} = -0.01 + 0.57x + 2.67x^2 - 4.08x^3 - 12.25x^4 + 7.41x^5 + 24.87x^6 - 3.79x^7 - 14.38x^8$

(a) #params = 3	(b) #params = 9	(c) #params = 9
MSE = 0.006	MSE = 0.035	MSE = 0
L2 norm = 0.90	L2 norm = 1.06	L2 norm = 32.69
L1 norm = 0.98	L1 norm = 2.32	L1 norm = 70.03

L1, L2의 차이에 대해 알기 : https://medium.com/data-science/visualizing-regularization-and-the-l1-and-l2-norms-d962aa769932

4. 지도학습 – 회귀

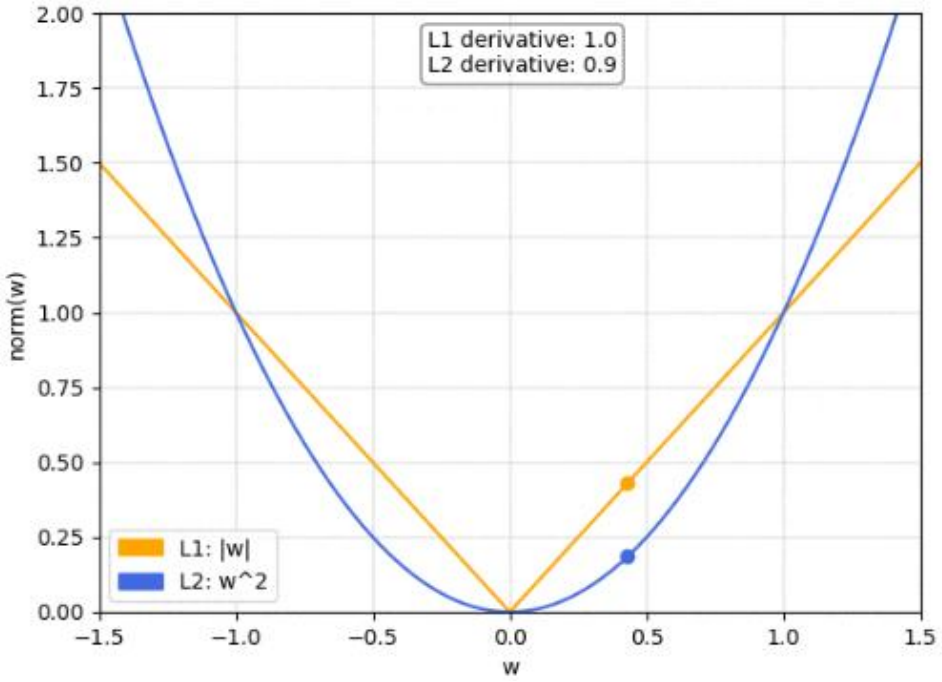
라쏘(Lasso) – L1 정규화

정규화(Normalization)란 데이터의 수치를 일정한 범위에 맞추어 변환해 주는 것.

$\beta = [2.0, -1.5, 0.0, 4.0]$

$$L_{lasso}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i^T\hat{\beta})^2 + \lambda\sum_{j=1}^{m}|\hat{\beta}_j|$$

람다가 크면	더 많은 계수를 0으로 만듦 예측보다는 희소성에 집중
람다가 0이면	일반 선형회귀와 동일함



- HOW ? : 선형회귀 모델에 L1 정규화 항을 추가하여, 불필요한 a를 0으로 만들어 더 나은 추론을 하게 한다.
- 장점 : 특성(feature)이 많은 데이터, 해석이 필요한 모델에 유용하게 사용 가능함
- 단점 : 비선형 모델에서 약함, 정보의 손실이 발생함(특성을 지움), 적절한 정규화 강도를 선택하기 어려울 수 있음

4. 지도학습 – 회귀

릿지(Lidge) – L2 정규화

$$\beta = [2.0,\ -1.5,\ 0.0,\ 4.0]$$

$$\beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 = 2.0^2 + (-1.5)^2 + 0 + 4.0^2 = 4 + 2.25 + 0 + 16 = 22.25$$

$$SSE_{L_2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{P} {\beta_j}^2$$

- HOW ? : 선형 회귀 모델에 L2 정규화 항을 추가하여, 가중치 a의 크기를 작게(0에 가깝게) 유지하게 하여 더 나은 추론을 하게 한다.
- 장점 : 모든 특성(feature)를 유지하며 과적합을 줄일 수 있고, 해가 항상 존재하고, 고차원 데이터에서 잘 작동함
- 단점 : 중요한 변수와 덜 중요한 변수를 모두 남기기 때문에 해석력이 떨어질 수 있고, L1보다는 복잡한 모델임

4. 지도학습 – 회귀

엘라스틱 넷(Elastic Net)

라쏘(L1)

릿지(L2)

(1 - a) 와 a를 더하면 = 1입니다. L1과 L2의 비중 조절을 위해 a를 사용한 것

$$\frac{\sum_{i=1}^{n}(y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

깔끔한 미분을 위해 ½를 붙여주는 것(수학적 관례)

$$L = \frac{1}{2}(y - \hat{y})^2 \Rightarrow \frac{dL}{d\hat{y}} = \hat{y} - y$$

- HOW ? : 선형 회귀 모델에 L1, L2 정규화 항을 동시에 추가하여, 불필요한 가중치 a는 0으로 만들고,
- 나머지 가중치의 크기는 작게 유지하여 희소성과 안정성을 동시에 상승시킨다.
- 장점 : 특성을 선택하고(L1), 안정화 하는(L2) 과정을 동시에 적용 가능함. 다중공선성이 높고, feature 수 〉 데이터 수 인 경우에 강력함
- 단점 : 하이퍼 파라미터가 2개라서 튜닝이 복잡하며, 직관적 해석이 어려울 수 있음

엘라스틱 넷 참고 : https://www.geeksforgeeks.org/elastic-net-regression-in-r-programming/