

# 1. 한눈에 보는 머신러닝

## 1.1 머신러닝이란?

## 아서 새뮤얼 Artuhr Samuel (1959)

컴퓨터 프로그램을 명시적으로 구현하는 대신 컴퓨터 스스로 학습하는 능력을 갖도록 하는 연구 분야

## 톰 미첼 Tom Mitchell (1977)

*과제  $T$ 에 대한 프로그램의 성능  $P$ 가 경험  $E$ 를 통해 향상되면 해당 "프로그램이 경험  $E$ 를 통해 학습한다" 라고 말한다.*

- 스팸 필터 예제
  - 작업  $T$  = 메일의 스팸 여부 판단
  - 경험  $E$  = 훈련 데이터
  - 성능  $P$  = 메일의 스팸 여부 판단의 정확도

## 데이터셋 용어

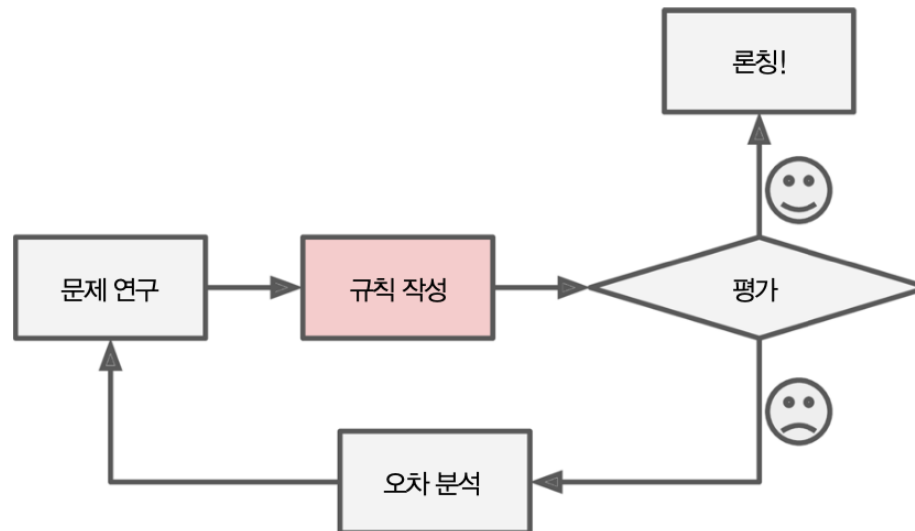
- **훈련셋**: 머신러닝 프로그램이 훈련(학습)하는 데 사용하는 데이터 집합
- **훈련 사례** 혹은 **샘플**: 각각의 훈련 데이터

## 1.2 머신러닝 활용

# 전통적인 프로그래밍

전통적 프로그래밍 다음 과정으로 진행된다.

1. 문제 연구: 문제 해결 알고리즘 연구
2. 규칙 작성: 알고리즘 구현
3. 평가: 구현된 프로그램 테스트
  - 테스트 통과: 프로그램 론칭
  - 테스트 실패: 오차 분석 후 1단계로 이동

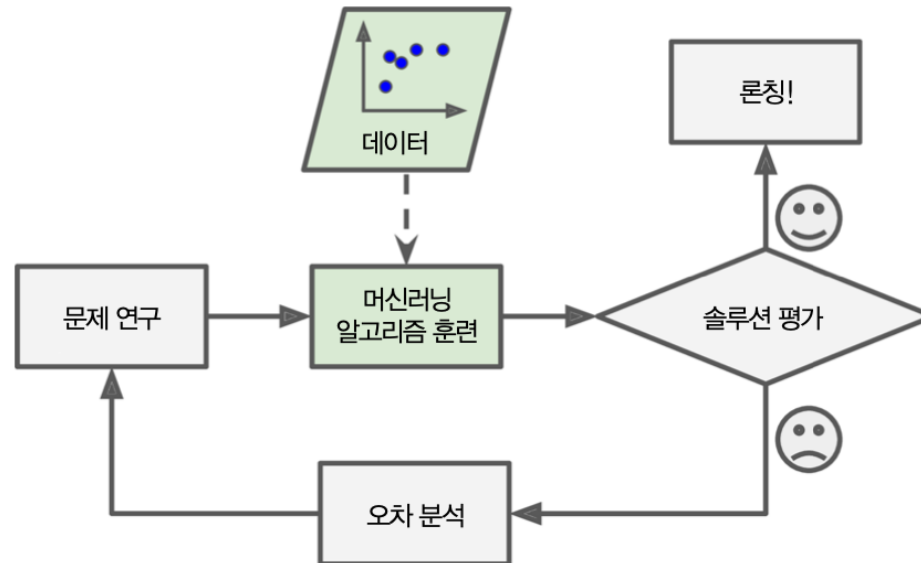


## 예제: 스팸 메일 분류

- 특정 단어가 들어가면 스팸 메일로 처리
- 프로그램이 론칭된 후 새로운 스팸단어가 사용될 때 스팸 메일 분류 실패
- 개발자가 새로운 규칙을 업데이트 시켜줘야 함
- 새로운 규칙이 생겼을 때 사용자가 매번 업데이트를 시켜줘야하기 때문에 유지 보수가 어려움



# 머신러닝 프로그래밍

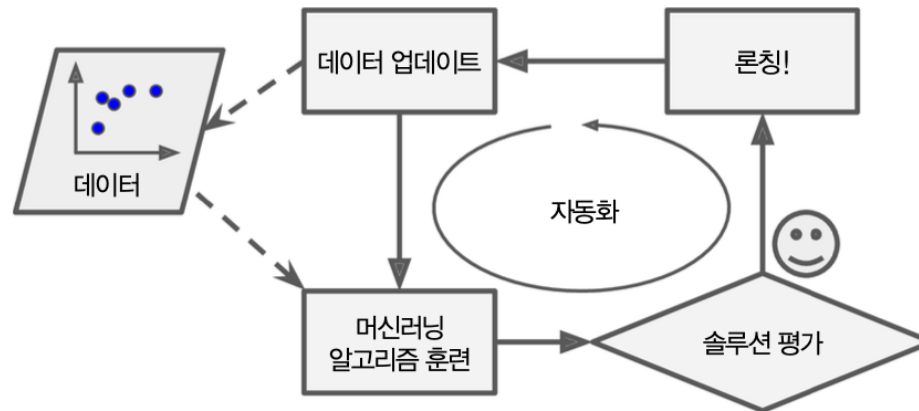


## 예제: 스팸 메일 분류

"광고", "투^^자", "무❤️료" 등의 표현이 스팸 메일에 자주 등장하는 경우 스팸 메일 분류기 기능 자동 업데이트 가능

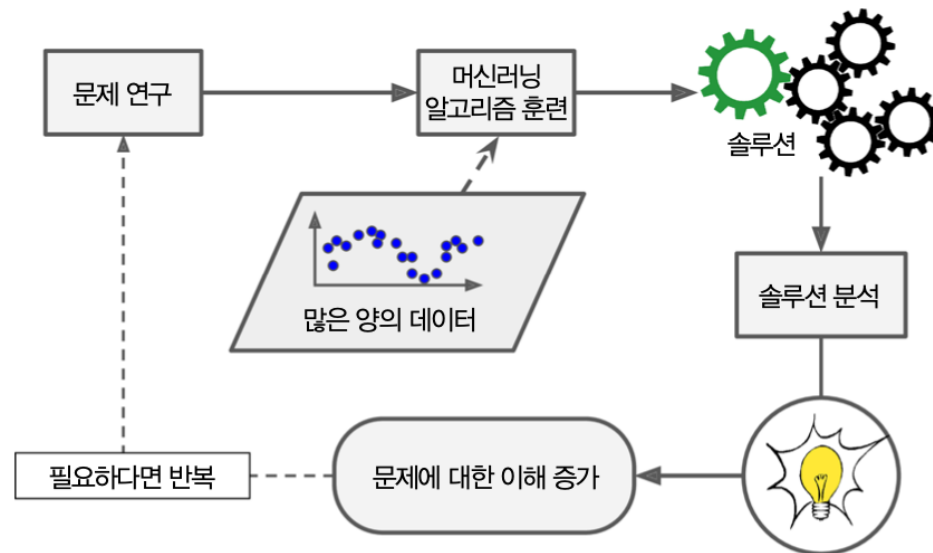
# 머신러닝 학습 자동화

- 머신러닝 작업 흐름의 전체를 머신러닝 파이프라인 또는 **MLOps(Machine Learning Operations, 머신러닝 운영)**라 부르며 자동화가 가능함.



# 머신러닝의 장점

- 스팸 메일 분류기 처럼 알고리즘에 대한 너무 많은 세부 튜닝과 매우 긴 규칙을 요구하는 문제를 해결할 수 있다.
- 음성 인식 등 전통적인 방식으로 해결하기에 너무 복잡한 문제를 해결할 수 있다.
- 새로운 데이터에 바로 적용이 가능한 시스템을 쉽게 재훈련할 수 있다.
- 머신러닝 프로그램으로 생성된 솔루션 분석을 통해 데이터에 대한 통찰을 얻을 수 있다.



## 1.3 머신러닝 시스템 유형

## 훈련 지도 여부

- 지도 학습
- 비지도 학습
- 준지도 학습
- 자기주도 학습
- 강화 학습

## 실시간 훈련 여부

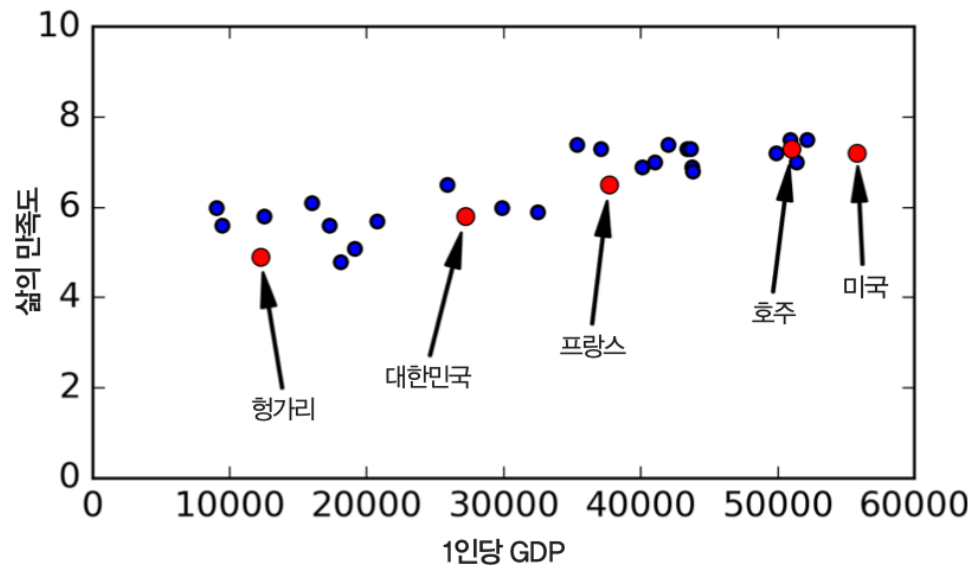
- 배치 학습
- 온라인 학습
- 외부 메모리 학습

## 예측 모델 사용 여부

- 사례 기반 학습
- 모델 기반 학습

## 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.

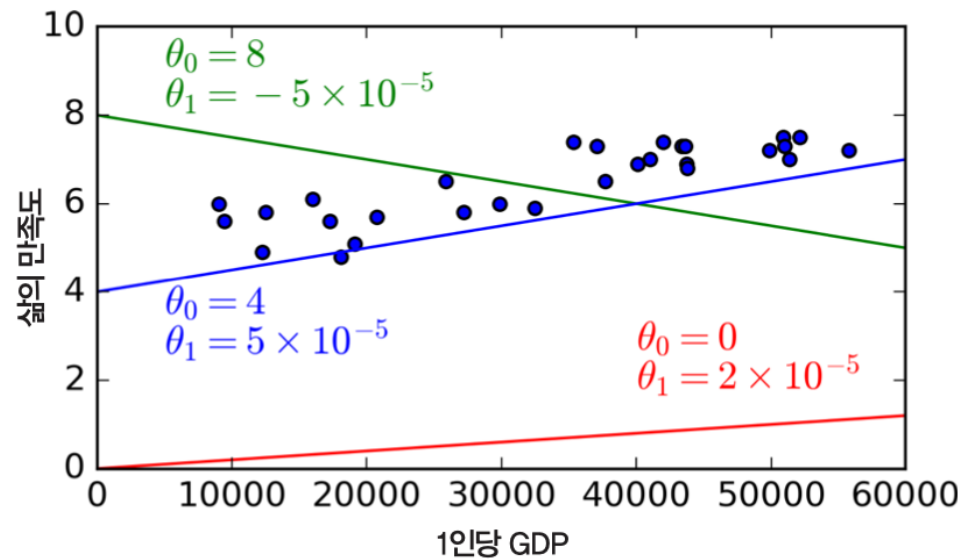




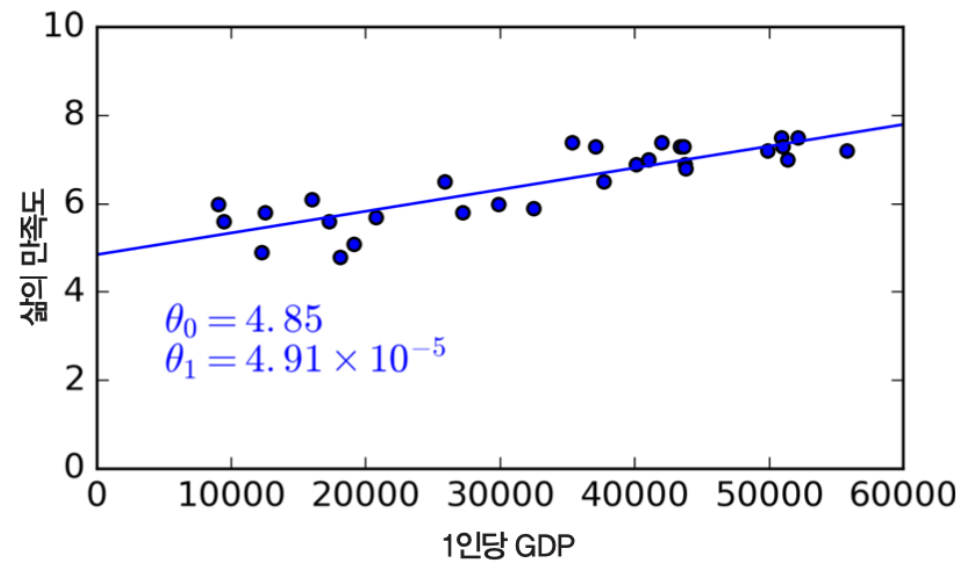
- 데이터를 대표하는 하나의 직선(선형 모델)을 찾기

$$\text{'삶의 만족도'} = \theta_0 + \theta_1 \times \text{'1인당 GDP'}$$

- 데이터를 대표할 수 있는 선형 방정식을 찾아야 함



- 아래 선형 모델이 최적!



## 분류 기준의 비배타성

- 분류 기준이 상호 배타적이지는 않음.
- 스팸 필터 예제
  - 지도 학습: 스팸 메일과 스팸이 아닌 메일로 이루어진 훈련셋으로 모델 학습 진행
  - 온라인 학습: 실시간 학습 가능
  - 모델 기반 학습: 훈련 결과로 생성된 모델을 이용하여 스팸 여부 판단

## 1.4 머신러닝 모델 훈련의 어려움

## 데이터 문제

- 충분하지 않은 양의 훈련 데이터
- 대표성 없는 훈련 데이터
- 저품질 데이터
- 특성 공학 필요

## 알고리즘 문제

- 과대적합
- 과소적합

## 1.5 테스트와 검증

## 테스트셋 활용 훈련된 모델 검증

- 훈련된 모델의 성능 평가: 테스트셋 활용
- 전체 데이터셋을 훈련셋(80%)와 테스트셋(20%)로 구분
  - 훈련셋: 모델 훈련용.
  - 테스트셋: 모델 테스트용
  - 데이터셋이 매우 크면 테스트셋 비율을 낮출 수 있음.
- 과대 적합: 훈련된 모델의 훈련셋에 대한 성능에 비해 테스트셋에 대한 성능이 낮은 경우
- 과소 적합: 훈련된 모델의 훈련셋에 대한 성능이 기대 이하인 경우