

1장 한눈에 보는 머신러닝

감사의 글

자료를 공개한 저자 오렐리앙 제롱과 강의자료를 지원한 한빛아카데미에게 진심어린 감사를 전합니다.

1.1 머신러닝이란?

- 아서 새뮤얼(Artuhr Samuel), 1959

머신러닝은 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야

- 톰 미첼(Tom Michell), 1977

*어떤 작업T에 대한 컴퓨터 프로그램의 성능을P로 측정했을 때 경험E로 인해 성능이 향상되었다면, 이 컴퓨터 프로그램은 **작업T와 성능 측정P에 대해 경험E로부터 학습한다**고 말한다.*

머신러닝 프로그램 예제

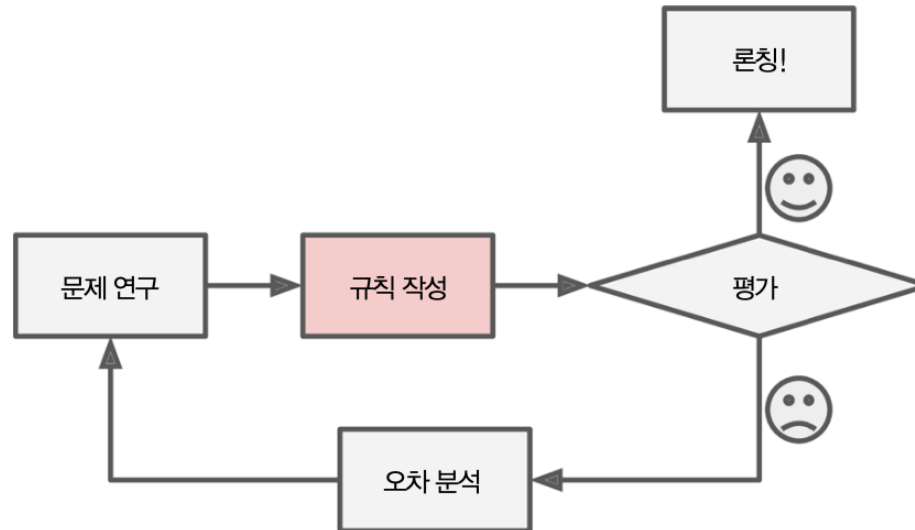
- 스팸 필터: 스팸(spam)과 스팸이 아닌 메일(ham)의 샘플을 이용하여 스팸 메일 구분법 학습

기본 용어

- **훈련 세트(training set)**: 머신러닝 프로그램이 훈련(학습)하는 데 사용하는 데이터 집합
- **훈련 사례** 혹은 **샘플**: 각각의 훈련 데이터
- **토미첼의 정의와의 연계**
 - 작업 T: 새로운 메일이 스팸 여부 판단
 - 경험 E: 훈련 데이터셋
 - 성능 P: 예를 들어, 정확히 분류된 메일의 비율

1.2 왜 머신러닝을 사용하는가?

전통적인 프로그래밍

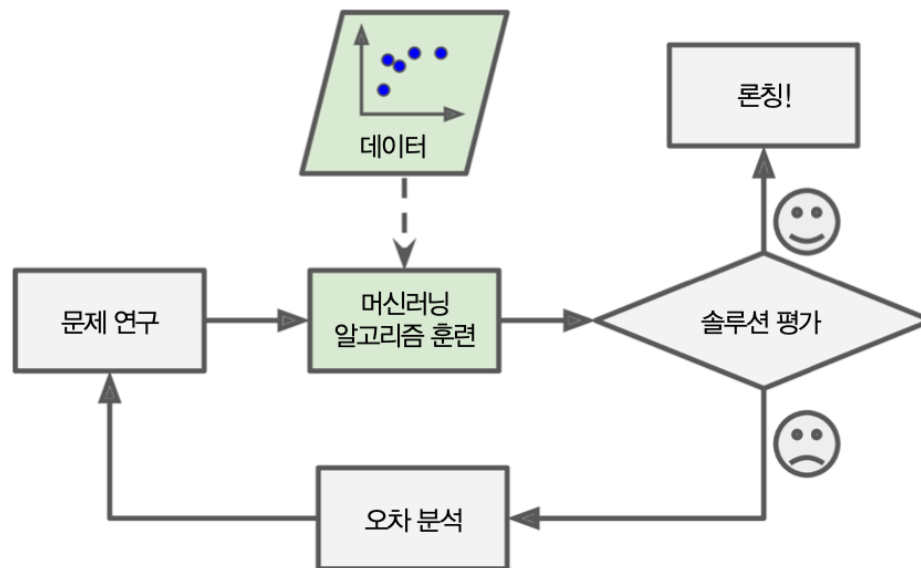


- 전통적인 프로그래밍 접근 방법은 다음과 같다.
 - **문제 연구**: 누군가가 문제를 해결하기 위해 해결책을 찾음
 - **규칙 작성**: 결정된 규칙을 개발자가 프로그램을 작성
 - **평가**: 만들어진 프로그램을 테스트
 - 문제가 없다면 **론칭**, 문제가 있다면 **오차를 분석**한 후 처음 과정부터 다시 실시

예제: 스팸 메일 분류

- 특정 단어가 들어가면 스팸 메일로 처리
- 프로그램이 론칭된 후 새로운 스팸단어가 생겼을 때 소프트웨어는 이 단어를 자동으로 분류할 수 없음
- 개발자가 새로운 규칙을 업데이트 시켜줘야 함
- 새로운 규칙이 생겼을 때 사용자가 매번 업데이트를 시켜줘야하기 때문에 유지 보수가 어려움

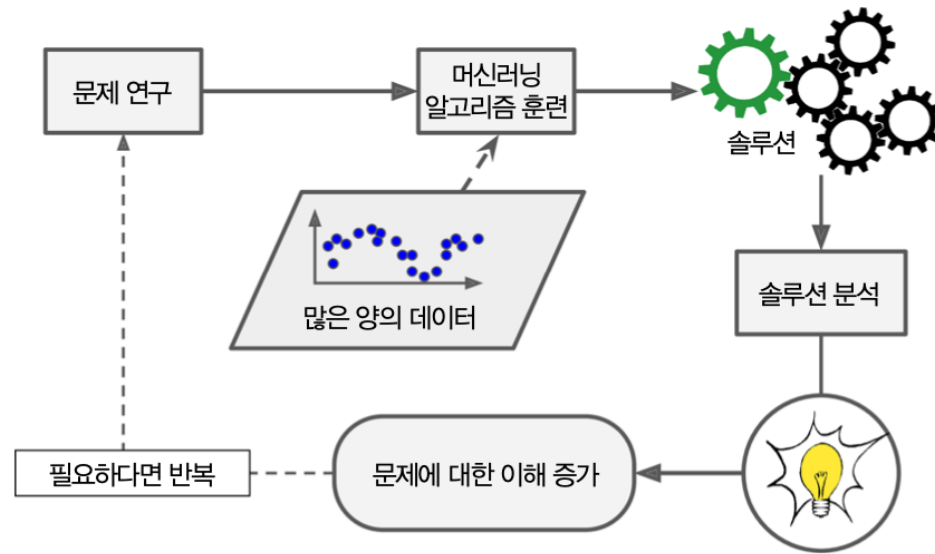
머신러닝



예제: 스팸 메일 분류

- 사용자가 스팸으로 지정한 메일에 '4U', 'For U', 'Buy Drugs' 등의 표현이 자주 등장하는 경우 그런 표현을 자동으로 인식하고 메일을 스팸으로 분류하도록 프로그램 스스로 학습

머신러닝의 장점



- 전통적인 방식으로는 해결 방법이 없는 너무 복잡한 문제 해결
- 새로운 데이터에 쉽게 적응 가능한 시스템
- 데이터 마이닝(data mining): 복잡한 문제와 대량의 데이터에서 통찰 얻기

1.3 적용 사례

대표적인 머신러닝 적용 사례

- 이미지 분류 작업: 생산 라인에서 제품 이미지를 분석해 자동으로 분류
- 시맨틱 분할 작업: 뇌를 스캔하여 종양 진단
- 텍스트 분류(자연어 처리): 자동으로 뉴스 기사 분류
- 텍스트 분류: 토론 포럼에서 부정적인 코멘트를 자동으로 구분
- 텍스트 요약: 긴 문서를 자동으로 요약
- 자연어 이해 : 챗봇(chatbot) 또는 개인 비서 만들기

- 회귀 분석: 회사의 내년도 수익을 예측하기
- 음성 인식: 음성 명령에 반응하는 앱
- 이상치 탐지: 신용 카드 부정 거래 감지
- 군집 작업: 구매 이력을 기반 고객 분류 후 다른 마케팅 전략 계획
- 데이터 시각화: 고차원의 복잡한 데이터셋을 그래프로 효율적 표현
- 추천 시스템: 과거 구매 이력 관심 상품 추천
- 강화 학습: 지능형 게임 봇(bot) 만들기

1.4 머신러닝 시스템 종류

머신러닝 시스템 분류 기준

- 기준 1: 훈련 지도 여부
 - 지도 학습
 - 비지도 학습
 - 준지도 학습
 - 강화 학습
- 기준 2: 실시간 훈련 여부
 - 온라인 학습
 - 배치 학습
- 기준 3: 예측 모델 사용 여부
 - 사례 기반 학습
 - 모델 기반 학습

분류 기준의 비배타성

- 분류 기준이 상호 배타적이지는 않음.
- 스팸 필터 예제
 - 심층 신경망 모델 활용 실시간 스팸 메일 분류 학습 가능
 - 지도 학습 + 온라인 학습 + 모델 기반 학습

훈련 지도 여부 구분

지도 학습

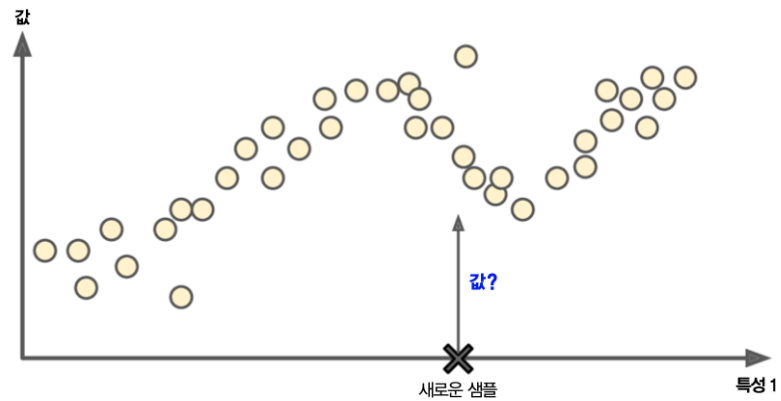
- 훈련 데이터에 **레이블(label)**이라는 답 포함
 - 레이블 대신에 **타겟(target)**이란 표현도 사용됨.
- 대표적 지도 학습
 - 분류
 - 회귀

분류



- 특성을 사용한 데이터 분류
- 예제: 스팸 필터
 - 특성: 소속 정보, 특정 단어 포함 여부 등
 - 레이블(타겟): 스팸 또는 햄

회귀



- 특성을 사용하여 타겟(target) 수치 예측
- 예제: 중고차 가격 예측
 - 특성: 주행거리, 연식, 브랜드 등
 - 타겟: 중고차 가격

중요한 지도학습 알고리즘들

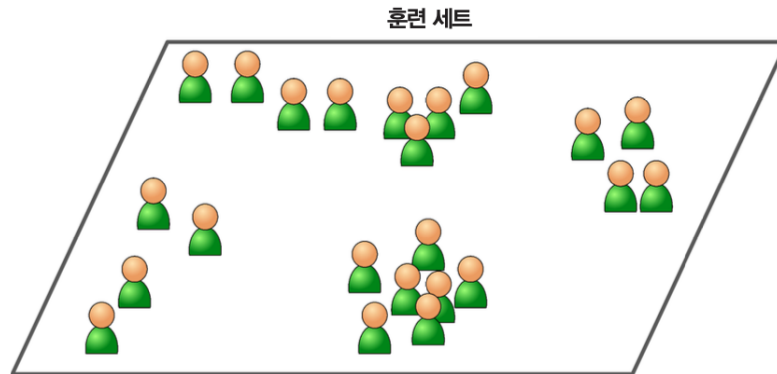
- k-최근접 이웃(k-NNs)
- 선형 회귀(linear regression)
- 로지스틱 회귀(logistic regression)
- 서포트 벡터 머신(support vector machines, SVCs)
- 결정 트리(decision trees)와 랜덤 포레스트(random forests)
- 신경망(neural networks)

주의사항

- 일부 회귀/분류 알고리즘을 분류/회귀에 사용 가능
- 예제: 로지스틱 회귀, SVM 등등

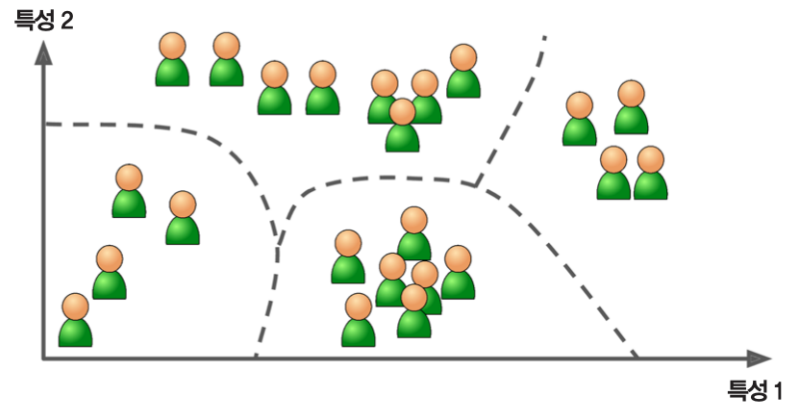
비지도 학습

- 레이블 없는 훈련 데이터를 이용하여 시스템 스스로 학습



- 대표적 비지도 학습
 - 군집
 - 시각화
 - 차원 축소
 - 연관 규칙 학습

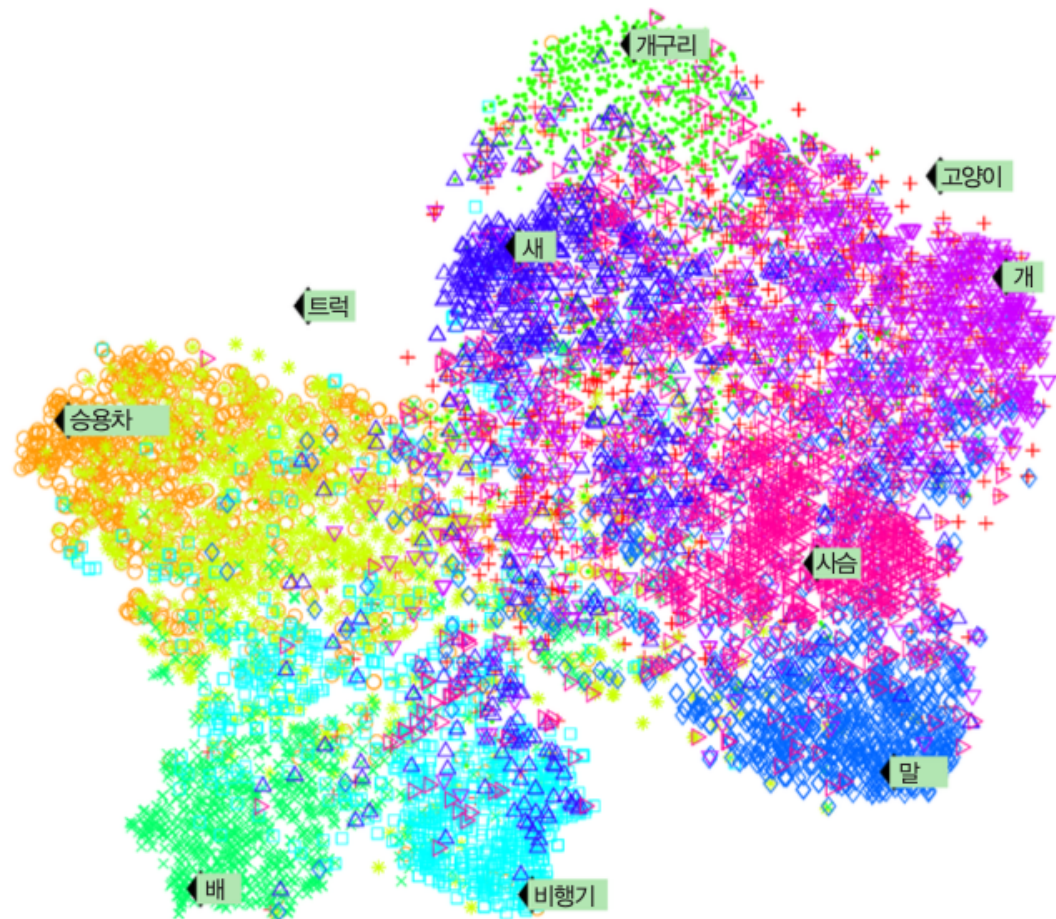
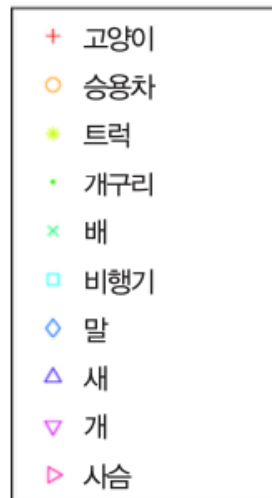
군집



- 데이터를 비슷한 특징을 가진 몇 개의 그룹으로 나누는 것
- 예제: 블로그 방문자들을 그룹으로 묶기: 남성, 여성, 주말, 주중, 만화책, SF 등등
- 대표적 군집 알고리즘
 - k-평균
 - DBSCAN
 - 계층 군집 분석

시각화

- 다차원 특성을 가진 데이터셋을 2D 또는 3D로 표현하기
- 시각화를 하기 위해서는 데이터의 특성을 2가지로 줄여야함
- 데이터가 구성 패턴에 대한 정보 획득 가능



차원 축소

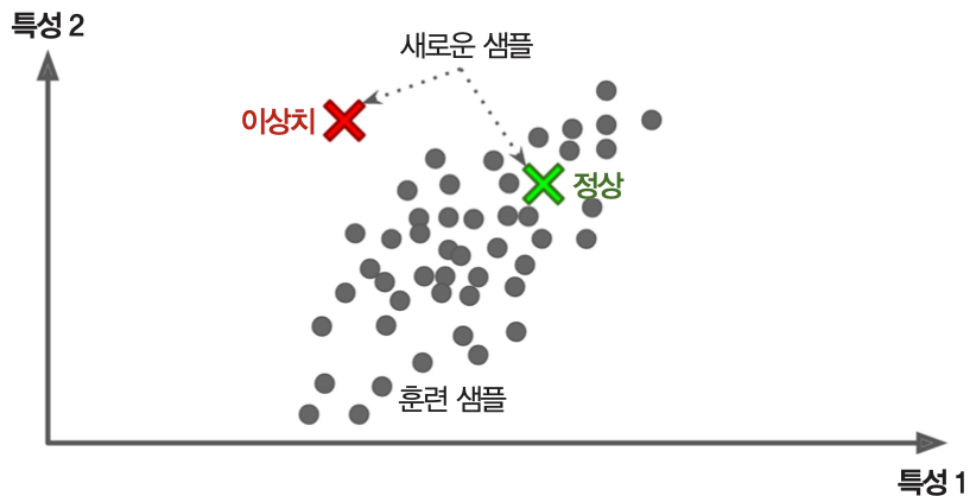
- 데이터의 특성 수 줄이기
- 예제
 - 특성 추출: 상관관계가 있는 여러 특성을 하나로 합치기
 - 자동차의 주행거리와 연식은 상관관계가 높음. 따라서 차의 '마모정도'라는 하나의 특성으로 합칠 수 있음.
- 차원 축소의 장점: 머신러닝 알고리즘의 성능 향상
 - 훈련 실행 속도 빨라짐
 - 메모리 사용 공간 줄어듦

시각화와 차원축소 알고리즘

- 주성분 분석(PCA)
- 커널 PCA
- 지역적 선형 임베딩
- t-SNE

이상치 탐지(Outlier detection)

- 정상 샘플을 이용하여 훈련 후 입력 샘플의 정상여부 판단 후 이상치(outliers) 자동 제거



- 예제: 부정거래 사용 감지, 제조 결함 잡아내기 등등

특이치 탐지(Novelty detection)

- 전혀 '오염되지 않은'(clean) 훈련 세트 활용 후 훈련 세트에 포함된 데이터와 달라 보이는 데이터 감지하기

이상치 탐지 vs. 특이치 탐지

- 예제: 수 천장의 강아지 사진에 치와와 사진이 1%정도 포함되어 있는 경우
- 특이치 탐지 알고리즘은 새로운 치와와 사진을 특이한 것으로 간주하지 않음.
- 반면에 비정상 탐지 알고리즘은 새로운 치와와 사진을 다른 강아지들과 다른 종으로 간주할 수 있음.

연관 규칙 학습

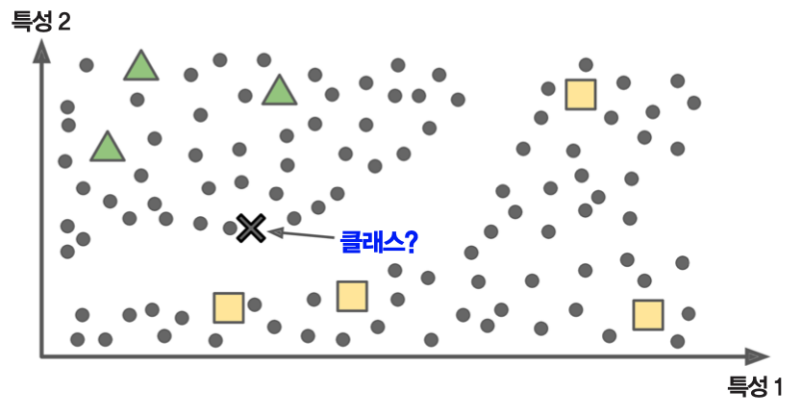
- 데이터 특성 간의 흥미로운 관계 찾기
- 예제: 마트 판매 기록
 - 바비큐 소스와 감자 구매와 스테이크 구매 사의 연관성이 밝혀지면 상품을 서로 가까이 진열해야 함.

준지도 학습

- 레이블이 적용된 적은 수의 샘플이 주어졌을 때 유용함.
- 비지도 학습을 통해 군집을 분류한 후 샘플들을 활용해 지도 학습 실행
- 대부분 지도 학습과 비지도 학습 혼합 사용

준지도 학습 예제

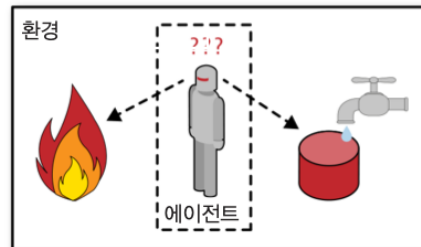
- 아래 그림 참조: 새로운 사례 x 를 세모에 더 가깝다고 판단함.



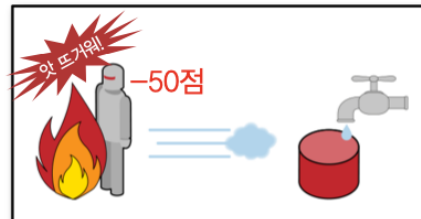
- 구글 포토 호스팅: 가족 사진 몇 장에만 레이블 적용. 이후 모든 사진에서 가족사진 확인 가능.

강화 학습

- 에이전트(학습 시스템)가 취한 행동에 대해 보상 또는 벌점을 주어 가장 큰 보상을 받는 방향으로 유도하기



- 1 관찰
- 2 정책에 따라 행동을 선택

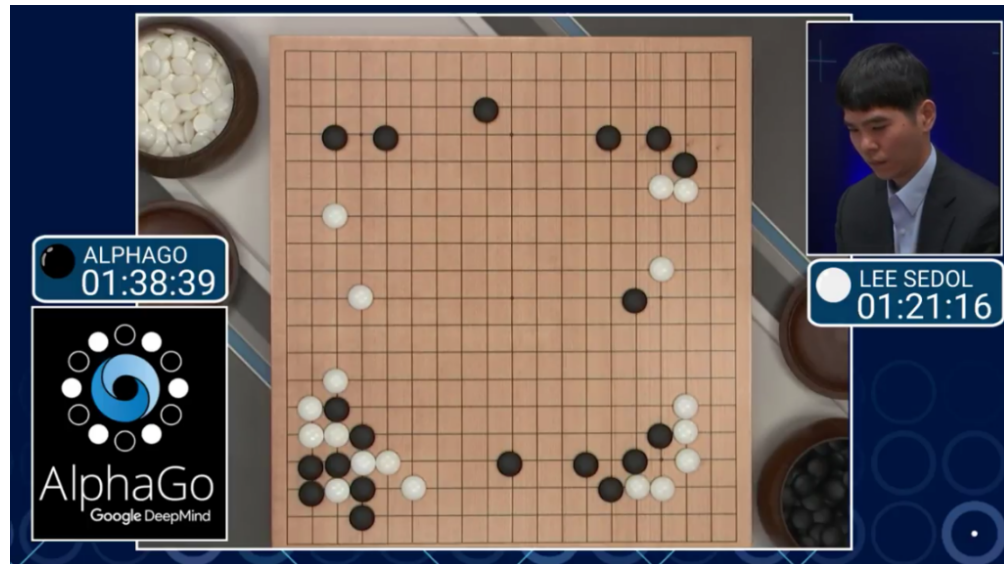


- 3 행동 실행!
- 4 보상이나 벌점을 받음



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

- 예제: 딥마인드(DeepMind)의 알파고(AlphaGo)



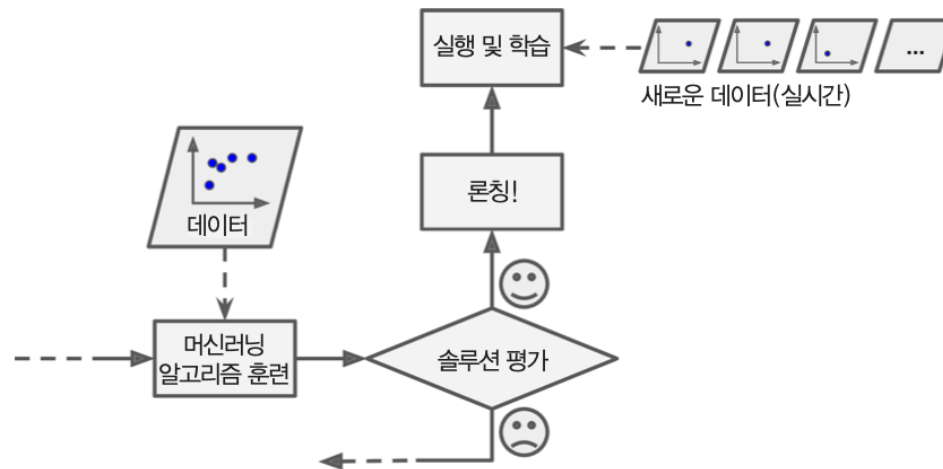
실시간 훈련 여부 구분

배치 학습(batch learning)

- 주어진 훈련 세트 전체를 사용해 오프라인에서 훈련
- 먼저 시스템을 훈련시킨 후 더 이상의 학습 없이 제품 시스템에 적용
- 단점
 - 컴퓨팅 자원(cpu, gpu, 메모리, 저장장치 등)이 충분한 경우에만 사용 가능
 - 새로운 데이터가 들어오면 처음부터 새롭게 학습해야 함.

온라인 학습(online learning)

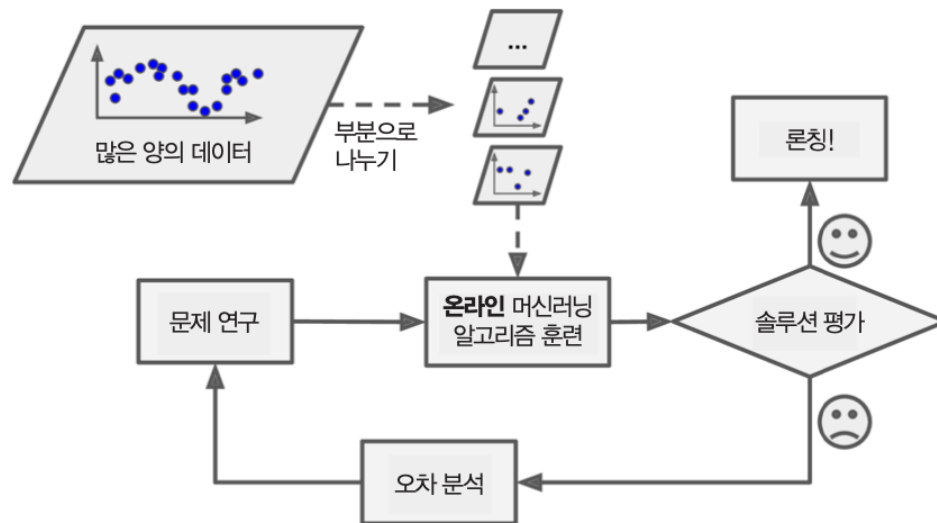
- 하나씩 또는 적은 양의 데이터 묶음(미니배치, mini-batch)를 사용해 점진적으로 훈련



- 단점
 - 나쁜 데이터가 주입되는 경우 시스템 성능이 점진적으로 떨어질 수 있음.
 - 지속적인 시스템 모니터링 필요

- 예제

- 주식가격 시스템 등 실시간 반영이 중요한 시스템
- 스마트폰 등 제한된 자원의 시스템
- 외부 메모리 학습: 매우 큰 데이터셋 활용하는 시스템



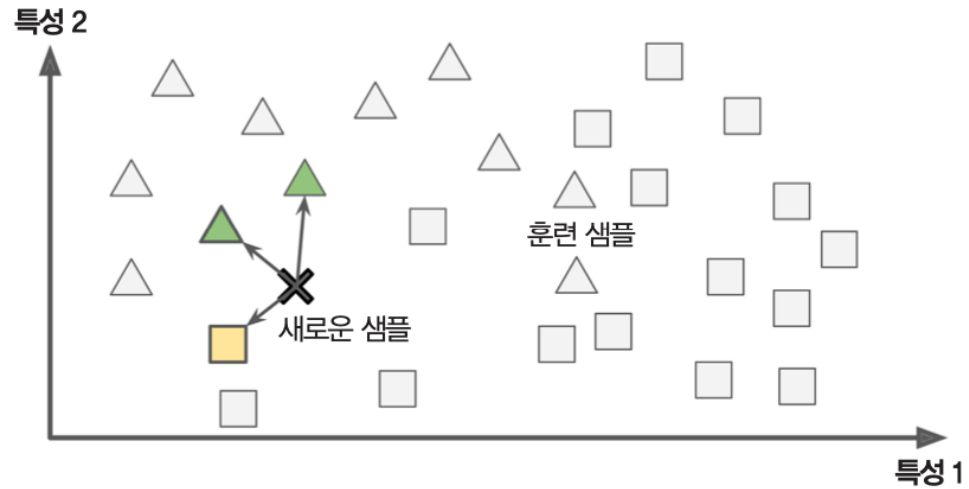
예측 모델 사용 여부 구분

- 훈련 모델의 일반화(**generalization**) 방식에 따른 분류
- 일반화 = '새로운 데이터에 대한 예측'

사례 기반 학습

- 샘플을 기억하는 것이 훈련의 전부
- 예측을 위해 기존 샘플과의 유사도 측정
- 예제: k-최근접 이웃(k-NN, k-nearest neighbors) 알고리즘

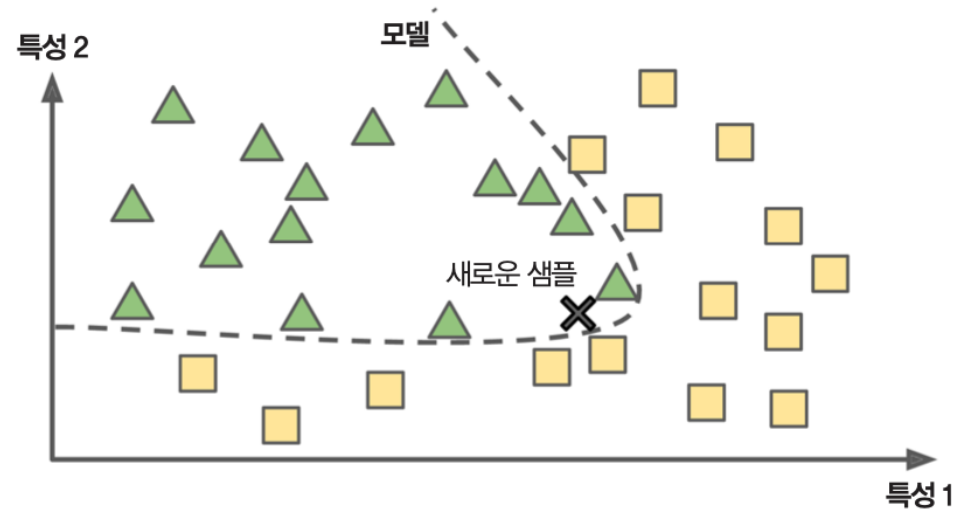
- k-NN 활용 예제: 새로운 샘플 x가 기존에 세모인 샘플과의 유사도가 높기 때문에 세모로 분류.



모델 기반 학습

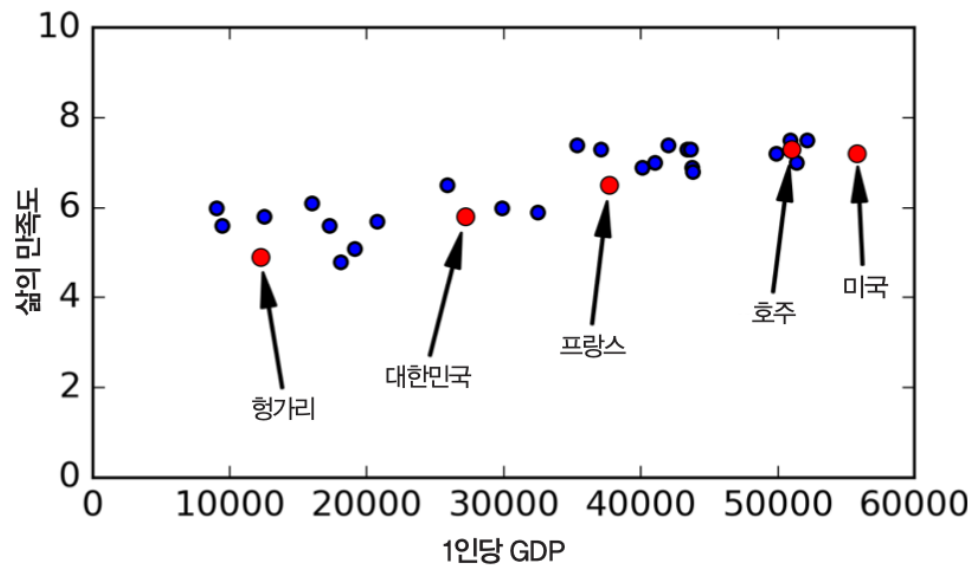
- 모델을 미리 지정한 후 훈련 세트를 사용해서 모델을 훈련시킴
- 훈련된 모델을 사용해 새로운 데이터에 대한 예측 실행
- 예제: 이 책에서 다루는 대부분의 알고리즘

- 예제: 학습된 모델을 이용하여 새로운 데이터 x 를 세로 클래스로 분류



선형 모델 학습 예제

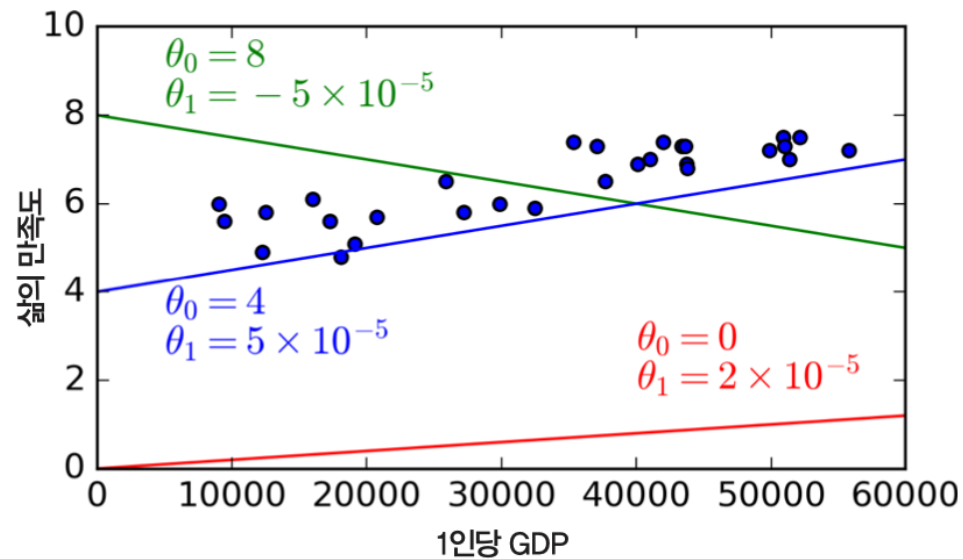
- 목표: OECD 국가의 1인당 GDP(1인당 국가총생산)와 삶의 만족도 사이의 관계 파악
- 1인당 GDP가 증가할 수록 삶의 만족도가 선형으로 증가하는 것처럼 보임.



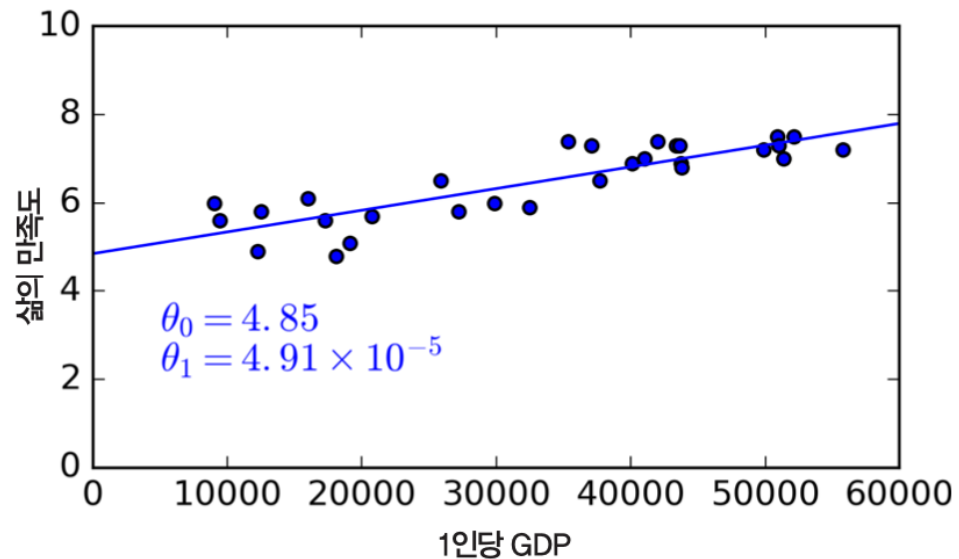
- 데이터를 대표하는 하나의 직선(선형 모델)을 찾기

$$\text{'삶의만족도'} = \theta_0 + \theta_1 \times \text{'1인당GDP'}$$

- 데이터를 대표할 수 있는 선형 방정식을 찾아야 함



- 학습되는 모델의 성능 평가 기준을 측정하여 가장 적합한 모델 학습
 - 효용 함수: 모델이 얼마나 좋은지 측정
 - 비용 함수: 모델이 얼마나 나쁜지 측정
- 아래 선형 모델이 최적!

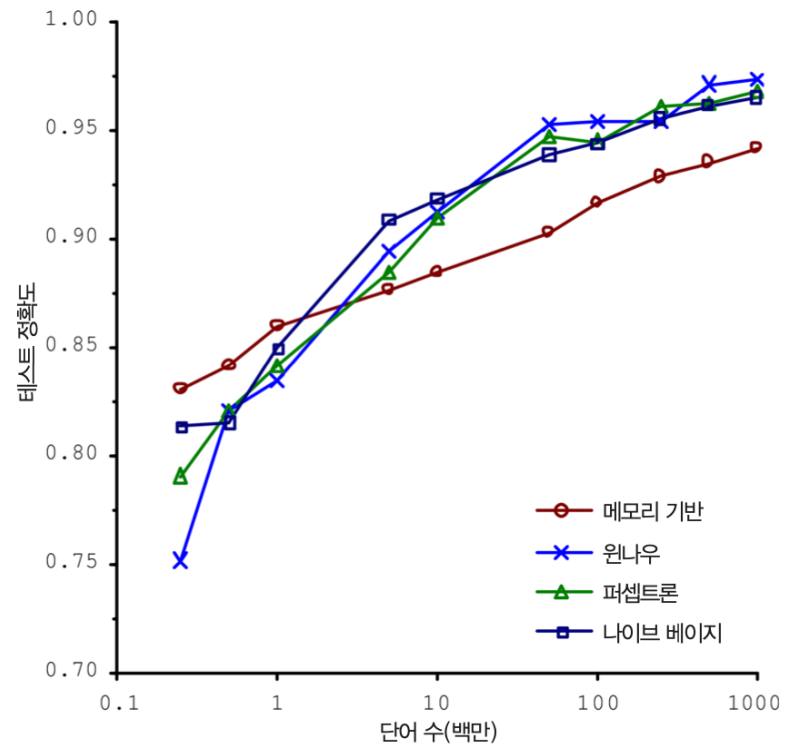


1.5 머신러닝의 주요 도전 과제

충분하지 않은 양의 훈련 데이터

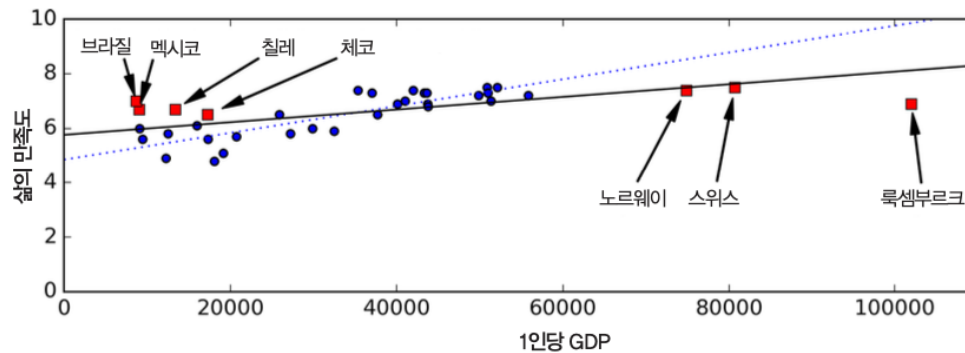
- 간단한 문제라도 수천 개 이상의 데이터가 필요
- 이미지나 음성 인식 같은 문제는 수백만 개가 필요할 수도 있음
- 데이터가 부족하면 알고리즘 성능 향상 어려움

- 일반적으로 데이터가 많을 수록 모델의 성능 높아짐.



대표성 없는 훈련 데이터

- 샘플링 잡음: 우연에 의해 추가된 대표성이 없는 데이터
- 샘플링 편향: 표본 추출 방법이 잘못되어 한 쪽으로 쏠린 대표성이 없는 데이터
- 예제: 1인당 GDP와 삶의 만족도 관계
 - 잡음: 빨강 네모 데이터가 추가 될 경우 선형 모델 달라짐.
 - 편향: OECD 국가중에서 이름에 영어 알파벳 W가 포함된 국가들은 삶의 만족도가 매우 높음. 하지만 일반화는 불가능.



낮은 품질의 데이터 처리

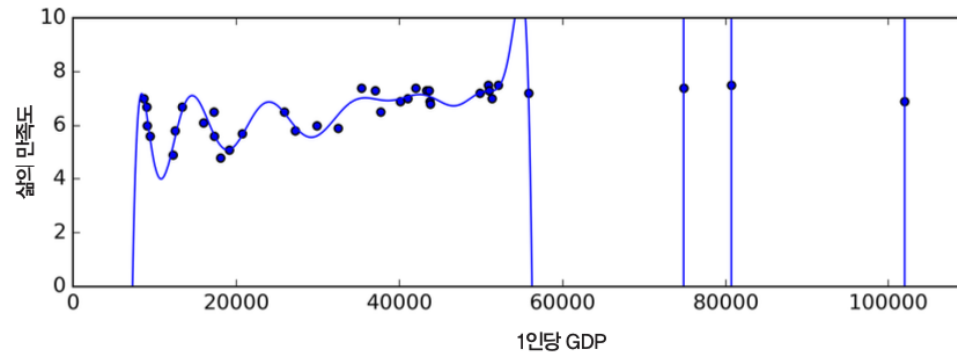
- 이상치 샘플이라면 고치거나 무시
- 특성이 누락되었다면
 - 해당 특성을 제외
 - 해당 샘플을 제외
 - 누락된 값을 채움
 - 해당 특성을 넣은 경우와 뺀 경우 각기 모델을 훈련

관련이 없는 특성

- 풀려는 문제에 관련이 높은 특성을 찾아야 함
- 특성 선택: 준비되어 있는 특성 중 가장 유용한 특성을 찾음
- 특성 추출: 특성을 조합하여 새로운 특성을 만들

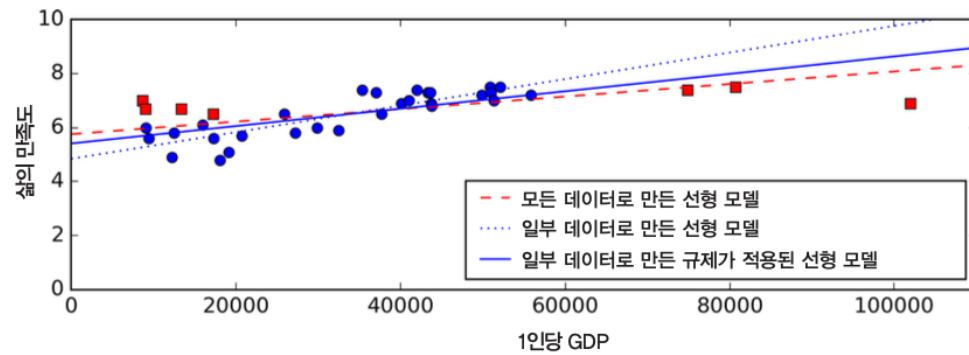
과대적합

- 훈련 세트에 특화되어 일반화 성능이 떨어지는 현상



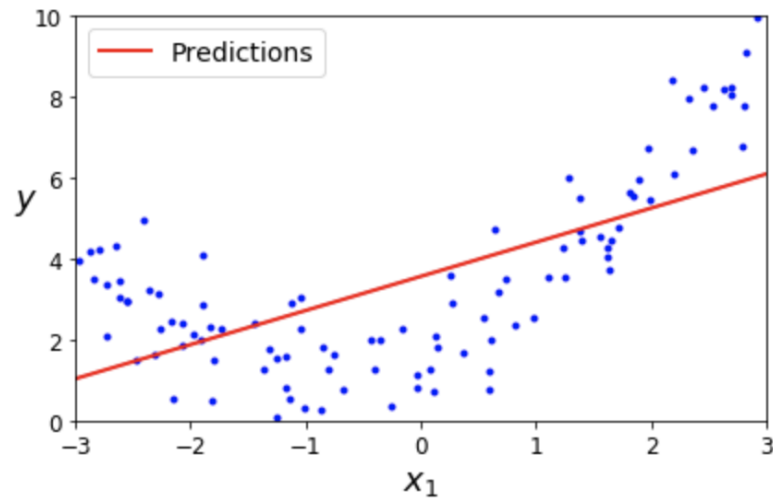
- 규제를 적용해 과대적합을 감소시킬 수 있음
- 파라미터를 조정되는 과정에 규제 적용

- 파랑 점선이 규제를 적용해 훈련된 선형 모델임.



과소적합

- 모델이 너무 단순해서 훈련 세트를 잘 학습하지 못함



- 해결 방법
 - 보다 많은 모델 파라미터를 사용하는 모델 적용
 - 보다 좋은 특성 활용
 - 보다 규제 강도 적용

1.6 테스트와 검증

검증

- 훈련된 모델의 성능 평가: 테스트 세트 활용
- 전체 데이터셋을 훈련 세트(80%)와 테스트 세트(20%)로 구분
 - 훈련 세트: 모델 훈련용.
 - 테스트 세트: 모델 테스트용
 - 데이터셋이 매우 크면 테스트 세트 비율을 낮출 수 있음.
- 검증 기준: **일반화 오차**
 - 새로운 샘플에 대한 오류 비율
 - 학습된 모델의 일반화 성능의 기준
- 과대 적합: 훈련 오차에 비해 일반화 오차가 높은 경우

하이퍼파라미터(hyper-parameter)

- 알고리즘 학습 모델을 지정에 사용되는 파라미터
- 훈련 과정에 변하는 파라미터가 아님
- 하이퍼파라미터를 조절하면서 가장 좋은 성능의 모델 선정

교차 검증

- 예비표본(홀드아웃, holdout) 검증
 - 예비표본(검증세트): 훈련 세트의 일부로 만들어진 데이터셋
 - 다양한 하이퍼파라미터 값을 사용하는 후보 모델을 평가하는 용도로 예비표본을 활용하는 기법
- 교차 검증
 - 여러 개의 검증세트를 사용한 반복적인 예비표본 검증 적용 기법
 - 장점: 교차 검증 후 모든 모델의 평가를 평균하면 훨씬 정확한 성능 측정 가능
 - 단점: 훈련 시간이 검증 세트의 개수에 비례해 늘어남

검증 예제: 데이터 불일치

- 모델 훈련에 사용된 데이터가 실전에 사용되는 데이터를 완벽하게 대변하지 못하는 경우
- 예제: 꽃이름 확인 알고리즘
 - 인터넷으로 구한 꽃사진으로 모델 훈련
 - 이후 직접 촬영한 사진으로 진행한 성능측정이 낮게 나오면 **데이터 불일치** 가능성 높음

- 데이터 불일치 여부 확인 방법
 - 훈련-개발 세트: 예를 들어, 인터넷에서 다운로드한 꽃사진의 일부로 이루어진 데이터셋
 - 훈련-개발 세트를 제외한 나머지 꽃사진으로 모델 훈련 후, 훈련-개발 세트를 이용한 성능 평가 진행
- 훈련-개발 세트에 대한 평가가 좋은 경우: 과대적합 아님
 - 훈련-개발 세트에 평가는 좋지만 (실제 찍은 사진으로 이루어진) 검증 세트에 대한 평가 나쁜 경우: 데이터 불일치
 - 다운로드한 사진을 실제 찍은 사진처럼 보이도록 전처리 한 후에 다시 훈련시키면 성능 향상시킬 수 있음.
- 훈련-개발 세트에 대한 평가가 나쁜 경우: 과대적합
 - 모델에 규제를 적용하거나 더 많은 훈련 데이터 활용해야 함.