인공지능

Fundamentals

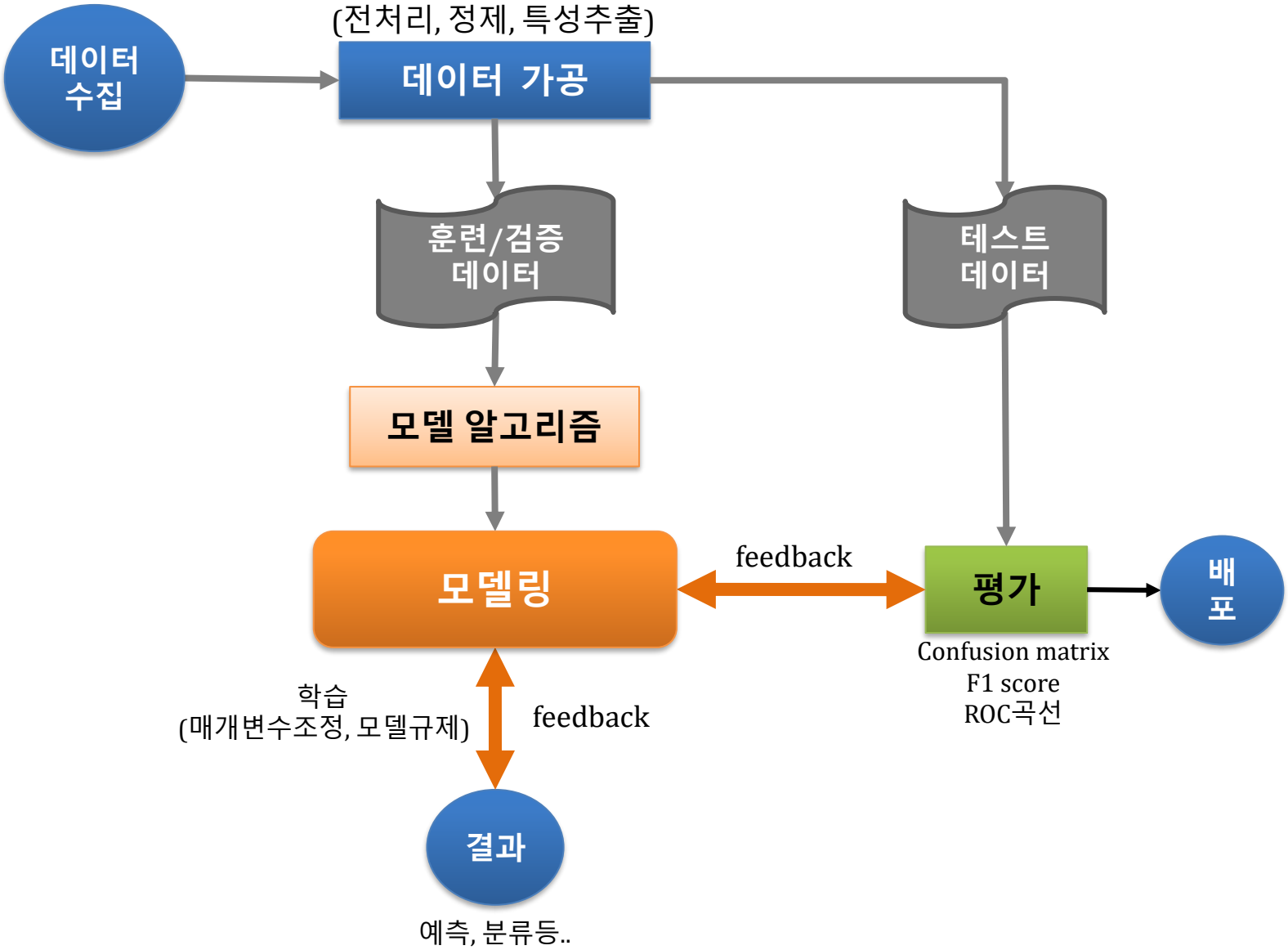
김선녕(ksycafe@gmail.com)

Data Set

Learning Types

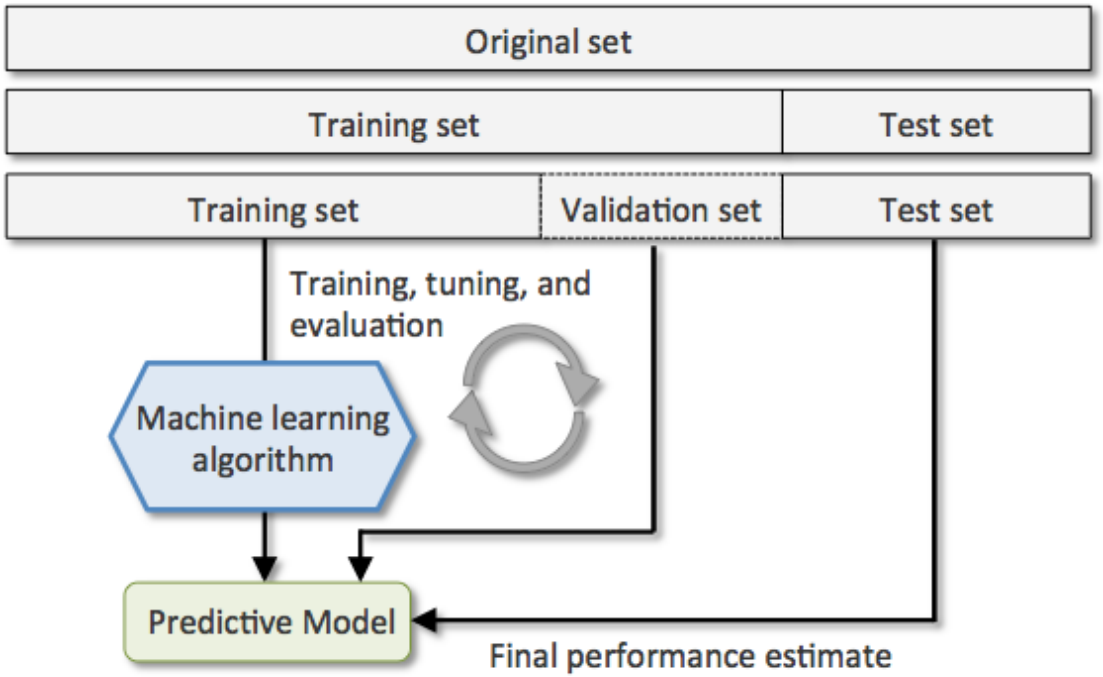
Fitting/One-Hot Encoding

머신러닝 절차

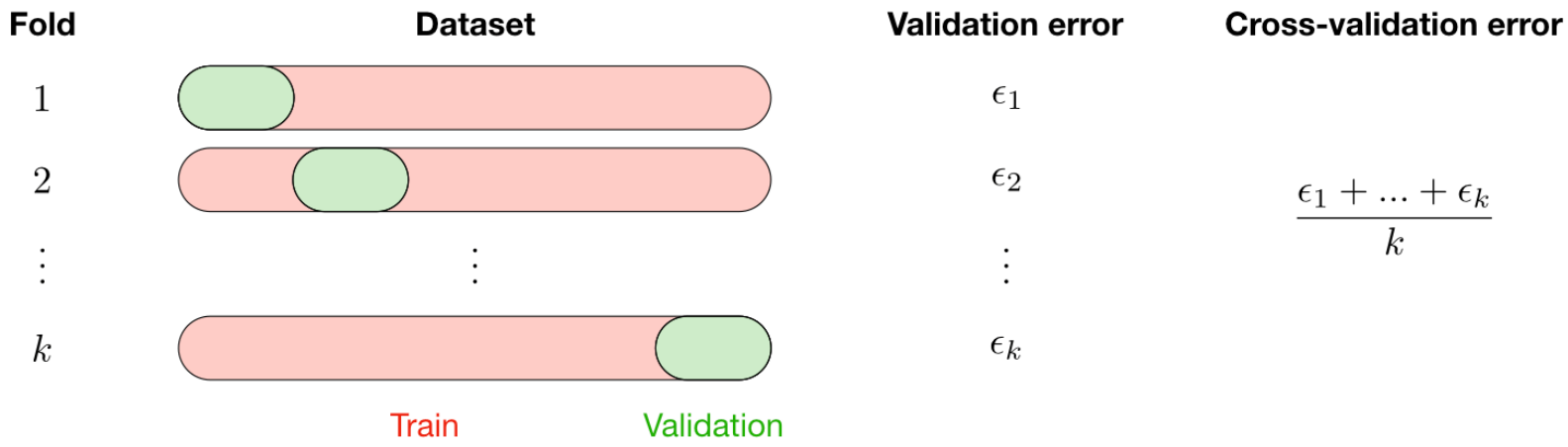

데이터
수집
(전처리, 정제, 특성추출)
데이터 가공
훈련/검증
데이터
테스트
데이터
모델 알고리즘
모델링
feedback
평가
배
포
Confusion matrix
F1 score
ROC곡선
학습
(매개변수조정, 모델규제)
feedback
결과
예측, 분류등..

데이터셋

- 훈련데이터($Training\ data$) : 머신러닝 모델을 만들 때 사용(실제데이터)
- 검증데이터($Validation\ Data$) : 훈련데이터에서 분할. $learning\ rate$ 또는 $regularization,\ parameter$ 등을 튜닝하는 데 사용. 모델 성능개선
- 테스트데이터($Test\ data$): 최종 성능 측정. 모델이 얼마나 잘 작동하는 지 측정하는 데 사용
- 분할방법
  - 훈련데이터 : 테스트데이터 = 7: 3 혹은 7.5 : 2.5
  - 훈련데이터 : 검증데이터 : 테스트데이터 = 6 : 2 : 2 혹은 7 : 1.5 : 1.5



K-겹 교차검증(k-hold cross-validation)



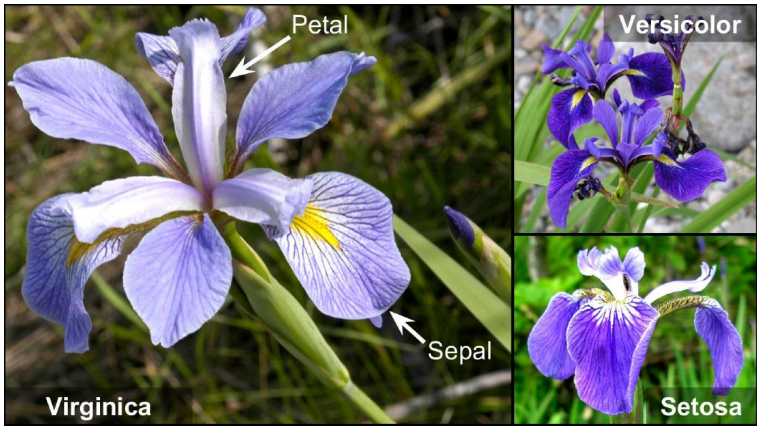
Haberman Survival Data Set(https://archive.ics.uci.edu/ml/datasets/)

- 1958년~1970년사이의 유방암 수술을 받은 환자의 생존 사례(University of Chicago's Billings Hospital)
- 총 건수 : 306건
- 속성(Attribute)
  - 환자의 나이
  - 수술년도
  - 양성수
- 분류(class) : 생존상태
  - 1 : 5년 이상 생존한 환자
  - 2 : 5년 이내 사망한 환자

```
30,64,1,1
30,62,3,1
30,65,0,1
31,59,2,1
31,65,4,1
33,58,10,1
33,60,0,1
34,59,0,2
34,66,9,2
34,58,30,1
.
.
.
77,65,3,1
78,65,1,2
83,58,2,2
```

Iris Database(https://archive.ics.uci.edu/ml/datasets/iris)

- 피셔(Fisher, 통계학)교수가 1936년 3종의 붓꽃을 50송이씩(총 150 샘플) 채취하여 만들었다.
- 속성(Attribute) : 4 개
  - 1. 꽃받침 길이(sepal length in cm)
  - 2. 꽃받침 너비(sepal width in cm)
  - 3. 꽃잎 길이(petal length in cm)
  - 4. 꽃잎 너비(petal width in cm)
- 종류(class) : setosa, versicolour, virginica



Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
				I. versicolor
				I. versicolor
6.9	3.1	4.9	1.5	*I. versicolor*
5.5	2.3	4.0	1.3	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*

Wine Data Set(https://archive.ics.uci.edu/ml/datasets/wine)

- 이탈리아의 같은 지역에서 재배되는 세 가지 품종에서 파생된 와인의 화학적 분석결과
- 총 건수 : 178건(각각 59, 71, 48)
- 속성(Attribute) : 13 개
  - Alcohol
  - Malic acid
  - Ash
  - Alcalinity of ash
  - Magnesium
  - Total phenols
  - Flavanoids
  - Nonflavanoid phenols
  - Proanthocyanins
  - Color intensity
  - Hue
  - OD280/OD315 of diluted wines
  - Proline
- 종류(class) : 3

```
1,14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065
1,13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050
1,13.16,2.36,2.67,18.6,101,2.8,3.24,.3,2.81,5.68,1.03,3.17,1185
..
..
2,12.37,.94,1.36,10.6,88,1.98,.57,.28,.42,1.95,1.05,1.82,520
2,12.33,1.1,2.28,16,101,2.05,1.09,.63,.41,3.27,1.25,1.67,680
2,12.64,1.36,2.02,16.8,100,2.02,1.41,.53,.62,5.75,.98,1.59,450
..
..
3,12.86,1.35,2.32,18,122,1.51,1.25,.21,.94,4.1,.76,1.29,630
3,12.88,2.99,2.4,20,104,1.3,1.22,.24,.83,5.4,.74,1.42,530
3,12.81,2.31,2.4,24,98,1.15,1.09,.27,.83,5.7,.66,1.36,560
```

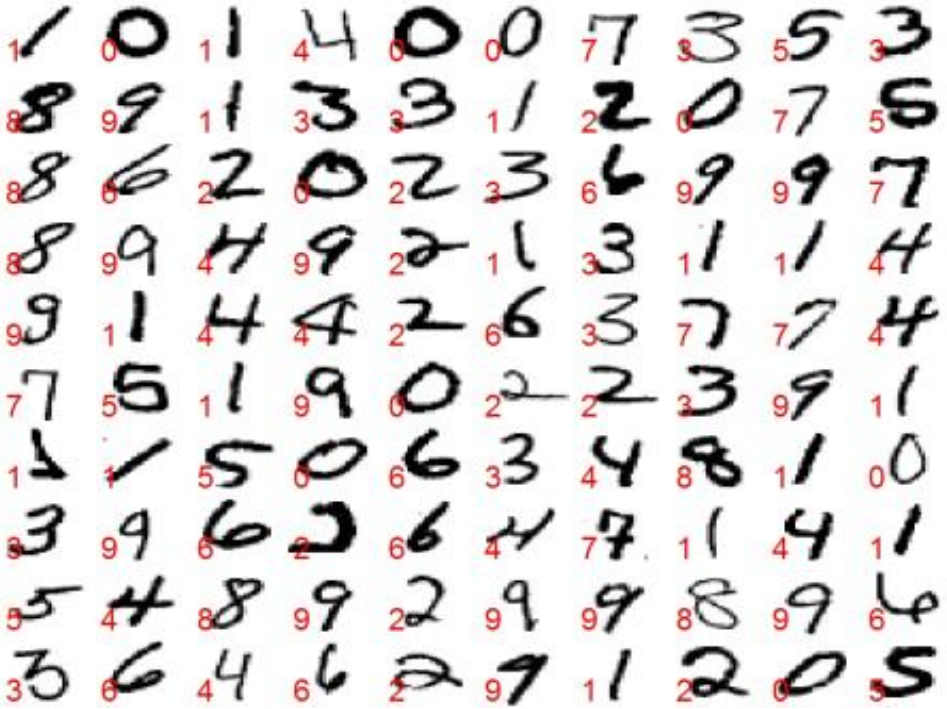
Breast Cancer Wisconsin dataset

- https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)
- 총건수 : 569건
- 속성(Attribute) : 31
  - ID number
  - radius (mean of distances from center to points on the perimeter) 반경
  - texture (standard deviation of gray-scale values) 질감
  - perimeter 둘레
  - area 면적
  - smoothness (local variation in radius lengths) 매끄러움
  - compactness (perimeter^2 / area - 1.0) 조그만 정도
  - concavity (severity of concave portions of the contour) 오목함
  - concave points (number of concave portions of the contour) 오목함 점의수
  - symmetry 대칭
  - fractal dimension ("coastline approximation" - 1) 프렉탈 차원
- 종류(class) : 2
  - M = malignant(악성) : 212건
  - B = benign(양성) : 357건

0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

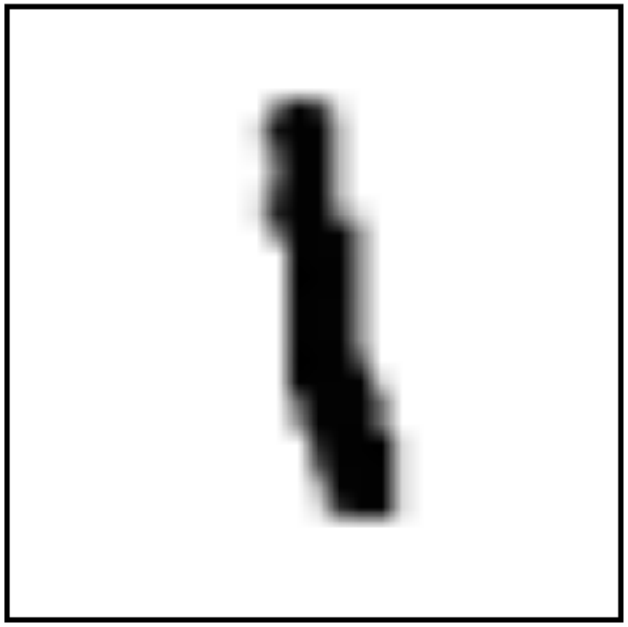
MNIST Database(http://yann.lecun.com/exdb/mnist/)

- $MNIST$: $Modified\ National\ Institute\ of\ Standards\ and\ Technology\ database$
- 미국표준국$(NIST)$에서 수집한 필기 숫자$(handwritten\ digits)$ 데이터
- 훈련데이터$(training\ set)$ 60,000, 테스트데이터$(test\ set)$ 10,000
  - $train-images-idx3-ubyte.gz$: $training\ set\ images$ (9,912,422 $bytes$)
  - $train-labels-idx1-ubyte.gz$: $training\ set\ labels$ (28,881 $bytes$)
  - $t10k-images-idx3-ubyte.gz$: $test\ set\ images$ (1,648,877 $bytes$)
  - $t10k-labels-idx1-ubyte.gz$: $test\ set\ labels$ (4,542 $bytes$)



MNIST Image구조

- 각 이미지는 $28 pixels\ by\ 28\ pixles$
  - $28 \times 28 = 784\ numbers$
  - $28 \times 28 = 784$ 차원 벡터
  - 벡터의 각 구성 요소는 0과 1 사이의 값으로 픽셀의 강도를 표시



≈

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

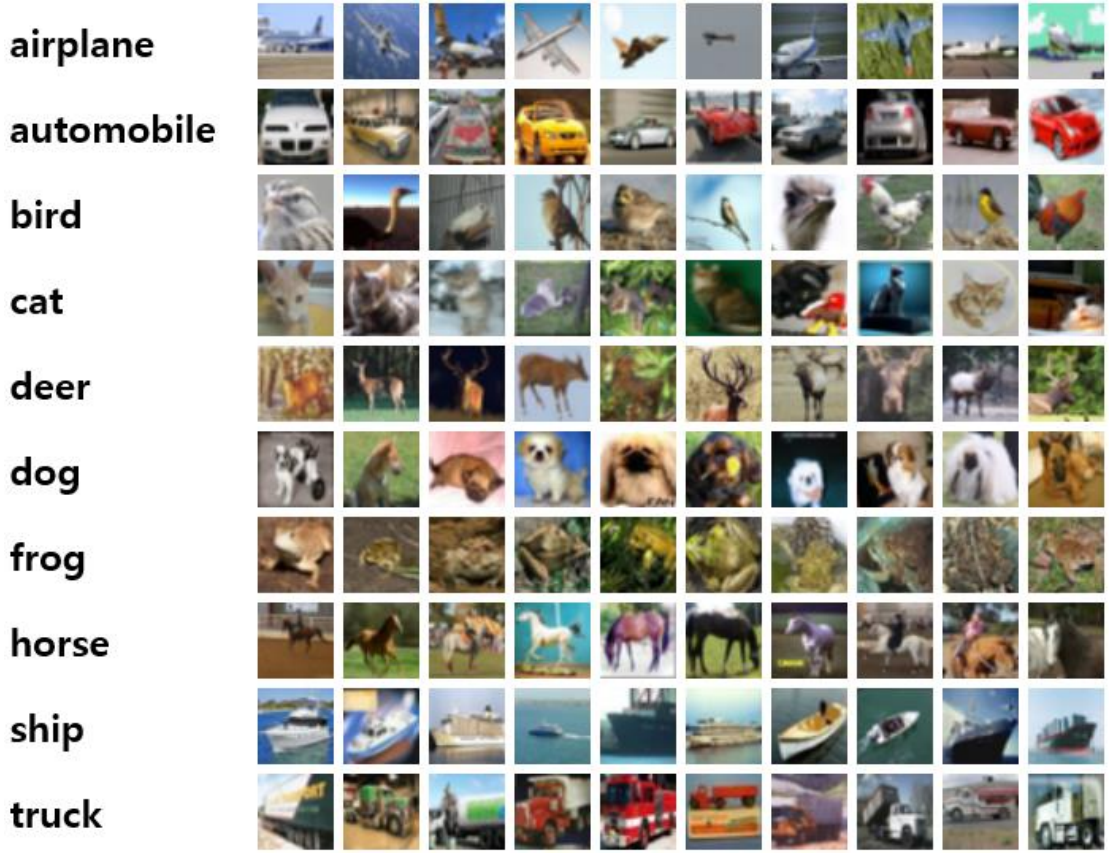
Fashion-MNIST

- *A dataset of Zalando's article images*
- *https://github.com/zalandoresearch/fashion − mnist*
- 훈련데이터(*training set*) 60,000, 테스트데이터(*test set*) 10,000

Label	*Description*
0	*T − shirt/top*
1	*Trouser*
2	*Pullover*
3	*Dress*
4	*Coat*
5	*Sandal*
6	*Shirt*
7	*Sneaker*
8	*Bag*
9	*Ankle boot*

CIFAR-10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html)

- 60,000 $(32 * 32$ *color images*$)$
  - $\mathbf{50,000}$ *training images*
  - $\mathbf{10,000}$ *testing images*
- **10** *classes* : 6,000 *images per class*



CIFAR-100 dataset (https://www.cs.toronto.edu/~kriz/cifar.html)

- 100 *classes*(20 *superclasses group*) *containing* 600 *images each*.
- 500 *training images and* 100 *testing images per class*.

20 Superclass	100 Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

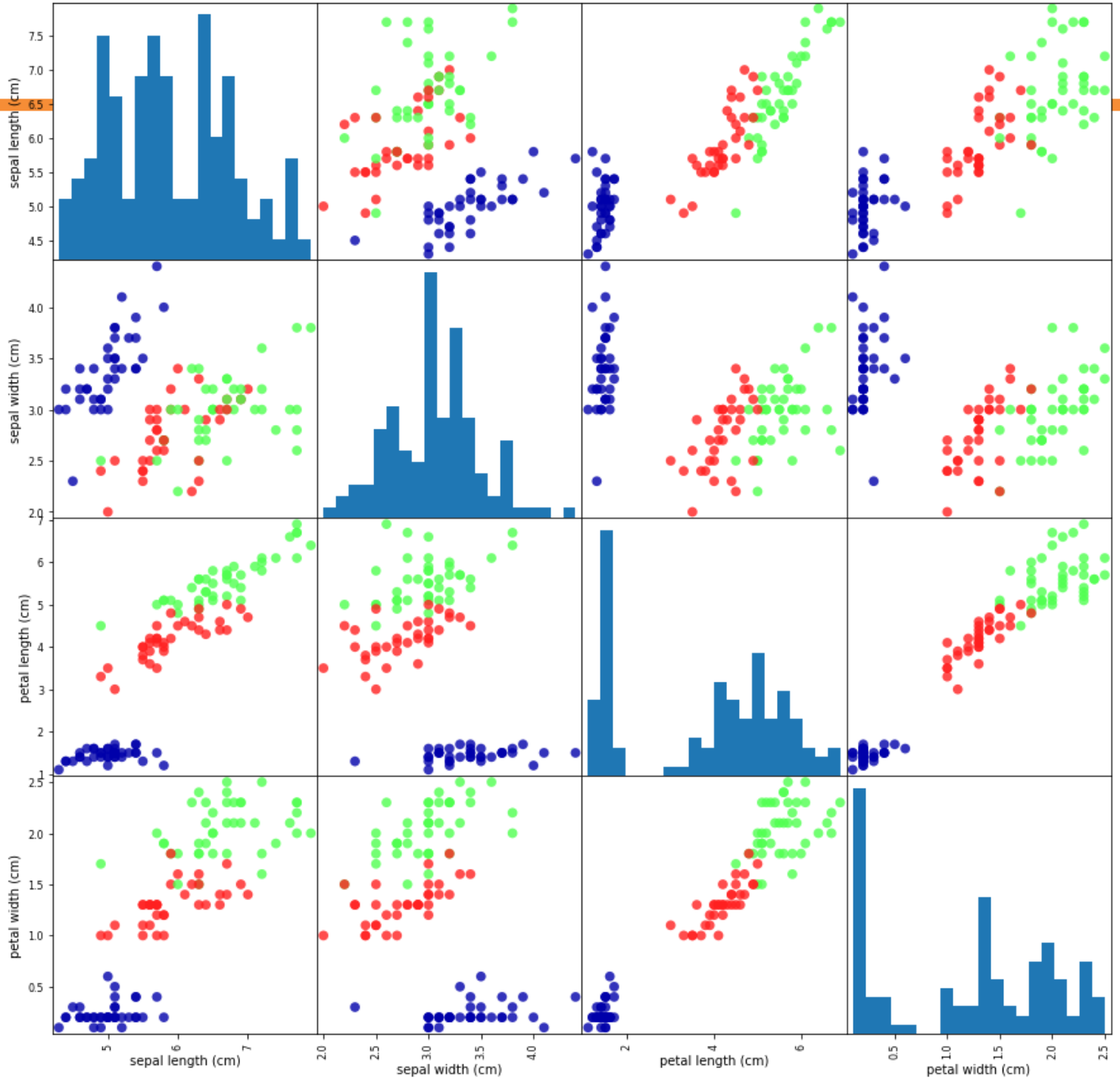
데이터 시각화 : 산점도

- 데이터에서 한 특성을 x축에 놓고 다른 하나는 y축에 놓아 각 데이터 포인트를 하나의 점으로 나타내는 그래프
- 3개 이상의 특성을 표현하기 어렵다.
- 대신 모든 특성을 짝지어 만드는 산점도 행렬(scatter matrix)을 사용할 수 있다

붓꽃 산점도 행렬

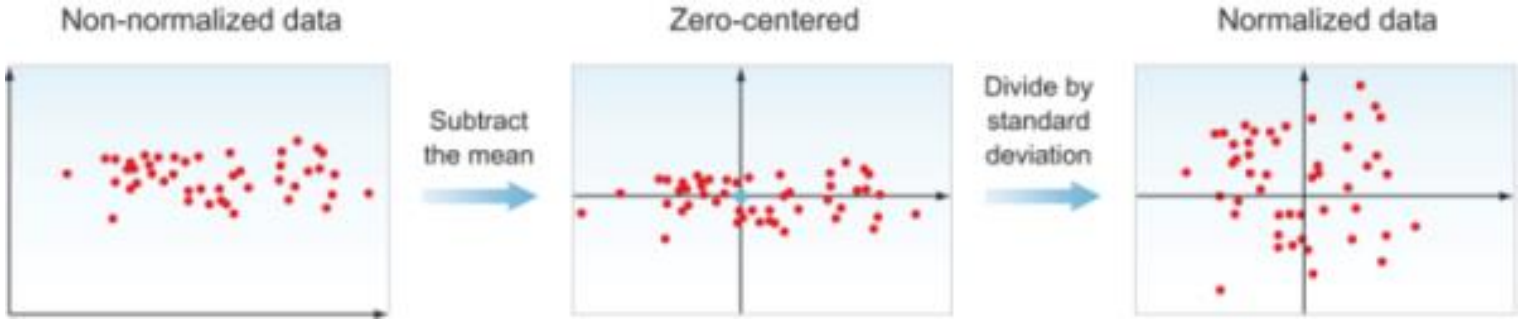

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import pandas as pd
import mglearn

iris_dataset = load_iris()
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], random_state=0)

iris_dataframe = pd.DataFrame(X_train, columns = iris_dataset.feature_names)
pd.plotting.scatter_matrix(iris_dataframe, c=y_train, figsize=(15, 15),
                           marker='o', hist_kwds={'bins':20}, s=60,
                           alpha =.8, cmap=mglearn.cm3)
```

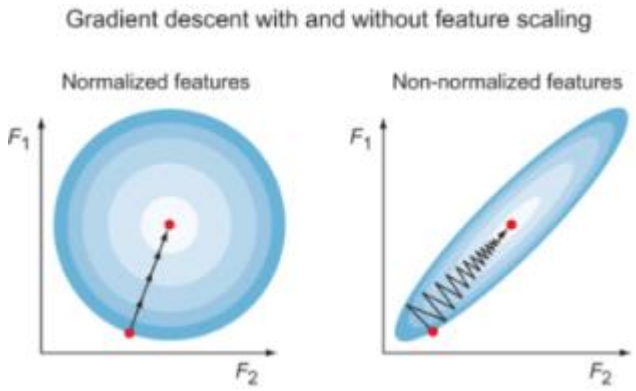
Ref : Introduction to machine learning with python – O'RELLY


sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

데이터 정규화(Data normalization)

- 값은 작게 : [0, 1] 구간
- 특성의 평균을 0으로 만들고 표준편차 1로 만든다
- (예) 사이킷런의 StandardScaler를 사용



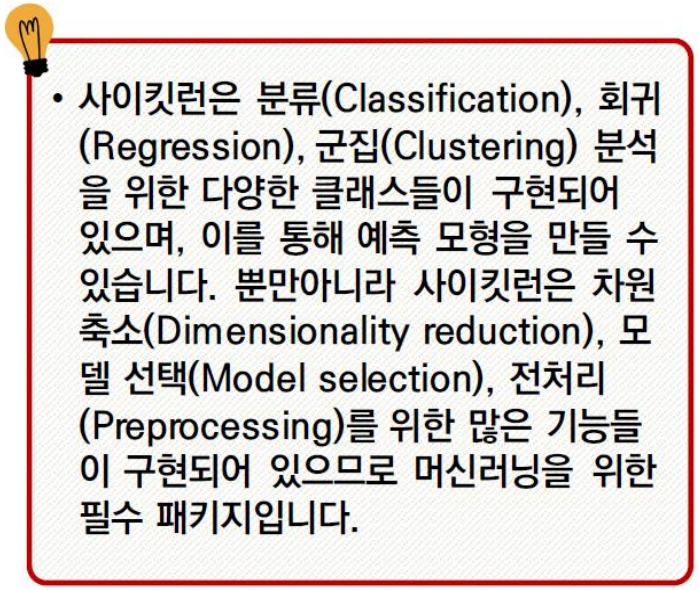


다차원 특징 공간

- 3차원 : Haberman Survival Data Set
  - Attribute : 3
- 4차원 : Iris Data Set
  - Attribute : 4
- 13차원 : Wine Data Set
  - Attribute : 13
- 784차원 : MNIST Data Set
  - Attribute : 784(28×28)
- 3072차원 : CIFAR-10 dataset
  - Attribute : 3072(32× 32 × 3)

scikit-learn 패키지

- 예측분석을 위한 간단하고 효율적인 도구
- 상업적으로 사용 가능한 오픈소스 BSD 라이센스이므로 모든 사람이 사용할 수 있음
- NumPy(넘파이), SciPy(사이파이) 및 matplotlib(맷플롯립) 기반



- 사이킷런은 분류(Classification), 회귀(Regression), 군집(Clustering) 분석을 위한 다양한 클래스들이 구현되어 있으며, 이를 통해 예측 모형을 만들 수 있습니다. 뿐만아니라 사이킷런은 차원 축소(Dimensionality reduction), 모델 선택(Model selection), 전처리(Preprocessing)를 위한 많은 기능들이 구현되어 있으므로 머신러닝을 위한 필수 패키지입니다.

데이터셋 분리


```
1 import seaborn as sns
2 iris = sns.load_dataset("iris")
3 X = iris.iloc[:, :-1]
4 y = iris.iloc[:, -1]
```

```
1 from sklearn.model_selection import train_test_split
2 train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.3)
```

```
1 train_X.shape, test_X.shape, train_y.shape, test_y.shape
```

```
((105, 4), (45, 4), (105,), (45,))
```

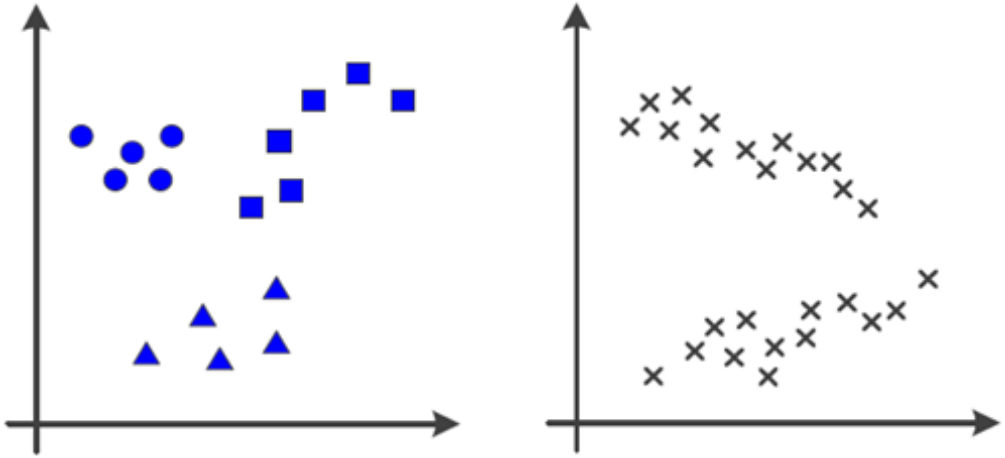
```
1 test_y.value_counts()
```

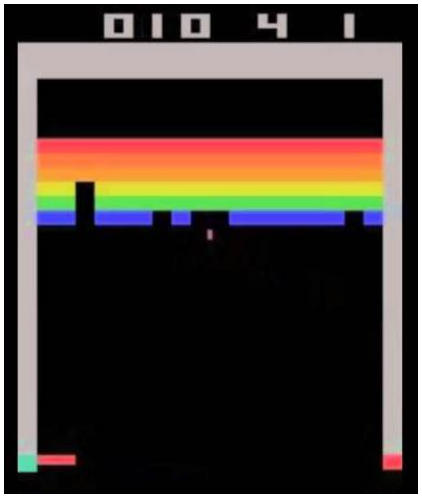
```
setosa        19
versicolor    13
virginica     13
Name: species, dtype: int64
```


Data Set
Learning Types
Fitting/One-Hot Encoding

학습의 유형

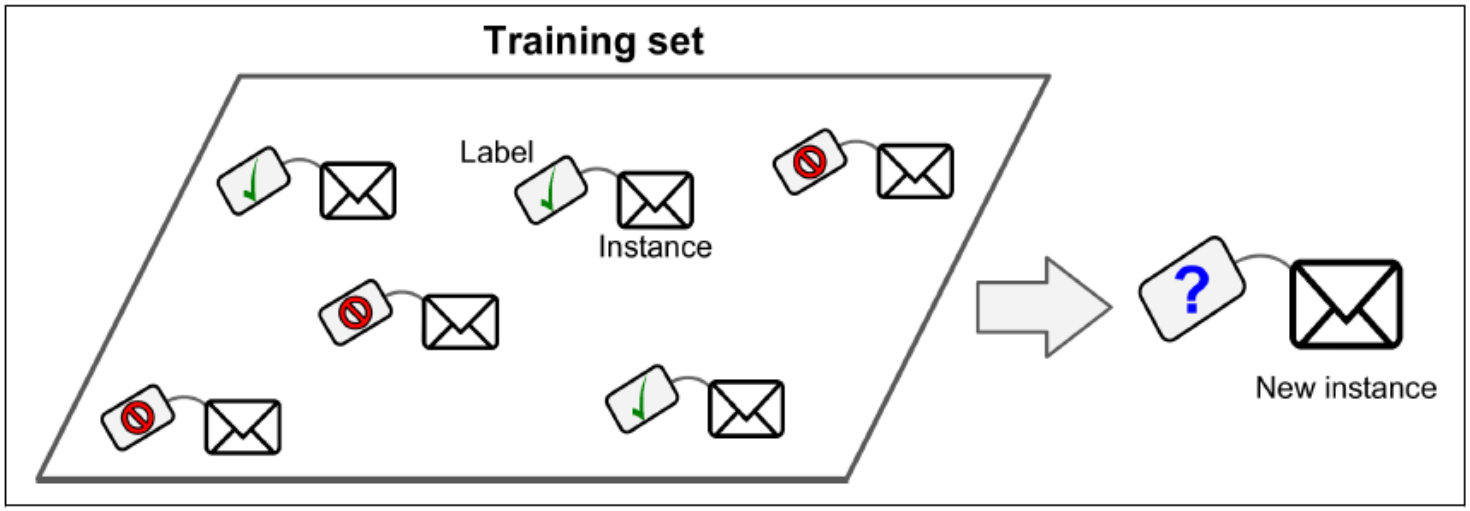
- 지도학습(*Supervised Learning*)
- 비지도학습(*Unsupervised Learning*)
- 준지도 학습(*Semi − supervised Learning*)
- 강화학습(*Reinforcement Learning*)





지도학습(*Supervised Learning*)

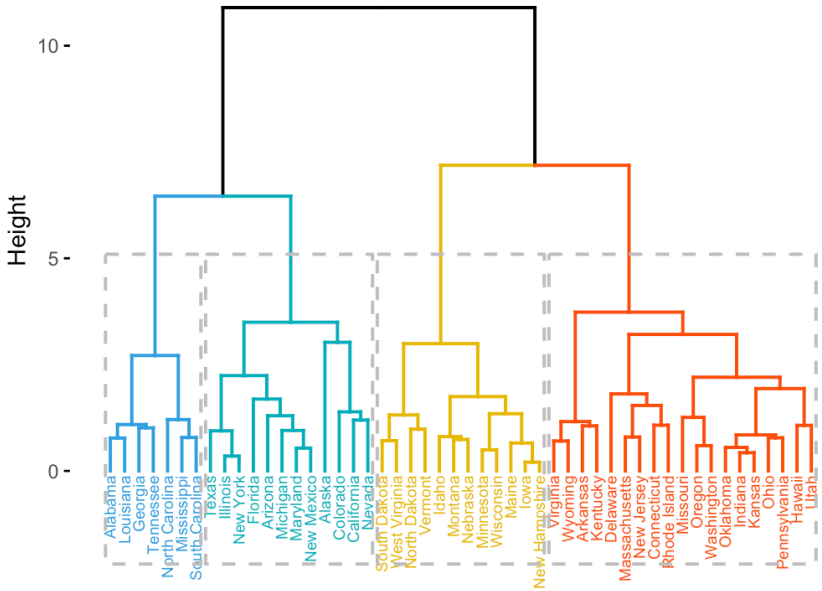
- 모든 훈련 데이터에 레이블(*Label*, 정답) 정보를 갖는다
- 대표적인 알고리즘
  - $k-$ 최근접 이웃($k-nearest\ neighbors$)
  - 선형회귀($linear\ regression$)
  - 로지스틱회귀($logistic\ regression$)
  - 서포트벡터머신($SVM, support\ vector\ machine$)
  - 결정트리($decision\ tree$)와 랜덤 포레스트($random\ forest$)
  - 신경망($neural\ networks$)

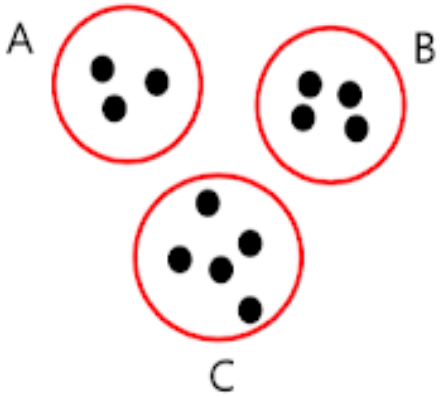


Ref : hands on machine learning 2nd - oreily

비지도 학습(*Unsupervised Learning*)

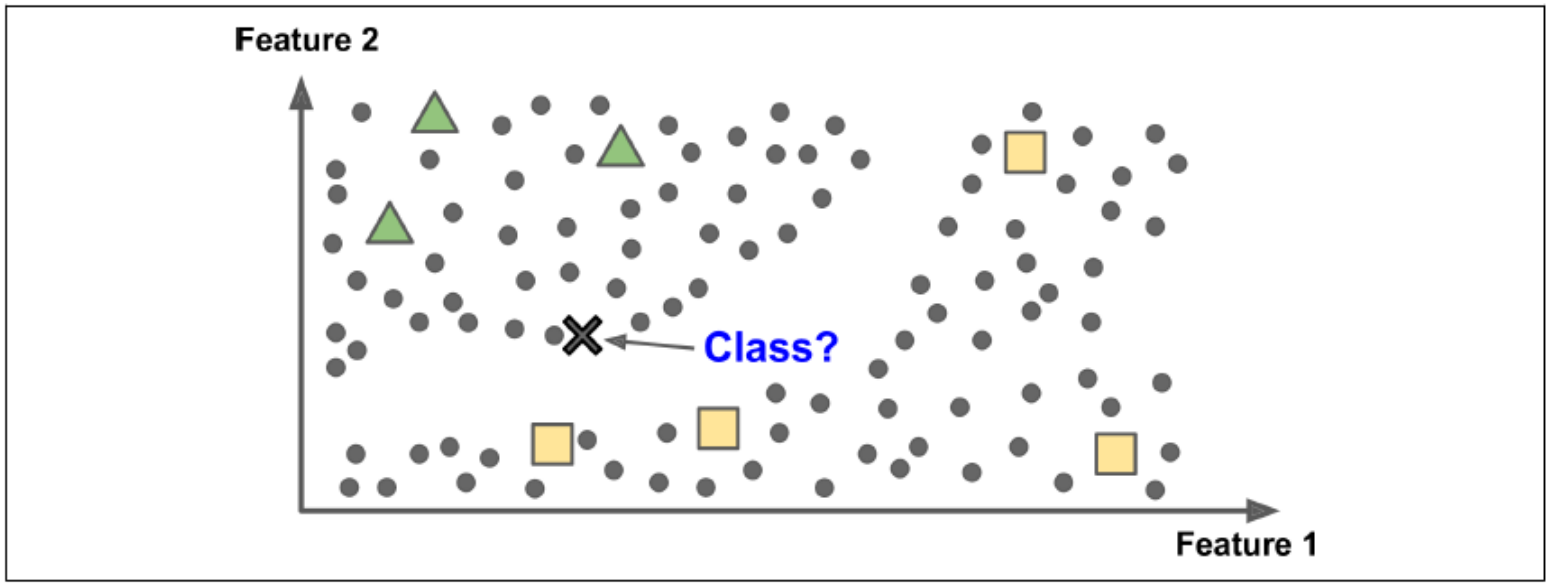
- 모든 훈련 데이터에 레이블 정보가 없다
- 군집(*clustering*)
  - k –평균($k-means$)
  - 병합 군집(*agglomerative clustering*) : 계층군집분석(*hierarchical clustering*)
  - DBSCAN(*Density – based spatial clustering of applications with noise*)
- 이상치탐지(*outlier detection*)
- 차원축소(*dimension reduction*) : 주성분분석(*PCA*)





준지도 학습(*Semi − supervised Learning*)

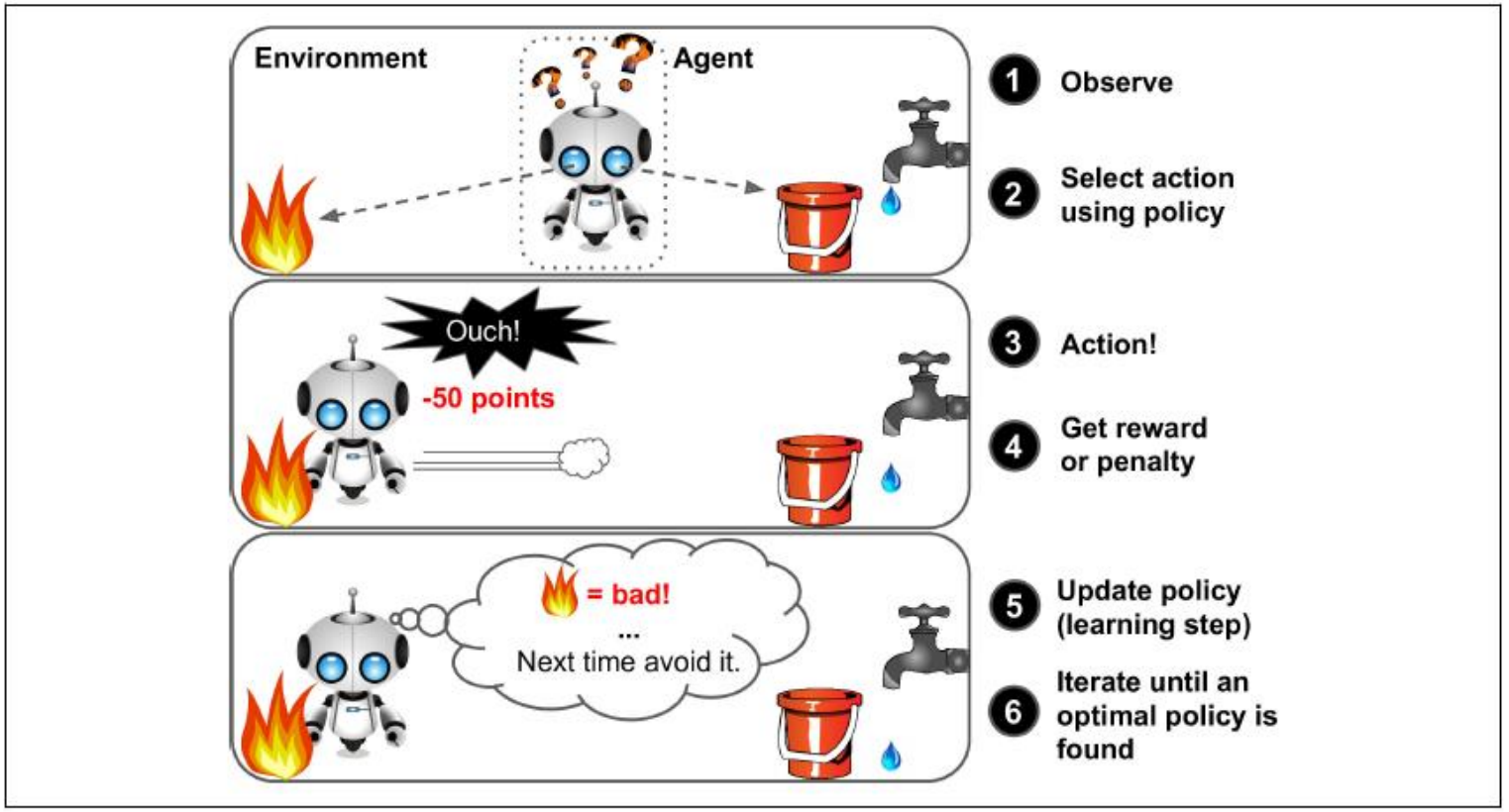
- 많은 데이터에 레이블을 다는 것은 시간과 비용이 많이 소요된다
- 따라서, 일부 데이터에만 레이블이 있다
- 지도학습 + 비지도학습



두개의 클래스(삼각형과 사각형)를 사용한 준비도 학습

강화학습

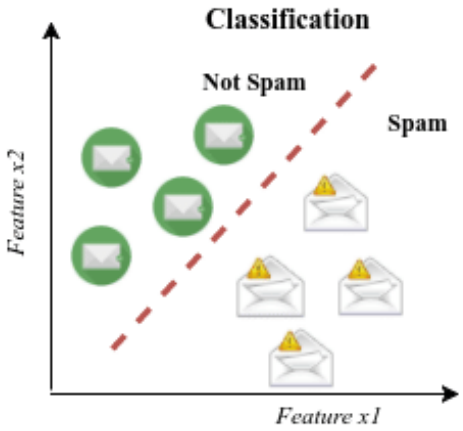
- 행동실행에 따라 보상이나 벌점을 받음 - ③④
- 정책수정(학습) : 최적의 정책을 찾을 때까지 반복 - ⑤⑥

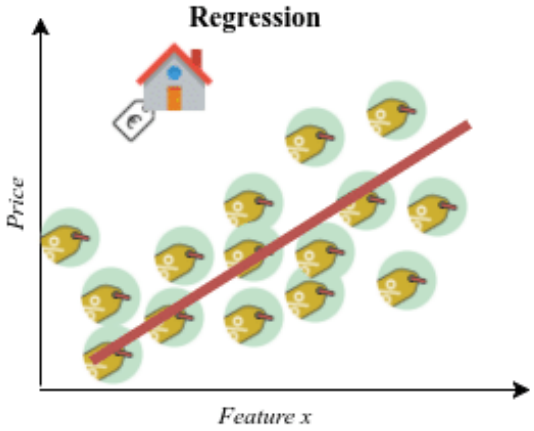


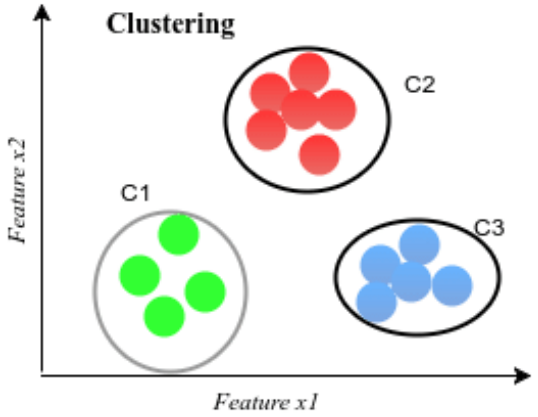
Ref : hands on machine learning 2nd - oreily

군집 vs 분류 vs 회귀

- 군집(Clustering) : 비지도학습
- 분류(Classification) : 지도학습
- 회귀(Regression) : 지도학습(범주)







https://www.researchgate.net/publication/319093376_DEEP_GENERATIVE_MODELS_FOR_SYNTHETIC_RETINAL_IMAGE_GENERATION

Data Set
Learning Types
Fitting/One-Hot Encoding

One-hot encoding

- 다중 분류에서 인코딩 방법으로 출력 값의 형태가 정답 레이블은 $\mathbf{1}(\boldsymbol{true})$, 그 외의 레이블은 $\mathbf{0}(\boldsymbol{false})$인 배열
- 한 개의 요소만 1이고 나머지는 0인 N차원의 벡터로 표현
- 파이썬 : `from sklearn.preprocessing import LabelEncoder`

(예1) [0,0,0,1,0] 4번째 인덱스만 1이고 나머지는 0. 즉, 4번째 인덱스가 정답.

(예2)

Index	Job
1	Police
2	Doctor
3	Student
4	Teacher
5	Driver

One hot encoded data
[1 0 0 0 0]
[0 1 0 0 0]
[0 0 1 0 0]
[0 0 0 1 0]
[0 0 0 0 1]

One-hot encoding 변환 샘플

CIFAR10	*MNIST Zip*	*MNIST Fashion*	*One − Hot*
airplane	0	*T − shirt/top*	[1 0 0 0 0 0 0 0 0 0]
automobile	1	*Trouser*	[0 1 0 0 0 0 0 0 0 0]
bird	2	*Pullover*	[0 0 1 0 0 0 0 0 0 0]
cat	3	*Dress*	[0 0 0 1 0 0 0 0 0 0]
deer	4	*Coat*	[0 0 0 0 1 0 0 0 0 0]
dog	5	*Sandal*	[0 0 0 0 0 1 0 0 0 0]
frog	6	*Shirt*	[0 0 0 0 0 0 1 0 0 0]
horse	7	*Sneaker*	[0 0 0 0 0 0 0 1 0 0]
ship	8	*Bag*	[0 0 0 0 0 0 0 0 1 0]
truck	9	*Ankle boot*	[0 0 0 0 0 0 0 0 0 1]