06

What is Machine Learning

Introduction

❖ Concepts of AI, ML, and DL

- Artificial intelligence
  - A technology that embodies the intellectual abilities of humans through computers
- Classification of artificial intelligence
  - Strong AI: AI with performance beyond human capabilities
  - Weak AI: AI designed for use as a tool in certain areas

강인공지능

약인공지능

Introduction

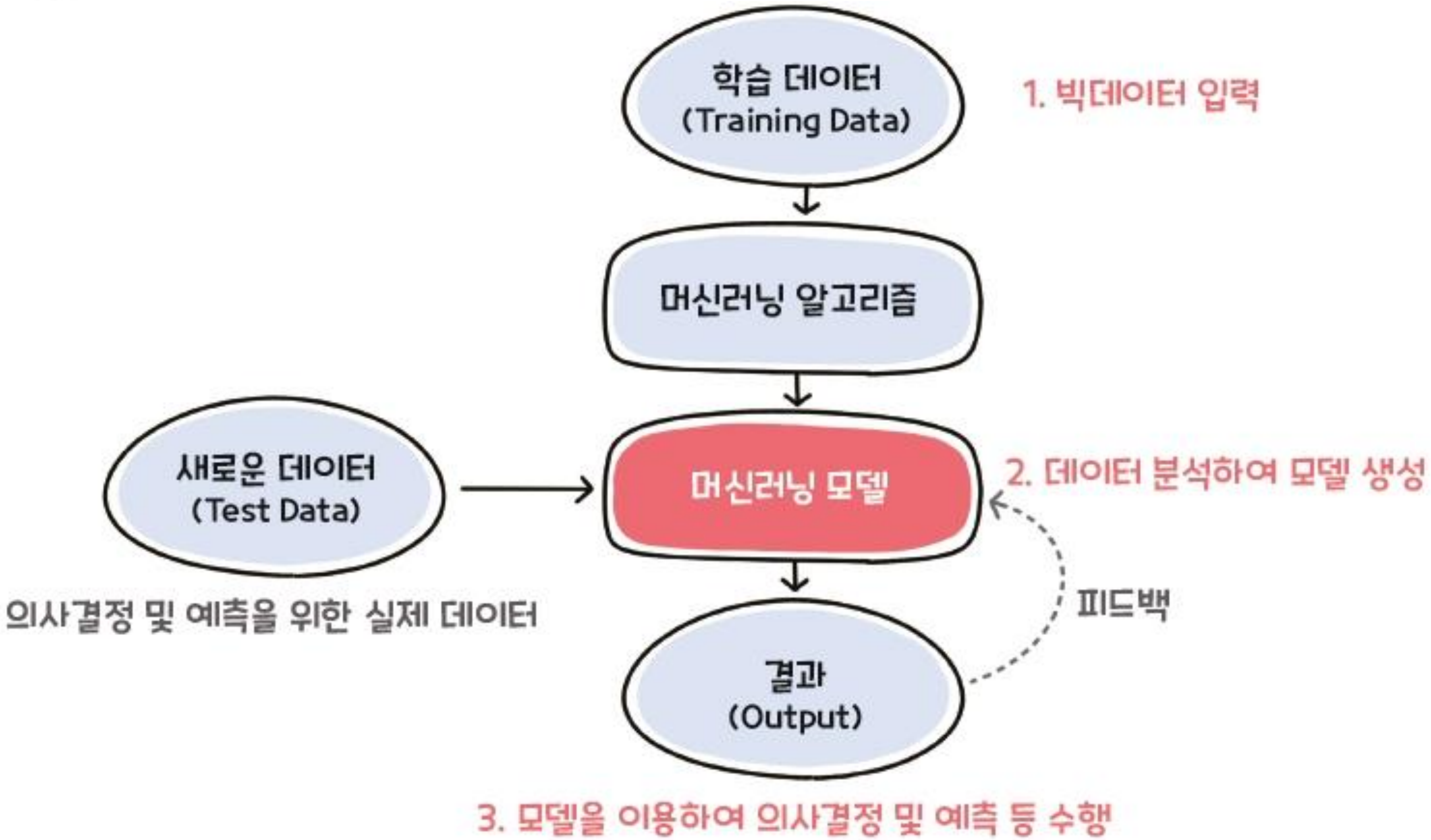
❖ Concepts of AI, ML, and DL

- Machine Learning
  - A technology that allows computers to learn like humans so that computers themselves can discover new rules without human help
  - Machine learning basically analyzes data using algorithms, learns through analysis, and makes judgments or predictions based on what is learned
  - Machine learning is the process of self-learning and processing data
    - Insert Big Data
    - Analyze data to create a model
    - Use models to make decisions, predictions, etc

Introduction

❖ Concepts of AI, ML, and DL

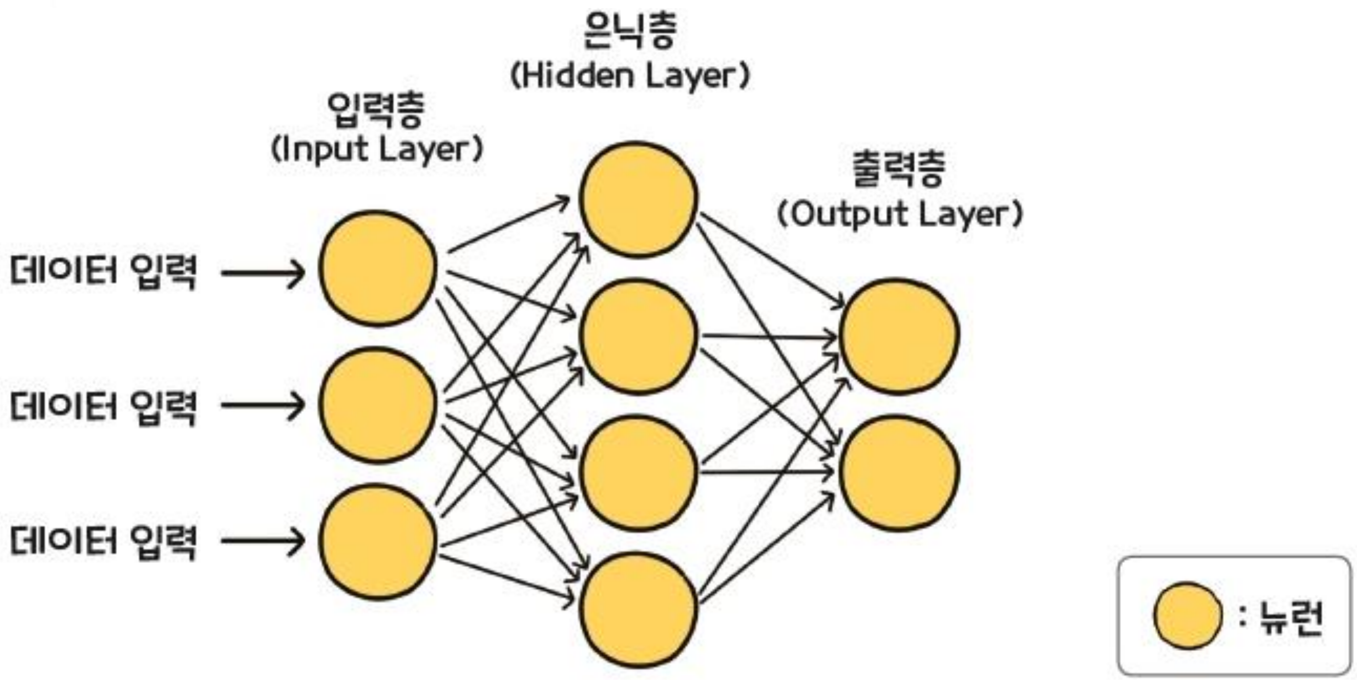
- Machine Learning



Introduction

❖ Concepts of AI, ML, and DL

- Deep Learning
  - ANN, Artificial Neural Network
    - A network of interconnected neurons

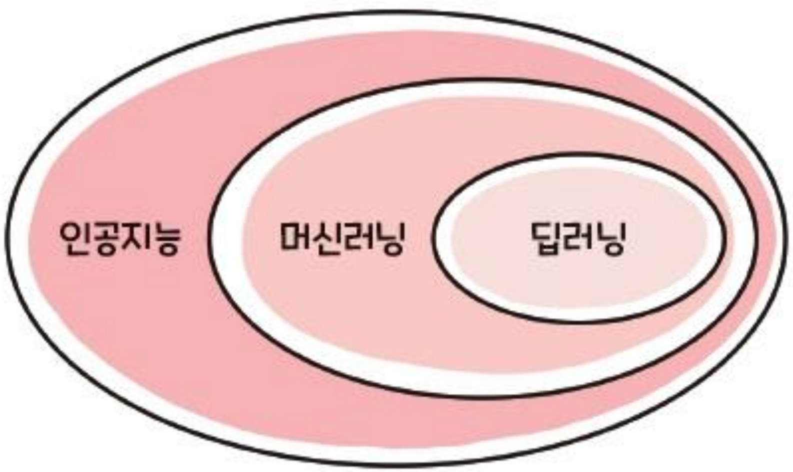


Introduction

❖ Concepts of AI, ML, and DL

▪ Deep Learning

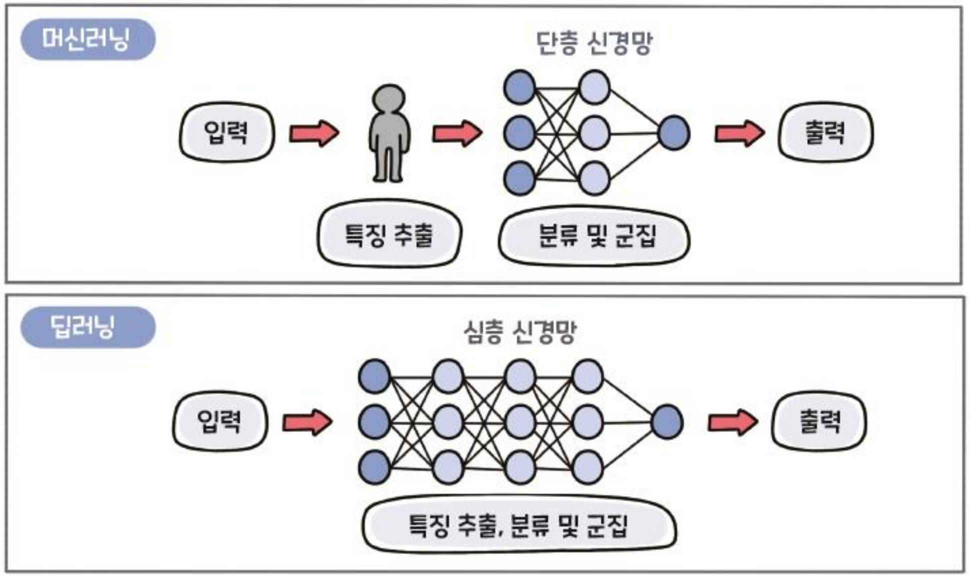
- Technology for performing machine learning using artificial neural networks with multiple hidden layers
- “Deep” in deep learning means deep layers of continuous neural networks
- Performance increases as this neural network deepens



Introduction

❖ Difference between ML and DL

- Human in the loop
  - Machine learning involves some degree of intervention, such as human informing the learning data of labels (corrects) or extracting the characteristics of the data
  - Deep learning learns on its own without human intervention



Introduction

❖ Difference between ML and DL

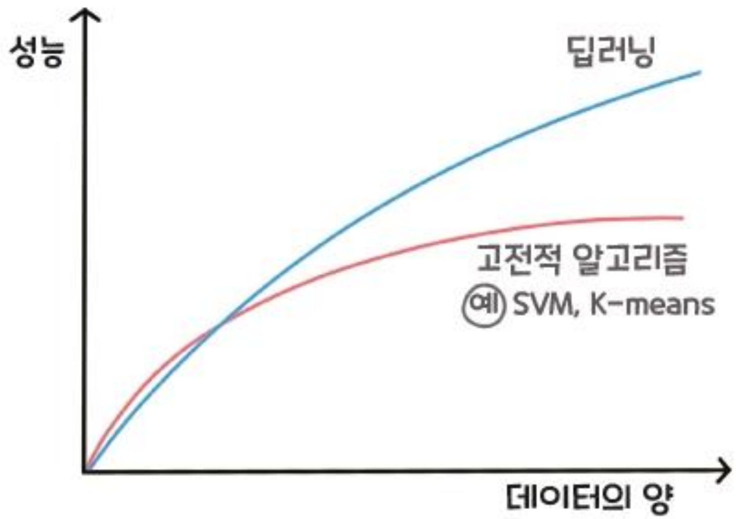
- Feature extraction
  - In machine learning, in order for a computer to learn on its own, it has to convert human-recognized data into computer-aware data,
  - For this task, it finds out what characteristics each data has and converts the data into vector

[특징 추출의 예]

Introduction

❖ Difference between ML and DL

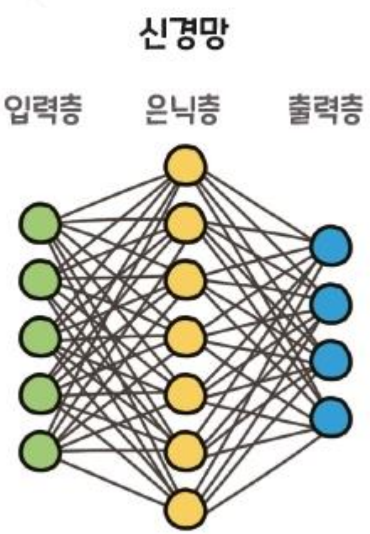
- Data dependencies
  - Deep learning directly extracts important features to solve a given problem
  - So if you don't have enough data, you can't extract the exact features
  - On the other hand, if sufficient data is given, it performs well enough to identify important features that humans do not recognize

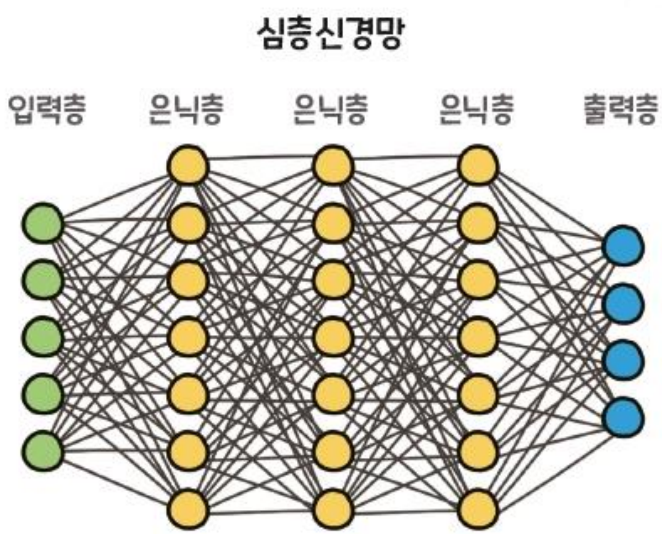


Introduction

❖ Difference between ML and DL

- Using neural network
  - Deep learning uses a deep neural network to extract features from input data and derive results (prediction or classification) on its own
  - The use of deep neural networks is a distinct characteristic of deep learning





Introduction

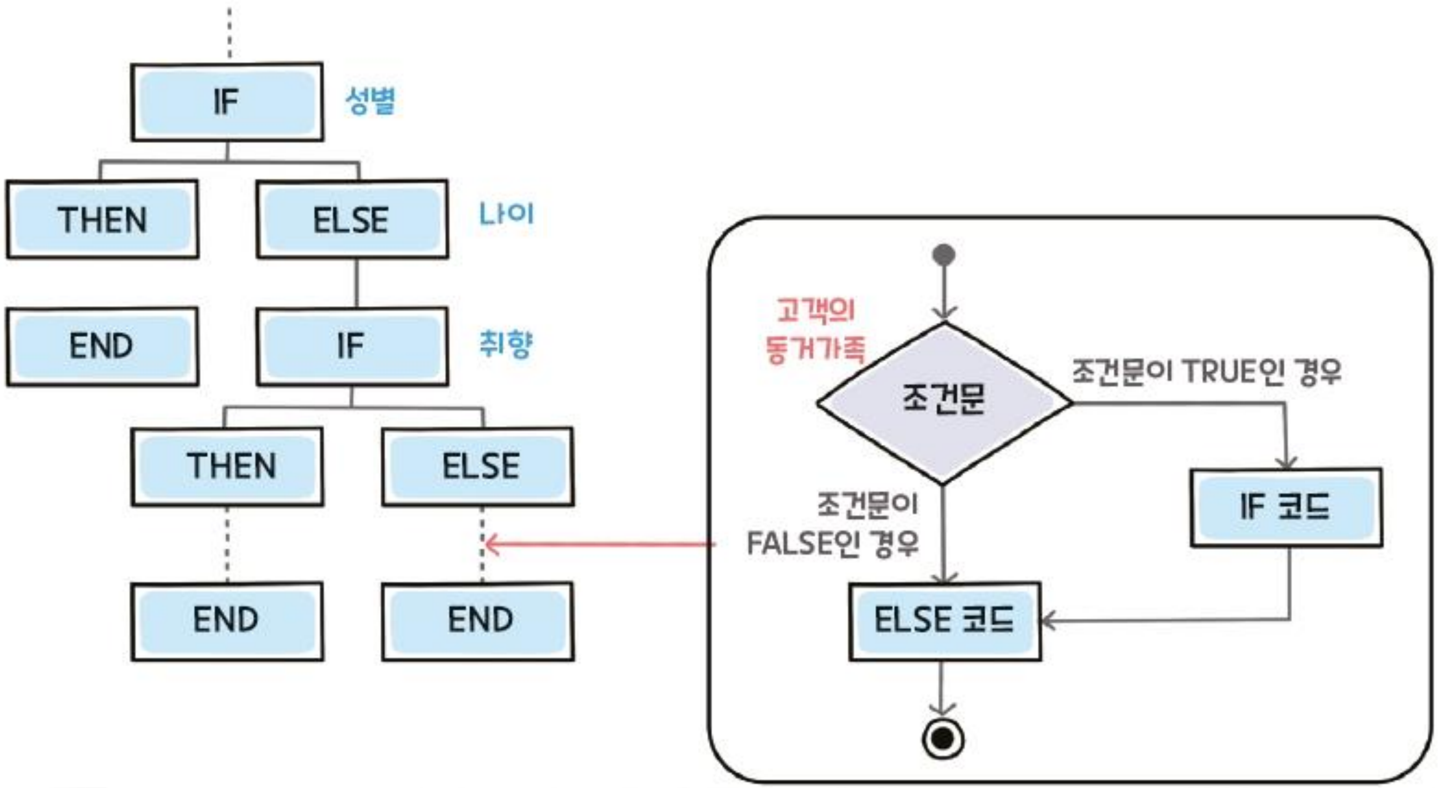
❖ Difference between ML and DL

- Using neural network

구분	머신러닝	딥러닝
필요한 데이터의 양	적은 양의 데이터도 가능	빅데이터
정확도	낮음	높음
훈련 시간	짧은 시간 안에 가능	오래 걸림
하드웨어	CPU만으로도 가능	GPU
하이퍼파라미터 튜닝	제한적	다양한 방법으로 튜닝 가능

Why Machine Learning?

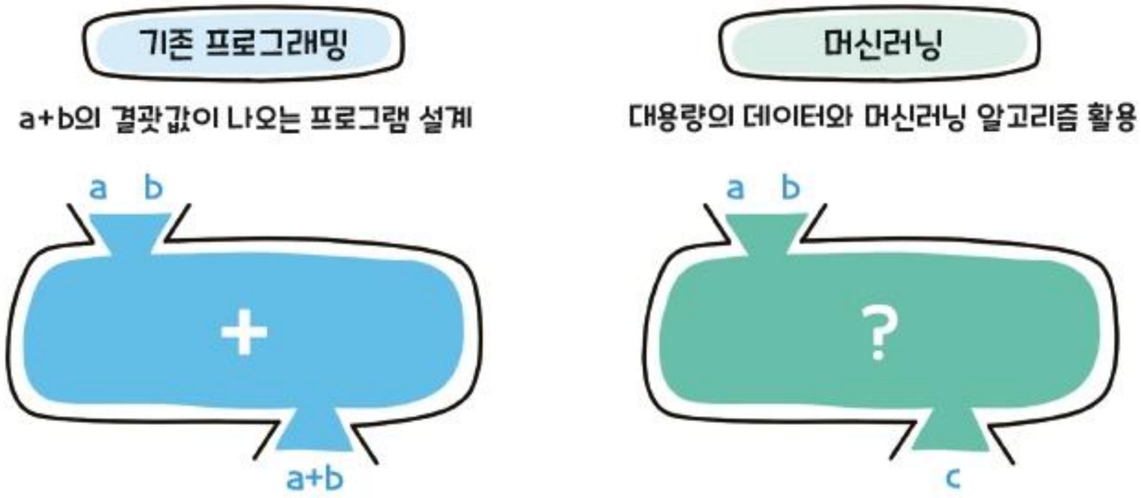
❖ Limitation of basic programming



Why Machine Learning?

❖ Usability of machine learning

- But it's not the right time to make a quick decision
  - Using machine learning to solve this problem
- Machine learning is a very useful solution when large amounts of data and many variables are involved, and programs with conventional rules cannot solve complex tasks or problems



Kinds of Machin Learning

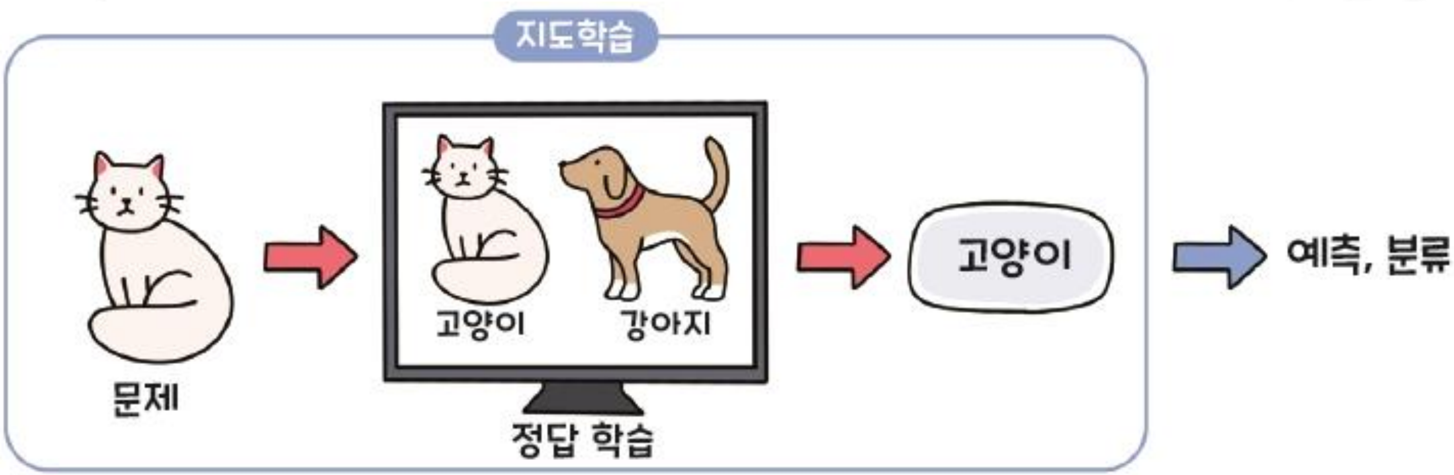
❖ Categorization for training

- Supervised learning : classification and regression
- Unsupervised learning : Clustering
- Reinforcement learning : Use rewards for actions taken in the environment to conduct learning

Kinds of Machin Learning

❖ Supervised learning

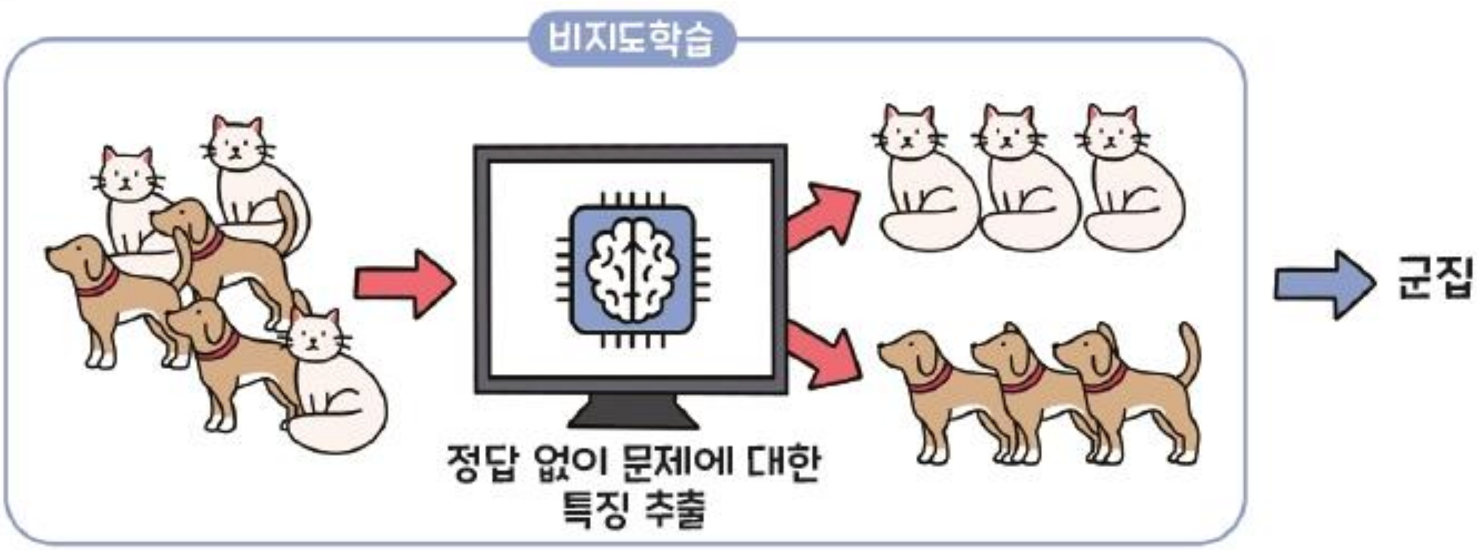
- Learning to predict the right answer to an unknown problem by learning questions and answers together
- The models used in supervised learning include prediction and classification



Kinds of Machin Learning

❖ Unsupervised learning

- A form of computer learning without the help
- Computer uses training data to find regularity between data



Kinds of Machin Learning

❖ Unsupervised learning

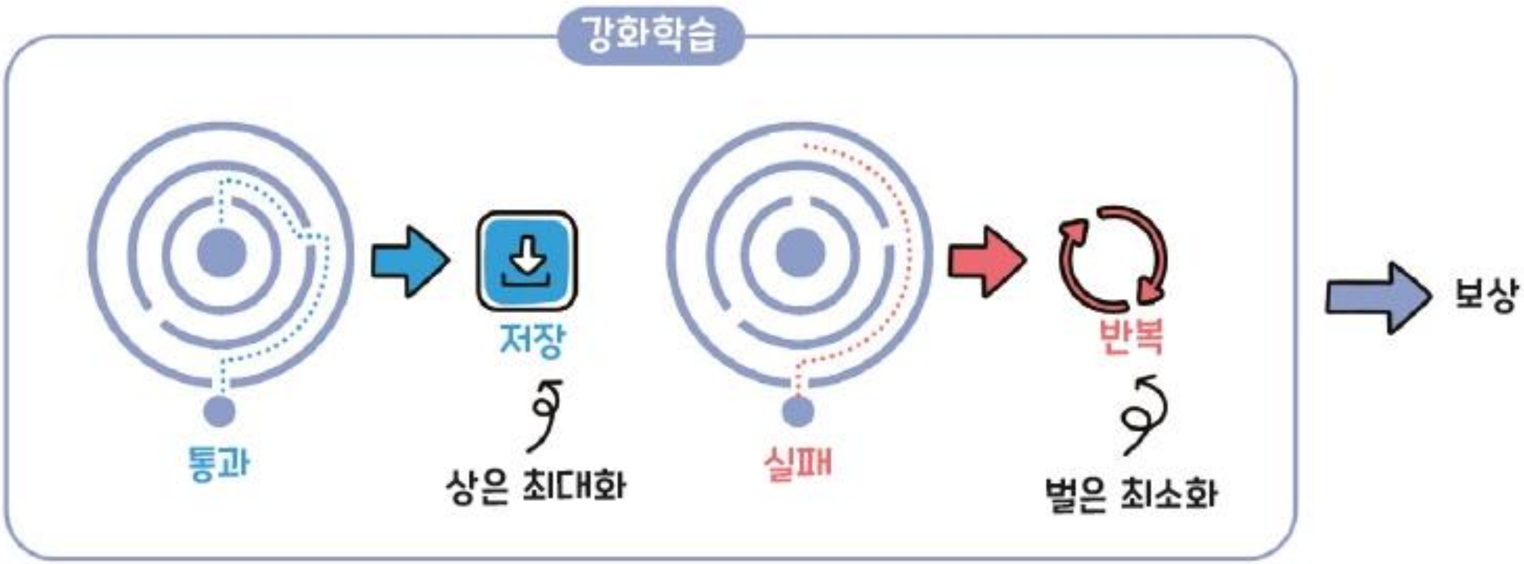
- Unlike supervised learning, which identified the relationship between x (input data) and y (labels in supervised learning), unsupervised learning identifies the relationship between x by itself
- In other words, the difference between y (label)
  - Clustering is a model used in unsupervised learning

구분	지도학습	비지도학습
필요한 데이터 종류	x(학습 데이터), y(레이블)	x(학습 데이터)

Kinds of Machin Learning

❖ Reinforcement learning

- Learning to be rewarded for what you've done
- How computers learn to choose the best behavior for a given state



Kinds of Machin Learning

❖ Reinforcement learning

- Agent: Subject to act in a given problem situation
- State : Current situation
- Action: Options that the player can take
- Rewards: Benefits that follow when a player does something
- Environment: means the problem itself
- Observation : Information about the environment (viewing and listening) collected by the agent

Kinds of Machin Learning

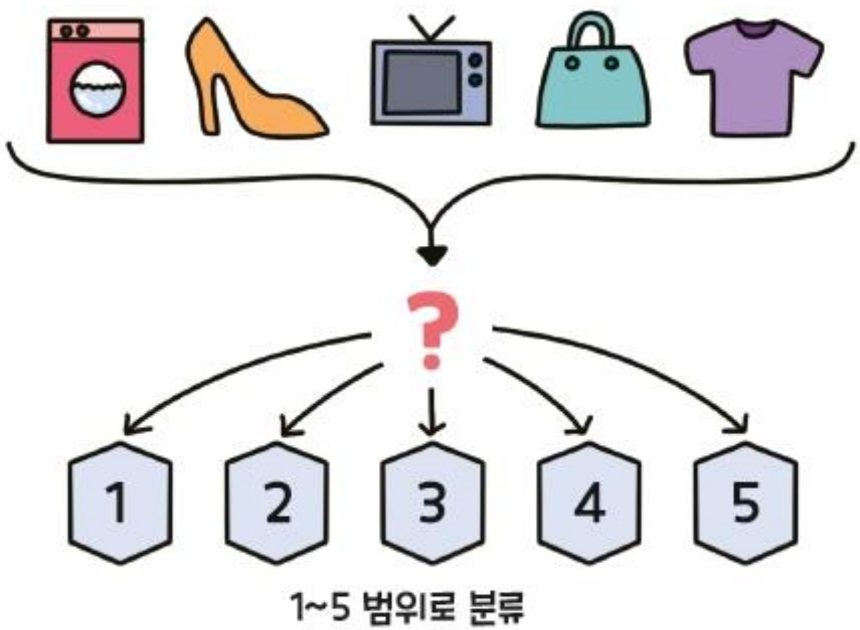
❖ Reinforcement learning

- Depending on the behavior chosen by the agent in a given environment, you are rewarded if the behavior is the right choice, and punished if the behavior is the wrong choice
- Reinforcement learning allows the agent to keep an eye on the status and learn (behavior) toward higher rewards

Types of Machine Learning Algorithm

❖ Classification

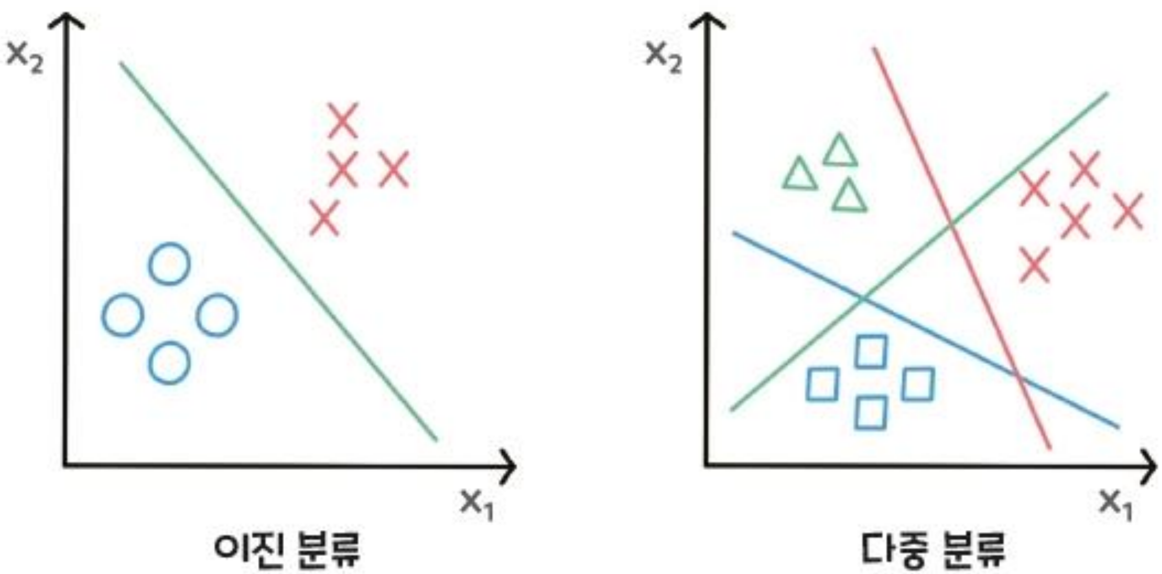
- A technique for learning labeled data, classifying data with similar properties, and finding out which group the newly entered data belongs to



Types of Machine Learning Algorithm

❖ Classification

- Binary classification : Categorize data into 2 groups
- Multiclass classification : Categorize data into 3 or more groups



Types of Machine Learning Algorithm

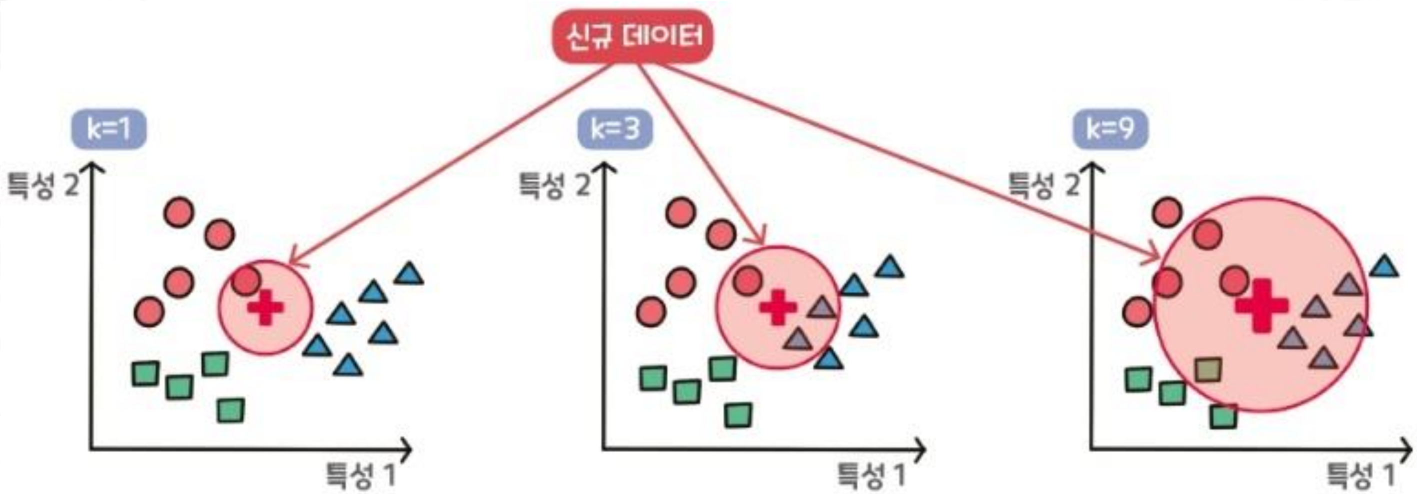
❖ Classification (Algorithm)

- K-neighbor nearest
- Support vector machine
- Decision tree

Types of Machine Learning Algorithm

❖ Classification (Algorithm)

- K-neighbor nearest
  - Algorithms to classify which of the existing groups of data (K groups) belongs to when new data comes in
  - (Example) When new data is entered when K=1, new data is classified as a red circle, when K=3, and when K=9, it is classified as a blue triangle



Types of Machine Learning Algorithm

❖ Classification (Algorithm)

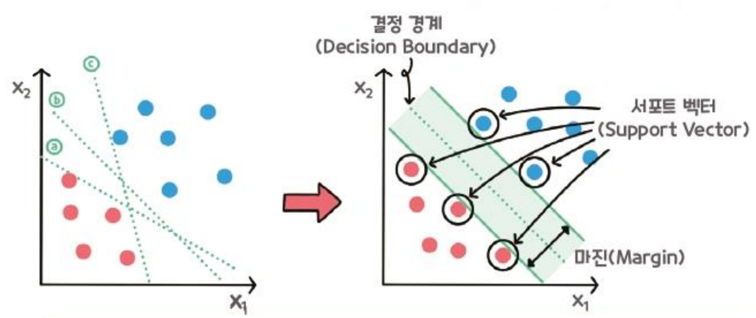
- K-neighbor nearest
  - KNNs are not significantly affected by the noise present in the learning data and are quite effective when the number of learning data is large
  - However, it is unclear which hyperparameters are suitable for analysis, so there is a disadvantage that researchers should randomly select according to each characteristic of the data

Types of Machine Learning Algorithm

❖ Classification (Algorithm)

▪ Support vector machine

- Categorize data in the direction of maximizing margin, which means margin between two categories
- SVMs find and classify lines that maximize margins, so larger margins are more likely to be classified even if new data comes in
- SVM is easy to use and highly predictive
  - – However, it takes time to build a model and the results are less descriptive

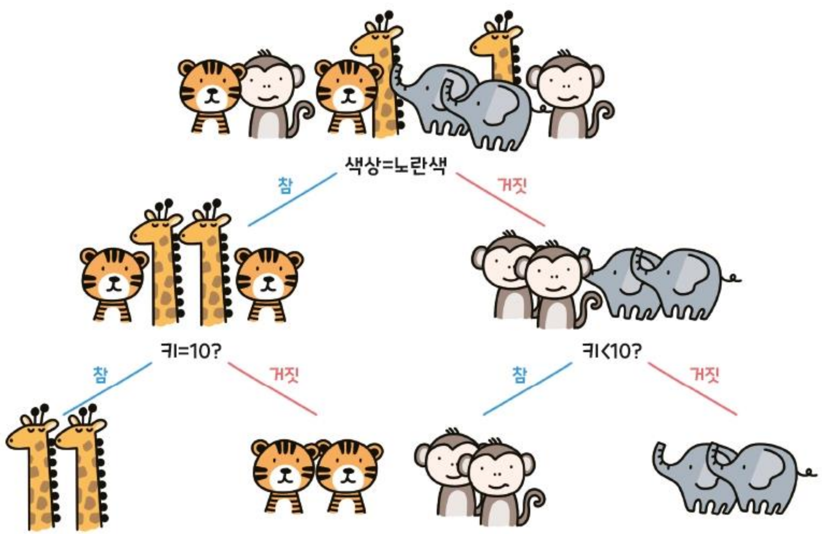


· 결정 경계(Decision Boundary) : 분류를 위한 기준선
· 서포트 벡터(Support Vector) : 결정 경계와 가장 가까운 위치에 있는 데이터
· 마진(Margin) : 결정 경계와 서포트 벡터 사이의 거리

Types of Machine Learning Algorithm

❖ Classification (Algorithm)

- Decision tree
  - An analysis method for classifying decision-making rules into tree forms
  - It is called 'decision tree' because the method of starting from the upper node and expanding to the lower node according to the classification criteria resembles 'tree'



Types of Machine Learning Algorithm

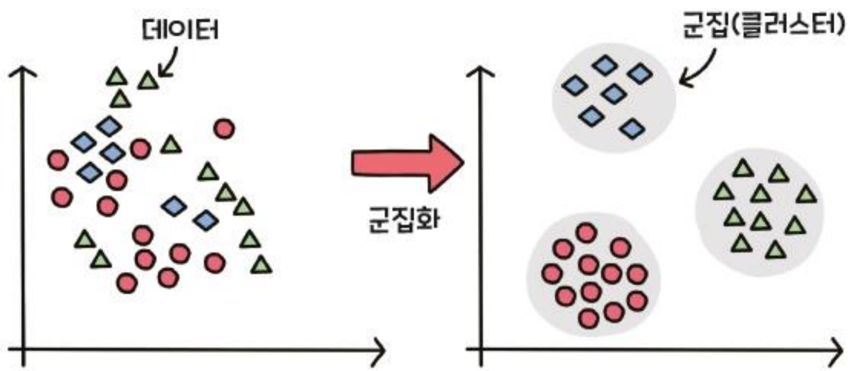
❖ Classification (Algorithm)

- Decision tree
  - Decision Tree is intuitive and easy to understand the analysis process
  - In the case of artificial neural networks, it is a black box model that is difficult to explain the analysis results, while decision trees can observe the analysis process with their eyes
  - Need for a clear explanation of the results

Types of Machine Learning Algorithm

❖ Unsupervised learning

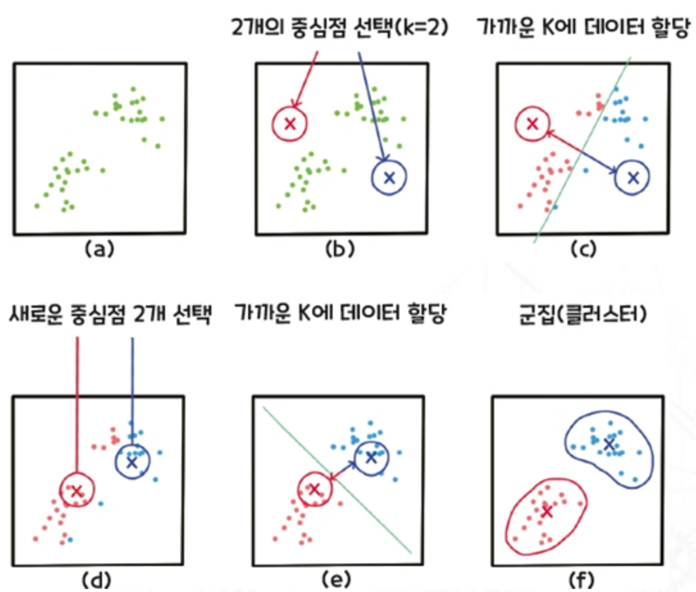
- Cluster
  - A group of data with similar characteristics
- Clustering
  - Classifying the data into clusters according to a similar degree when given the data
  - Various data are mixed together, but the clustering process groups similar data as shown in the graph on the right



Types of Machine Learning Algorithm

❖ Unsupervised learning

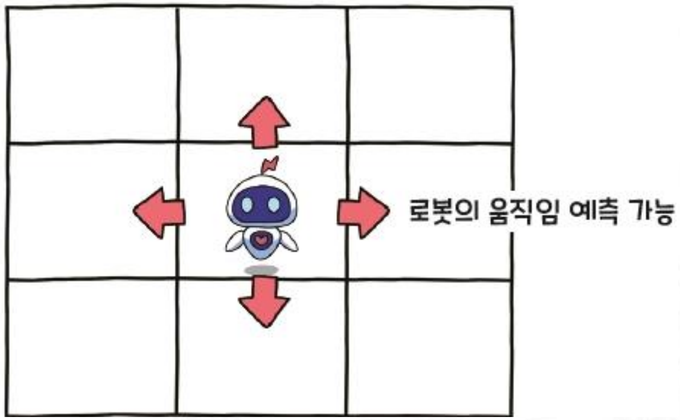
- K-means clustering
  - 'K' is the number of groups to be grouped from the given data
  - ‘Means’ means the average distance between the center of each cluster and the data
  - The center of the cluster is called centroids



Types of Machine Learning Algorithm

❖ Reinforcement learning

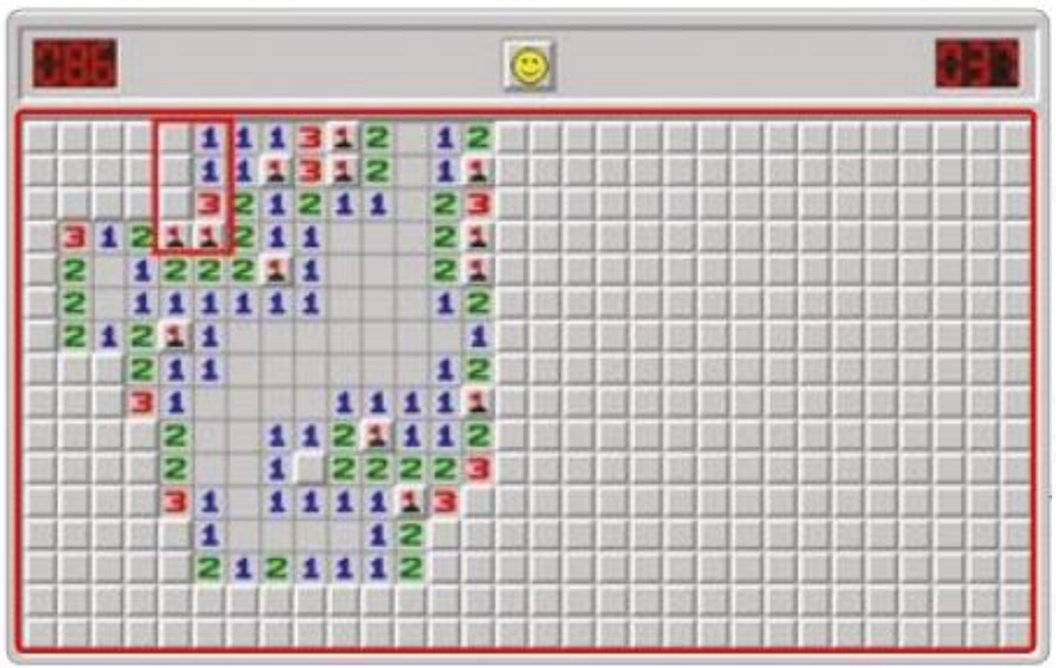
- Algorithm
  - Model-based algorithms refer to the probability that an action in the current state will result in the next state
  - Intuitive visibility of the robot's next state as it moves up, down, left, and right in a grid space
    - – Model-based algorithms can predict changes in state according to behavior, resulting in optimal solutions



Types of Machine Learning Algorithm

❖ Reinforcement learning

- Algorithm
  - Finding a policy that maximizes the rewards an agent receives through action



Examples

❖ Install the scikit-learn library

- pip install scikit-learn

TIP https://scikit-learn.org/dev/_downloads/scikit-learn-docs.pdf에 접속하면 가장 최신 버전의 사이킷 런 사용 설명서를 무료로 다운로드할 수 있다. 무려 2,500여 쪽에 달하는 방대한 문서다. 그렇다고 겁먹을 필요는 없다. 필요한 부분을 선택적으로 참조하면 된다.

Load ‘iris’ Dataset

❖ Load the dataset

프로그램 3-1(a) iris 데이터셋 읽기

```
01  from sklearn import datasets
02
03  d=datasets.load_iris()    # iris 데이터셋을 읽고
04  print(d.DESCR)            # 내용을 출력
```

- 01행: sklearn 모듈의 datasets 클래스를 불러옴
- 03행: load_iris 함수를 호출해 iris 데이터셋을 읽어 객체 d에 저장
- 04행: 객체 d의 DESCR 변수를 출력

❖ Terminology

- Dataset
- Feature vector
- Class

TIP 기계 학습이 사용하는 데이터는 여러 개의 샘플을 담고 있어서 데이터셋(data set)이라 부르기도 한다. 이 책에서는 데이터와 데이터셋을 엄밀히 구분하지 않고 함께 사용하는데, 데이터셋은 iris처럼 특정한 데이터를 가리킬 때 주로 사용한다.

Load 'iris' Dataset

```
Iris plants dataset
--------------------

**Data Set Characteristics:**
```

150개의 샘플

```
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
```

네 개의 특징(feature)

```
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
```

세 개의 부류

```
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ================ ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ================ ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ================ ==== ==== ======= ===== ====================
:Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
...
```

Load ‘iris’ Dataset

❖ ‘iris’ dataset

프로그램 3-1(b) iris의 내용 살펴보기

```
for i in range(0,len(d.data)):        # 샘플을 순서대로 출력
    print(i+1,d.data[i],d.target[i])
```

```
1 [5.1 3.5 1.4 0.2] 0
2 [4.9 3.  1.4 0.2] 0
3 [4.7 3.2 1.3 0.2] 0
4 [4.6 3.1 1.5 0.2] 0
...
51 [7.  3.2 4.7 1.4] 1
52 [6.4 3.2 4.5 1.5] 1
53 [6.9 3.1 4.9 1.5] 1
54 [5.5 2.3 4.  1.3] 1
...
101 [6.3 3.3 6.  2.5] 2
102 [5.8 2.7 5.1 1.9] 2
103 [7.1 3.  5.9 2.1] 2
104 [6.3 2.9 5.6 1.8] 2
...
```

d.target(레이블)

d.data(특징 벡터)

Representation of dataset

❖ Representing samples as feature vectors and labels

- Feature vectors are denoted by x 특징 벡터: $\mathbf{x}=(x_1,x_2,\cdots,x_d)$
  - d is the number of features called the dimension of the feature vector
- Labels are 0,1,2,...A value of ,c-1 or 1,2,...A value of ,c-1,c or one hot code
  - One hot code is a binary sequence with only one element
  - Ex) Setosa: (1,0,0), Versicolor: (0,1,0), Virginica: (0,0,1)

	특징 벡터 $\mathbf{x}=(x_1, x_2,\cdots, x_d)$	레이블(참값) y
샘플 1:	(5.1, 3.5, 1.4, 0.2)	0
샘플 2:	(4.9, 3.0, 1.4, 0.2)	0
…	…	…
샘플 51:	(7.0, 3.2, 4.7, 1.4)	1
샘플 52:	(6.4, 3.2, 4.5, 1.5)	1
…	…	…
샘플 101:	(6.3, 3.3, 6.0, 2.5)	2
샘플 102:	(5.8, 2.7, 5.1, 1.9)	2
…	…	…
샘플 n:	(5.9, 3.0, 5.1, 1.8)	2

iris 데이터셋
(n=150, d=4)

Data Distribution of Feature Space

❖ iris dataset

- Distribution of data in a three-dimensional space, excluding one data dimension

프로그램 3-2 iris 데이터의 분포를 특징 공간에 그리기

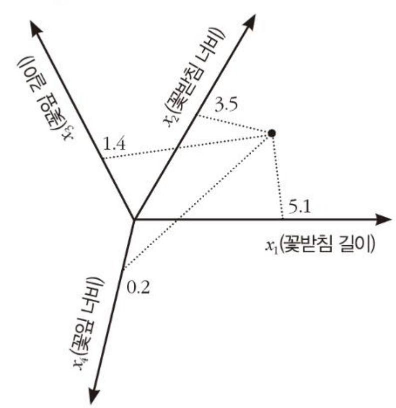
```
import plotly.express as px

df = px.data.iris()
fig = px.scatter_3d(df, x='sepal_length', y='sepal_width', z='petal_width',
color='species')        # petal_length를 제외하여 3차원 공간 구성
fig.show(renderer="browser")
```

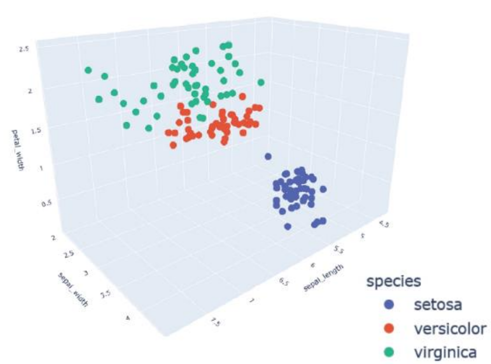
Data Distribution of Feature Space

❖ Observe the distribution of data in the feature space

- Setosa is distributed downward and Virginica is distributed upward for the vertical width
  - Petal width is excellent in discernment
- The segmental width axis overlaps a lot in three categories, so it is less sensible
- As a whole, the three classes occupy different areas of the three-dimensional space, with several samples overlapping



(a) 4차원 특징 공간(가상의 그림)



(b) 꽃잎 길이 축을 제외한 3차원 특징 공간

Data Distribution of Feature Space

NOTE 다차원 특징 공간

종이에 그릴 수 있는 공간은 3차원으로 제한되지만, 수학은 아주 높은 차원까지 다룰 수 있다. 예를 들어 2차원 상의 두 점 $\mathbf{x}=(x_1,x_2)$와 $\mathbf{y}=(y_1,y_2)$의 거리를 $d(\mathbf{x},\mathbf{y})=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$으로 계산할 수 있는데, 4차원 상의 두 점 $\mathbf{x}=(x_1,x_2,x_3,x_4)$와 $\mathbf{y}=(y_1,y_2,y_3,y_4)$의 거리는 $d(\mathbf{x},\mathbf{y})=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2+(x_4-y_4)^2}$로 계산할 수 있다.

일반적으로 d차원 상의 두 점의 거리는 $d(\mathbf{x},\mathbf{y})=\sqrt{\sum_{i=1}^{d}(x_i-y_i)^2}$ 로 계산한다. 기계 학습에서는 d=수백~수만에 달하는 매우 고차원 특징 공간의 데이터를 주로 다룬다.

Modeling and Prediction

❖ Using the support vector machine model

프로그램 3-1(c) iris에 기계 학습 적용: 모델링과 예측

```
07  from sklearn import svm          Hyperparameter
08
09  s=svm.SVC(gamma=0.1,C=10)                       # svm 분류 모델 SVC 객체 생성하고
10  s.fit(d.data,d.target) Training set             # iris 데이터로 학습
11
12  new_d=[[6.4,3.2,6.0,2.5],[7.1,3.1,4.7,1.35]]    # 101번째와 51번째 샘플을 변형하여
                                                      새로운 데이터 생성
13  res=s.predict(new_d) Test set
14  print("새로운 2개 샘플의 부류는", res)
```

```
새로운 2개 샘플의 부류는 [2 1]
```

- 09행: SVM의 분류기 모델 SVC 클래스의 객체를 생성하여 s에 저장
- 10행: 객체 s의 fit 함수는 훈련 집합을 가지고 학습을 수행
  (매개변수로 특징 벡터 iris.data와 레이블 iris,target을 설정)
- 13행: 객체 s의 predict 함수는 테스트 집합을 가지고 예측 수행

Performance Measurement

- The importance of objective performance measurements
  - Important when choosing a model
  - Important when deciding whether to install on-site
- Generalization capabilities
  - Performance on new data not used for learning
  - The most obvious way is to install it on-site and measure performance
  - Cost makes it difficult to apply it in real life
  - Requires wisdom to segment and use given data

Performance Measurement

❖ Confusion matrix

- Matrix recording the number of correct and incorrect classifications by class
  - n_{ij}는 모델이 i라고 예측했는데 실제 부류는 j인 샘플의 개수

		참값(그라운드 트루스)					
		부류 1	부류 2	…	부류 j	…	부류 c
예측한 부류	부류 1	n_{11}	n_{12}		n_{1j}		n_{1c}
	부류 2	n_{21}	n_{22}		n_{2j}		n_{2c}
	…						
	부류 i	n_{i1}	n_{i2}		n_{ij}		n_{ic}
	…						
	부류 c	n_{c1}	n_{c2}		n_{cj}		n_{cc}

(a) 부류가 c개인 경우

		그라운드 트루스	
		긍정	부정
예측값	긍정	TP	FP
	부정	FN	TN

(b) 부류가 2개인 경우

- Positive and negative negative in binary classification
- True positive, false negative, false positive, true negative

Performance Measurement

❖ Performance metric

- Accuracy

$$정확률 = \frac{맞힌\ 샘플\ 수}{전체\ 샘플\ 수} = \frac{대각선\ 샘플\ 수}{전체\ 샘플\ 수}$$

- Specificity and sensitivity

$$특이도 = \frac{TN}{TN+FP},\quad 민감도 = \frac{TP}{TP+FN}$$

- Precision and recall

$$정밀도 = \frac{TP}{TP+FP},\quad 재현율 = \frac{TP}{TP+FN}$$

Divide into Training/Validation/Test

❖ Training/Validation/Test

- Training set
  - Data used to learn machine learning models that provide both feature vector and label information
- Test Set
  - Data used to measure the performance of a learned model, which provides only feature vector information when predicting, and uses label information when measuring accuracy with prediction results

NOTE 하이퍼 매개변수 설정

하이퍼 매개변수hyper parameter란 모델의 동작을 제어하는 데 쓰는 변수이다. 모델의 학습을 시작하기 전에 설정해야 하는데, 적절한 값으로 설정해야 좋은 성능을 얻을 수 있다. 최적의 하이퍼 매개변수 값을 자동으로 설정하는 일을 하이퍼 매개변수 최적화(hyper parameter optimization)라 하는데, 이것은 기계 학습의 중요한 주제 중 하나다. 하이퍼 매개변수 최적화는 4.10절에서 다룬다.

Divide into Training/Validation/Test

❖ Divide the given data into training, validation, and test sets at an appropriate rate

- Model selection included: divided into training/validation/test sets
- Exclude model selection: split into training/test sets

Divide into Training/Validation/Test

❖ Exclude the model selection

- 08행: train_test_split 함수로 훈련 60%, 테스트 40%로 랜덤 분할
- 12행: 훈련 집합 x_train, y_train을 fit 함수에 주어 학습 수행
- 14행: 테스트 집합의 특징 벡터 x_test를 predict 함수에 주어 예측 수행
- 17~20행: 테스트 집합의 레이블 y_test를 가지고 혼동 행렬 계산

프로그램 3-5 필기 숫자 인식 – 훈련 집합으로 학습하고 테스트 집합으로 성능 측정

```
01 from sklearn import datasets
02 from sklearn import svm
03 from sklearn.model_selection import train_test_split
04 import numpy as np
05
06 # 데이터셋을 읽고 훈련 집합과 테스트 집합으로 분할
07 digit=datasets.load_digits()
08 x_train,x_test,y_train,y_test=train_test_split(digit.data,digit.target,train_size=0.6)
09
```

Divide into Training/Validation/Test

예) 부류 3에 속하는 75개 샘플 중 73개를 3, 1개를 2, 1개를 7로 인식

```python
# svm의 분류 모델 SVC를 학습
s=svm.SVC(gamma=0.001)
s.fit(x_train,y_train)

res=s.predict(x_test)

# 혼동 행렬 구함
conf=np.zeros((10,10))
for i in range(len(res)):
    conf[res[i]][y_test[i]]+=1
print(conf)

# 정확률 측정하고 출력
no_correct=0
for i in range(10):
    no_correct+=conf[i][i]
accuracy=no_correct/len(res)
print("테스트 집합에 대한 정확률은", accuracy*100, "%입니다.")
```

```
[[76.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0. 78.  0.  0.  0.  0.  0.  0.  3.  0.]
 [ 0.  0. 66.  1.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0. 73.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 63.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0. 70.  0.  0.  0.  2.]
 [ 0.  0.  0.  0.  0.  0. 77.  0.  0.  0.]
 [ 0.  0.  0.  1.  0.  0.  0. 77.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 74.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  0.  0. 56.]]

테스트 집합에 대한 정확률은 98.7482614742698l%입니다.
```

Cross-Validation

- Limitations of training/test set division
  - Likelihood of accidental high or accidental low accuracy
- k-fold cross validation
  - Use the training set divided into k subsets
  - Measure performance by learning with k-1 leaving one and then leaving it
  - Increase reliability by averaging k performance

Cross-Validation

Cross-Validation

프로그램 3-6 필기 숫자 인식 – 교차 검증으로 성능 측정

```python
from sklearn import datasets
from sklearn import svm
from sklearn.model_selection import cross_val_score
import numpy as np

digit=datasets.load_digits()
s=svm.SVC(gamma=0.001)
accuracies=cross_val_score(s,digit.data,digit.target,cv=5) # 5-겹 교차 검증

print(accuracies)
print("정확률(평균)=%0.3f, 표준편차=%0.3f"%(accuracies.mean()*100,accuracies.std()))
```

```
[0.97527473 0.95027624 0.98328691 0.99159664 0.95774648]
정확률(평균)=97.164, 표준편차=0.015
```

Challenging Issues for ML

❖ Need for a variety of data

- Computers can only recognize numbers 0 and 1 unlike humans
- In other words, the object in the image should be expressed as a number, and a separate processing (weight of machine learning) should be performed to recognize only the car among the objects
  - Through this complex process, a lot of data is needed to accurately recognize various types of cars

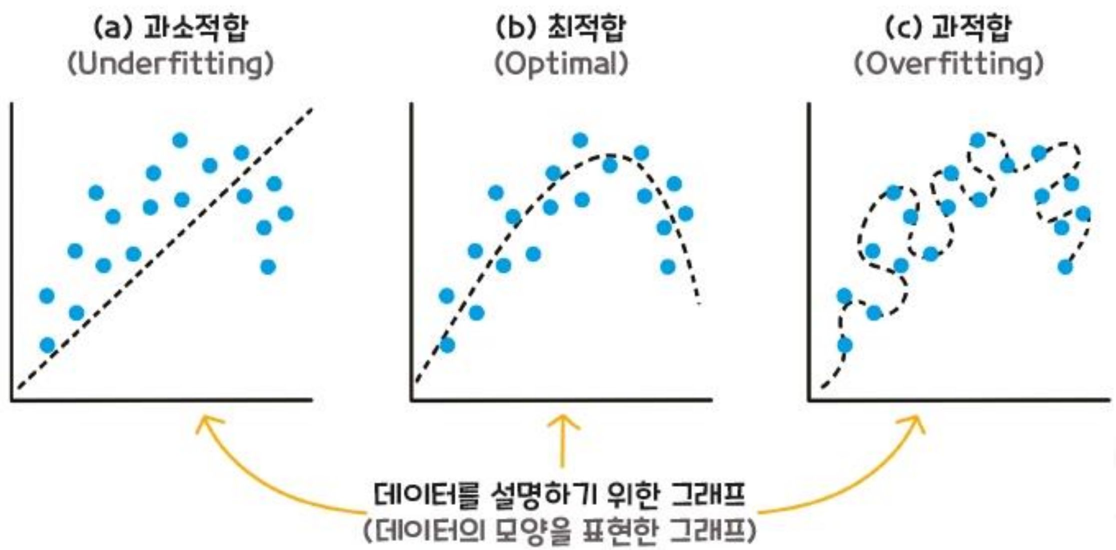
Challenging Issues for ML

❖ Overfitting

- Overlearning training data means poor performance when analyzing real data
- This occurs when data is significantly lacking compared to the complexity of the problem,
  - i.e. learning data does not cover the entire space where the problem is defined and focuses only in some cases

Challenging Issues for ML

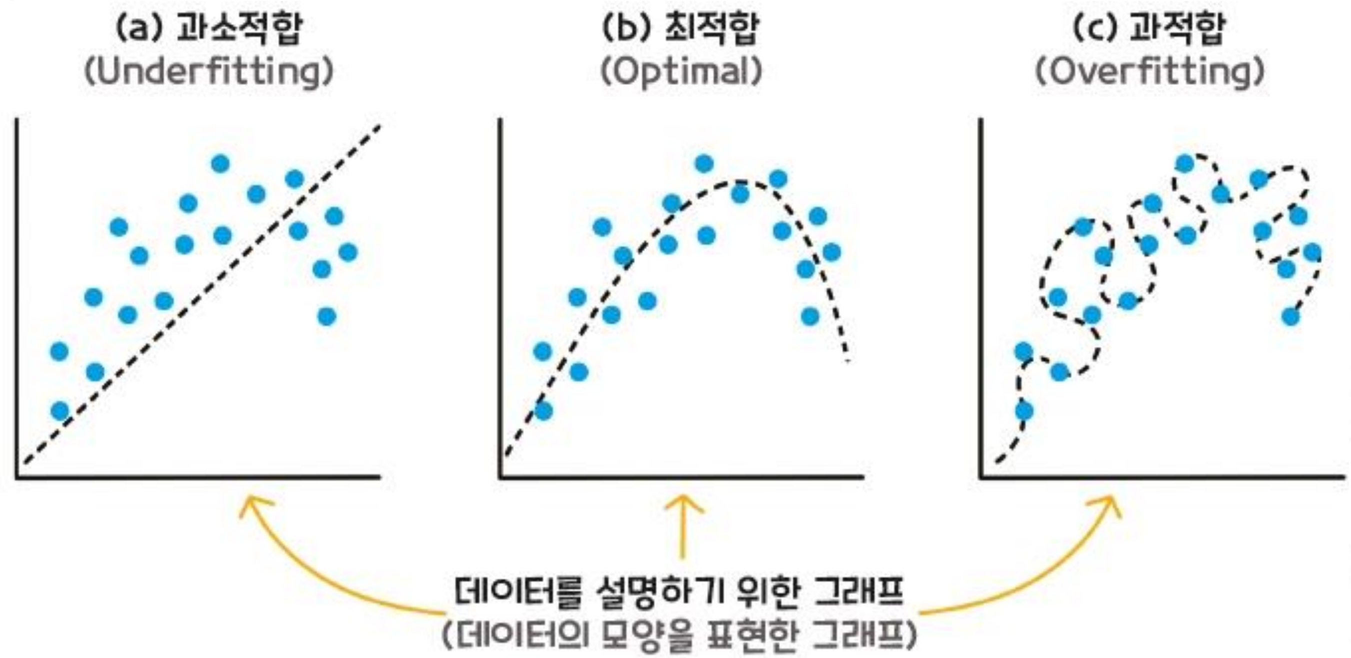
❖ Overfitting



– (a) 그래프 : 변수가 0인 쪽의 데이터 몇 개는 비교적 잘 근사하지만, 일정 시점 이후 데이터는 우하향하고 있음
– (b) 그래프 : 데이터와 비슷하게 우상향하고 있어 제대로 반영하고 있다고 볼 수 있음

Challenging Issues for ML

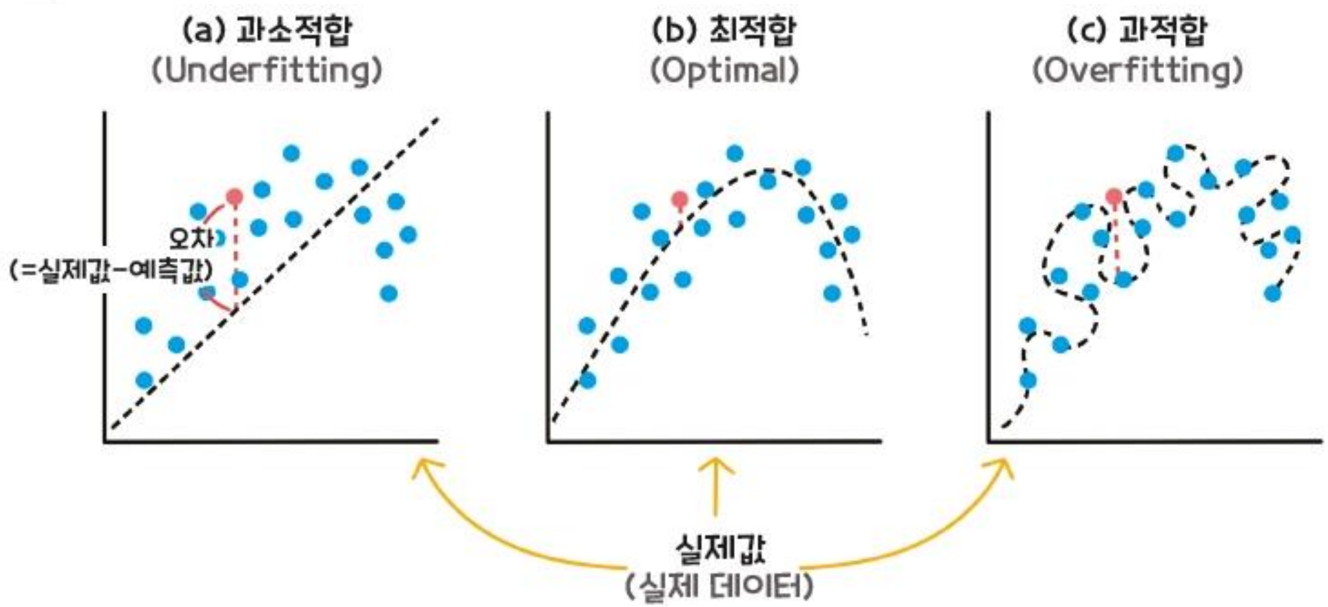
❖ Overfitting



- (c) 그래프 : 학습 데이터와 생성된 모델의 오차를 구해보면 0에 가까울 것임. 즉, 그래프가 모든 점을 지나고 있음
- 그렇다면 (b) 그래프보다 (c) 그래프가 더 좋은 그래프일까? 어떤 그래프가 좋은 그래프인지 알아보기 위해서는 실제 데이터값을 불러오면 됨

Challenging Issues for ML

❖ Overfitting



– 하나의 실제값을 불러왔을 때, 실제값과 모델이 내놓은 예측값의 차이인 오차가 가장 작은 그래프가 좋은 그래프라고 할 수 있음
– 확인해 보면 (b) 그래프의 오차가 가장 작으므로 최적합(Optimal), 즉 가장 좋은 그래프라고 할 수 있음