

---

# 인공지능과 신약개발을 위한 파이썬

5주차 머신 러닝의 이해

홍 성 은

[sungkenh@gmail.com](mailto:sungkenh@gmail.com)

# 목차

---

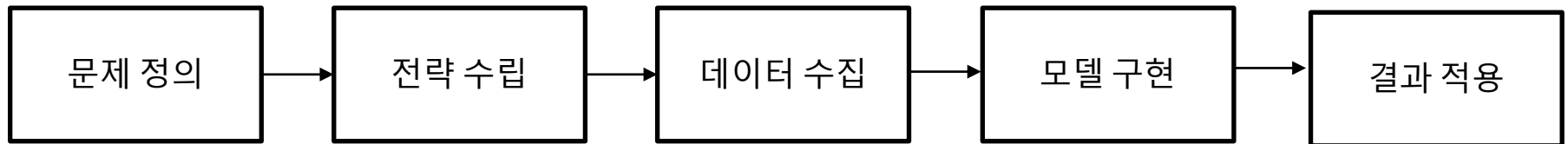
- 데이터 분석
- 머신러닝
- 회귀분석
- 분류
- 클러스터링

# 데이터 분석

---

- 데이터 분석 프로세스

- 문제 정의 : 해결하려는 문제를 명확히 정의하는 것
- 전략 수립 : 문제 해결을 위해 어떤 데이터를 어떻게 사용할지를 정함
- 데이터 수집 : 머신러닝에 필요한 데이터를 수집하는 것
- 모델 구현 : 분류, 회귀, 설명, 추천 등을 위한 머신러닝 모델을 구현
- 결과 적용 : 머신러닝 모델을 실제 상황에 적용하고 성능을 개선하는 것



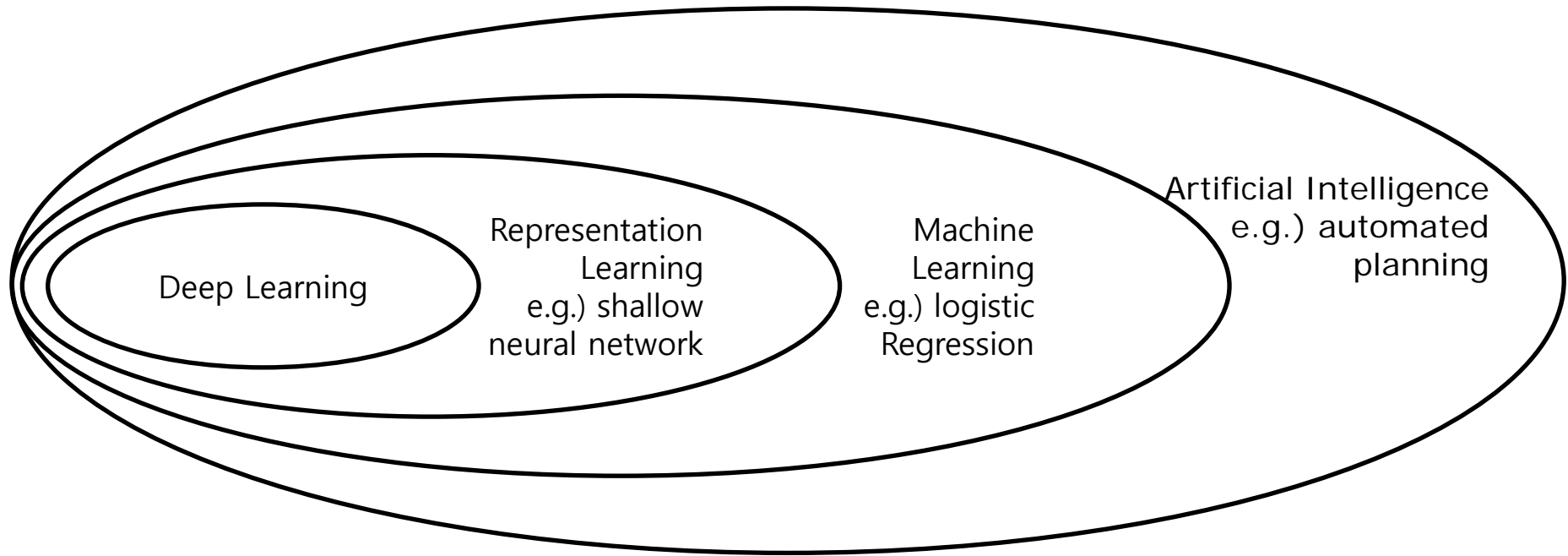
# 데이터 분석

- 데이터 수집
  - 전체 과정에서 70~80%의 시간을 소모함
  - 핵심 데이터를 확보했는지 여부
  - 데이터 품질
  - 잘못된 데이터 사용은 잘못된 결과를 도출
  - 수집 가능 여부 (보유 기관의 데이터 정책)
  - 수집 주기 (일회성, 한시간/하루/한달에 한번 등)
  - 비용 (무료 또는 유상, 통신 비용 등)
  - 데이터 포맷 변경, 호환성, 처리 비용
  - 정답 데이터 셋 확보 여부
  - 수집 아이디어...



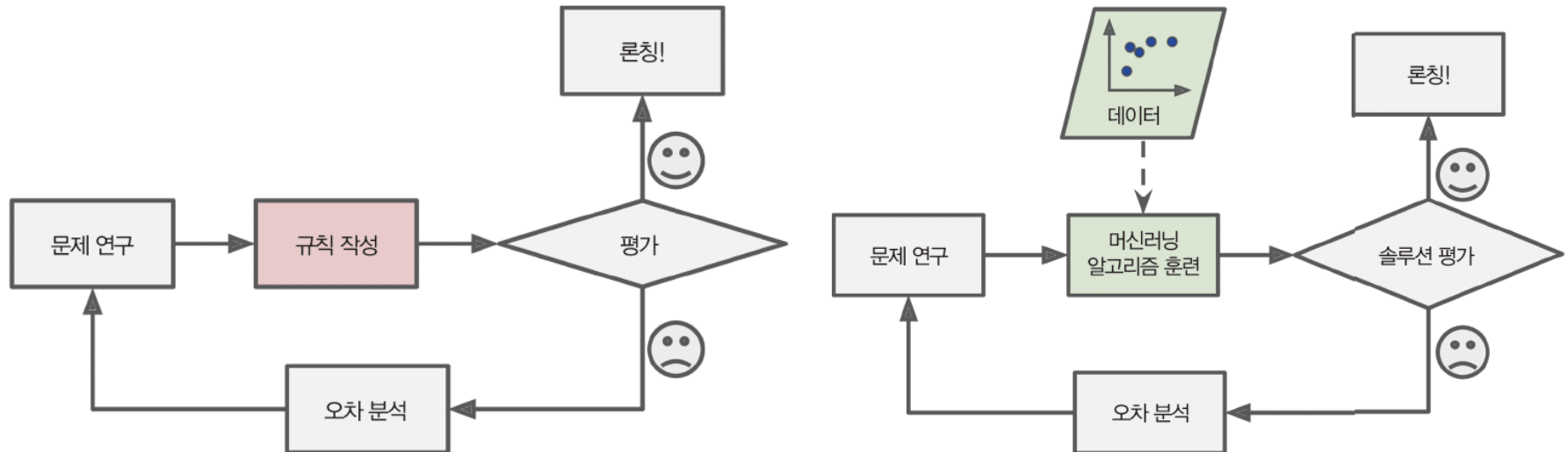
# 머신 러닝

---



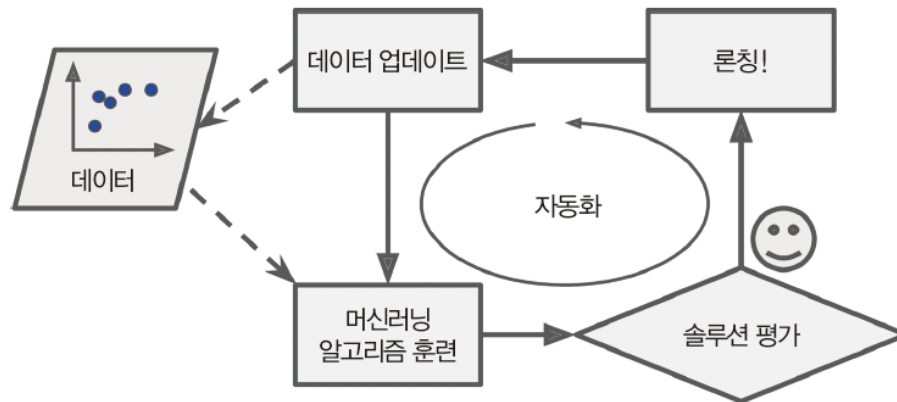
# 머신 러닝

- 머신러닝은 데이터에서부터 학습하도록 컴퓨터를 프로그래밍하는 과학
- 명시적 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구분야
- 딥러닝: 신경망을 기반으로 하는 머신러닝 기술
  - 마치 사람이 많은 정보에 접하면서 학습하듯이 컴퓨터도 데이터를 보고 학습하는 방법
  - 음성인식, 자동차 번호판 인식, 언어 번역, 채팅 대화, 글쓰기, 작곡 등 여러 분야에서 좋은 성과를 냄
- 스팸 필터
  - 스팸메일과 일반 메일의 샘플을 이용해 스팸 메일 구분법을 학습하는 머신러닝 프로그램
  - 시스템이 학습하는데 사용하는 데이터 샘플을 **훈련 데이터**, 각 훈련 데이터를 **훈련 사례(샘플)**
  - 정확히 분류된 메일의 비율의 성능을 **정확도**라고 함



# 머신 러닝

- 머신 러닝 작업 흐름



- 머신 러닝이 유용한 문제

- 음성인식

- One, Two를 구분하는 프로그램을 작성한다고 했을 경우
    - Two는 높은 피치의 T로 시작하기에 높은 피치의 강도를 측정하는 알고리즘을 하드코딩해서 두개를 구분할 수 있음
    - 소음이 있는 환경에서 수백만 명이 말하는 여러 언어로 된 수천개의 단어를 구분하는 것으로 확장하기 어려움
    - 각 단어를 녹음한 샘플을 사용해 스스로 학습하는 알고리즘을 작성하는 것이 가장 좋은 솔루션

# 머신 러닝

---

- 머신 러닝 해석

- 머신 러닝이 학습한 것을 조사할 수 있음
- 스팸 필터가 충분한 스팸 메일로 훈련되었다면 스팸을 예측하는데 가장 좋은 단어 및 단어의 조합을 확인할 수 있음
- 간혹, 예상치 못한 연관성이나 새로운 추세가 발견되기도 해서 해당 문제를 더 잘 이해하도록 도와 줌
- 머신 러닝 기술을 적용해서 대용량의 데이터를 분석하면 **겉으로 보이지 않는 패턴을 발견**할 수 있도록 해주는데 이를 **데이터 마이닝**이라고 함

- 머신 러닝 응용 분야

- 제품 이미지를 보고 자동으로 분류하기
- 자동으로 뉴스 기사를 분류하기
- 내년도 회사의 수익을 예측하기
- 음성을 듣고 이해하는 앱을 만들기
- 구매 이력을 기반으로 고객을 나누기



# 머신 러닝

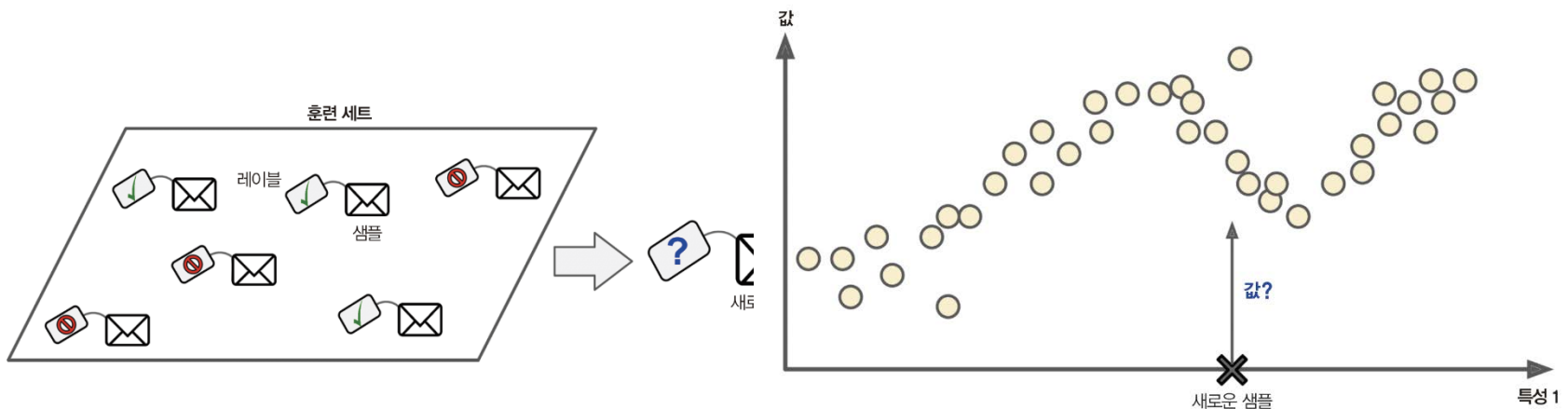
---

- 머신 러닝의 유형
  - 사람의 감독하에 훈련하는 것인지(지도, 비지도, 준지도, 강화학습)
  - 실시간으로 점진적인 학습을 하는지 아닌지(온라인 학습과 배치학습)
  - 단순히 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 훈련 데이터셋의 패턴을 발견하여 예측 모델을 만드는 것인지(사례 기반 학습, 모델 기반 학습)

# 머신 러닝

- 지도학습

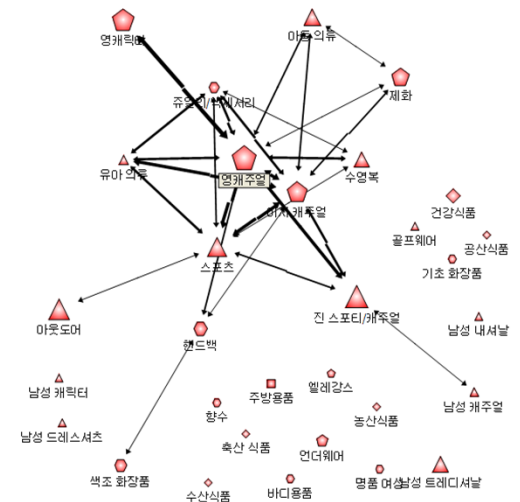
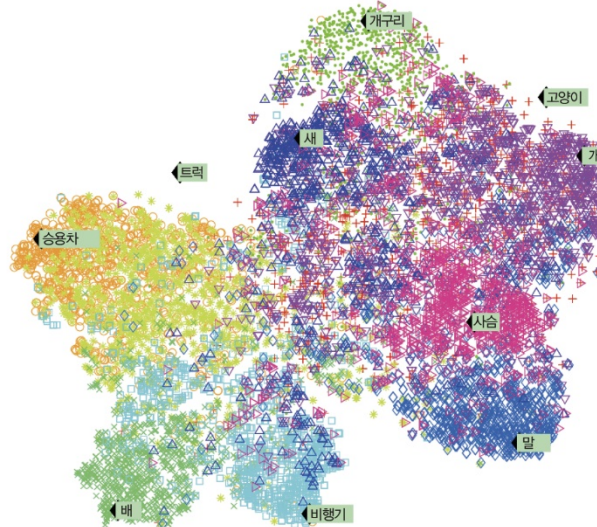
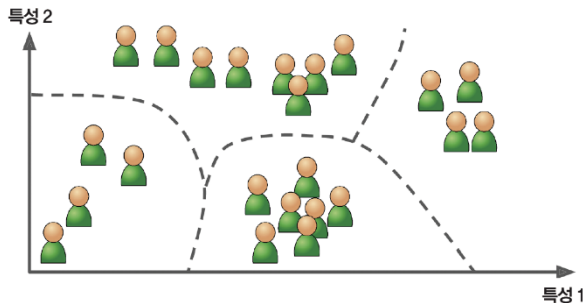
- 지도학습은 정답이 주어지고 정답을 예측하는데 사용
- 정답은 목적(target) 변수, 레이블이라고도 함
- 예측은 분류와 회귀로 나누어짐
- 분류
  - 분류(classification)란 어떤 항목(item)이 어느 그룹에 속하는지를 판별하는 기능을 말함
  - 두 가지 카테고리를 나누는 작업을 이진 분류(binary classification)라고 하고 세 개 이상의 클래스를 나누는 작업을 다중 분류(multiclass classification)라고 함
- 회귀
  - 수치를 예측하는 것을 회귀라고 한다.



# 머신 러닝

## • 비지도 학습

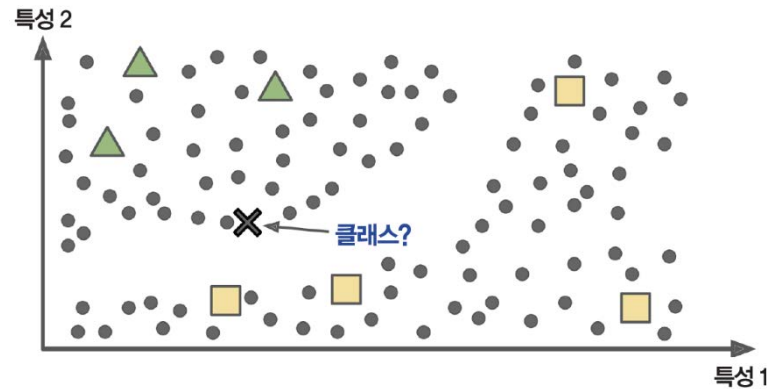
- 비지도 학습이란 정답이 없이 데이터로부터 중요한 의미를 찾아내는 머신 러닝 기법임
  - 군집화: 유사한 항목들을 같은 그룹으로 묶음
  - 차원 축소 및 시각화 : 머신 러닝에 사용할 특성의 수를 줄임
  - 연관 분석
    - 어떤 사건이 다른 사건과 얼마나 자주 동시에 발생하는지 파악
    - 자주 발생하는 패턴 찾기(상품의 연관성, 취향의 연관성 등 분석)
    - 같이 구매한 상품 분석(market basket analysis, 장바구니 분석)
    - 상품의 진열 배치 및 상품 프로모션(쿠폰 발행 등)에 활용



# 머신 러닝

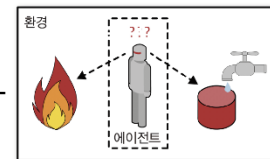
- 준지도 학습

- 데이터에 레이블을 다는 것이 시간과 비용이 많이 필요하기 때문에 레이블이 없는 샘플이 많고 레이블이 있는 샘플이 적음
- 정답이 일부만 있는 경우를 준지도 학습이라고 함



- 강화 학습

- 학습하는 시스템을 **에이전트**
- 환경을 관찰해서 행동을 실행하고 그 결과로 **보상, 벌점**부과
- 시간이 지나며 가장 큰 보상을 얻기 위해 **정책**이라는 전략을 스스로 학습



- 1 관찰
- 2 정책에 따라 행동을 선택



- 3 행동 실행!
- 4 보상이나 벌점을 받음



- 5 정책 수정(학습 단계)
- 6 최적의 정책을 찾을 때까지 반복

# 머신 러닝

---

- 온라인 학습
  - 데이터를 순차적으로 한 개 또는 **미니배치**라 부르는 작은 묶음 단위로 주입하여 시스템을 훈련
  - 매 학습 단계가 빠르고 비용이 적게 들어 데이터가 도착하는대로 바로 학습 가능
  - 연속적으로 데이터를 받고 빠른 변화에 스스로 적응해야 하는 시스템에 적합
  - 아주 큰 데이터를 학습하는 시스템에도 온라인 학습 알고리즘 사용 가능(데이터의 일부를 입력으로 학습하는 것을 반복)
  - **학습률**: 변화하는 데이터에 얼마나 빠르게 적응할 것인지
- 배치 학습(오프라인 학습)
  - 시스템이 점진적으로 학습할 수 없음
  - 가용한 데이터를 모두 사용해야 하므로 오프라인으로 수행
  - 시스템을 훈련시키고 제품 시스템에 적용하면 더 이상의 학습은 없음

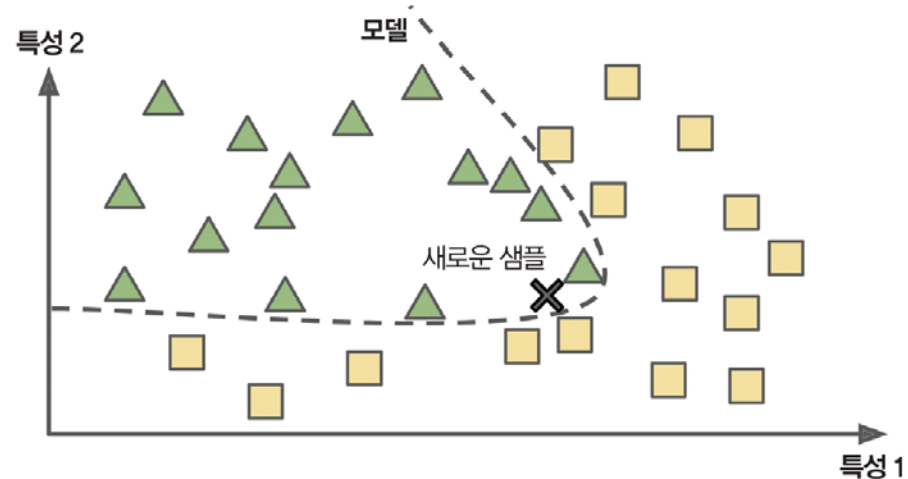
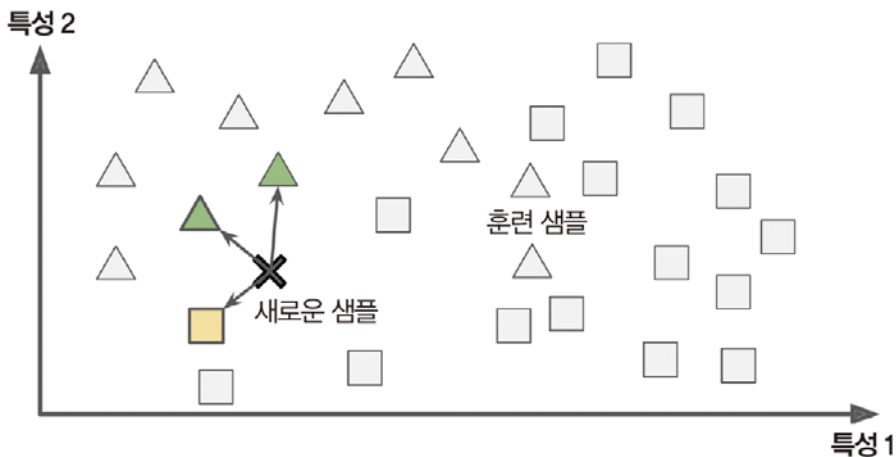
# 머신 러닝

- 사례 기반 학습

- 스팸 메일과 유사한 메일을 구분하도록 스팸 필터를 프로그래밍할 수 있음
- 두 메일 사이의 **유사도를 측정**(공통으로 포함된 단어의 수를 세는 것)
- 시스템이 훈련 샘플을 기억함으로써 학습하는 방식

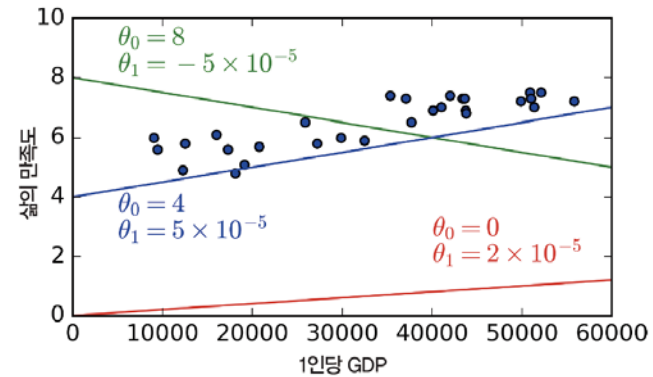
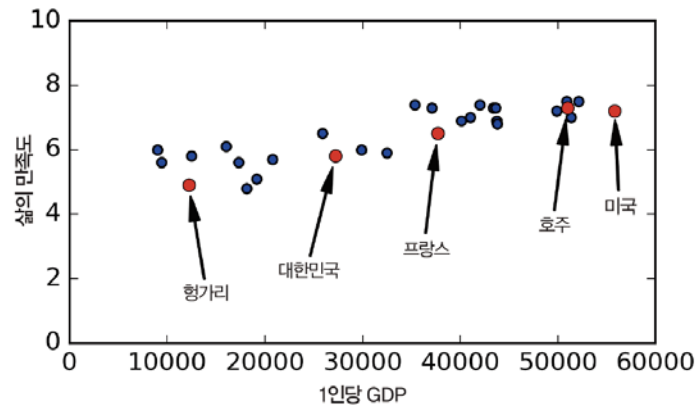
- 모델 기반 학습

- 샘플로부터 일반화시키는 다른 방법은 샘플들의 모델을 만들어 예측에 사용하는 것



# 머신 러닝

- 모델기반 학습 사례



$$\text{삶의\_만족도} = \theta_0 + \theta_1 \times \text{1인당\_GDP}$$

모델 파라미터

# 머신 러닝

---

- 머신 러닝 알고리즘

항목	머신 러닝 유형	알고리즘
지도학습	분류	kNN, 베이즈, 결정 트리, 랜덤포레스트, 로지스틱회귀, 그라디언트부스팅, 신경망
	회귀	선형회귀분석, SVM(서포트 벡터 머신), 신경망
비지도학습	군집화	k-means, DBSCAN, 계층적군집분석(HCA), 이상치 탐지와 특이치 탐지
	시각화와 차원축소	PCA, 시각화(T-sne), 지역적 선형 임베딩(LLE)
	연관 규칙	Aprori, Eclat



# 머신 러닝

- 머신 러닝 동작

- 머신 러닝은 모델(model)을 사용
  - 스팸 메일을 찾아내는 모델,
  - 누가 게임에서 이길지 예측하는 모델,
  - 내일 날씨를 예측하는 모델
- 과학에서는 어떤 현상을 설명하는 모델로 수식을 주로 사용
  - 모든 질량을 가진 모든 물체는 서로 끌어당긴다는 만유인력 법칙은 두 물체의 질량에 각각 비례하고, 두 물체의 거리의 자승에 반비례하는 수식으로 표현
  - 머신 러닝, AI 모델은 데이터 기반의 모델을 사용함

- 모델의 가치

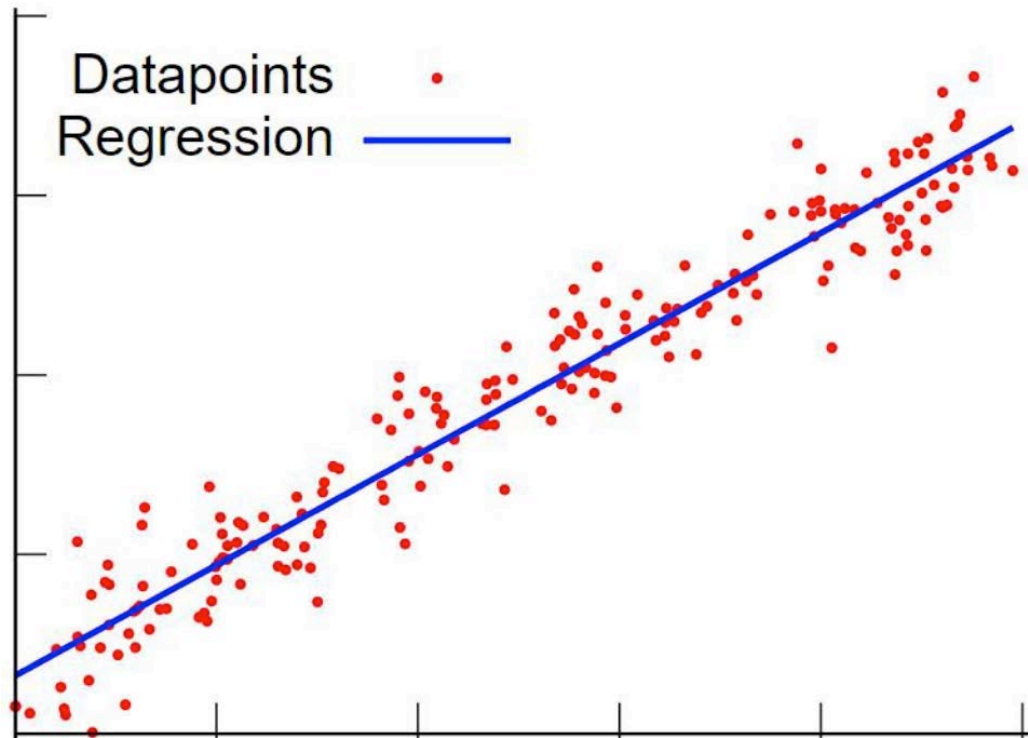
- 와인 품질 =  $12.145 + (0.00117 \times \text{겨울철 강수량})$   
 $+ (0.064 \times \text{재배철 평균기온}) - (0.00386 \times \text{수확기 강수량})$



# 머신 러닝

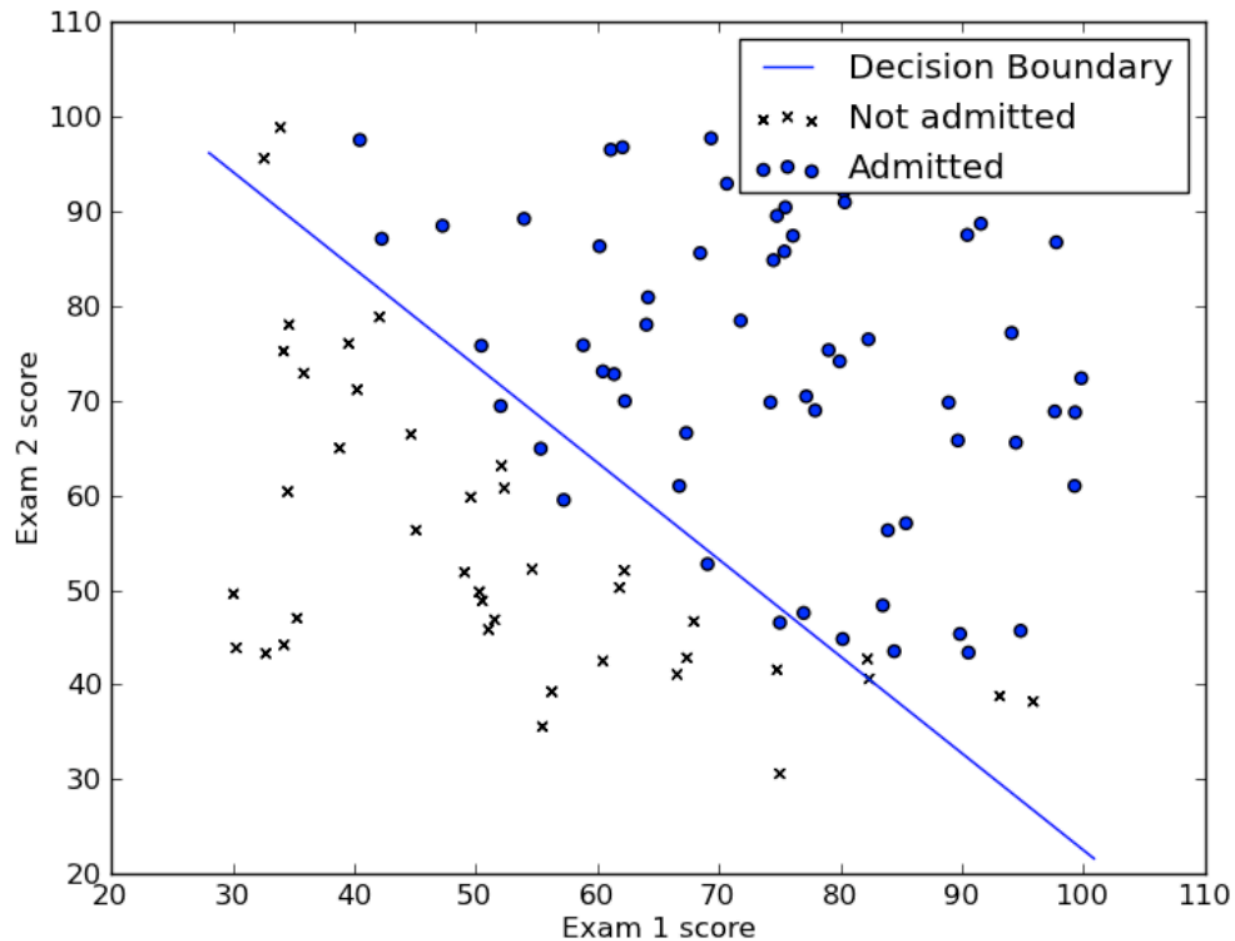
---

- 선형 회귀(regression)  $y = wX + b$



# 머신 러닝

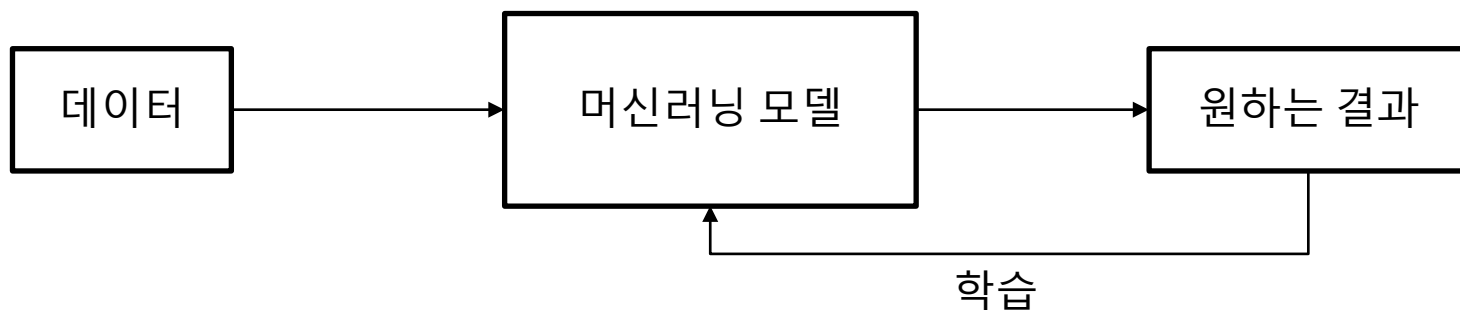
- 선형 분류(classification)  $ay + bx > c$



# 머신 러닝

---

- 모델의 특징
  - 머신 러닝에서는 데이터에 기반한 모델을 사용 (학습)
  - 현실 세계의 많은 현상은 수식으로 간단히 모델링하기 어렵고 과학적으로 증명할 수는 없음
- 모델 구조와 파라미터
  - 모델 구조: 모델의 동작을 규정하는 방법
  - 모델 파라미터: 모델이 잘 동작하도록 정한 가중치 등 계수
    - 예: 머리카락 길이
    - 모델의 구조는 프로그래머가 선택
    - 적절한 파라미터를 찾는 것은 머신 러닝 프로그램이 학습하여 찾음



# 머신 러닝

---

- 손실함수(비용 함수)

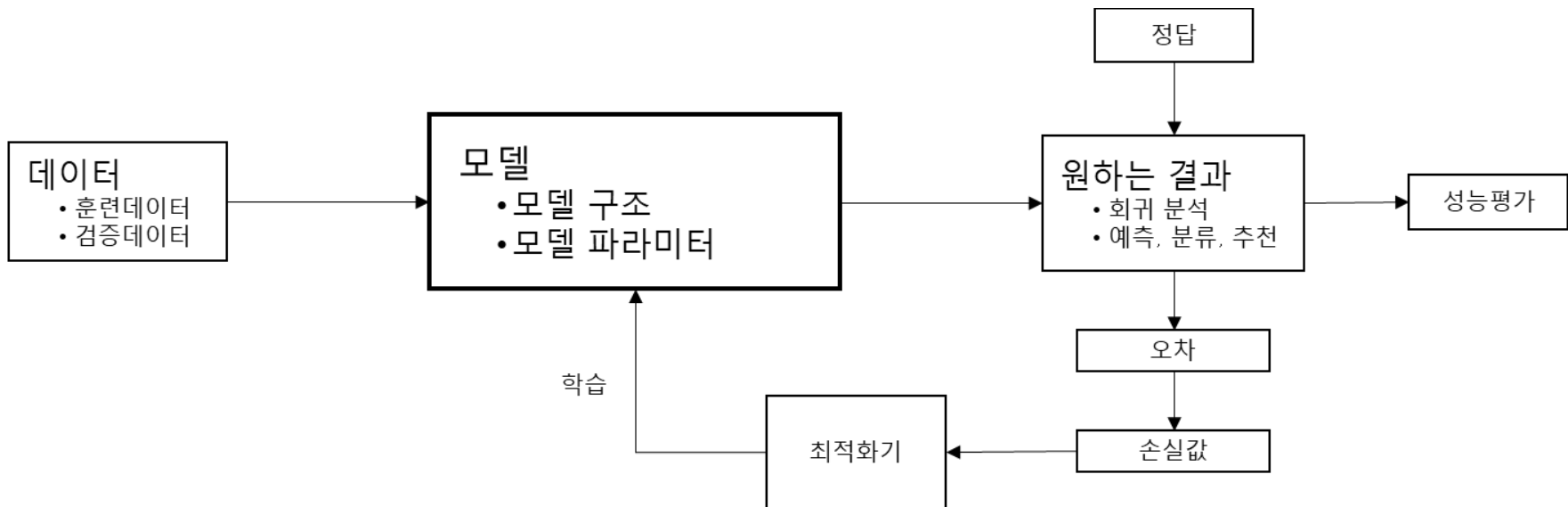
- 모델의 예측 값과 실제 값과의 차이, 즉 오차로부터 손실함수(loss function)을 계산함
- 이 손실함수를 줄이는 방향으로 모델을 최적화 (학습) 함
- 회귀분석에서 많이 사용하는 손실함수로는 오차 자승의 합의 평균치(MSE: mean square error)

$$MSE = \sum_{k=1}^N (y - \hat{y})^2$$

- N: 배치 크기
- 배치 크기 같은 설정 환경 변수를 하이퍼파라미터라고 함
- 하이퍼파라미터는 사람이 선택하는 변수이며, 기계 학습으로 자동으로 갱신되는 변수는 "파라미터"라고 함

# 머신 러닝

- 오차 손실함수, 최적화, 파라미터



# 머신 러닝

---

- 분류의 손실 함수
  - 분류에서는 손실함수로 MSE를 사용할 수 없음
  - 대신, 분류에서 정확도(accuracy)를 손실함수로 사용할 수 있음
  - 예를 들어 100명에 대해 남녀 분류를 시도하였으나 96명을 맞추고 4명을 틀렸다면 정확도는 0.96
  - 그러나 정확도를 손실함수로 사용하는데 다음과 같은 문제가 있음
- 카테고리 분포 불균형시 문제
  - 남자가 95명, 여자가 5명이 있는 그룹에서 남자는 1명을 잘 못 분류하고 여자는 3명을 잘 못 분류했다고 하면, 정확도는 여전히 0.96임
  - 손실을 제대로 측정하지 못함
  - 이를 보완하기 위해서 크로스 엔트로피(cross entropy)를 사용함

# 머신 러닝

---

- 크로스 엔트로피

$$CE = \sum_i p_i \log\left(\frac{1}{p_i'}\right)$$

- $p_i$ 는 어떤 사건이 일어날 실제 확률이고,  $p_i'$ 는 예측한 확률이다
- 남녀가 50명씩 같은 경우

$$CE = -0.5 \times \log\left(\frac{49}{50}\right) - 0.5 \times \log\left(\frac{47}{50}\right) = 0.02687$$

- 남자가 95명 여자가 5명인 경우

$$CE = -0.95 \times \log\left(\frac{94}{95}\right) - 0.05 \times \log\left(\frac{2}{5}\right) = 0.17609$$



# 머신 러닝

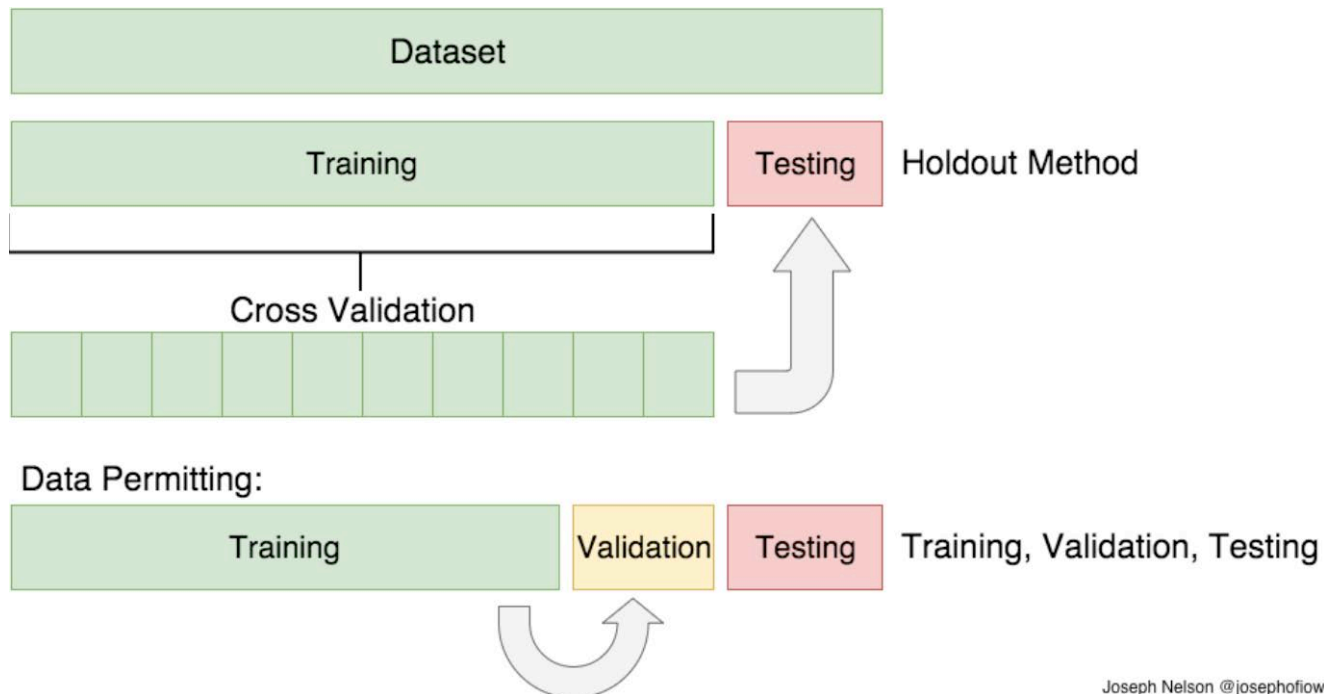
---

- 훈련과 검증
  - 모델이 데이터를 이용하여 학습하는 과정을 훈련 (training)이라고 함
  - 최적화 알고리즘에 의해서 파라미터(가중치 등)를 계속 갱신하여 모델의 예측 값이 실제 값에 수렴하도록 하는 것
  - 검증(validation) : 훈련된 모델이 잘 동작하는지 확인하는 과정
- 모델 동작이 얼마나 우수한지를 검증할 때는 훈련 데이터로해서는 안되며 훈련에 사용하지 않은, 새로운 검증 데이터(validation data)를 사용해야 함
- 보통 검증 데이터를 따로 제공하지 않으므로 훈련에 사용할 데이터의 일부를 검증용으로 미리 확보해야 함
- 훈련에 사용하지 않고 남겨 두었다가 모델이 제대로 동작하는지 테스트할 때 사용하는 데이터를 hold-out 데이터라고 함

# 머신 러닝

- 훈련과 검증

- 훈련 데이터 : 모델 파라미터를 훈련하는데 사용
- 검증 데이터 : 과대적합이나 과소적합을 검사하고 최적 모델 구조(하이퍼파라미터 등)를 찾는데 사용
- 테스트 데이터 : 모델의 성능을 최종적으로 테스트 하는데 사용



# 머신 러닝

- K-fold 교차 검증

주어진 훈련데이터



# 머신 러닝

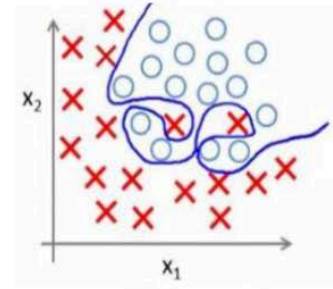
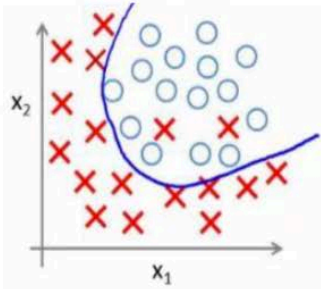
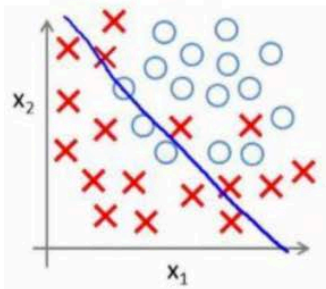
---

- 데이터의 대표성
  - 훈련 데이터가 미래에 나타날 가능성이 있는 모든 데이터의 특징을 반영하도록 구성해야 함
    - 예: 지리적, 인종적, 나이별, 소득별, 성별 등 균일성 유지
  - 훈련, 검증, 테스트 샘플 데이터가 전체 데이터의 특징을 계속 유지할 수 있어야 함

# 머신 러닝

- 과대적합(over fitting)

- 모델이 훈련 데이터에 대해서만 잘 동작하도록 훈련되어 새로운 데이터에 대해서는 오히려 잘 동작하지 못하는 것
- 과대적합된 모델은 훈련 데이터에 대해서는 매우 우수한 성능을 보이지만 일반화가 떨어짐
- 머신러닝에서는 과대적합을 피해서 일반적으로 잘 동작하게 모델을 만드는 것이 매우 중요함
  - 이를 모델의 일반화(generalization)라고 함



- 과소 적합(under fitting)

- 모델이 너무 간단하여 성능이 미흡한 경우
- 과소적합을 피하려면 좀 더 상세한 모델 구조를 사용해야 함
- 머신러닝에서는 과대적합과 과소적합을 모두 피해야 하며 최적의 예측을 수행하는 모델을 만드는 것이 중요함

# 머신 러닝

- 모델의 성능

- 모델의 성능을 평가하는 척도 필요
- 분류에서는 정확도(accuracy)를 성능 척도로 주로 사용
  - (참고) 분류에서 손실함수로 크로스 엔트로피를 주로 사용
- 손실함수와 성능 지표의 차이점
  - 손실함수를 정하는 목적은 모델을 훈련시킬 때의 기준으로 삼기 위해서임
  - 모델은 손실함수를 최소화 하는 방향으로 학습
  - 모델의 성능은 이렇게 만든 모델이 궁극적으로 얼마나 잘 동작하는지를 평가하는 척도임

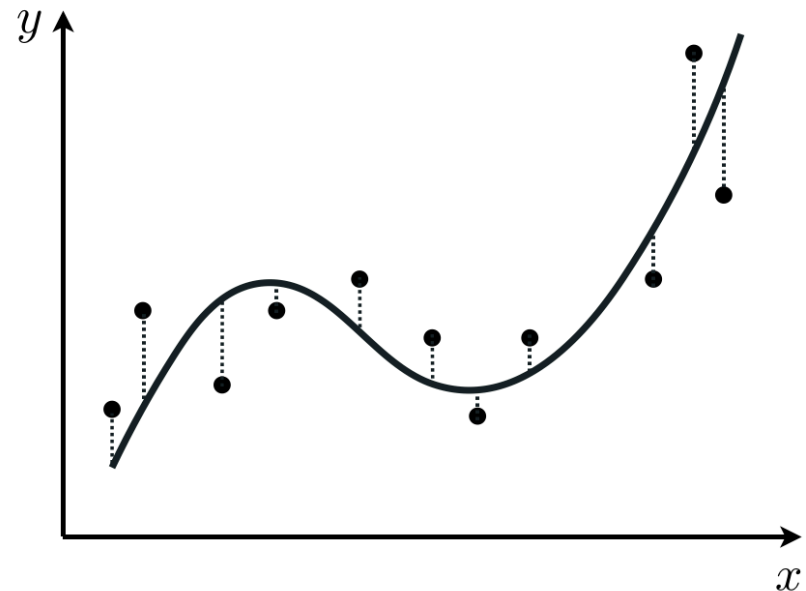
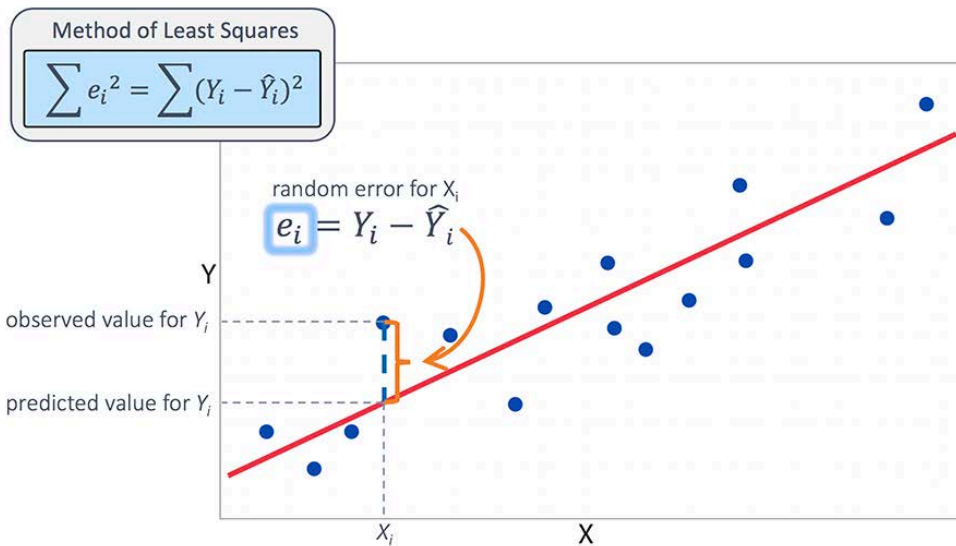
	손실함수	성능 지표
정 의	손실함수를 줄이는 방향으로 모델이 학습을 함	성능을 높이는 것이 머신러닝을 사용하는 목적임
회귀 모델	MSE (오차 자승의 평균)	R2
분류 모델	크로스 엔트로피	정 확 도 , 정 밀 도 , 재현률, F1점수

# 회귀 분석

- 회귀 분석

- 수치형 종속변수와 수치형 독립변수사이의 영향 또는 인과관계를 알 수 있는 분석
- 학습 데이터  $x$ 로 부터  $y$ 를 예측하는 함수  $f(x)$ 를 찾는 과정으로  $x$ 와  $y$ 는 모두 연속적인 수치 값
- 도출된 회귀식에서 직선의 기울기와 상수를 알 수 있는데, 이를 통해 독립변수의 변화에 따른 종속변수의 변화를 알 수 있는 것

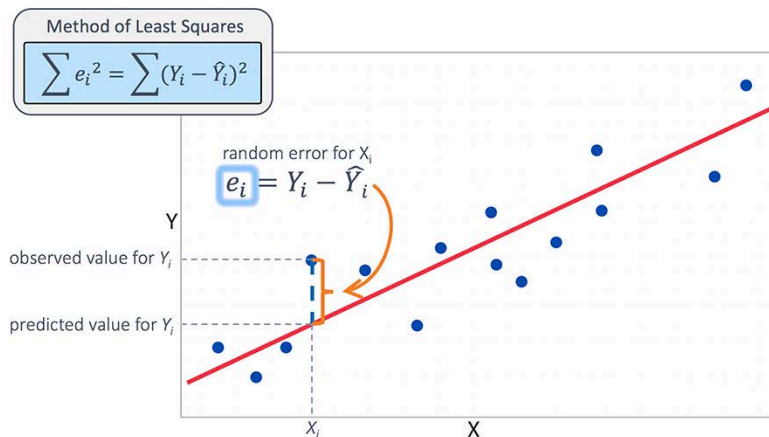
$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i + \dots + \beta_k X_i + \epsilon_i, (i = 1, \dots, n)$$



# 회귀 분석

- 회귀 문제와 손실함수

- MAE(Mean absolute Error): 원본 값과 예측 값에 대한 절대 오류의 평균
- MSE(Mean Squared Error): 원본 값과 예측 값에 대한 오류 제곱의 평균
- RMSE(Root MSE): MSE의 제곱근
- R-squared: 원본 값과 예측 값을 비교하여 회귀모델이 얼마나 잘 원본 값을 나타내는지 [0,1]



$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

$\hat{y}$  – predicted value of  $y$   
 $\bar{y}$  – mean value of  $y$



# 회귀 분석

- 단순 회귀 분석

- 하나의 수치형 설명변수가 하나의 수치형 종속변수에 어떤 인과관계 또는 영향을 미치는지에 대한 분석을 말함
- 많은 변수는 고려하지 않고 오직 하나의 종속변수(Y)와 하나의 독립변수(X)에 의해서만 시행
- Ex) 쇼핑몰의 입점 매장 수가 고객의 방문빈도에 어떤 영향을 미치는지를 확인하려 하는 상황에 우리는 단순회귀분석을 사용

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, (i = 1, \dots, n)$$

$$\hat{Y} = b_0 + b_1 X$$

	총매출액	방문빈도	1회평균매출액	쿠폰사용횟수	거래기간
0	12717240	109	116672	4	1093
1	12802210	22	581919	20	1002
2	12815010	27	474630	11	1066
3	13038990	24	543291	5	1069
4	13072260	37	353304	9	1077

# 회귀 분석

- 단순 회귀 분석

- P쇼핑몰에서 최근 인기있는 S브랜드매장을 입점시킨 결과, 고객들의 전반적인 방문빈도가 늘어났다고 함
- 늘어난 방문빈도가 실제 총 매출액에 영향을 미치는지 알아보고자

```
model = smf.ols(formula = '총매출액 ~ 방문빈도', data = data)
result = model.fit()
result.summary()
```

OLS Regression Results

$$\hat{Y}_i = 13490000 + 144800 \times \text{'방문빈도'}$$

Dep. Variable:	총매출액	R-squared:	0.191
Model:	OLS	Adj. R-squared:	0.184
Method:	Least Squares	F-statistic:	26.64
Date:	Mon, 25 Feb 2019	Prob (F-statistic):	1.06e-06
Time:	15:56:14	Log-Likelihood:	-2015.2
No. Observations:	115	AIC:	4034.
Df Residuals:	113	BIC:	4040.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	1.349e+07	1.89e+06	7.129	0.000	9.74e+06	1.72e+07
방문빈도	1.448e+05	2.81e+04	5.161	0.000	8.92e+04	2e+05

# 회귀 분석

- 다중 회귀 분석

- 하나의 수치형 종속변수와 2개이상의 수치형 독립변수 사이의 영향 또는 인과관계를 설명하는 분석을 말함
- 예를 들어, 아빠의 키, 엄마의 키, 할머니의 키(독립변수)가 자녀의 키(종속변수)에 어떤 관계를 갖는지 확인하려 할 때 우리는 다중회귀분석을 사용함

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_i X_i + \epsilon_i$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_i X_i$$

- 변수선택법: 다중회귀분석을 할 때는 여러 개의 독립변수를 선택하므로 최적화된 회귀모형을 만들기 위해서는 어떤 변수들을 독립변수에 넣을 것인지를 잘 판단해야 함

독립변수 선택 방법	
입력	사용하고자하는 독립변수를 모두 입력하는 방식
전진	모형적합에 가장 큰 영향을 미치는 독립변수를 순서대로 추가하는 방식
후진	모형적합에 가장 약하게 영향을 미치는 독립변수를 순서대로 제거하는 방식
단계 선택	<u>전진선택</u> 과 <u>후진선택</u> 을 결합한 방식이다. <u>전진선택</u> 을 사용하여 독립변수를 추가한 뒤, 다중공선성을 판단하여 후진방식으로 독립변수를 제거하는 방식이다.

# 회귀 분석

- 다중 회귀 분석

```
model = smf.ols(formula = '방문빈도 ~ 거래기간 + 총매출액 + 쿠폰사용횟수', data = data)
result = model.fit()
result.summary()
```

$$\hat{Y}_i = -55.174 + 0.0786 * \text{거래기간} + 0.000001 * \text{총매출액}$$

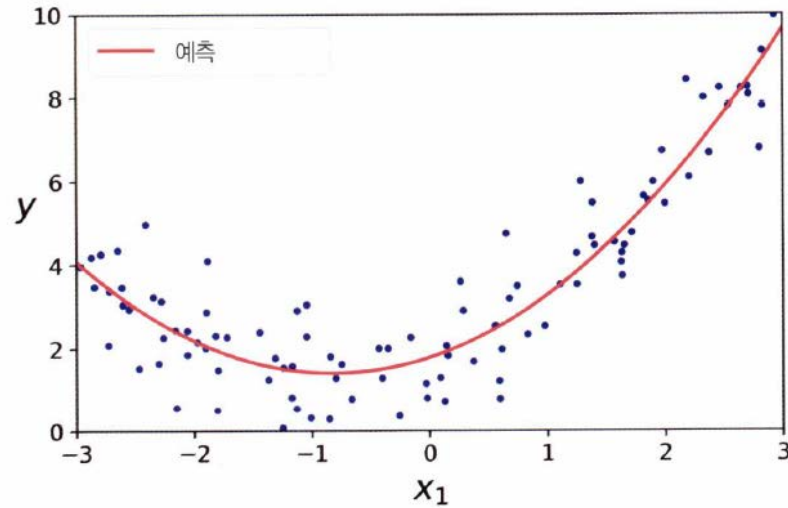
OLS Regression Results

Dep. Variable:	방문빈도	R-squared:	0.245
Model:	OLS	Adj. R-squared:	0.225
Method:	Least Squares	F-statistic:	12.02
Date:	Mon, 25 Feb 2019	Prob (F-statistic):	7.14e-07
Time:	15:56:14	Log-Likelihood:	-549.34
No. Observations:	115	AIC:	1107.
Df Residuals:	111	BIC:	1118.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-55.1744	31.260	-1.765	0.080	-117.117	6.768
거래기간	0.0786	0.029	2.665	0.009	0.020	0.137
총매출액	1.215e-06	2.51e-07	4.834	0.000	7.17e-07	1.71e-06
쿠폰사용횟수	0.2959	0.312	0.948	0.345	-0.323	0.915

# 회귀 분석

- 다항 회귀 분석
  - 한 특성과 예측값의 관계가 선형이 아닌 2차, 3차 이상의 관계를 갖는 회귀 방법



# 회귀 분석

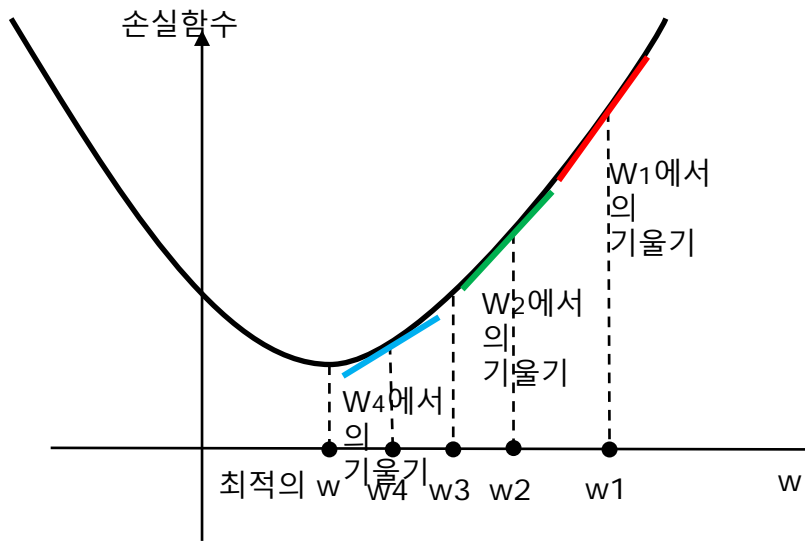
- 최적화(경사 하강법)

- 가장 일반적인 최적화 알고리즘: (Gradient Descent)
- 손실함수를 계수에 관한 그래프로 그렸을 때 최소값으로 빨리 도달하기 위해서는 현재 위치에서의 기울기(미분 값)에 비례하여 반대방향으로 이동하는 방식

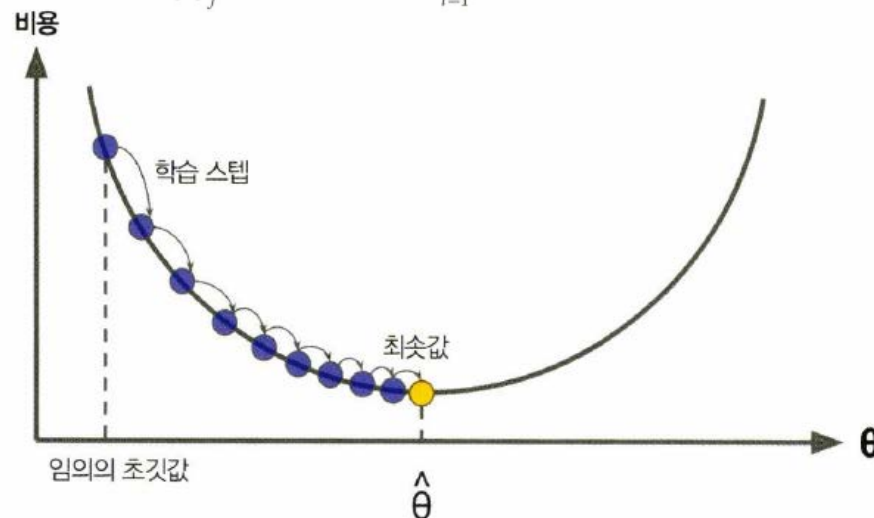
- 경사 하강법 특징

- 경사 하강법을 적용하려면 특성 변수들을 모두 동일한 방식으로 스케일링해야 한다.
- 특성 값마다 크기의 편차가 크면 특정 변수에 너무 종속되어 동작할 수 있고 이로 인해 수렴속도가 직선이 되지 않고 오래 걸릴 수가 있다.

$$W_i = W_{i-1} - \eta \text{Grad}(i)$$



$$\frac{\partial}{\partial \theta_j} \text{MSE}(\theta) = \frac{2}{m} \sum_{i=1}^m (\theta^T \mathbf{x}^{(i)} - y^{(i)}) x_j^{(i)}$$



# 회귀 분석

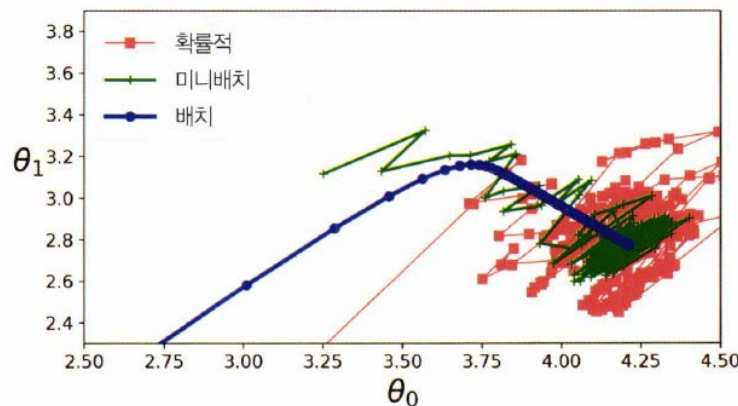
- 경사 하강법 종류

- 배치(Batch) GD

- 일반적으로 배치 GD방식을 많이 사용하는데, 적절한 크기의 배치단위로 입력 신호를 나누어 경사 하강법을 적용하는 방식임

- SGD (확률적 경사 하강법)

- 한 번에 한 샘플씩 랜덤하게 골라서 훈련에 사용하는 방법이다.
    - 즉 샘플을 하나만 보고 계수를 조정함
    - 계산량이 적어 동작속도가 빠르고, 랜덤한 방향으로 학습을 하므로 전역 최소치를 가능성이 높아짐
    - 매 샘플이 너무 랜덤하여 방향성을 잃고 수렴하는데 시간이 오래 걸릴 가능성도 있음



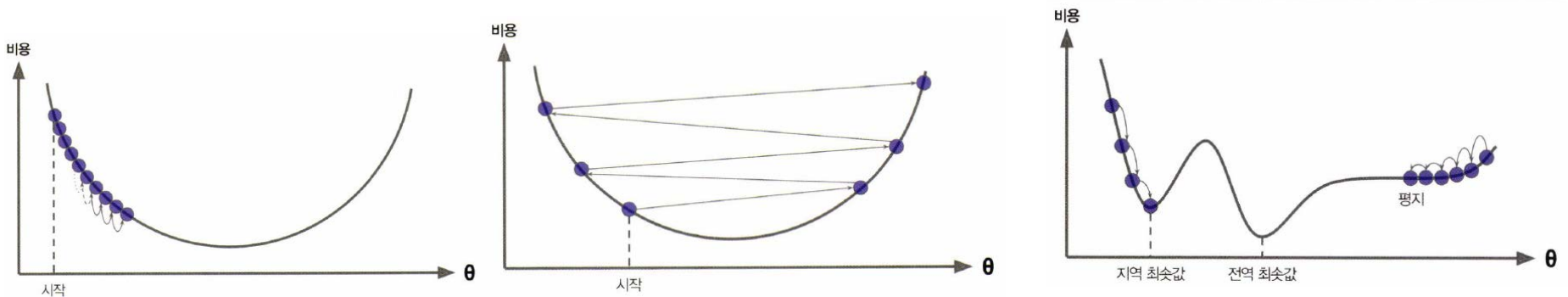
# 회귀 분석

- 학습률

- 계수를 업데이트 하는 속도를 조정하는 변수
- 학습률이 너무 작으면 수렴하는데 시간이 오래 걸리지만 최저점에 도달했을 때 흔들림 없이 안정적인 값을 얻게 되고,
- 학습률을 너무 크게 정하면 학습하는 속도는 빠르나 자칫하면 최저점으로 수렴하지 못하고 발산하거나 수렴하더라도 흔들리는 오차가 남아있을 수 있음

- 학습 스케줄(learning schedule) 기법

- 초기에는 학습률을 크게 정하고 (학습률을 빠르게 하고) 오차가 줄어들면 학습률을 줄여서 안정상태(steady state)의 오차를 줄이는 방법



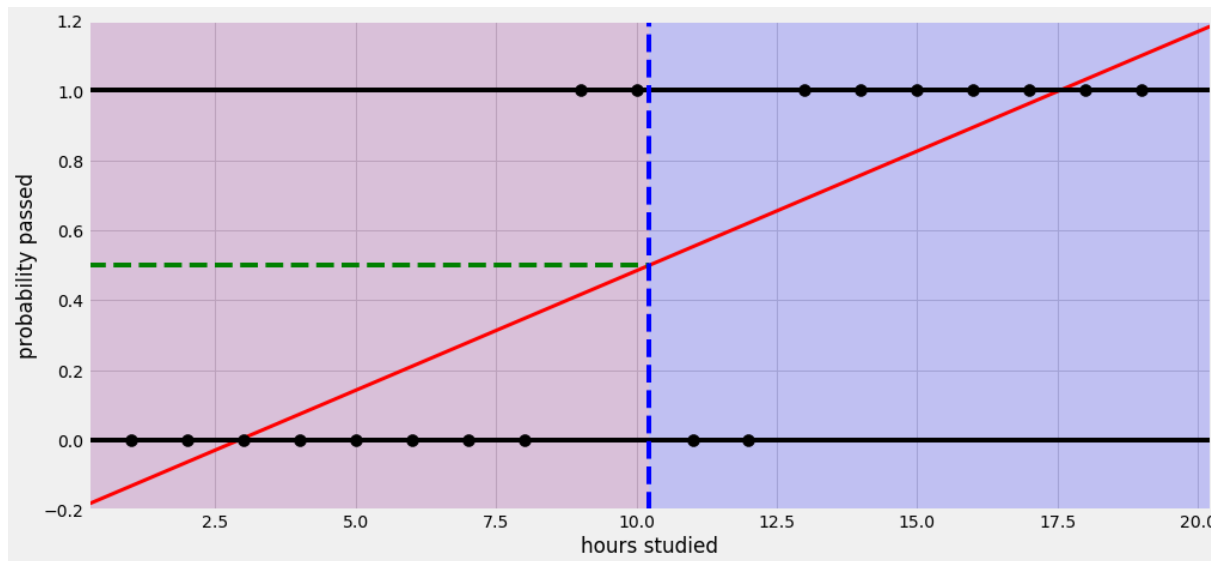


# 회귀 분석

- 로지스틱 회귀

- 임의의 범위를 갖는 값으로부터 0과 1사이의 값을 예측하거나 이진 분류에 사용하는 알고리즘임
- 로지스틱 회귀분석은 보통 독립 변수와 종속 변수의 관계를 S형 커브로 매핑함(선형 회귀분석 사용이 불가능한 경우)
- 신용도 판단, 연간 구매량 기준 우수 고객 여부 판단, 평가 지표 기준 합격 여부 판단, 건강 지표에 따른 건강 여부, 팀의 승리/패배 여부 예측 등 여러 경우에 사용함

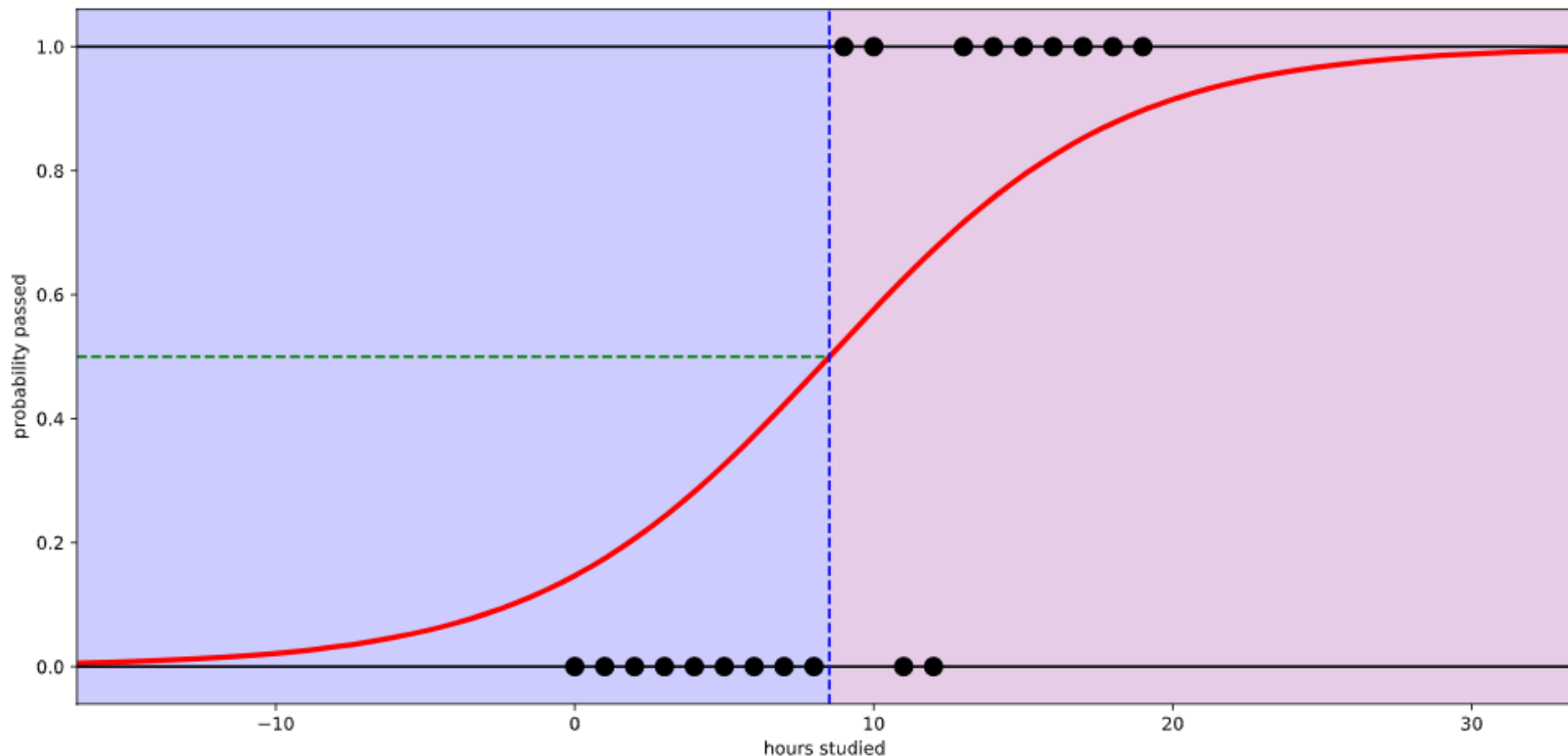
- 공부시간과 합격 여부



# 회귀 분석

- 로지스틱 회귀

- 데이터가 특정 범주에 속할 확률을 예측하는 단계
- 모든 속성(feature)들의 계수(coefficient)와 절편(intercept)을 0으로 초기화
- 각 속성들의 값(value)에 계수(coefficient)를 곱해서 log-odds를 구함
- Log-odds를 sigmoid 함수에 넣어서  $[0,1]$  범위의 확률을 구함



# 회귀 분석

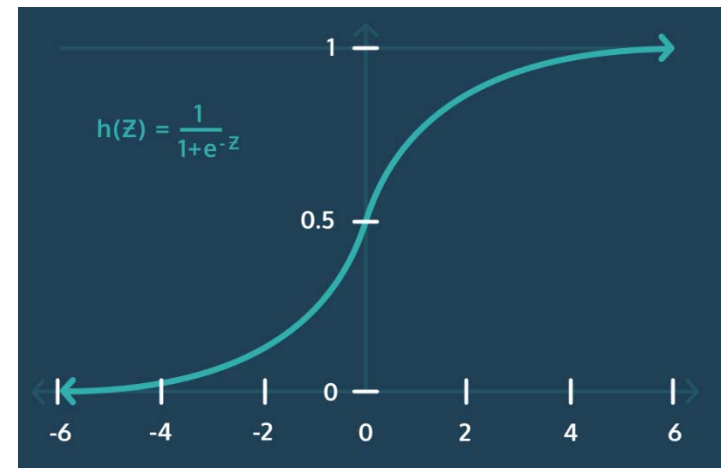
- 로지스틱 회귀

- odds: 사건이 발생할 확률을 발생하지 않을 확률로 나눈 값

$$Odds = \frac{P(event\ occurring)}{P(event\ not\ occurring)}$$

$$Odds\ of\ passing = \frac{0.7}{0.3} = 2.\overline{33} \quad Log\ odds\ of\ passing = \log(2.\overline{33}) = 0.847$$

$$z = b_0 + b_1x_1 + \cdots + b_nx_n$$



# 회귀 분석

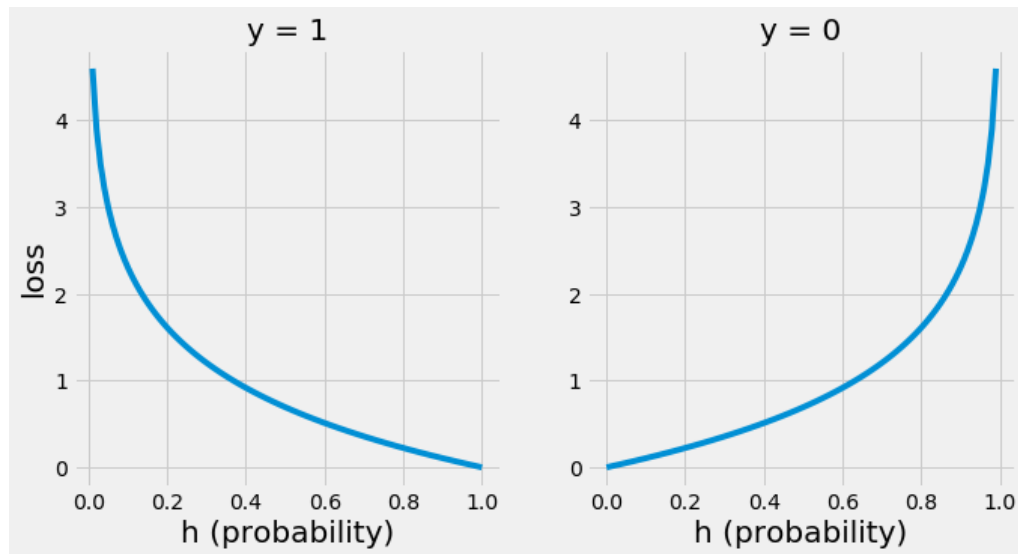
- 로지스틱 회귀

- 비용 함수(손실 함수): 로지스틱 회귀가 확률을 제대로 예측해주는지, 즉 구해놓은 속성들의 계수(coefficients)와 절편(intercept)이 적절한지 확인하기 위해 손실(Loss)을 고려
- 모델의 "적합성"을 평가하기 위해 각 데이터 샘플의 손실(모델 예측이 얼마나 잘못되었는지)을 계산한 다음 그것들의 평균화 해야 함

$$-\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h(z^{(i)})) + (1 - y^{(i)}) \log(1 - h(z^{(i)}))]$$

$$loss_{y=1} = -\log(h(z^{(i)}))$$

$$loss_{y=0} = -\log(1 - h(z^{(i)}))$$



# 회귀 분석

- 다항 로지스틱 회귀(소프트맥스 회귀)

- 앞에서는 이진 분류, 즉 합격/불합격 등 두 개의 레이블을 가진 경우에 로지스틱 회귀를 사용하는 예를 소개했음
- 그런데 2개가 아니라 3개 이상의 클래스 중에 하나를 예측해야 하는 경우는 다항 로지스틱 회귀(multinomial logistic regression)를 이용함
- 소프트맥스 (softmax) 함수를 사용함
  - $k$  : 범주의 수
  - $s(\mathbf{x})$  : 샘플  $\mathbf{x}$ 에 대한 각 범주의 점수를 담고 있는 벡터
  - $\sigma(s(\mathbf{x}))_k$  : 이 샘플이 범주  $k$ 에 속할 확률

$$s_k(\mathbf{x}) = (\boldsymbol{\theta}^{(k)})^T \mathbf{x} \quad \hat{p}_k = \sigma(s(\mathbf{x}))_k = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^K \exp(s_j(\mathbf{x}))}$$

- Argmax

- 이 연산은 함수를 최대화하는 변수의 값을 반환한다(numpy에도 비슷한 함수가 있는데 array에서 최댓값을 가지는 원소의 index를 반환)

$$\hat{y} = \operatorname{argmax}_k \sigma(s(\mathbf{x}))_k = \operatorname{argmax}_k s_k(\mathbf{x}) = \operatorname{argmax}_k ((\boldsymbol{\theta}^{(k)})^T \mathbf{x})$$

# 회귀 분석

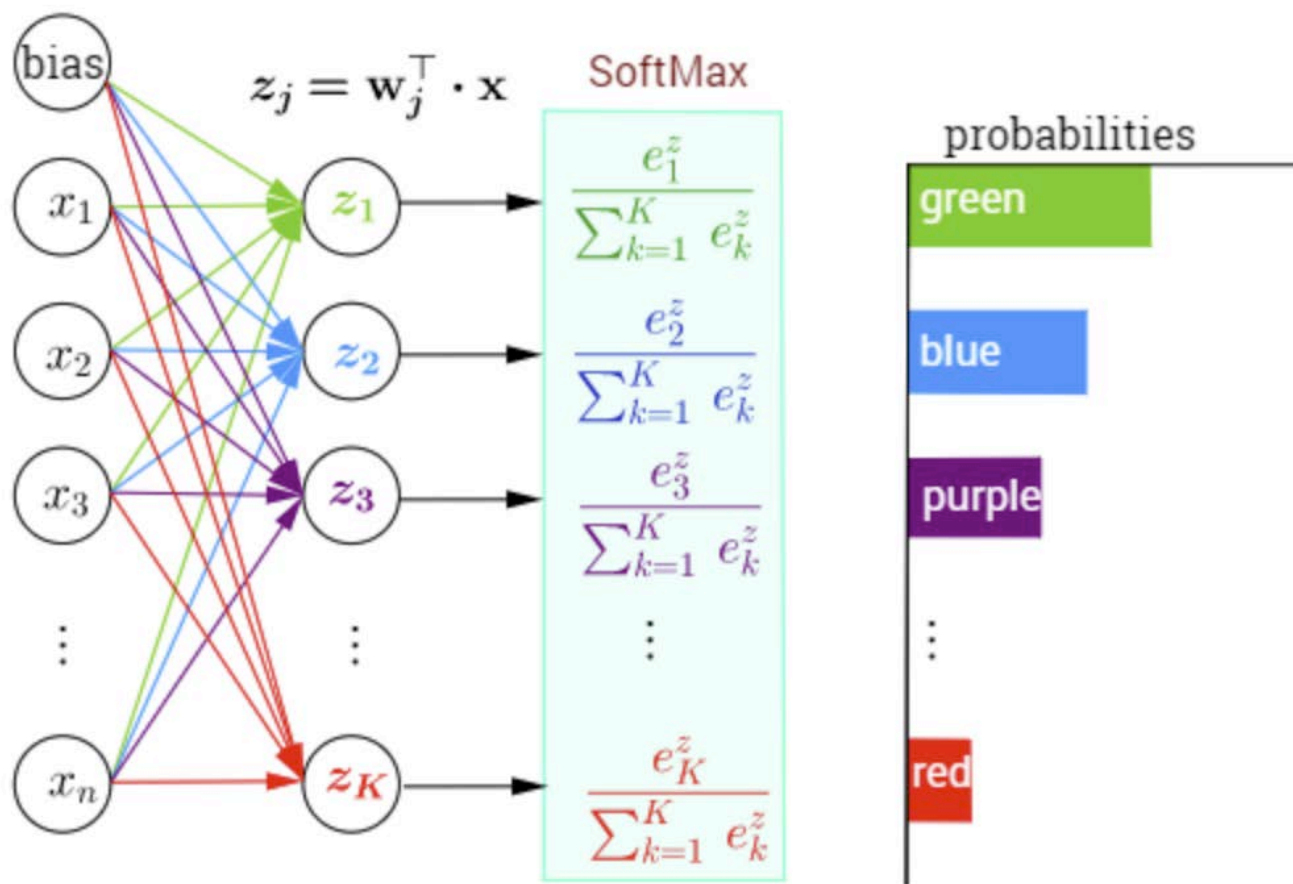
---

- 다항 로지스틱 회귀(소프트맥스 회귀)
  - 비용 함수 : 크로스 엔트로피

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{p}_k^{(i)})$$

# 회귀 분석

- 소프트맥스

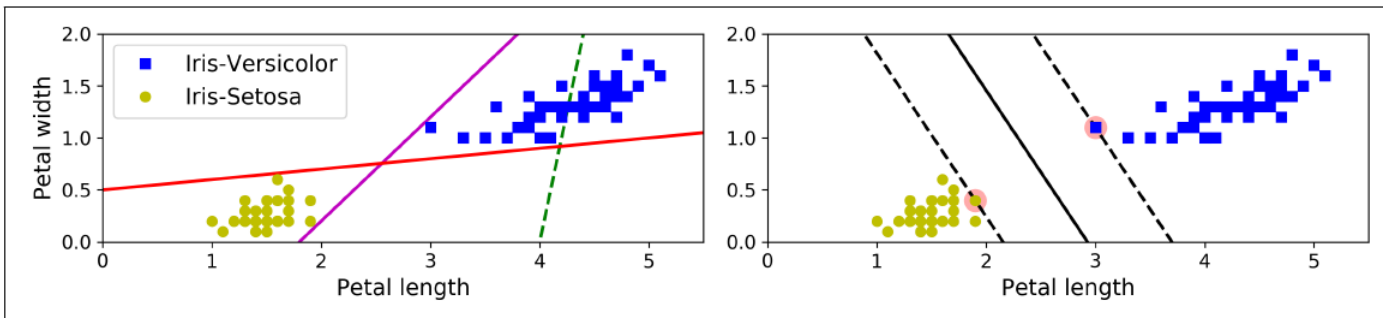


# 회귀 분석

- 서포트 벡터 머신(SVM)

- SVM은 비선형, 선형 분류, 회귀, 이상치 탐색을 하는데 사용할 수 있는 강력한 ML 모델 중 하나
- SVM은 특히 복잡한 문제에 잘 맞으며 작거나 중간 크기의 데이터 셋에 적합

- 선형 SVM(라지 마진 분류)



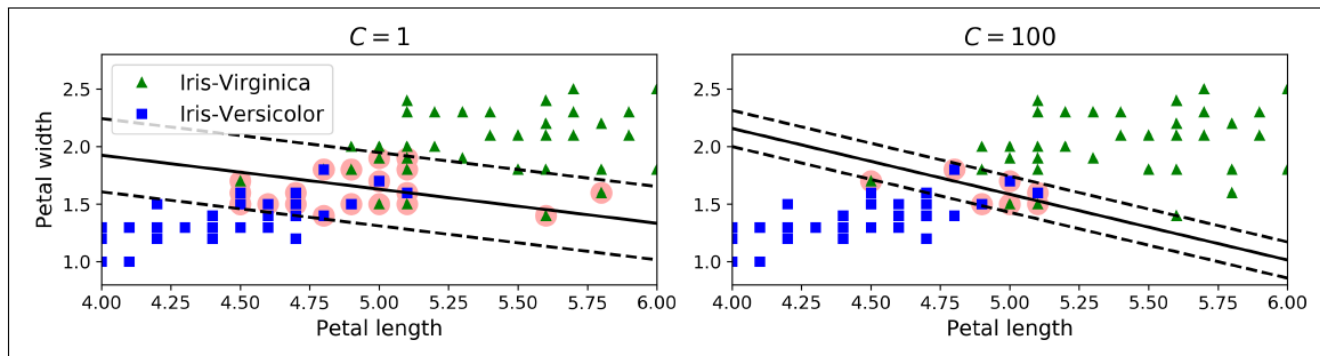
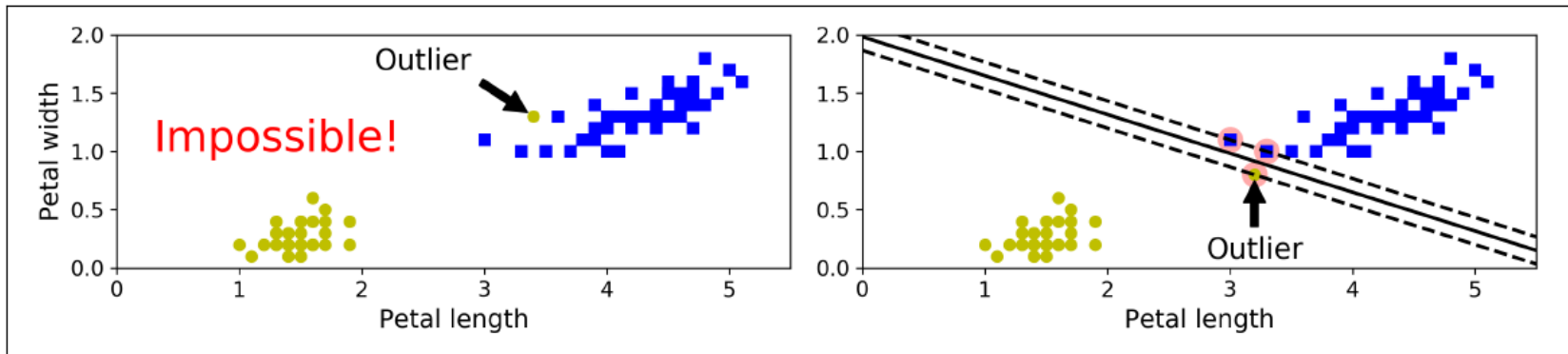


# 회귀 분석

- 서포트 벡터 머신(SVM)

- 소프트 마진 분류

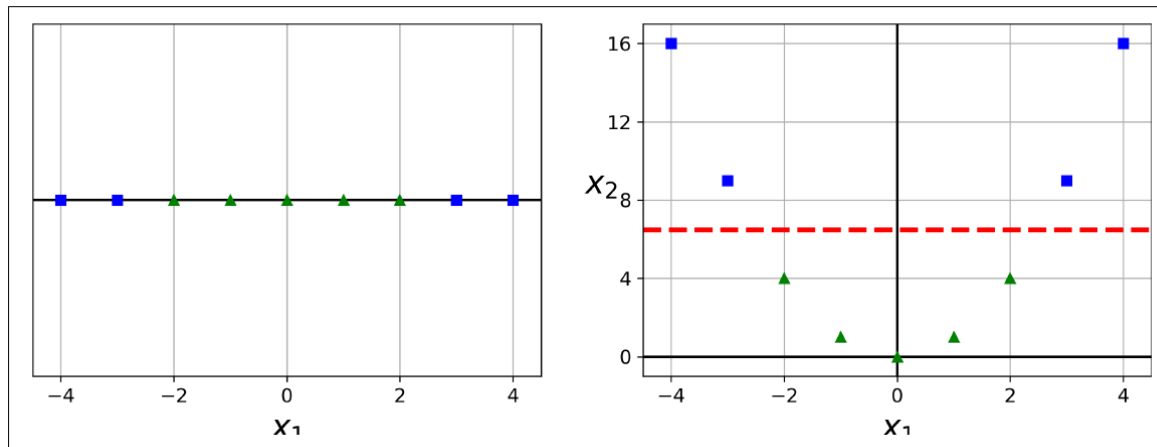
- 모든 샘플이 마진 바깥쪽에 올바르게 분류되어 있는 경우를 **하드마진 분류**
      - 데이터가 선형적으로 구분되어야 제대로 동작
      - 이상치에 민감
    - 도로의 폭을 가능한 넓게 유지하는 것과 마진 오류 사이의 적절한 균형을 찾는 **소프트 마진 분류**



# 회귀 분석

- 비선형 SVM 분류

- 많은 경우의 데이터는 직선으로 분류되지 않음
- 특징점들을 추가해서 직선으로 분류 가능하게 만들면 됨  $x_2 = x_1^2$



```
("svm_clf", SVC(kernel="poly", degree=3, coef0=1, C=5))
```

