Ch01_한눈에 보는 머신러닝_0714

▼ **내용**

- 지도학습과 비지도학습
- 온라인 학습과 배치 학습
- 사례 기반 학습과 모델 기반 학습

▼ **1.4 머신러닝 시스템의 종류**

▼ 요약

1) 지도학습, 비지도 학습, 준지도 학습, 강화 학습

: 사람의 감독 하에 훈련하는 것인지 아닌지의 여부

2) 배치 학습 vs 온라인 학습

: 입력 데이터의 스트림부터 실시간으로 점진적으로 학습할 수 있는지의 여부

3) 사례 기반 학습 vs 모델 기반 학습

: 단순하게 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것 vs 훈련 데이터셋에서 패턴을 발견하여 예측 모델을 만드는 것

* 각 범주들은 서로 배타적이면서 원하는 대로 연결 가능하다.

ex) 온라인 학습이면서 모델 기반이면서 지도 학습 시스템일 수 있다.

▼ **1.4.1 지도 학습, 비지도 학습, 준지도 학습, 강화 학습**

: 사람의 감독 하에 훈련하는 것인지 아닌지의 여부

> **지도 학습 (supervised learning)**

: 알고리즘에 주입하는 훈련 데이터에 레이블(원하는 답)이 포함되어 있다.

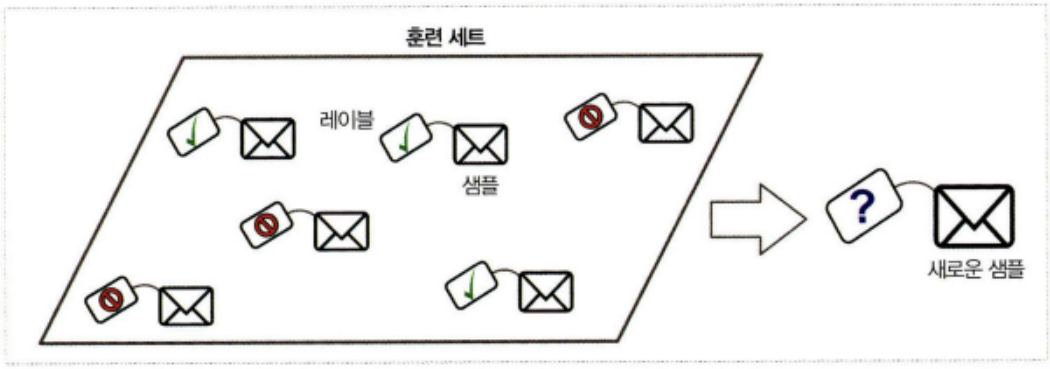


그림 1-5) 스팸 분류를 위한 레이블된 훈련 세트(지도 학습의 예)

지도 학습 작업

- **분류(classification)**

: 주어진 데이터를 클래스 별로 구별해 내는 과정으로, 다양한 분류 알고리즘을 통해 데이터와 데이터의 레이블 값을 학습시키고 모델을 생성한다. 데이터가 주어졌을 때 학습된 모델을 통해 어느 범주에 속한 데이터인지 판단하고 예측하게 된다.

1) 회귀(regression)

: 특성(예측변수)를 이용해 타깃 수치를 예측하는 알고리즘이다.

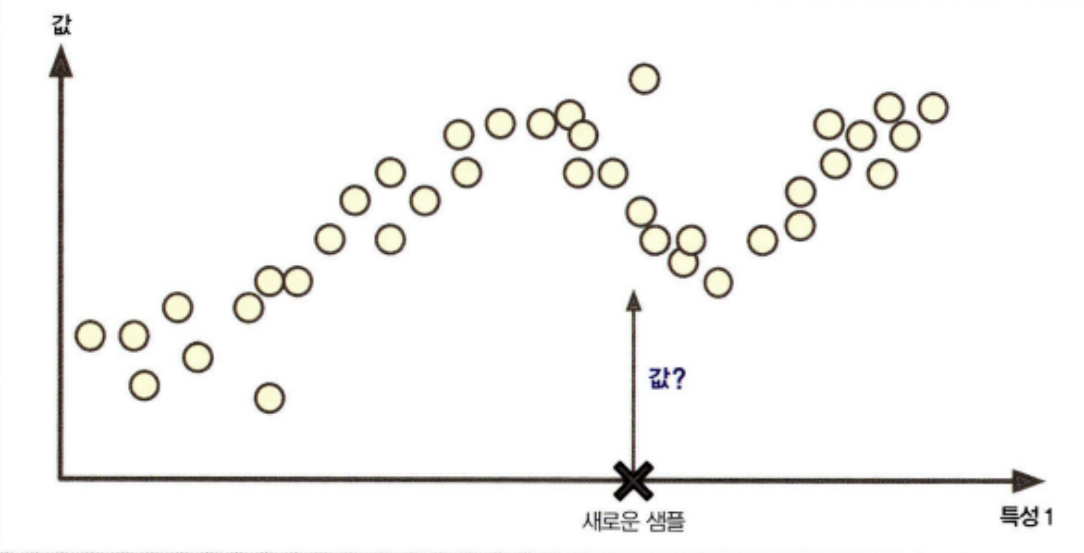


그림 1-6) 회귀 문제

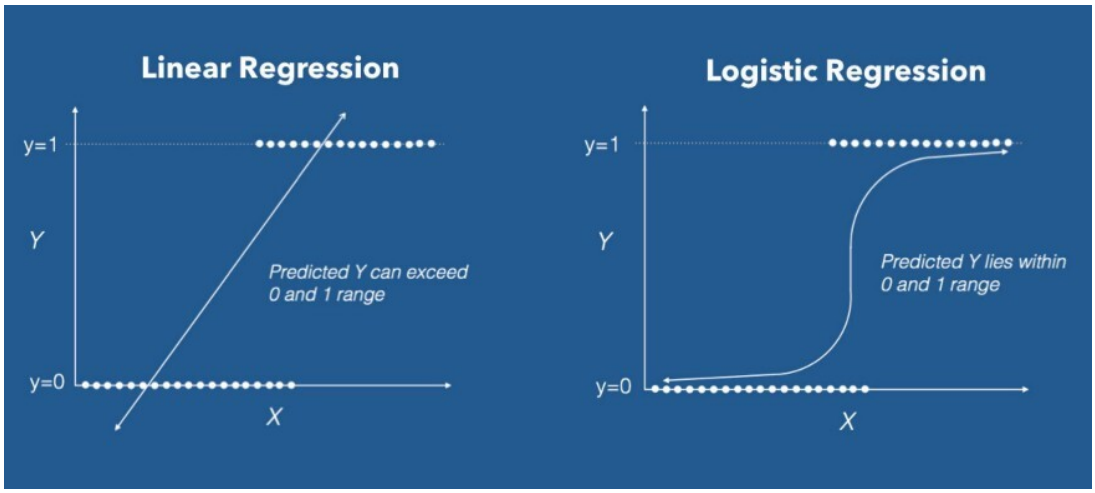
2) 로지스틱 회귀(logistic regression)

: 분류에 널리 쓰이며, 일반적이고 효과적인 분류 알고리즘이다. 범주형 자료를 분류하여 클래스에 속할 확률을 출력하는 알고리즘이다. 독립변수와 종속변수의 선형 관계성을 기반으로 만들어진다.

- **선형 회귀(Linear Regression) vs 로지스틱 회귀(Logistic Regression)**

: 공통적으로 독립변수와 종속변수의 관계로 생성되며, 데이터 값을 0에서 1사이의 값으로 매핑시켜 사용한다.

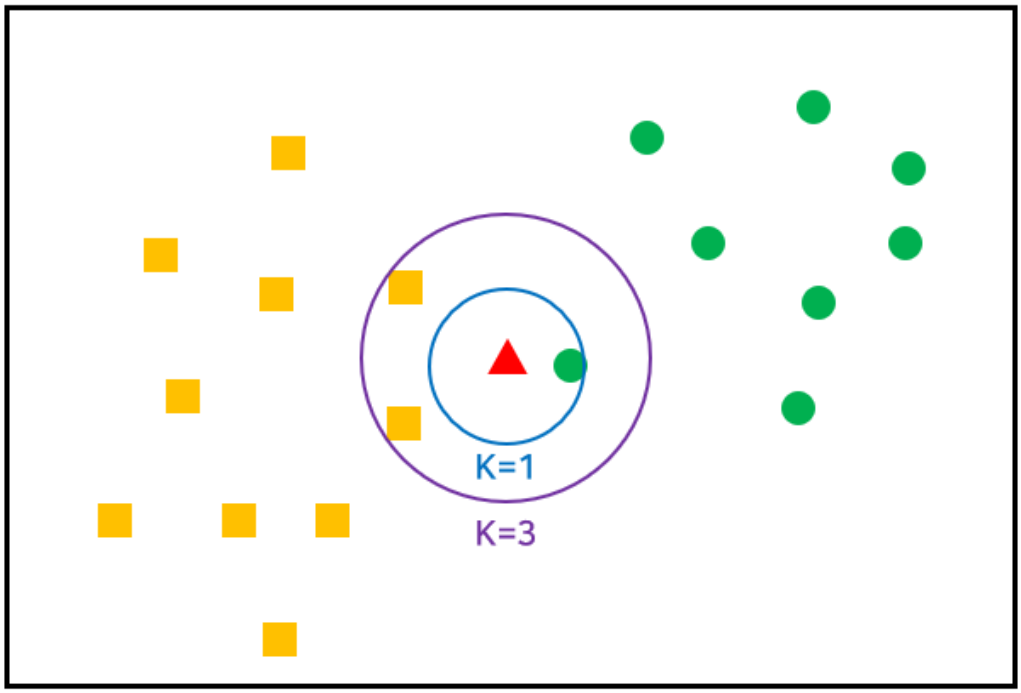
: 선형 회귀는 연속형 변수를 예측하는 데 사용하는 반면 로지스틱 회귀는 범주형 변수를 예측하는데 사용된다.



- **그 외의 지도 학습 알고리즘 종류**

- linear regression, logistic regression

- KNN(k-nearest heighbors)

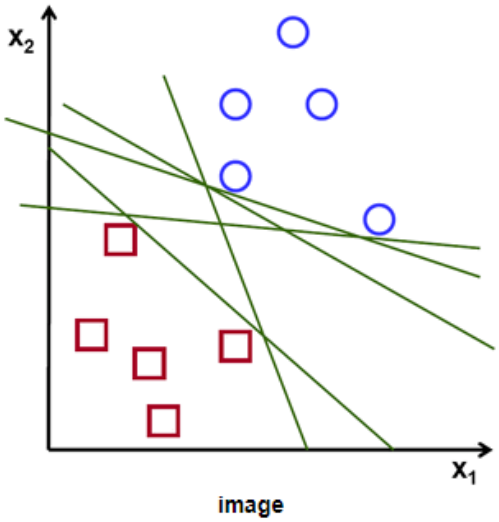


: 주변의 가장 가까운 K개의 데이터를 보고 데이터가 속할 그룹을 판단하는 알고리즘이다.

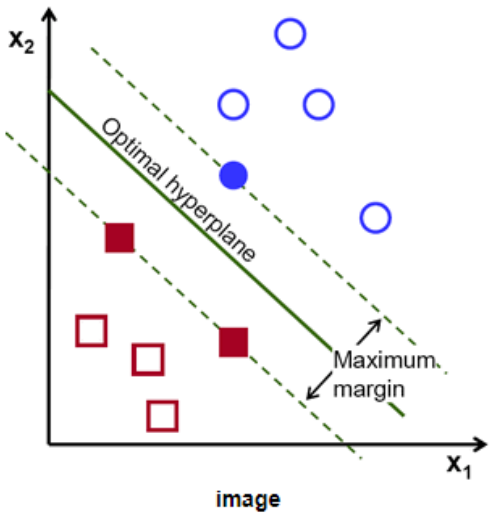
K-NN 알고리즘은, 단순히 훈련 데이터셋을 그냥 저장하는 것이 모델을 만드는 과정의 전부이다. 이때, 거리 측정에는 유클리드 거리(Euclidean distance)를 사용한다.

- support vector machine (SVM)

: 기본적으로 두 개의 그룹(데이터)을 분리하는 방법으로, 데이터들과 거리가 가장 먼 초평면(hyperplane)을 선택하여 분리하는 방법이다.



image

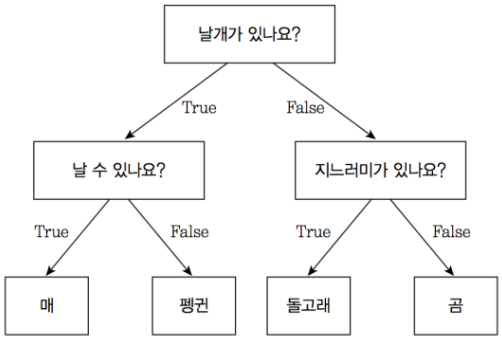


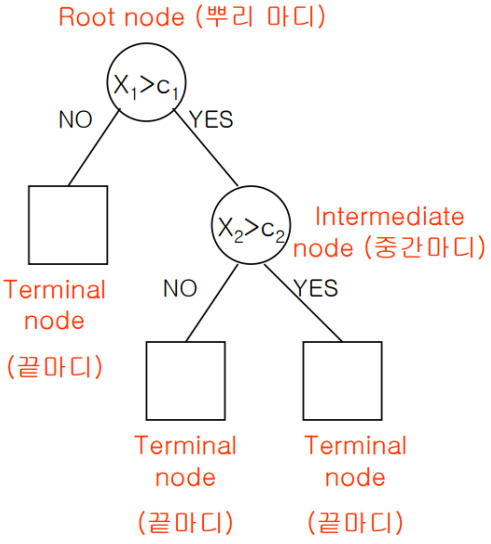
image

데이터를 분리하기 위해 필요한 직선이 한쪽 데이터로 치우쳐져 있으면 데이터에 변동이나 이즈가 있을 때 제대로 구분할 수 없기 때문에, Maximum margin을 갖는 'Optimal hyperplane'을 구는 방향으로 분류를 수행한다. 이때, margin은 초평면과 가장 가까이 있는 데이터와의 거리를 의미한다.

- decision tree

: 특정 기준(질문)에 따라 데이터를 구분하는 모델이다. 한번의 분기 때마다 변수 영역을 두 개로 구분한다.

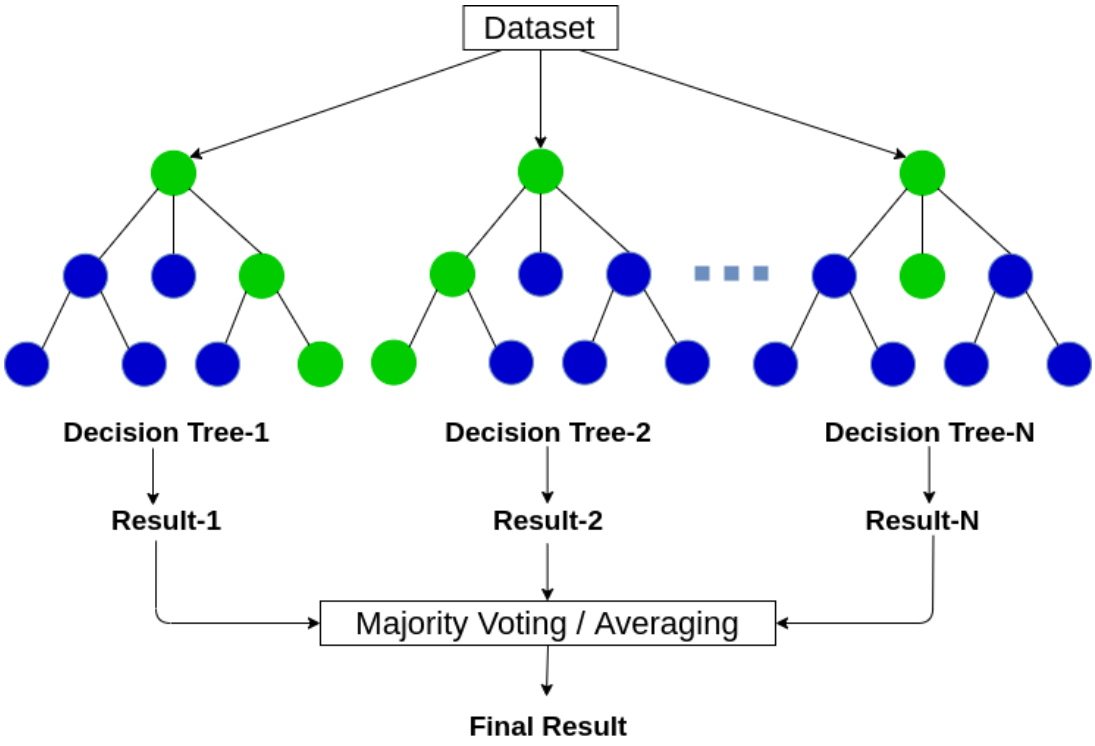




: 결정 트리에서 질문이나 정답을 담은 네모 상자를 노드(Node)라고 한다. 맨 처음 분류 기준 (즉, 첫 질문)을 Root Node라고 하고, 맨 마지막 노드를 Terminal Node 혹은 Leaf Node라고 한다. 이때, 질문이 너무 많아지면 오버피팅 문제가 발생한다.

- random forest

: 여러 개의 결정 트리를 만들고, 여러 결정 트리들이 각각 내린 예측 값들 중 가장 많이 나온 값을 최종 예측값으로 정한다. 의사결정 트리의 오버피팅 한계를 극복할 수 있다.



- neural network

비지도 학습 (unsupervised learning)

: 훈련 데이터에 레이블이 없는 경우로, 시스템이 아무런 도움 없이 학습해야 한다.

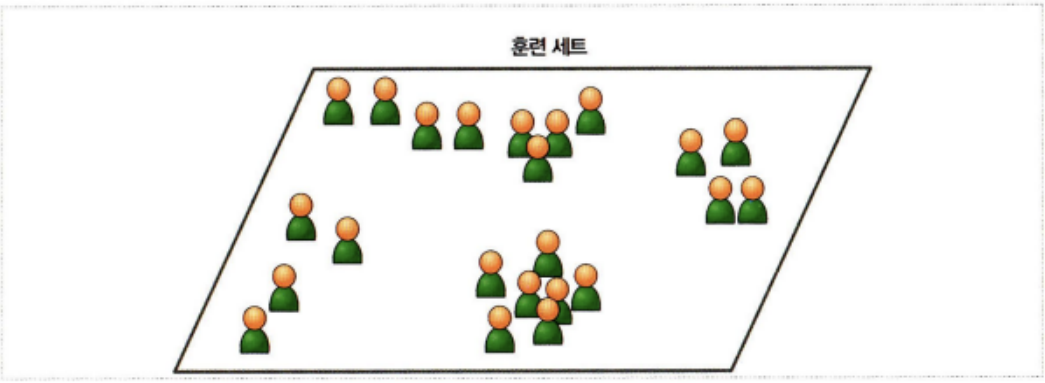


그림 1-7) 비지도 학습에서 레이블이 없는 훈련 세트

- **비지도 학습 알고리즘**

 # clustering
• k-means (k-평균)
• DBSCAN
• hierarchical cluster analysis (계층 군집)
• outlier detection (이상치 탐지)

: 학습 알고리즘에 넣기 전 데이터 셋에서 이상한 값을 자동으로 제거하는 것
• novelty detection(특이치 탐지)

: 훈련 세트에 있는 모든 샘플과 달라 보이는 새로운 샘플을 탐지하는 것이 목적
• one-class SVM
• isolation forest

 # visualization and dimension reduction (시각화와 차원축소)
• principal component analysis (PCA; 주성분 분석)
• kernel PCA
• locally-linear embedding (LLE)
• t-distributed stochastic neighbor embedding (t-SNE)

• association rule learning
• Apriori
• Eclat

- **비지도 학습의 예시**

1) 군집화

ex) 블로그 방문자에 대한 데이터가 많이 존재할 때, 비슷한 방문자들을 그룹으로 묶기 위해 군집 알고리즘을 적용하고자 한다. 하지만 방문자가 어떤 그룹에 속하는지 알고리즘에 알려줄 수 있는 데이터 포인트가 없을 때, 알고리즘이 스스로 방문자 사이의 연결 고리를 찾는다. 이때 계층 군집 알고리즘을 사용하면 각 그룹을 더 작은 그룹으로 세분화 할 수 있다.

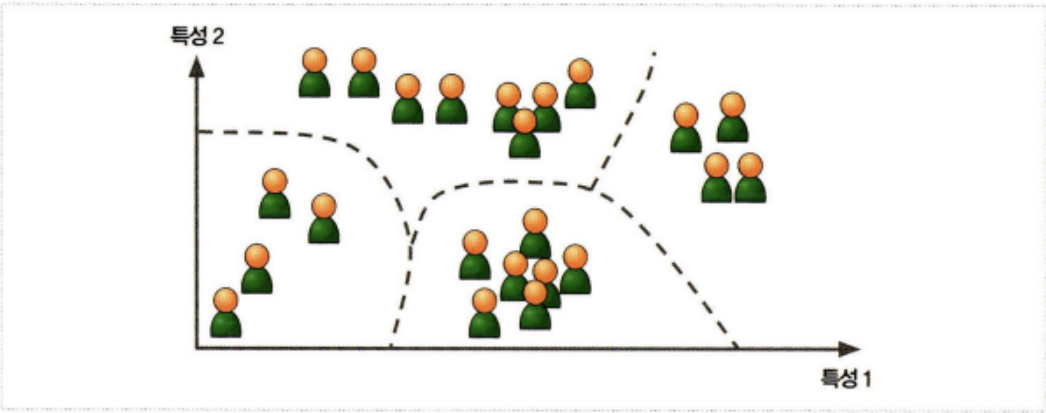


그림 1-8) 군집

2) 시각화 알고리즘

: 레이블이 없는 대규모의 고차원 데이터를 넣으면 도식화가 가능한 2D, 3D 표현을 만들어준다. 이 알고리즘은 가능한 한 구조를 그대로 유지하려 하므로 데이터가 어떻게 조직화 되어 있는지 이해할 수 있고, 예상하지 못한 패턴을 발견할 수도 있다.

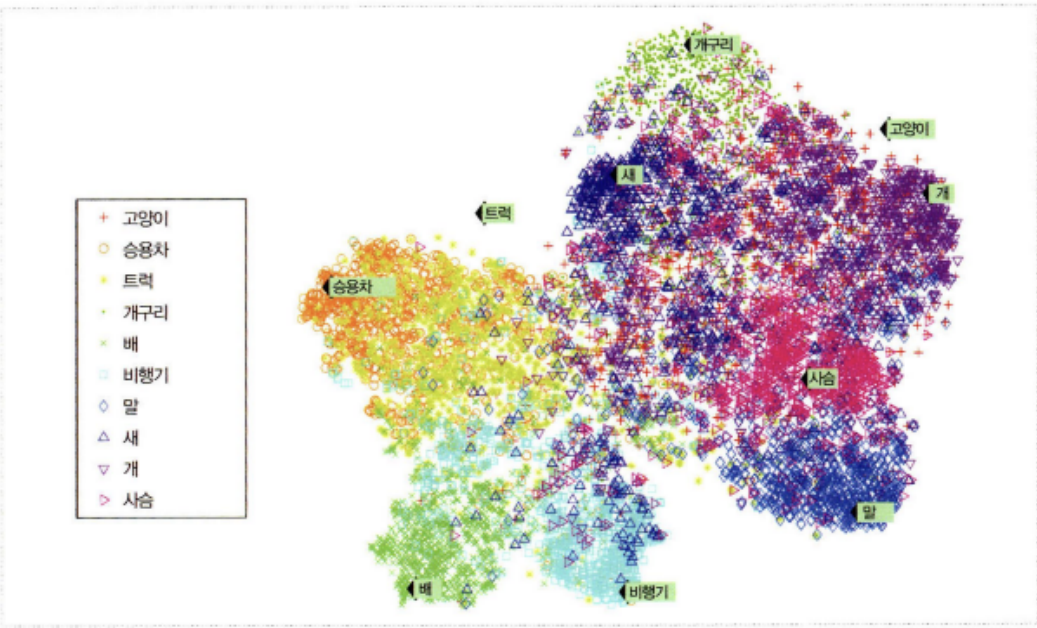


그림 1-9 의미 있는 군집을 강조한 t-SNE 시각화의 예[8]

3) 차원 축소

: 너무 많은 정보를 잃지 않으면서 데이터를 간소화하는 것. 상관 관계가 있는 여러 특성을 하나로 합치는 것이다.

4) 이상치 탐지

: 학습 알고리즘에 넣기 전 데이터셋에서 이상한 값을 자동으로 제거하는 것. 시스템은 훈련하는 동안 대부분 정상 샘플을 만나 이를 인식하도록 훈련된다. 그다음 새로운 샘플을 보고 정상 데이터인지 혹은 이상치인지 판단한다.

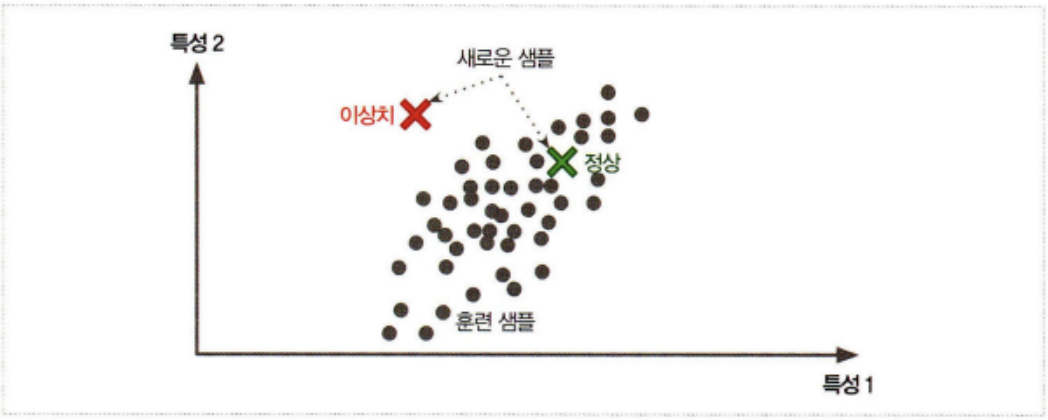


그림 1-10 이상치 탐지

5) 특이치 탐지

: 훈련 세트에 있는 모든 샘플과 달라 보이는 새로운 샘플을 탐지하는 것이 목적이다. 알고리즘으로 감지하고 싶은 모든 샘플을 제거한 매우 깨끗한 훈련 세트가 필요하다.

- **이상치 탐지 vs 특이치 탐지**

:강아지 사진 수천 장 중에 1%가 치와와 사진이면, 특이치 탐지 알고리즘은 치와와 사진을 새로운 특이한 것으로 처리하지 못한다. 반면, 이상치 탐지 알고리즘은 치와와 사진을 다른 강아지와 다르다고 인식하여 이상치로 분류한다.

준지도 학습 (semisupervised learning)

: 일부만 레이블이 있는 데이터를 다루는 알고리즘으로, 지도 학습과 비지도 학습의 조합으로 이루어져 있다.

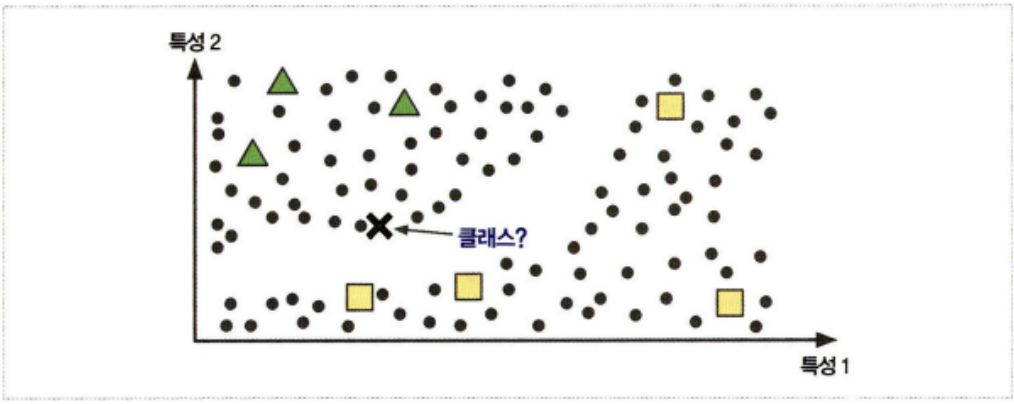


그림 1-11 두 개의 클래스(삼각형과 사각형)를 사용한 준지도 학습: 새로운 샘플(곱셈 기호)이 레이블이 있는 사각형 클래스에 더 가깝지만 레이블이 없는 샘플(원)이 이 샘플을 삼각형 클래스로 분류하는 데 도움을 줍니다.

- **심층 신뢰 신경망(DNN ; deep neural network)**

: 여러 겹으로 쌓은 restricted Boltzmann machine (RBM; 제한된 볼츠만 머신) 비지도 학

습에 기초하는 방법으로, 순차적으로 훈련한 후 지도 학습 방식으로 세밀하게 조정된

다.

강화 학습 (reinforcement learning)

: Agent (학습 시스템) 환경을 관찰해서 행동을 실행하고 그 결과로 보상 또는 벌점을 받는다. 시간이 지나면서 가장 큰 보상을 얻기 위해 최상의 전략인 정책(policy)을 스스로 학습한다. 이때, 정책은 주어진 상황에서 agent 가 어떤 행동을 해야 할지를 정의한다.

ex) Deepmind의 Alphago 프로그램

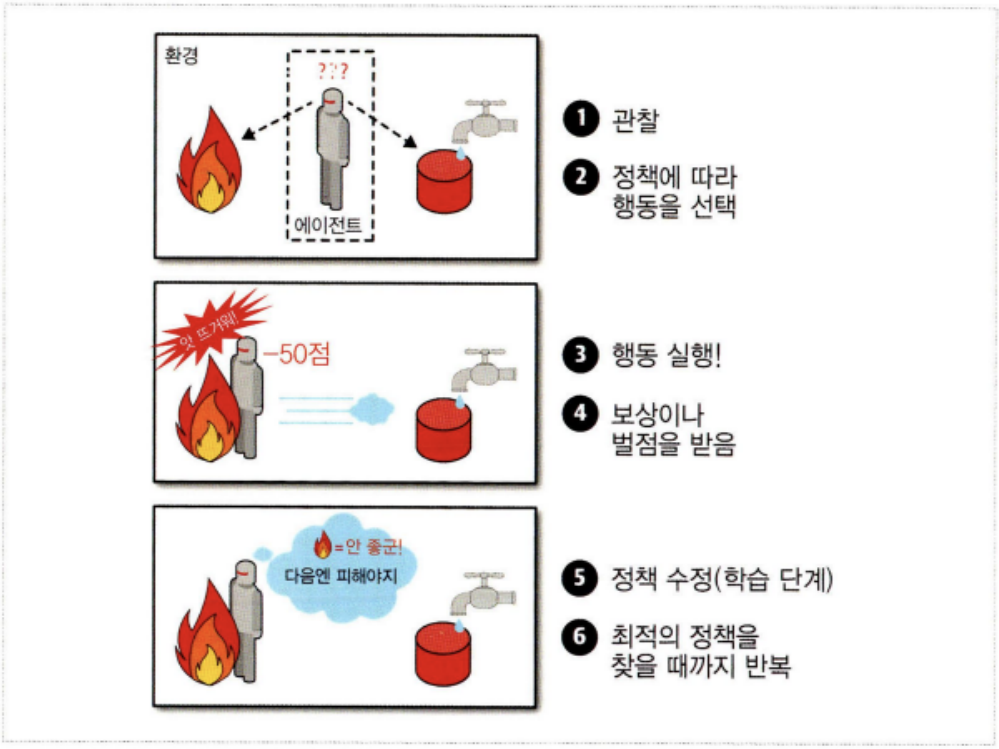


그림 1-12 강화 학습

▼ 1.4.2 배치학습과 온라인 학습

- 기준 : 입력 데이터의 스트림부터 실시간으로 점진적으로 학습할 수 있는지의 여부

배치 학습 (batch learning)

: 오프라인 학습이라고도 하며, 입력 데이터의 스트림부터 시스템이 점진적으로 학습할 수 없고, 가용한 데이터를 모두 사용해서 훈련해야 한다.

: 먼저 시스템을 훈련 시키고 그런 다음 제품 시스템에 적용하면 더 이상의 학습 없이 실행된다. 즉, 학습한 것을 단지 적용만 한다.

: 새로운 데이터에 대해 학습하려면 새로운 데이터뿐만 아니라 이전 데이터도 모두 포함한 전체 데이터를 사용해 처음부터 다시 학습시켜야 합니다.

: 시간과 자원을 많이 소모하므로 보통 오프라인에서 수행된다.

온라인 학습

: 데이터를 순차적으로 한 개씩 또는 미니 배치(mini-batch : 작은 묶음 단위)로 주입하여 시스

템을 훈련시킨다.

: 연속적으로 데이터를 받고 빠른 변화에 스스로 적응해야 하는 시스템에 적합하다.

: 아주 큰 데이터셋을 학습하는 외부 메모리 학습(out of core 학습)에도 사용할 수 있다. 알고리

즘이 데이터의 일부를 읽어 들이고 훈련 단계를 수행한 후, 전체 데이터가 모두 적용될 때 까지

이 과정을 반복한다.

: 데이터 양이 너무 많아 배치 학습 알고리즘을 사용하기 어려운 경우에 적용된다.

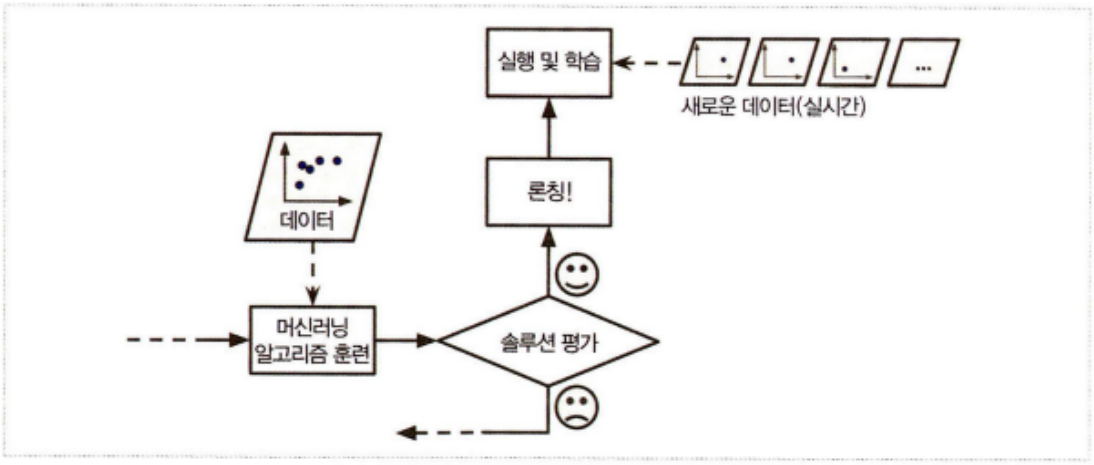


그림 1-13 온라인 학습에서 모델을 훈련하고 제품에 론칭한 뒤에도 새로운 데이터가 들어오면 계속 학습합니다.

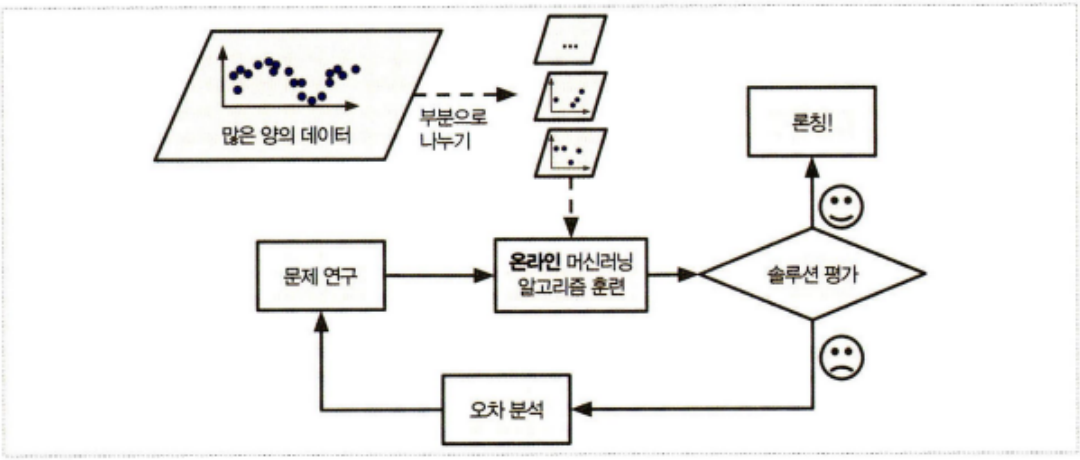


그림 1-14 온라인 학습을 사용한 대량의 데이터 처리

- **학습률(learning rate)**

- 온라인 학습 시스템에서 중요한 파라미터로, 변화하는 데이터를 얼마나 바르게 적응하는지를 나타낸다.

  - 높은 학습률 : 시스템이 데이터에 빠르게 적응하지만, 예전 데이터를 빨리 잊음

  - 낮은 학습률 : 시스템이 느리게 학습하지만, 새로운 데이터의 잡음이나 대표성 없는 데이터에 덜 민감하다

- **온라인 학습의 문제점**

: 시스템에 나쁜 데이터가 주입 되었을 때 시스템 성능이 점진적으로 감소한다.

▼ 1.4.3 사례 기반 힉습과 모델 기반 학습

- 기준 : 어떻게 일반화 되는가. 머신러닝은 주어진 훈련 데이터로 학습하고 훈련 데이터에서는 본 적 없는 새로운 데이터에서 좋은 예측을 만들어야 (일반화 되어야) 한다.

사례 기반 학습 (instance-based learning)

: 시스템이 훈련 샘플을 기억함으로써 학습하고, 유사도(similarity) 측정을 사용해 새로운 데이터와 학습한 샘플을 비교하는 식으로 일반화한다.

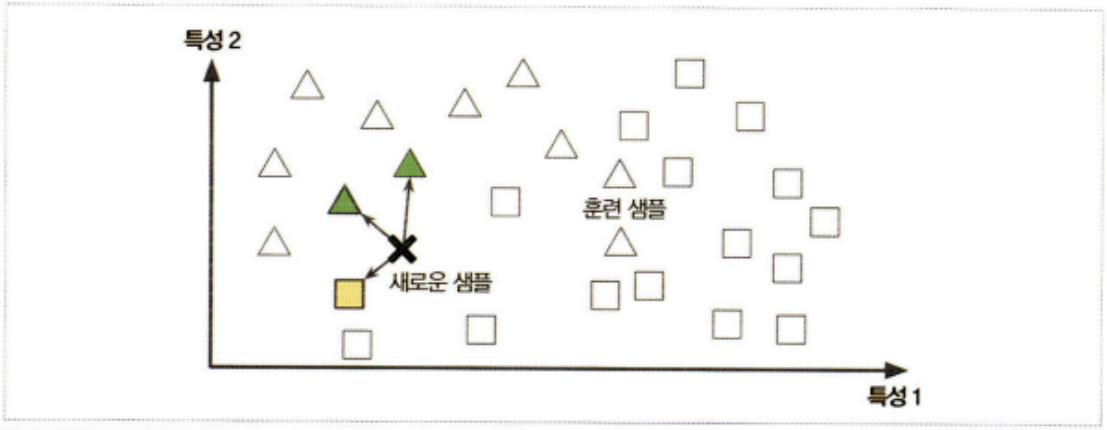


그림 1-15 사례 기반 학습

모델 기반 학습 (model-based learning)

: 샘플들의 모델을 만들어 예측에 사용하는 것

ex) GDP 에 따라 삶의 만족도가 어떻게 달라지는지를 알아보기 위해 선형 모형 이용
선형 모델 | Quality of life = $\theta 0 + \theta 1$*GDP

- 모델의 파라미터($\theta 0$, $\theta 1$)를 조정하여 어떤 선형 함수를 표현하는 모델을 얻을 수 있다.

- 모델을 사용하기 전 파라미터($\theta 0$, $\theta 1$)를 정의 해야 한다.

- **측정 지표** : 모델이 최상의 성능을 내도록 하는 값을 찾기 위해 사용한다.

- 효용 함수(utility function) 또는 적합도 함수(fitness function) : 모델이 얼마나 좋은지 측정

- 비용 함수(loss function) : 모델이 얼마나 나쁜지 측정

- 선형 회귀에서는 보통 선형 모델의 예측과 훈련 데이터 사이의 거리를 재는 비용 함수를 사용하며, 이 거리를 최소화 하는 것이 목표이다.

- **모델의 훈련(training)**

: 선형 회귀 알고리즘에서, 알고리즘에 훈련 데이터를 공급하여 데이터에 가장 잘 맞는 선형 모델의 파라미터를 찾는 것

- **예측 (prediction)**

: 훈련을 통해 얻어진 모델을 사용하여 예측