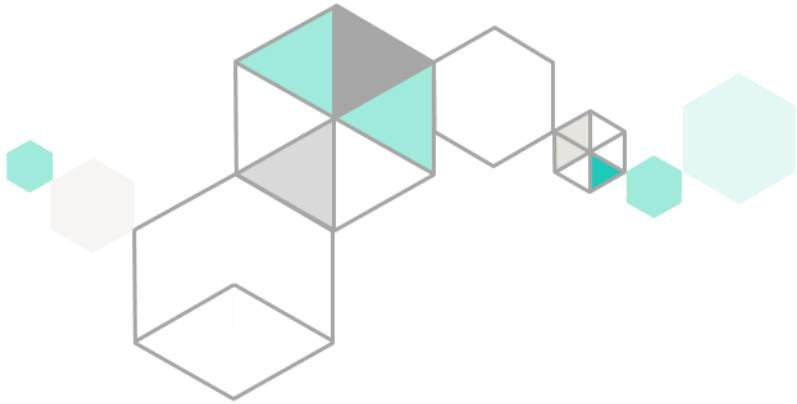
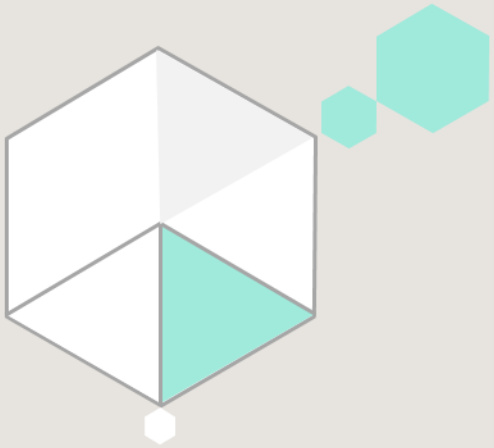


손바닥ML

With Scikit-Learn & Tensorflow

2017. 11. 15



Textbook

Hands-On Machine Learning with Scikit-Learn and TensorFlow - Aurélien Géron(1st, 2017)

Github

https://github.com/ageron/handson-ml

Contents

Part I. The Fundamentals of Machine

- What is Machine Learning? What problems does it try to solve? What are the main categories and fundamental concepts of Machine Learning systems?
- The main steps in a typical Machine Learning project
- Learning by fitting a model to data
- Optimizing a cost function
- Handling, cleaning, and peparing data
- Selecting and engineering features
- Selecting a model and tuning hyperparameters using cross-validation
- The main challenges of Machine Learning, in particular underfitting and overfitting (the bias/variance tradeoff)
- Reducing the dimensionality of the training data to fight the curse of dimensionality

Part I. The Fundamentals of Machine

The most common learning algorithms
- Linear and Polynomial Regression
- Logistic Regression
- k-Nearest Neighbors
- Support Vector Machines
- Decision Trees
- Random Forests
- Ensemble methods.

Part II, Neural Networks and Deep Learning

- What are neural nets? What are they good for?
- Building and training neural nets using TensorFlow
- The most important neural net architectures
  - feedforward neural nets
- convolutional nets, recurrent nets, long short-term memory (LSTM)
- nets, and autoencoders
- Techniques for training deep neural nets
- Scaling neural networks for huge datasets
- Reinforcement learning
- The first part is based mostly on Scikit-Learn while the second part uses
- TensorFlow

References

- Andrew Ng, ML course on Coursera
- Geoffrey Hinton, course on neural networks and Deep Learning

- Joel Grus, Data Science from Scratch (O'Reilly)
- Stephen Marsland, Machine Learning: An Algorithmic Perspective (Chapman and Hall)
- Sebastian Raschka, Python Machine Learning (Packt Publishing)
- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin, Learning from Data (AMLBook)
- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition (Pearson)

- kaggle.com

Part I. The Fundamentals of ML

Ch.1 The ML Landscape

What Is Machine Learning?

- **Machine Learning** is the science (and art) of programming computers so they can learn from data.

- Here is a slightly more general definition:

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Some terminology

spam 메일을 분류하는 문제를 생각해 보자.

이메일을 우리는 spam 또는 nonspam(=ham)으로 분류하려고 한다.

우리가 갖고 있는 메일을 **training data** 라고 한다.

각각의 training data 를 **training instance**(or **sample**) 이라고 한다.
(즉, training data = a set of training data)

이러한 tarining data 로 학습을 하여 실제로 spam을 맞추는 확률을 **accuracy**라고 한다.(in classification problem)

Why we use Machine Learning?

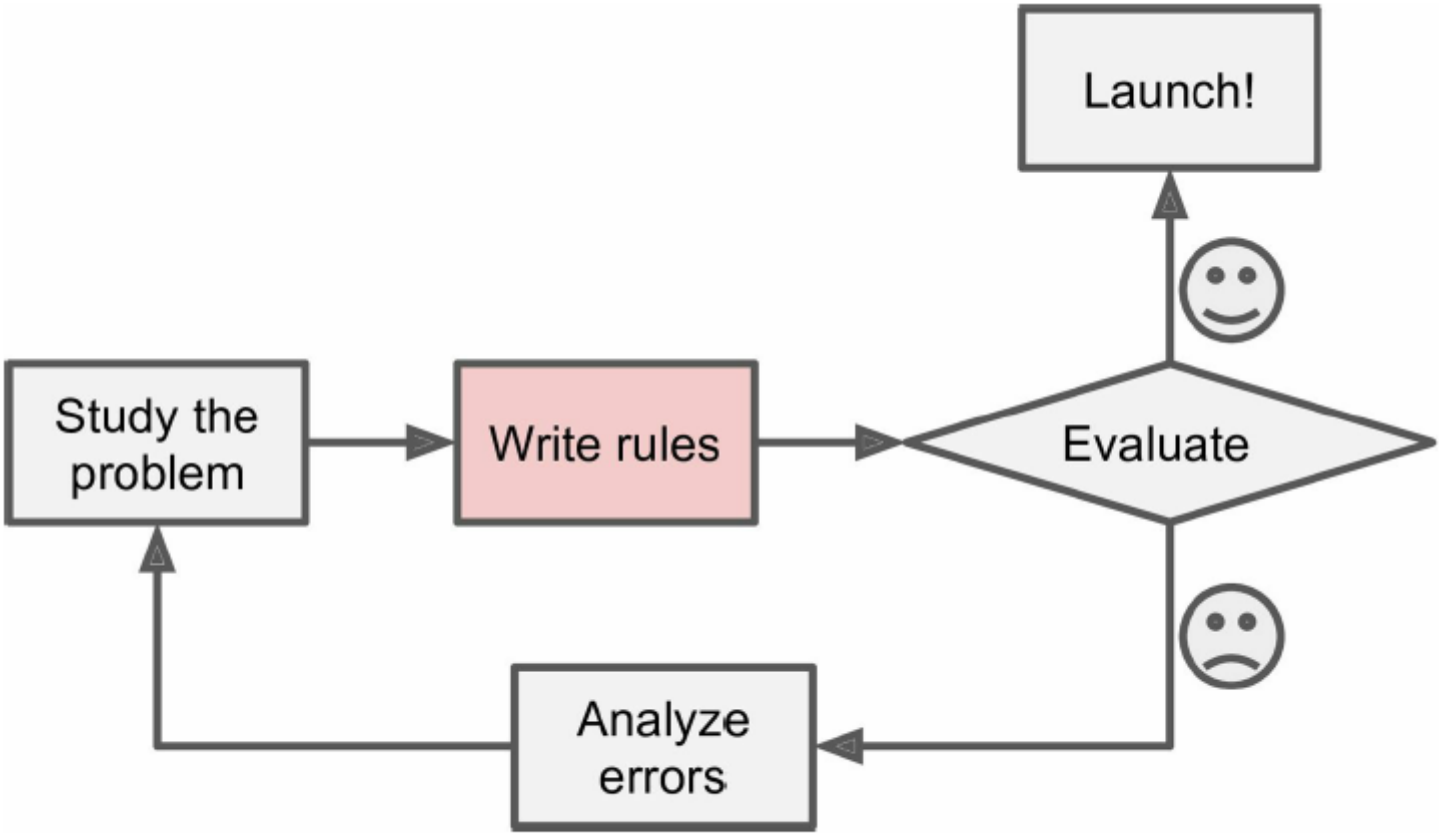


Figure 1-1. The traditional approach

Why we use Machine Learning?

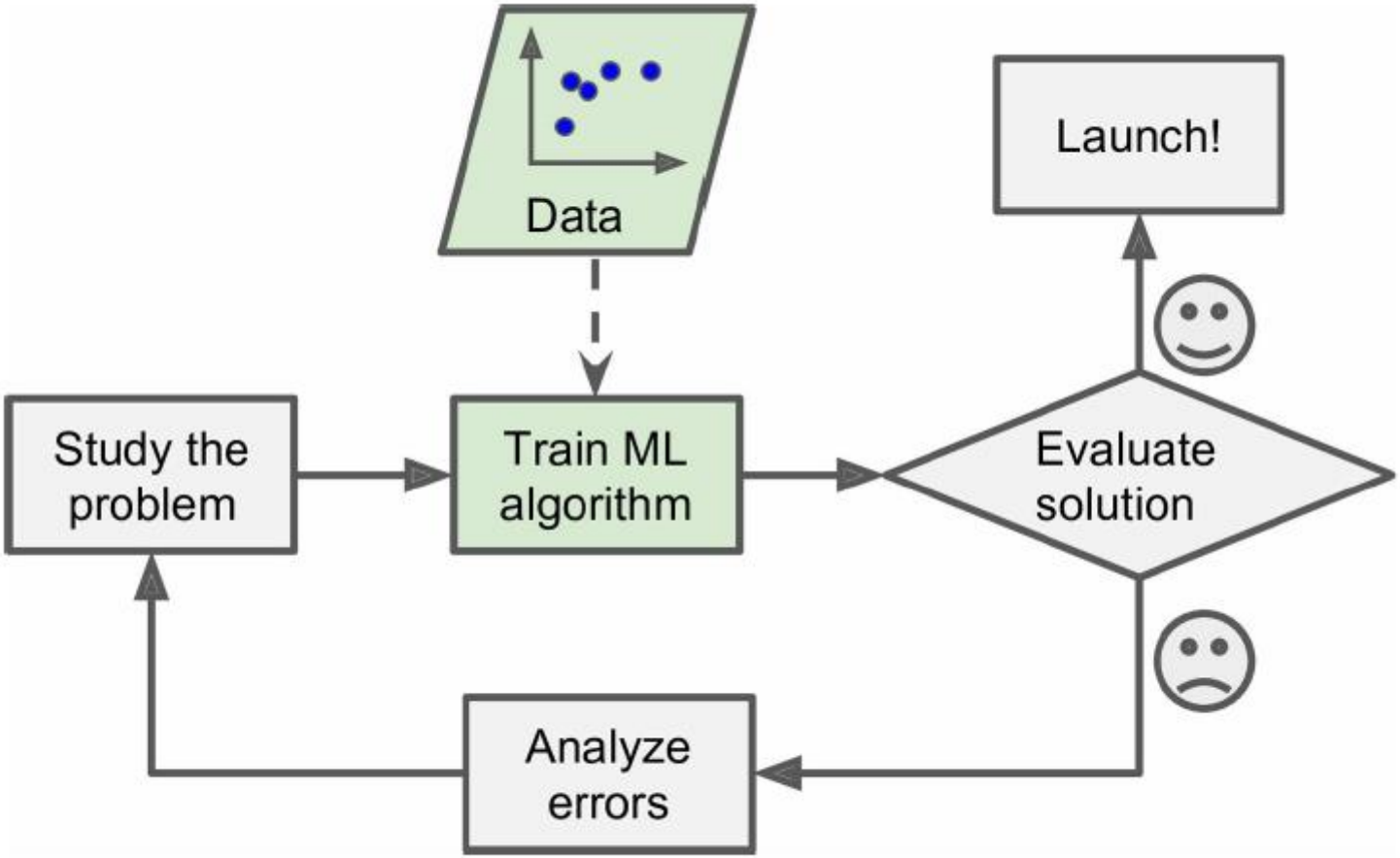


Figure 1-2. Machine Learning approach

Why we use Machine Learning?

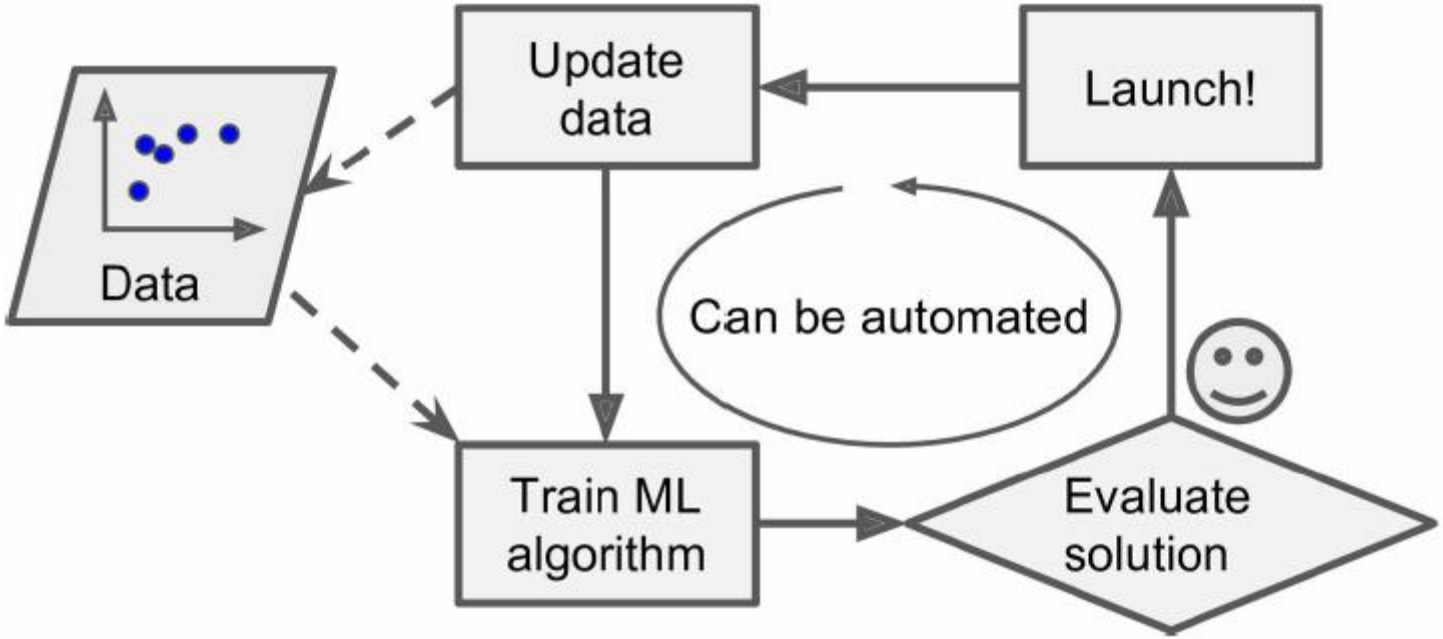


Figure 1-3. Automatically adapting to change

Why we use Machine Learning?

대량의 데이터에서 눈에 보이지 않는 관계를 발견할 수 있다(data mining)

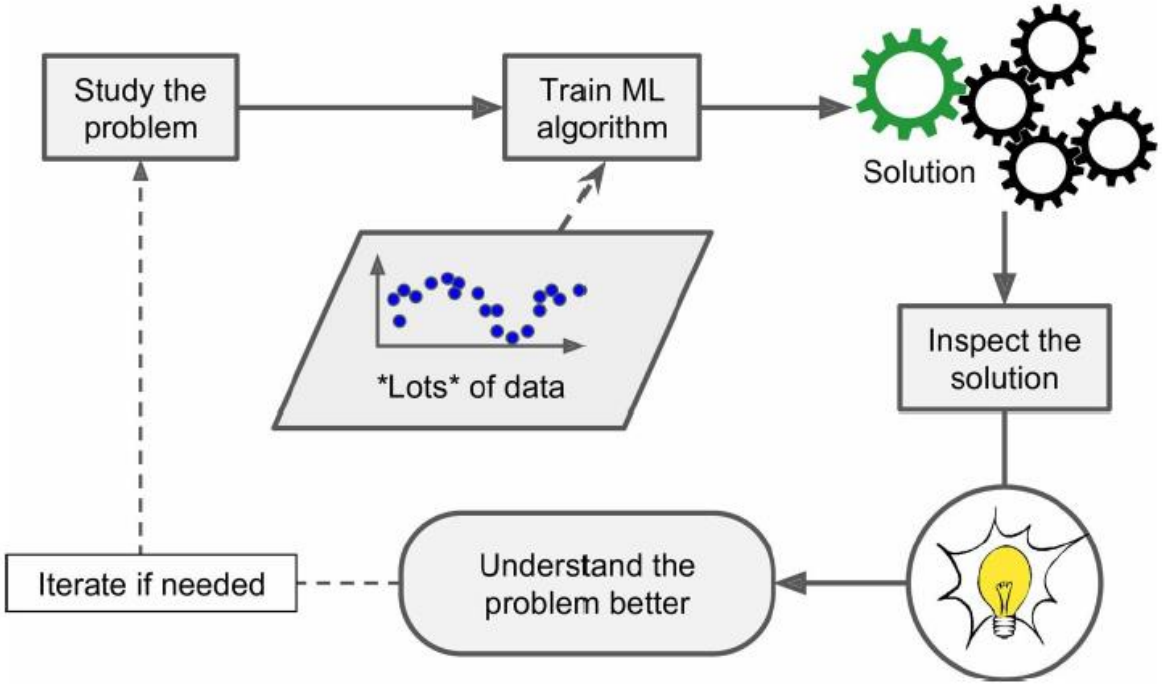


Figure 1-4. Machine Learning can help humans learn

Types of Machine Learning Systems

- There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on:
- Whether or not they are trained with human supervision (supervised, unsupervised, semisupervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus modelbased learning)

Supervised learning

In supervised learning, the training data you feed to the algorithm includes the desired solutions, called **labels**

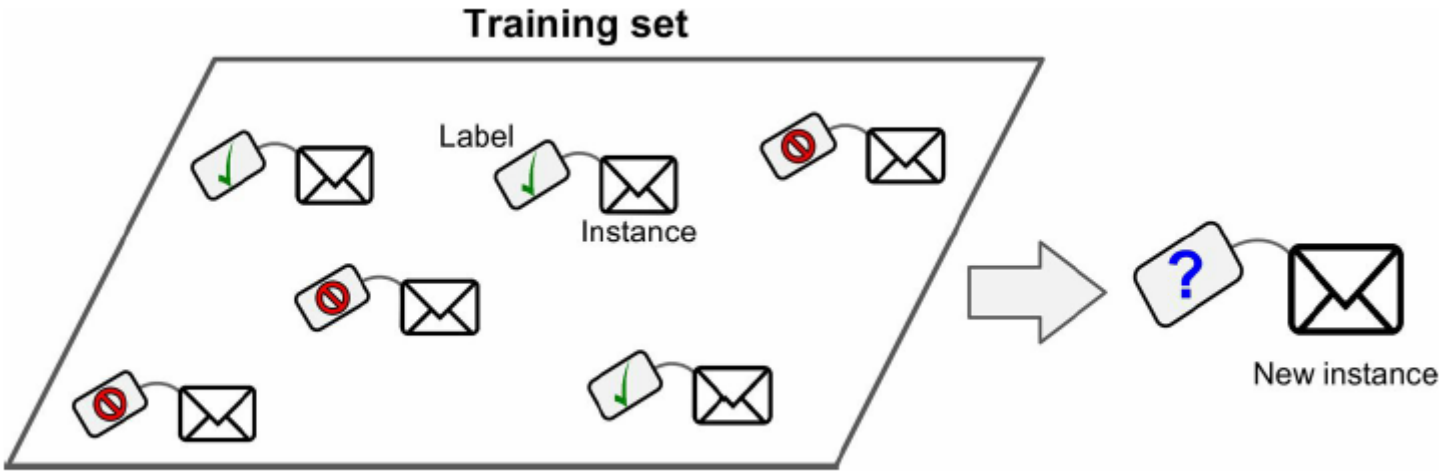


Figure 1-5. A labeled training set for supervised learning (e.g., spam classification)

Supervised learning

- **Classification** : label 이 있는 training data 를 learning 해서 새로운 데이터를 맞추는 문제(이때 label의 type 이 category(=nominal data) 이야 classification 이라고 한다)
- label 이 숫자이면 feature(=training instance) 를 **predictor**,
  label 이 숫자인 classification 문제를 **regression** 이라고 한다.
- Note that some regression algorithms can be used for
  as well, and vice versa
  ex) **Logistic Regression**

The most important supervised learning algorithms

- Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Supervised learning

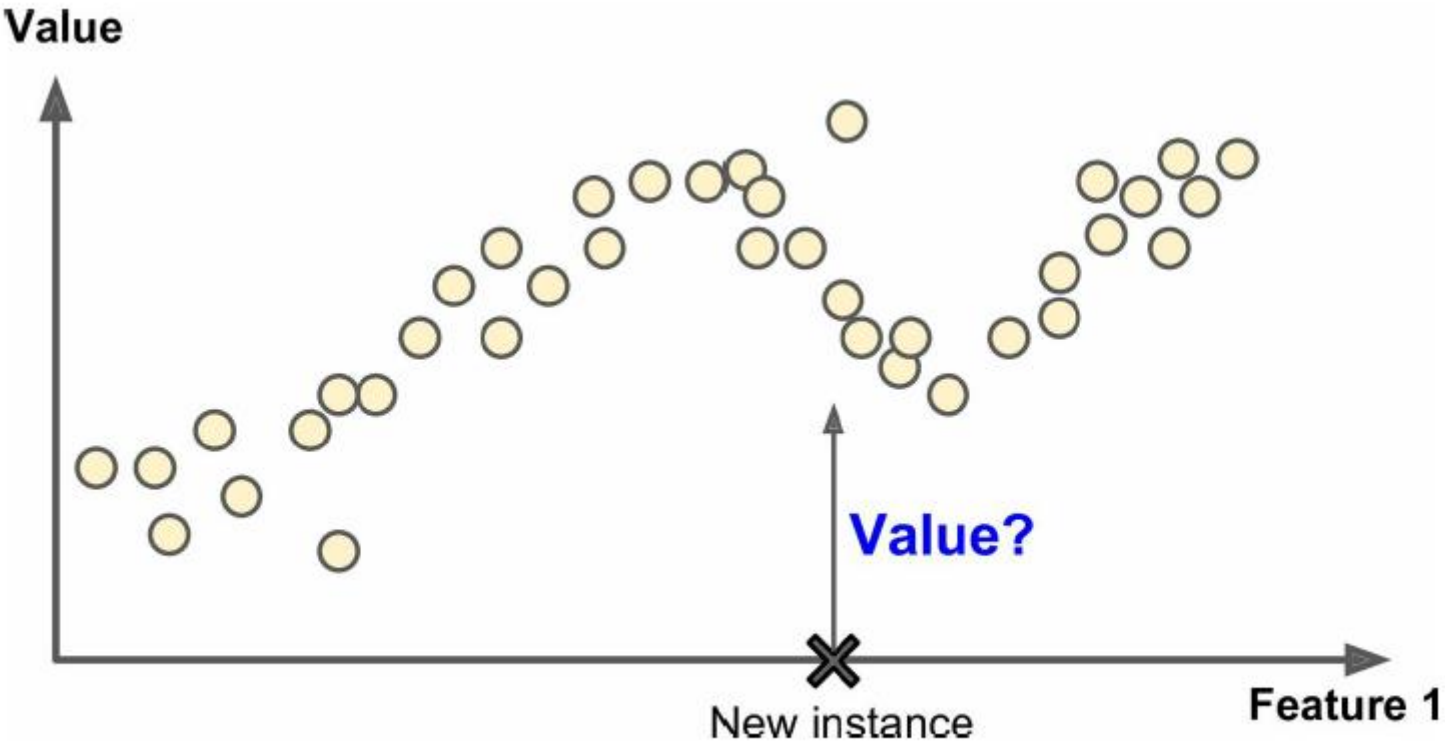


Figure 1-6. Regression

Unsupervised learning

- In unsupervised learning, the training data is unlabeled

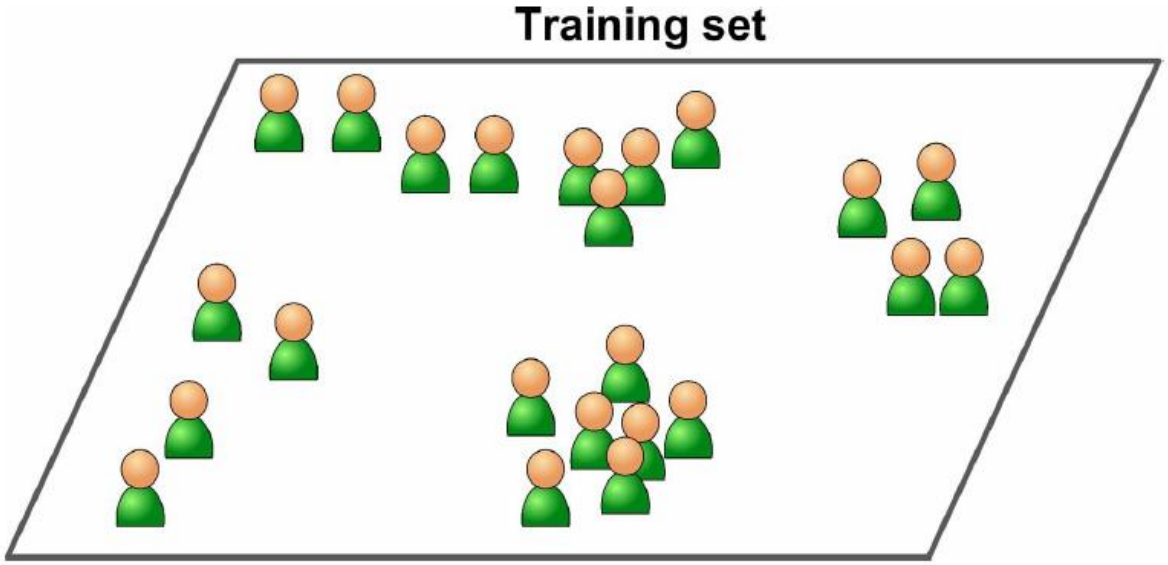


Figure 1-7. An unlabeled training set for unsupervised learning

Unsupervised learning

- Clustering
  - k-Means
  - Hierarchical Cluster Analysis (HCA)
  - Expectation Maximization

- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)

- Association rule learning
  - Apriori
  - Eclat

Unsupervised learning

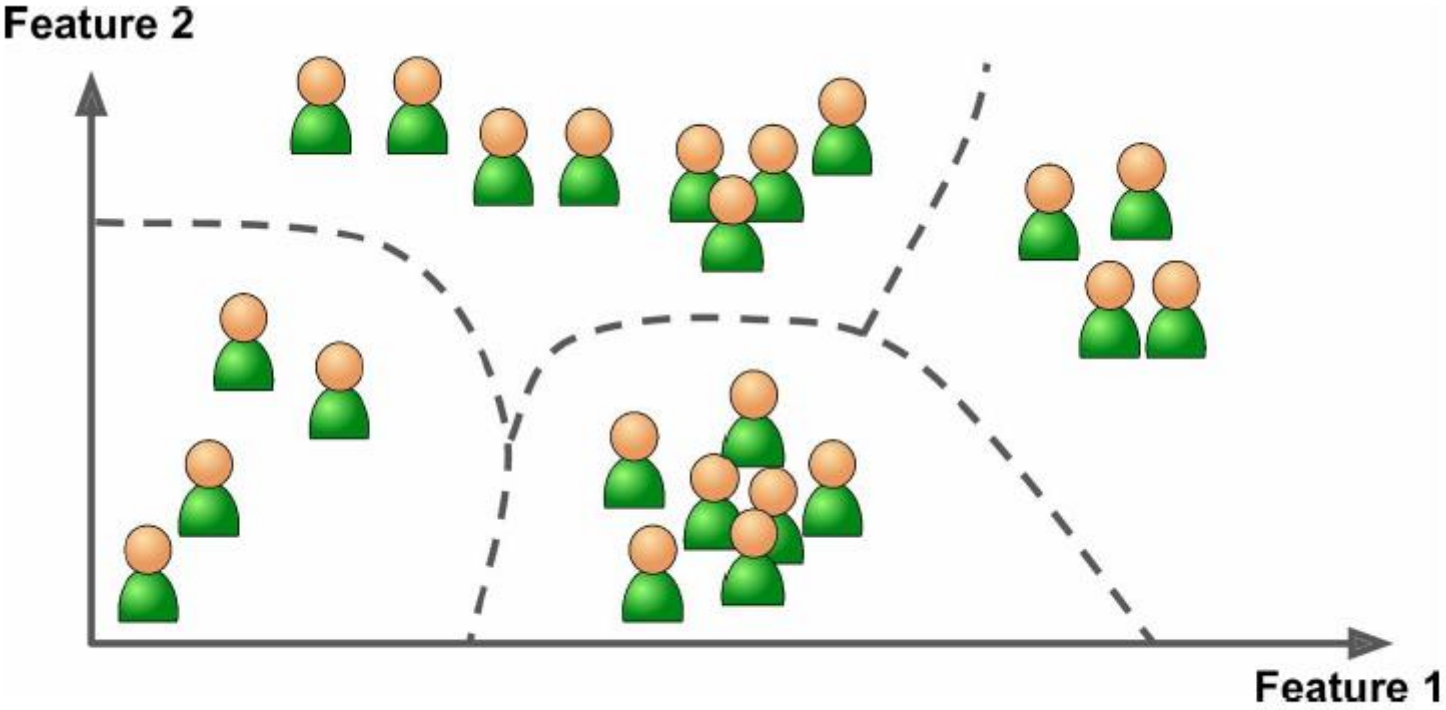


Figure 1-8. Clustering

Unsupervised learning

- A related task is **dimensionality reduction**, in which the goal is to simplify the data without losing too much information.
  ex) PCA, SVD, NMF etc

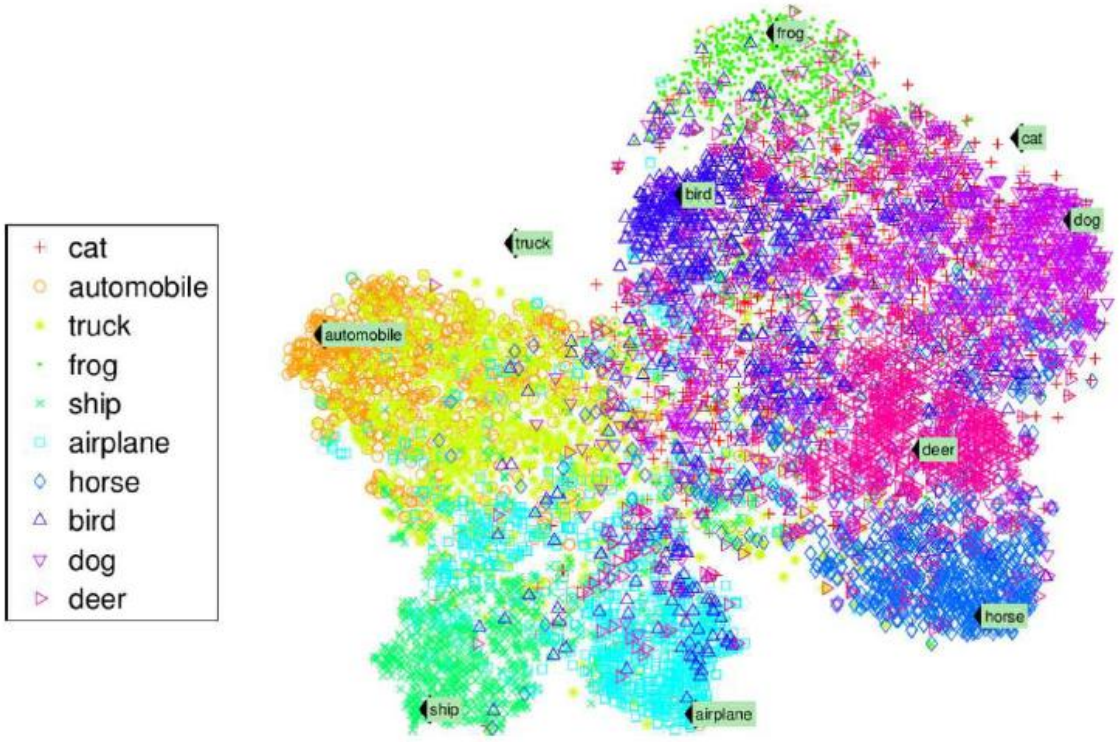


Figure 1-9. Example of a t-SNE visualization highlighting semantic clusters[3]

Unsupervised learning

- Yet another important unsupervised task is anomaly detection : for example, detecting unusual credit card transactions to prevent fraud, catching manufacturing defects, or automatically removing outliers from a dataset before feeding it to another learning algorithm.

- Finally, another common unsupervised task is association rule learning, in which the goal is to dig into large amounts of data and discover interesting relations between attributes.

Semisupervised learning

- Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called **semisupervised**

- Most semisupervised learning algorithms are combinations of unsupervised and supervised algorithms. For example, deep belief networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on top of one another. RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

- 모델의 likelihood 를 높이기 위해 EM 알고리즘 사용

Reinforcement Learning

- Reinforcement Learning is a very different beast. The learning system, called an agent in this context, can observe the select and perform actions, and get rewards in return (or penalties the form of negative rewards, as in Figure 1-12). It must then by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

Reinforcement Learning

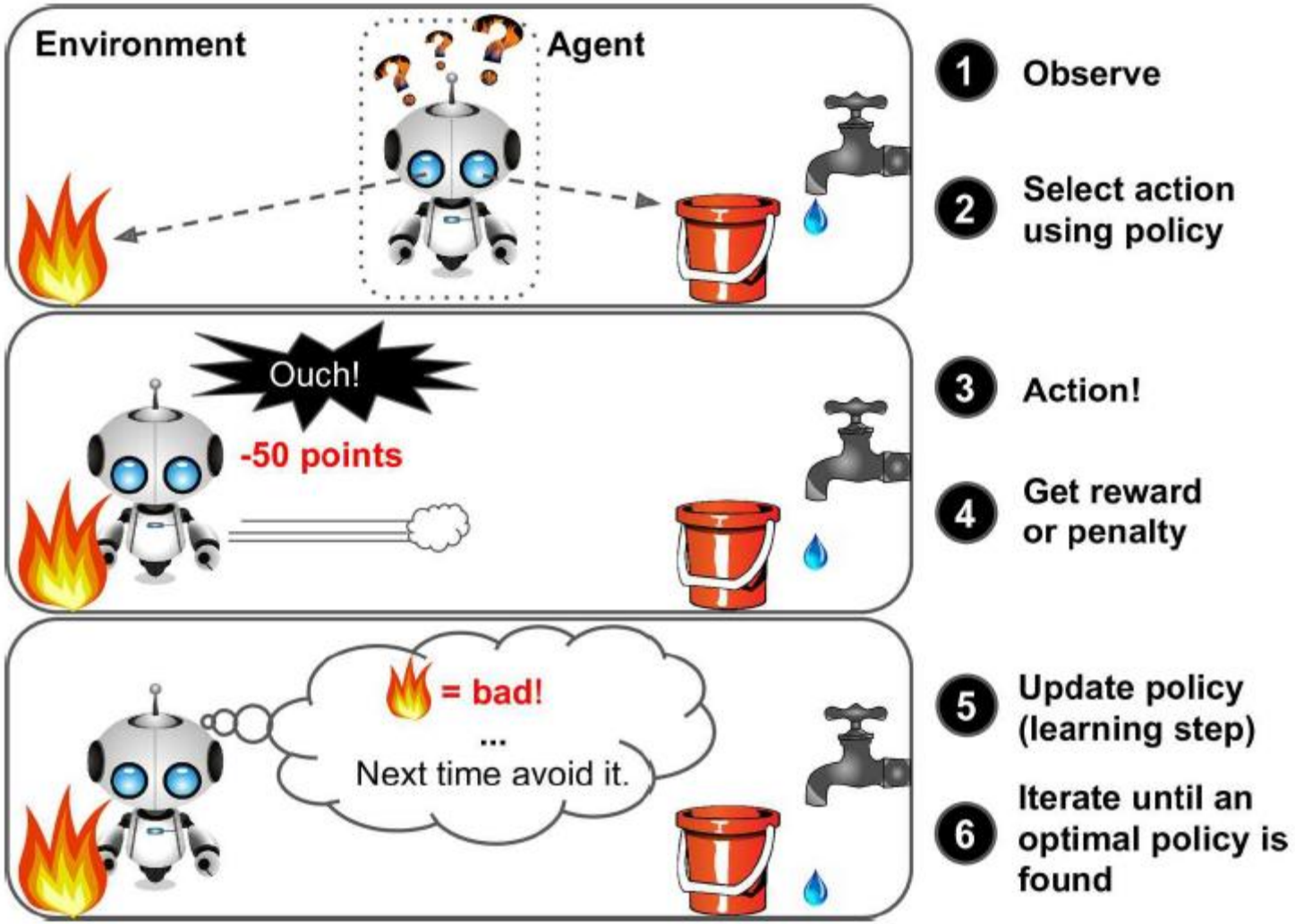


Figure 1-12. Reinforcement Learning

Batch and Online Learning

- Batch Learning(=offline learning)
  - Incoming data 로 부터 점진적으로 학습 불가능한 시스템
  - 모든 데이터를 학습해야 한다.
  - 새로운 데이터가 추가되면 기존의 데이터와 함께 다시 학습시켜야 한다.

- Online Learning
  - Incoming data 로 부터 점진적으로 학습 가능한 시스템
  - 연속적으로 들어오는 개별의 데이터 또는 mini-batch라고 불리는 소그룹으로 된 데이터가 들어올 때 점진적으로 학습이 가능한 학습 시스템
    ex) Markov chain 을 사용해서 기존의 데이터와의 관계 사용

Batch and Online Learning

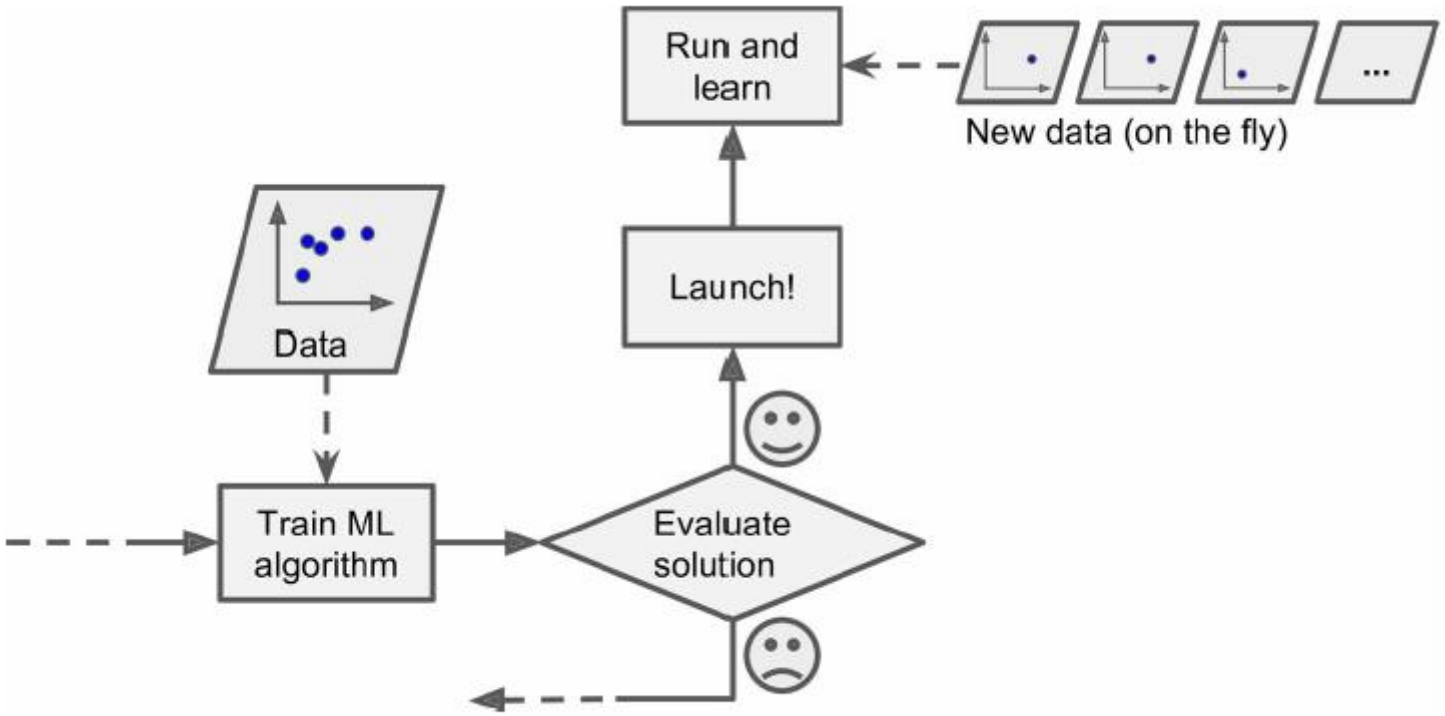


Figure 1-13. Online learning

Instance-Based vs Model-Based Learning

- Instance-Based Learning : labeled data set 이 있고, 새로운 data 가 들어왔을 때, 기존의 data와 distiance 가 작은 data를 n 개 뽑아 그들의 spam 여부를 보고 spam 인지 아닌지 체크하는 방법

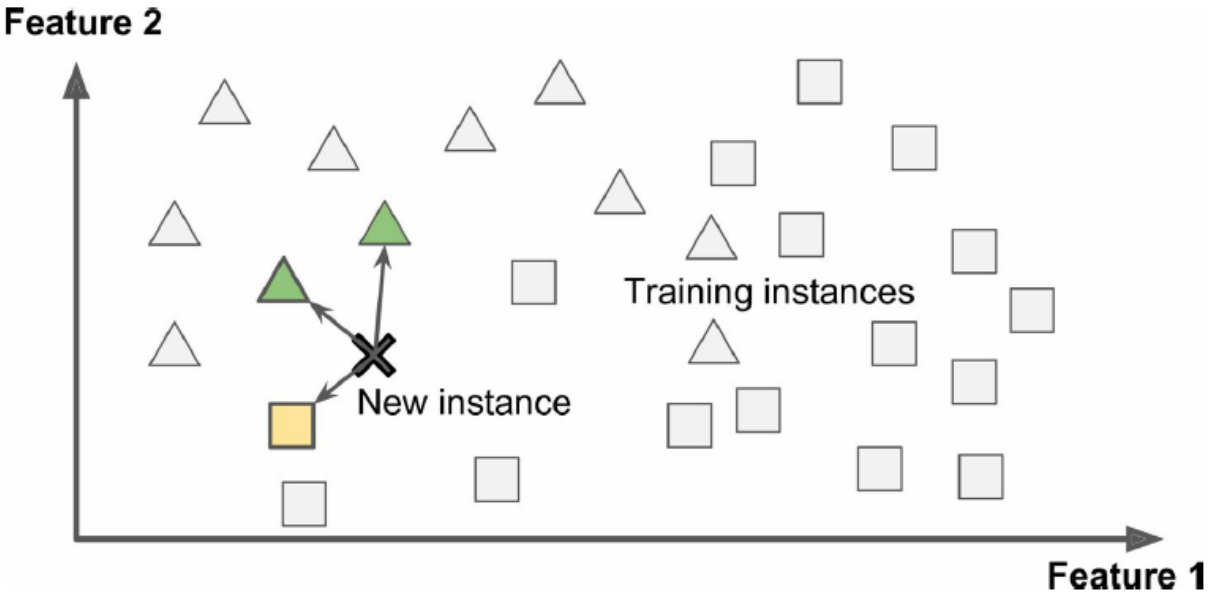


Figure 1-15. Instance-based learning

Instance-Based vs Model-Based Learning

- Model-Based Learning : training data 로 model 을 만들고, 새로운 데이터가 들어오면 모델에 넣어서 spam 인지 아닌지 판단하는 방법

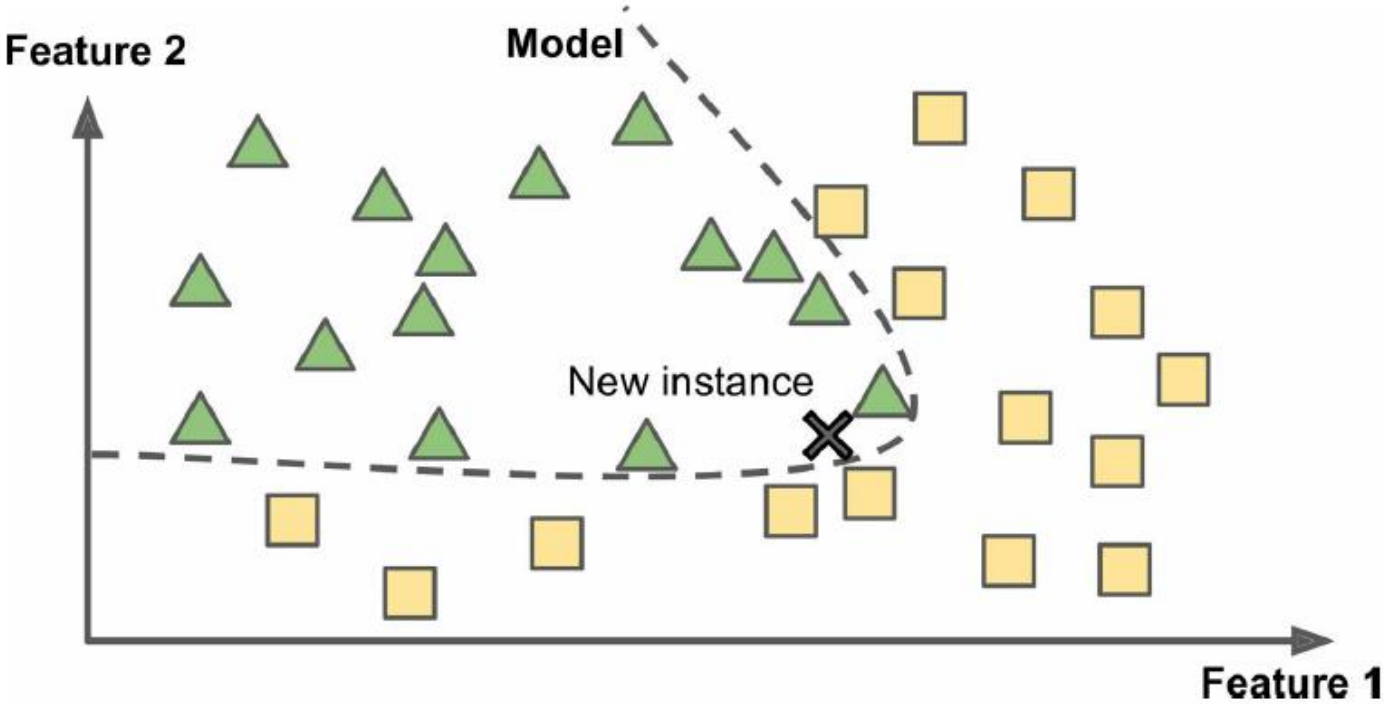


Figure 1-16. Model-based learning

Instance-Based vs Model-Based Learning

- How can you know which values will make your model perform best? To answer this question, you need to specify a performance measure. You can either define a utility function (or fitness function) that measures how good your model is, or you can define a cost function that measures how bad it is.

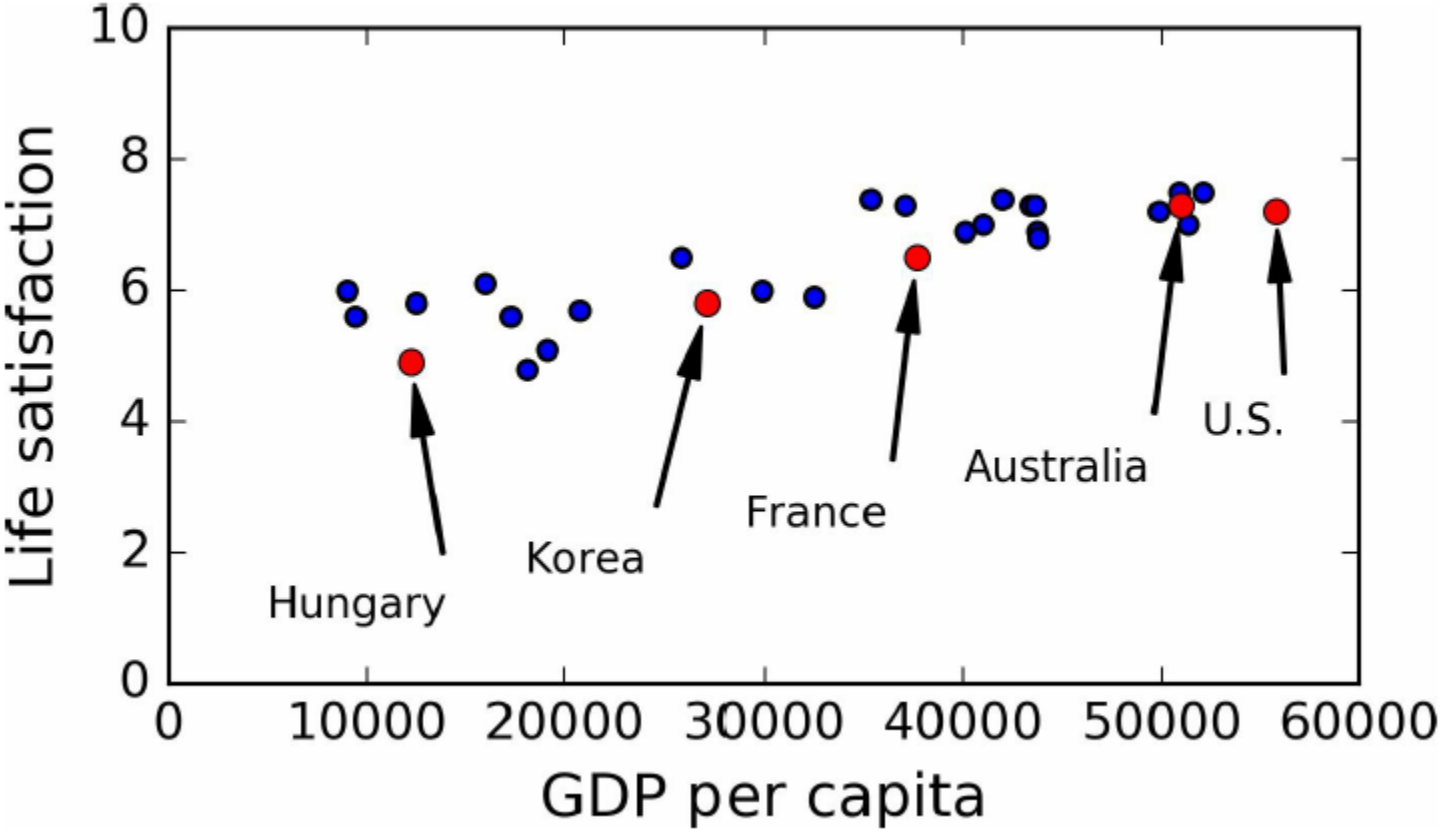
- ex. RMSE(Root Mean Square Error), MSE(Mean Squared Error), MAE(Mean Absolute Error)

$$\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{\mathrm{E}((\hat{\theta} - \theta)^2)}.$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Example

Example. Life satisfaction



Example

Equation 1-1. A simple linear model

$$life\_satisfaction = \theta_0 + \theta_1 \times GDP\_per\_capita$$

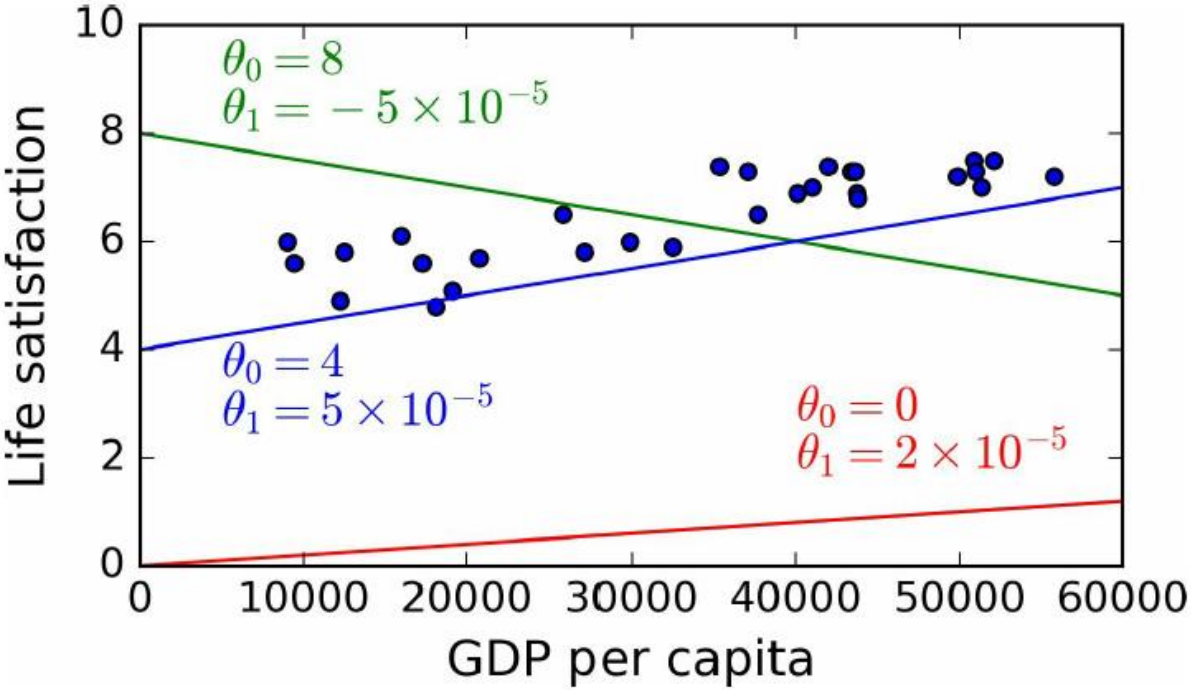


Figure 1-18. A few possible linear models

Example

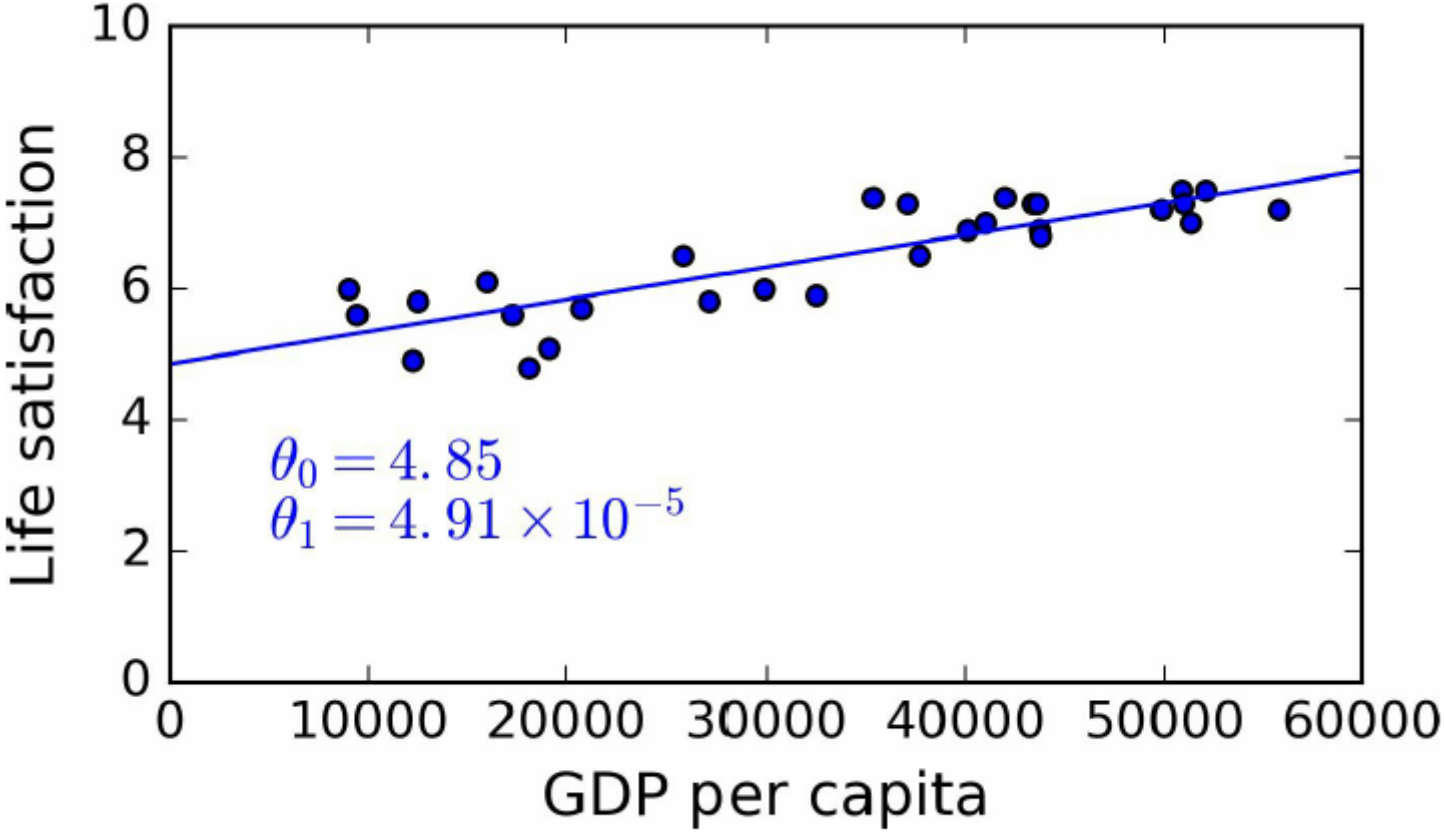


Figure 1-19. The linear model that fits the training data best

Main Challenges of Machine Learning

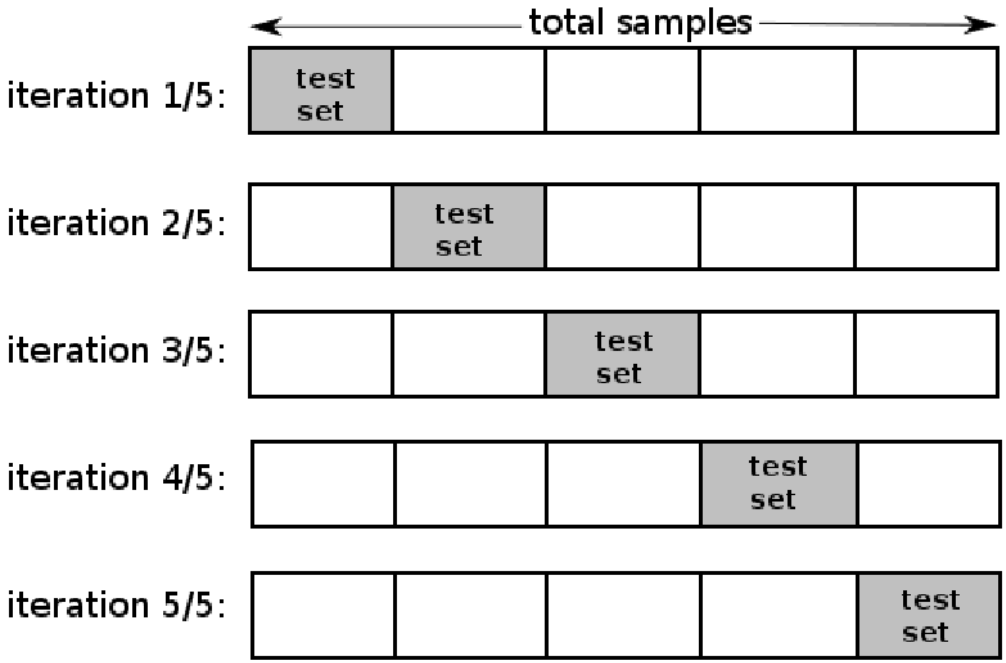
In short, since your main task is to select a learning algorithm and train it on some data, the two things that can go wrong are "bad algorithm" and "bad data."

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

Testing and Validating

모델이 생성되면 생성된 모델이 잘 작동하는지 Testing 이 필요하다!

Cross Validation : 주어진 데이터를 training set 과 testing set 으로 여러 번 나누어서 모델의 정확도(accuracy)를 체크하는 방법



Thank you