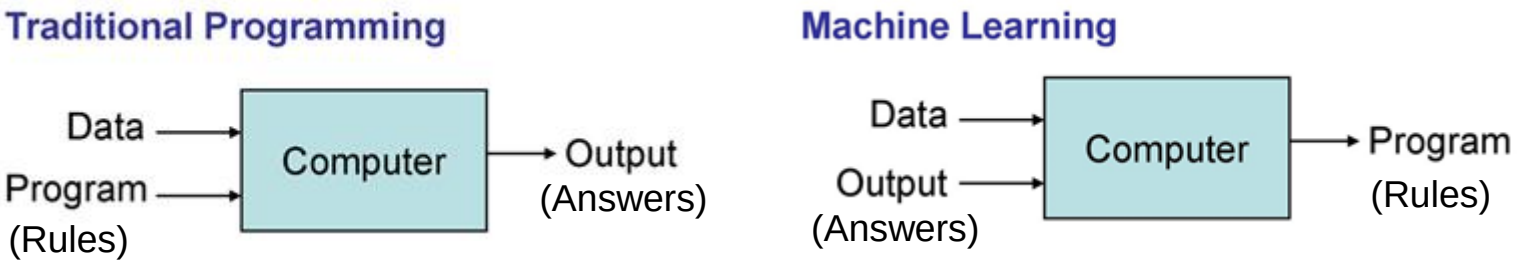
Introduction to Machine Learning

머신러닝(Machine Learning)이란 ?

- Limitations of explicit programming
  - Spam filter: many rules
  - Automatic driving: too many rules

- Machine learning
  - "Field of study that gives computers the ability to learn without being explicitly programmed", Arthur Samuel (1959)



왜 머신러닝을 사용하는가?

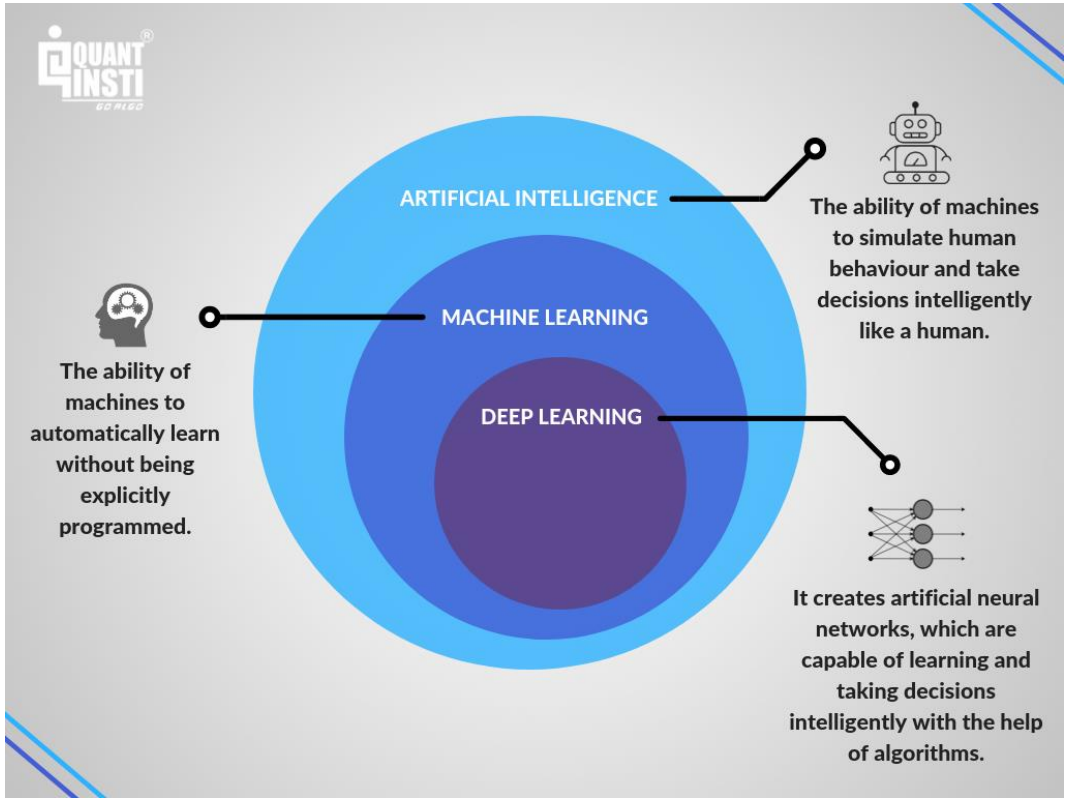
- 기존 솔루션으로는 많은 hand-tuning과 규칙이 필요한 문제의 경우 머신러닝을 이용하면 간단한 코드로 더 나은 성능을 얻을 수 있다.
- 전통적인 방법으로는 해결이 불가능한 복잡한 문제에 있어서 머신러닝이 새로운 해결책이 될 수 있다.
- 머신러닝은 변화가 심한 환경에서 새롭게 생성되는 데이터에 적응력이 뛰어나다.
- 머신러닝은 복잡한 문제와 대용량 데이터로부터 통찰을 얻을 수 있게 해준다.

Source: "Hands-On Machine Learning with Scikit-Learn and TensorFlow" , urélien Géron

혼동되어 사용되는 용어

- Data Mining
  - Data analysis processes that apply ML techniques to solving real world problems

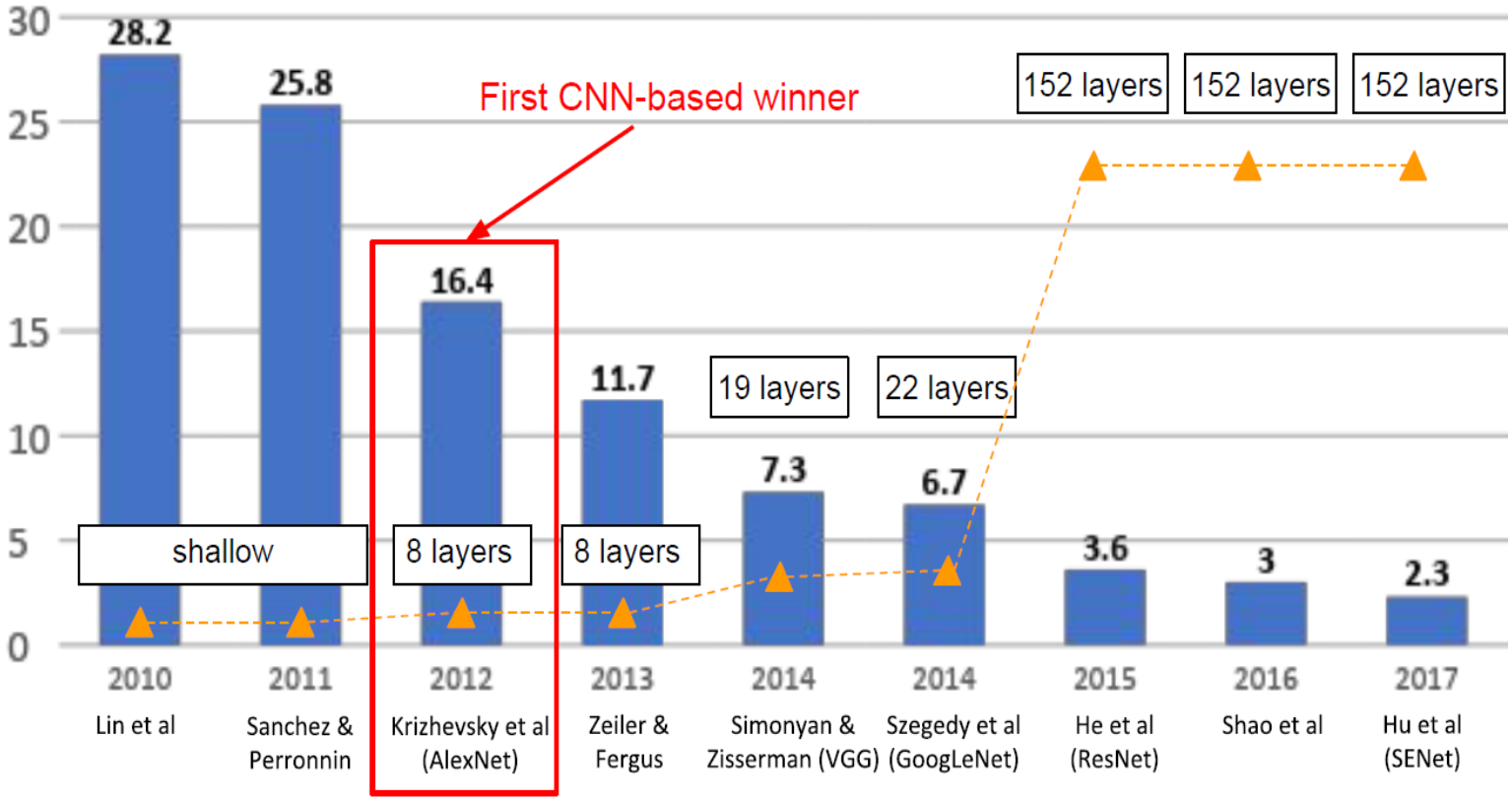
- AI & Deep Learning



인공지능 3대 분야

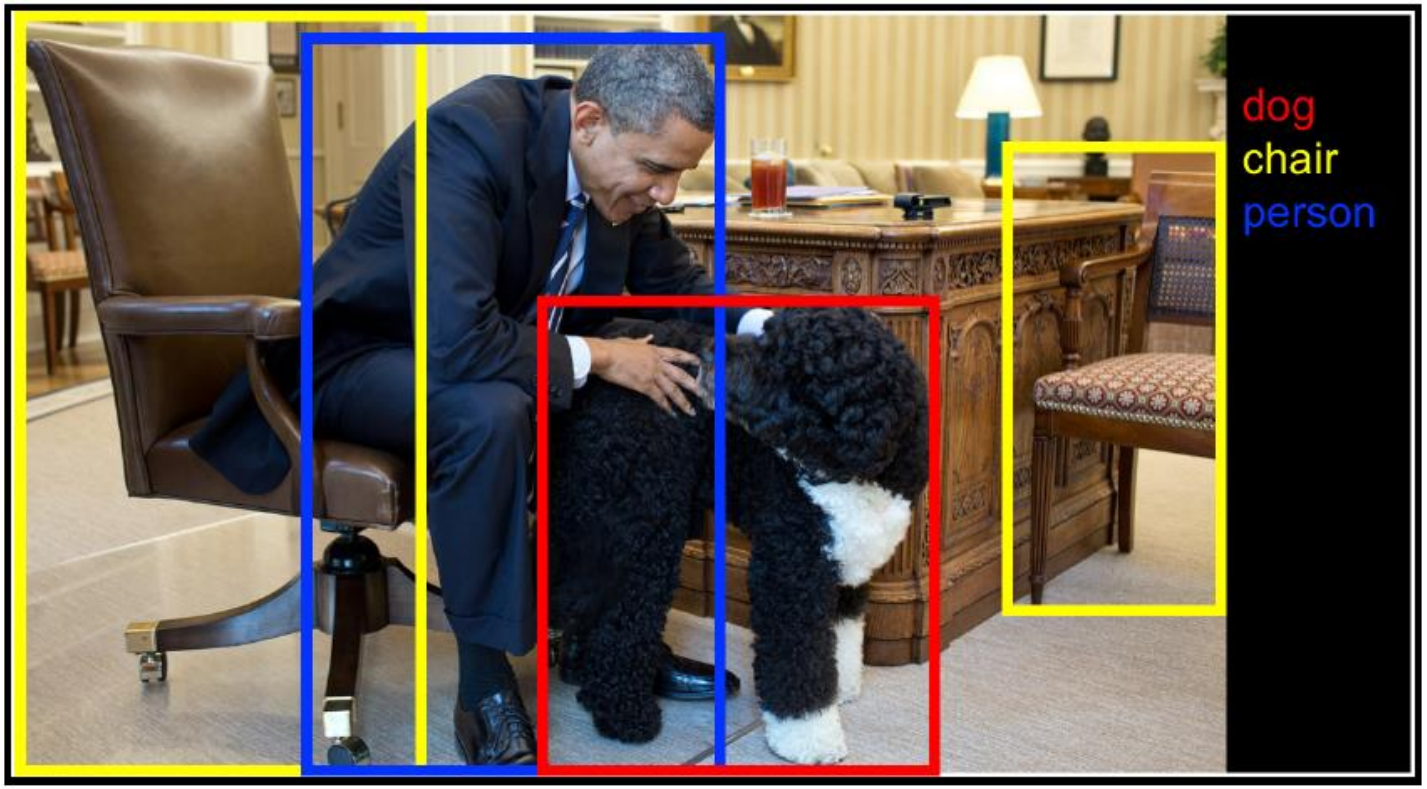
- Computer Vision – Image Recognition
- Automatic Speech Recognition (ASR)
- Natural Language Processing – Machine Translation

- ILSVRC ImageNet challenge
  - The images are large (256 pixels high) and there are 1,000 classes, some of which are really subtle (try distinguishing 120 dog breeds)
  - The top-5 error rate for image classification fell from over 26% to barely over 3% in just five years
    (human error rate: 5.1%)



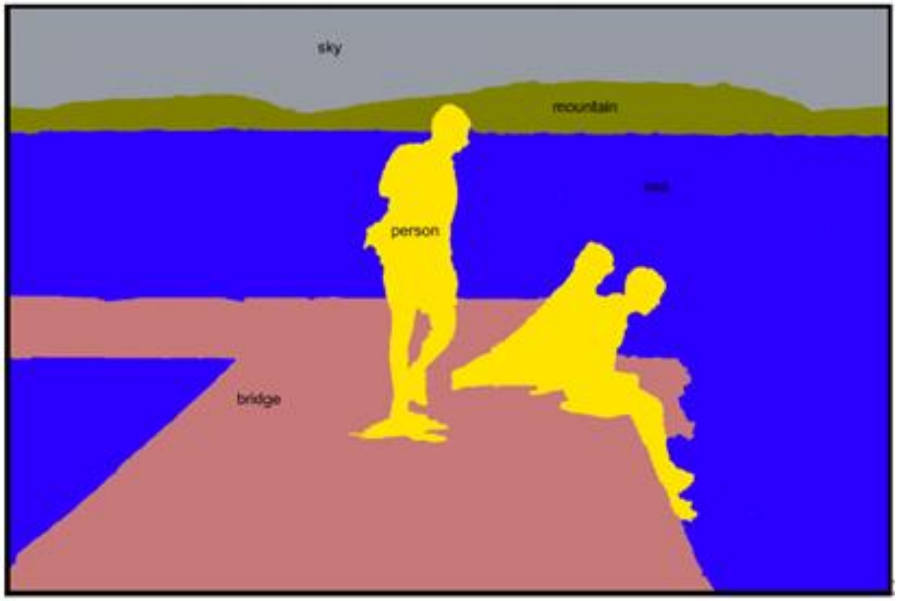
source: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture09.pdf

Deep Learning for Computer Vision - Object Detection



Source: "Deep Learning for Computer Vision", Shanmugamani

Deep Learning for Computer Vision - Semantic Segmentation

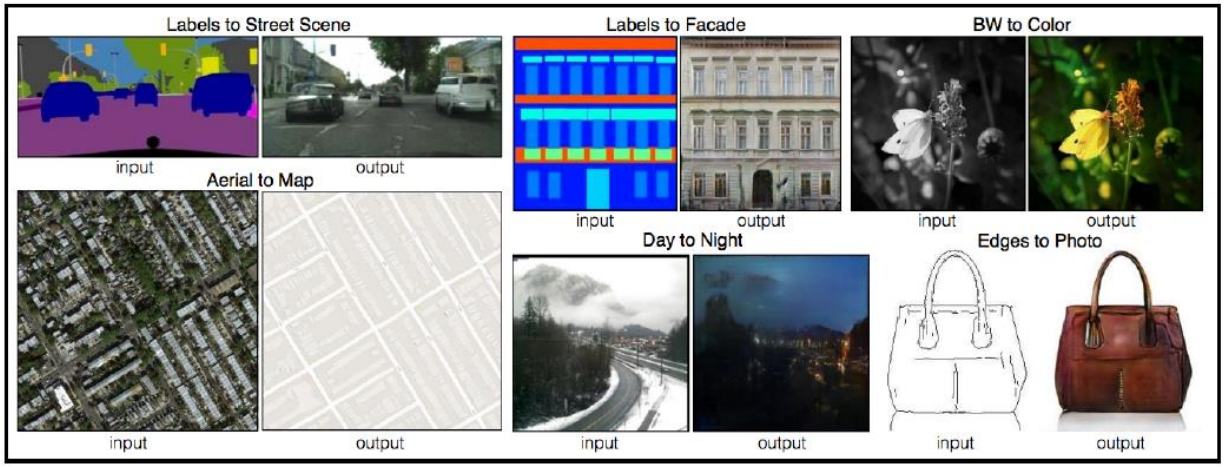


Source: "Deep Learning for Computer Vision", Shanmugamani

Deep Learning for Computer Vision - Image Captioning

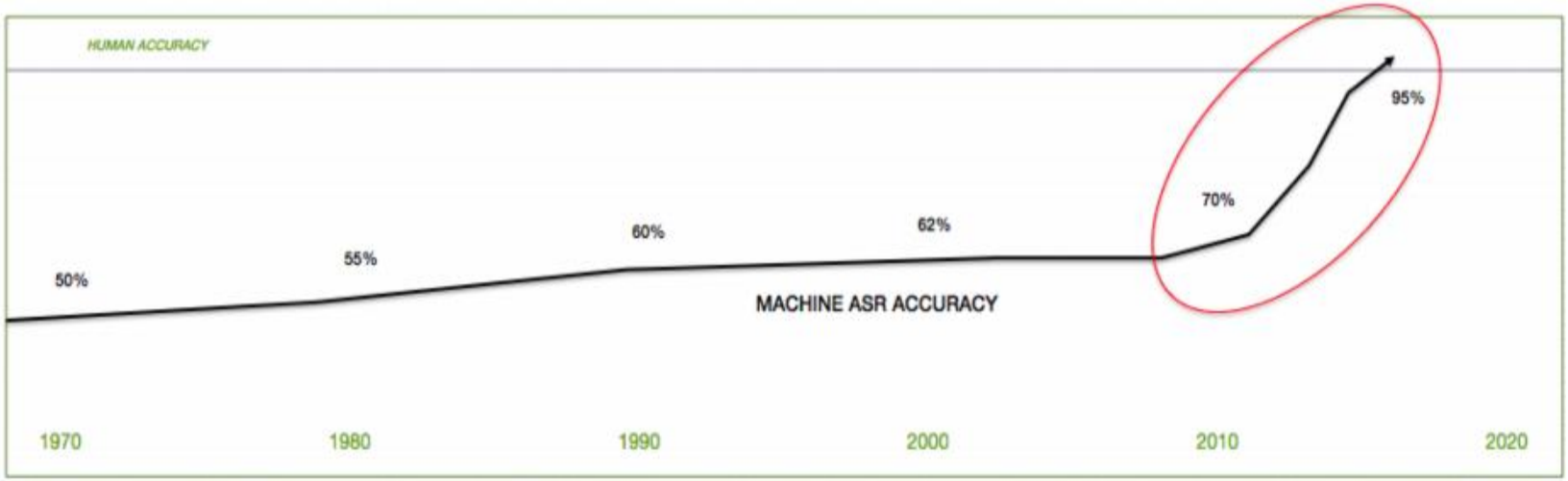
Source: "Deep Learning for Computer Vision", Shanmugamani

Deep Learning for Computer Vision - Image Generation



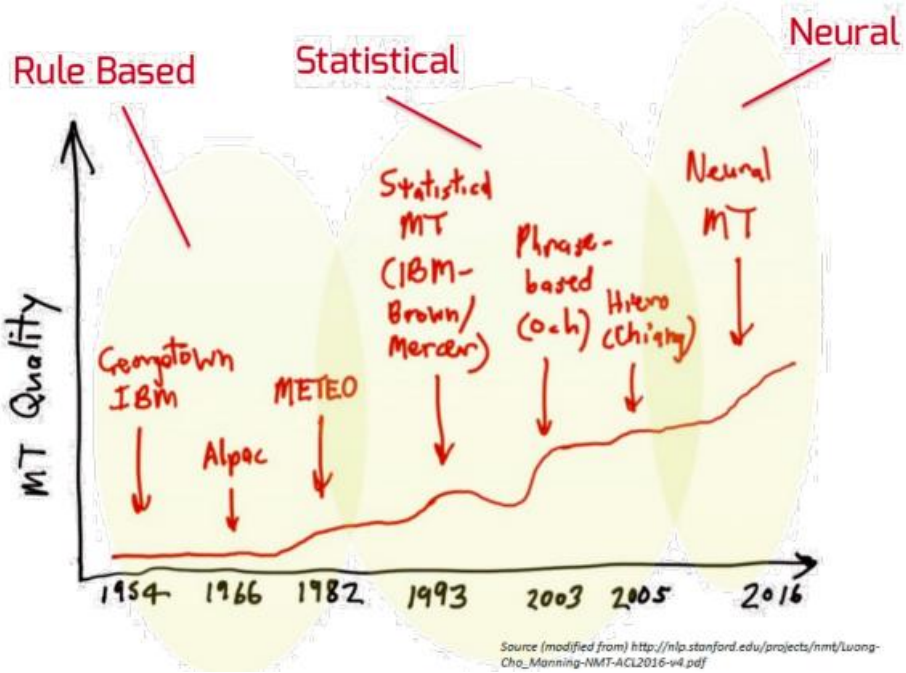
Source: "Deep Learning for Computer Vision", Shanmugamani

Speech Recognition



- 2012년부터 딥러닝을 활용하여 큰 발전
- 오히려 이 분야에서는 vision분야에 비해서 딥러닝 기술을 활용하여 상용화에까지 성공한 더욱 인상적인 사례

Neural Machine Translation

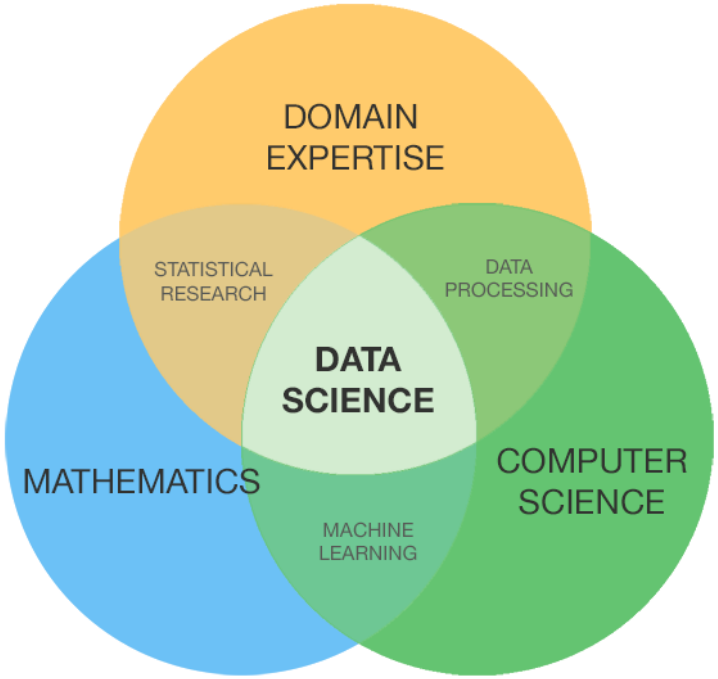


- 2014년 Sequence-to-sequence(seq2seq)가 소개됨
- 기계번역은 가장 늦게 혁명이 이루어졌지만, 가장 먼저 딥러닝만을 사용해 상용화가 된 분야
- 현재의 상용 기계번역 시스템은 모두 딥러닝으로 대체

Data Science & Data Scientist

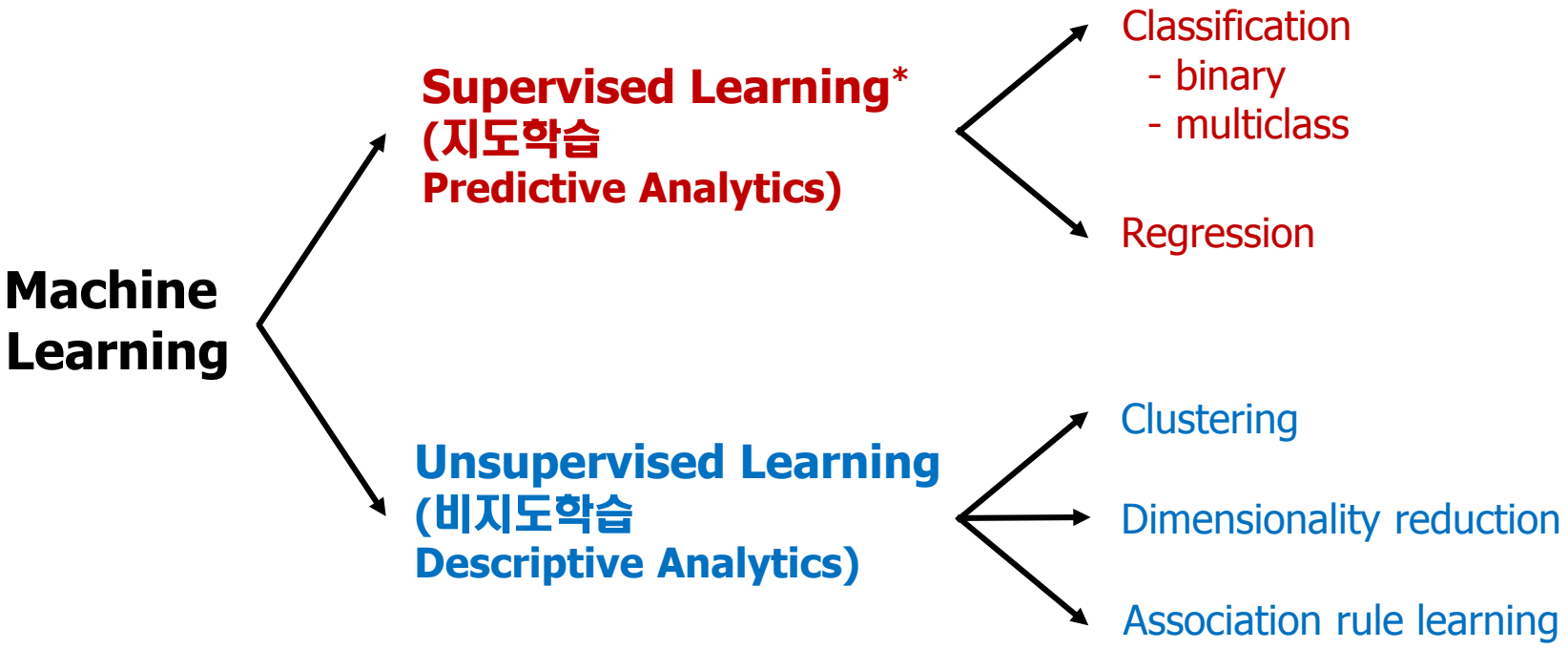
- Data Science
  - 데이터의 수집, 저장, 처리, 그리고 분석과 활용을 연구하는 다학제적 학문

cf. 통계학 – 통계의 관찰 및 그 분석 방법을 연구하는 학문



- Data Scientist
  - someone who is better at statistics than any SW engineer & better at SW engineering than any statistician

머신러닝의 유형



☞ *Some researchers classify Reinforcement Learning into a type of machine learning*

* *전체 머신러닝 사례의 **95%** 이상을 차지*

Y X

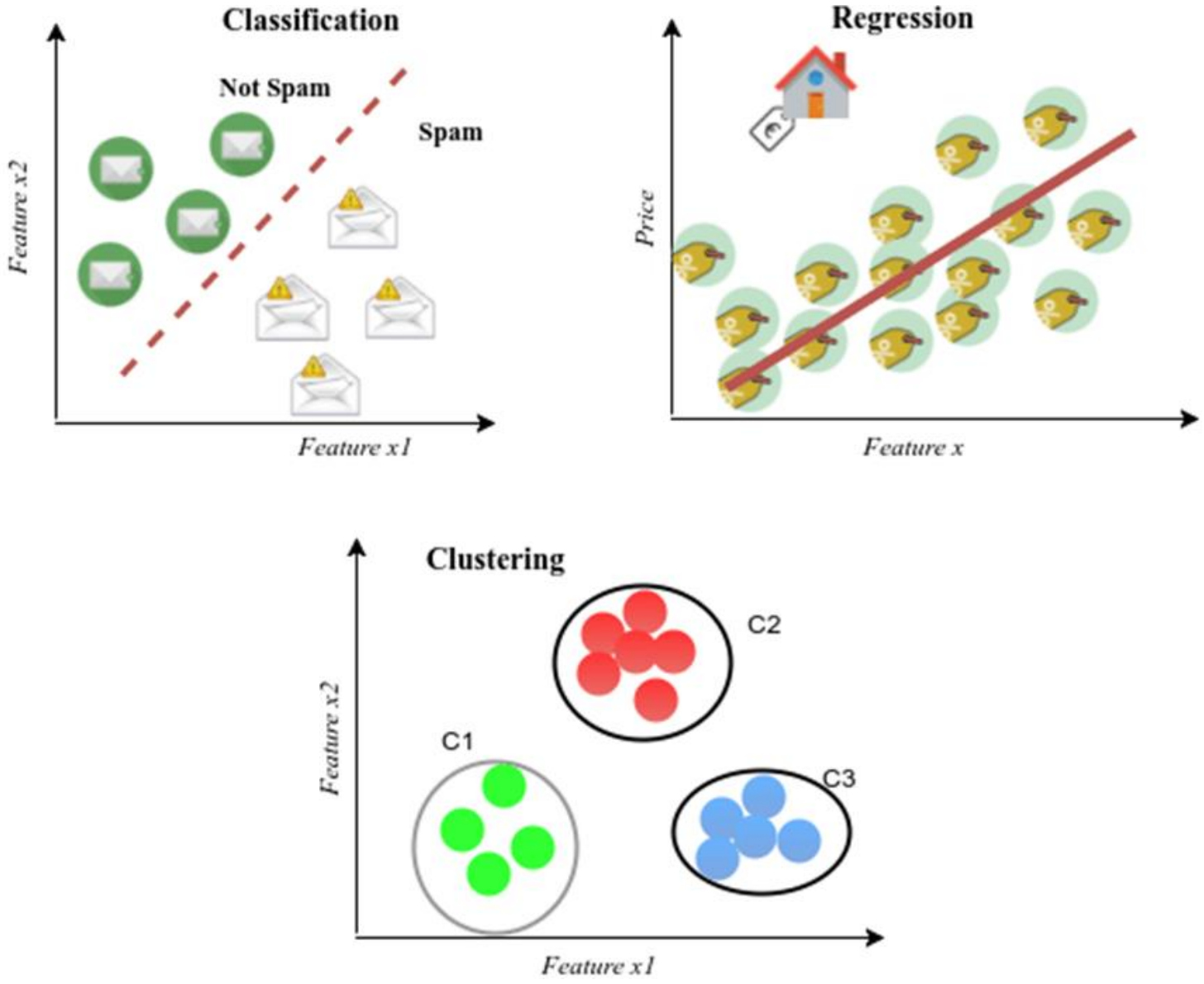
PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhe	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
19	0	3	Vander Planke, Mrs. Julius (Emelia Mari	female	31	1	0	345763	18		S
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C
21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26		S

타이타닉 호 침몰 당시의 승객 명단 데이터

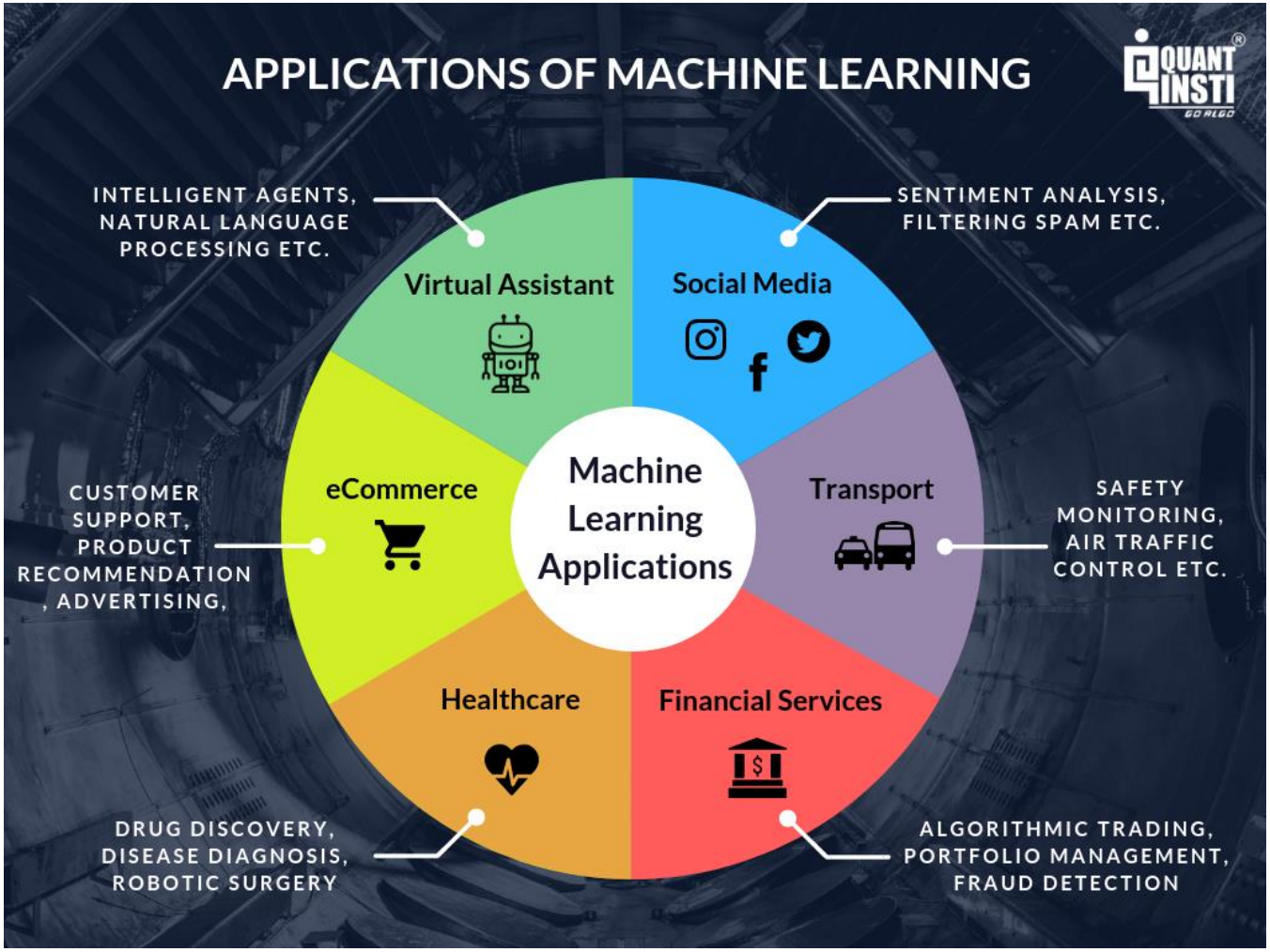
- Survived: 생존 여부 => 0 = No, 1 = Yes
- pclass: 티켓 등급 => 1 = 1st, 2 = 2nd, 3 = 3rd
- Sex: 성별
- Age: 나이
- Sibsp: 함께 탑승한 형제자매, 배우자의 수
- Parch: 함께 탑승한 부모, 자식의 수
- Ticket: 티켓 번호
- Fare: 운임
- Cabin: 객실 번호
- Embarked: 탑승 항구 => C = Cherbourg, Q = Queenstown, S = Southampton

Kaggle Kernels

https://www.kaggle.com/c/titanic/kernels?sortBy=voteCount&group=everyone&pageSize=20&competitionId=3136


Classification
Not Spam
Spam
Feature x2
Feature x1
Regression
Price
Feature x
Clustering
C1
C2
C3
Feature x2
Feature x1

머신러닝 응용분야


APPLICATIONS OF MACHINE LEARNING
QUANT INSTI
INTELLIGENT AGENTS, NATURAL LANGUAGE PROCESSING ETC.
SENTIMENT ANALYSIS, FILTERING SPAM ETC.
Virtual Assistant
Social Media
Machine Learning Applications
eCommerce
Transport
CUSTOMER SUPPORT, PRODUCT RECOMMENDATION , ADVERTISING,
SAFETY MONITORING, AIR TRAFFIC CONTROL ETC.
Healthcare
Financial Services
DRUG DISCOVERY, DISEASE DIAGNOSIS, ROBOTIC SURGERY
ALGORITHMIC TRADING, PORTFOLIO MANAGEMENT, FRAUD DETECTION

지도학습이 발견한 이상하고 놀라운 사실

- 배너광고를 본 사람 중 61%는 그와 관련된 검색을 할 가능성이 높다.
- 맥 사용자들은 상대적으로 더 비싼 호텔을 예약한다.
- 금융 및 주식 사이트는 오후1시 직후가 접속 피크다.
- 이메일 주소는 그 사람의 충성도를 암시한다.
- 신용카드를 술집에서 자주 사용한다면 결제대금을 연체할 위험이 높다.
- 신용등급이 낮을수록 자동차 사고를 낼 확률이 높다.
- 이미 여러 개의 계좌를 개설한 고객들에게 우편물을 보내면 오히려 그들이 더 많은 계좌를 개설할 가능성을 낮춘다.
- 약정기간이 끝났는지 온라인으로 알아보는 고객은 경쟁업체로 넘어갈 가능성이 높다.
- 직무순환을 많이 한 직원일수록 회사에 더 오래 다니는 경향이 있다.
- 스테이플러는 일자리를 의미한다.

Source: "예측 분석이다 (빅데이터의 다음 단계는)", 에릭 시겔

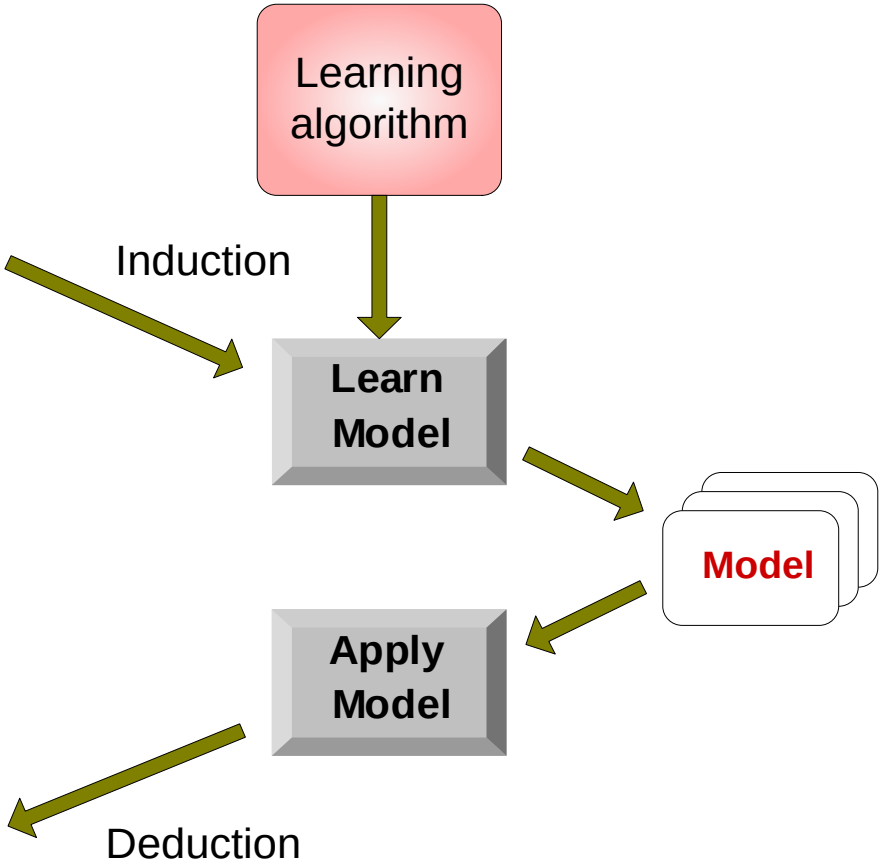
Naming Conventions

- **Features = predictor variables = independent variables**
- **Class = target variable = dependent variable**

분류(Classification)분석의 개념

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

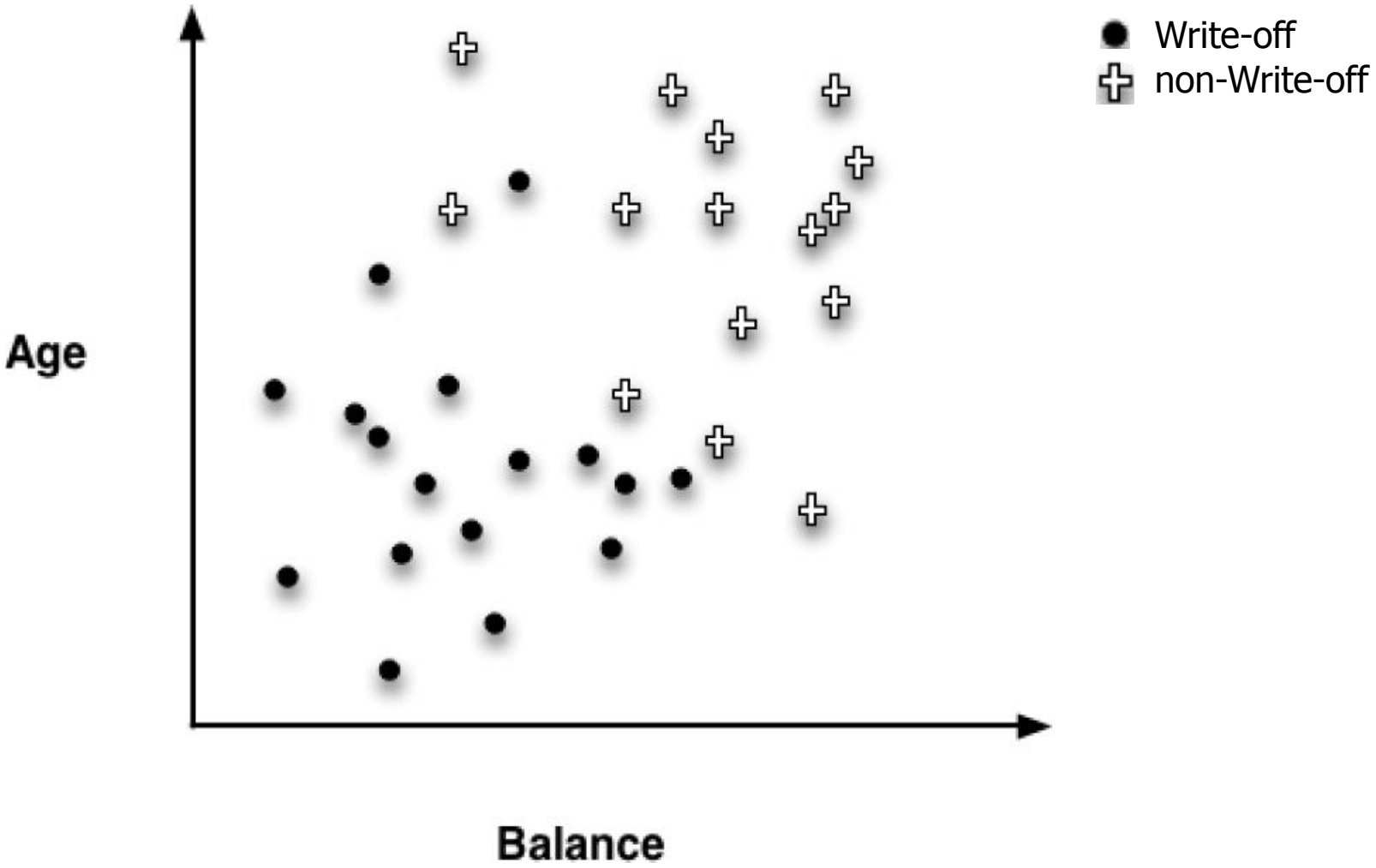


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

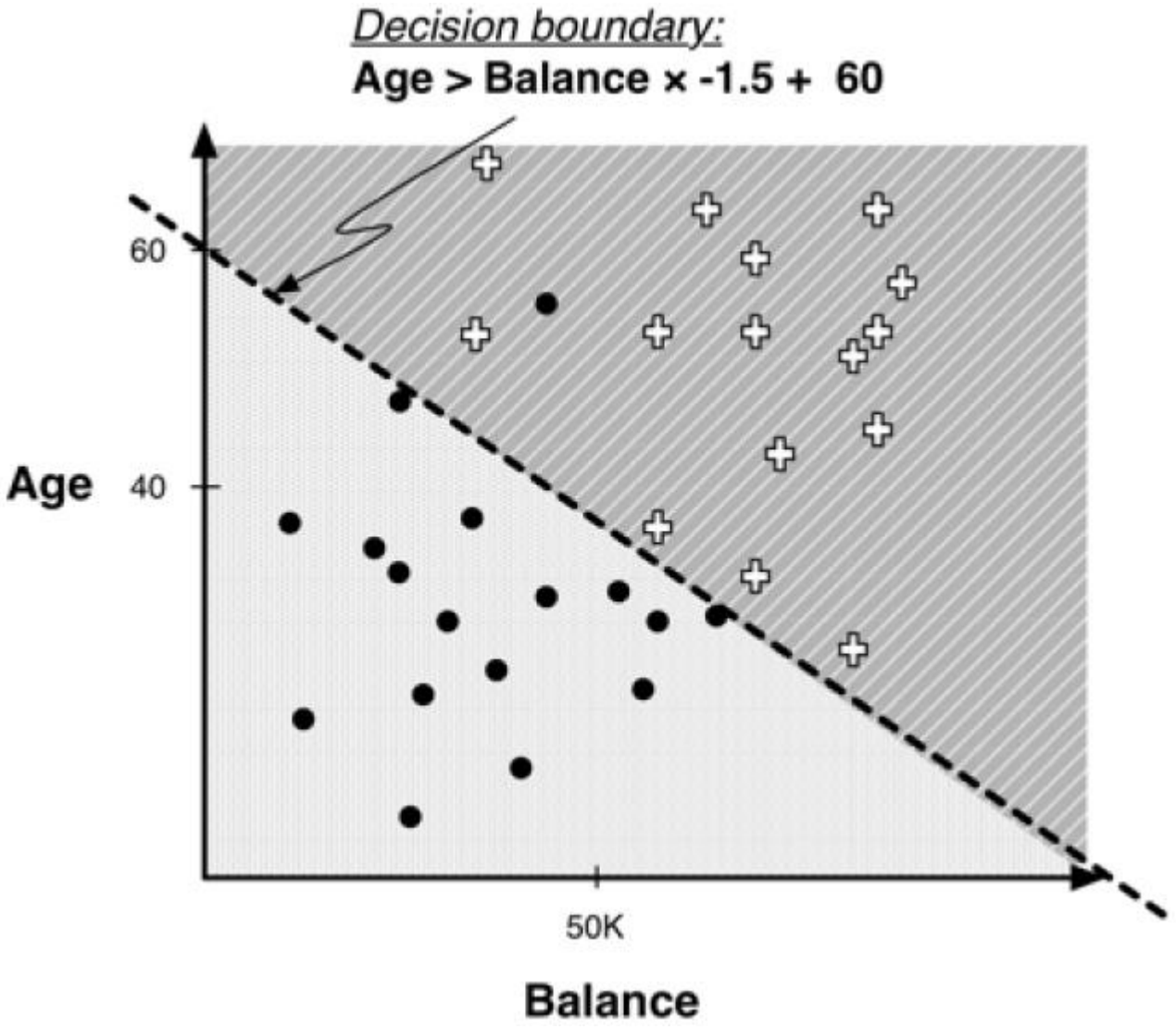
Source: Introduction to Data Mining by Tan, Steinbach & Kumar

분류분석 기법 (시각적 해석)



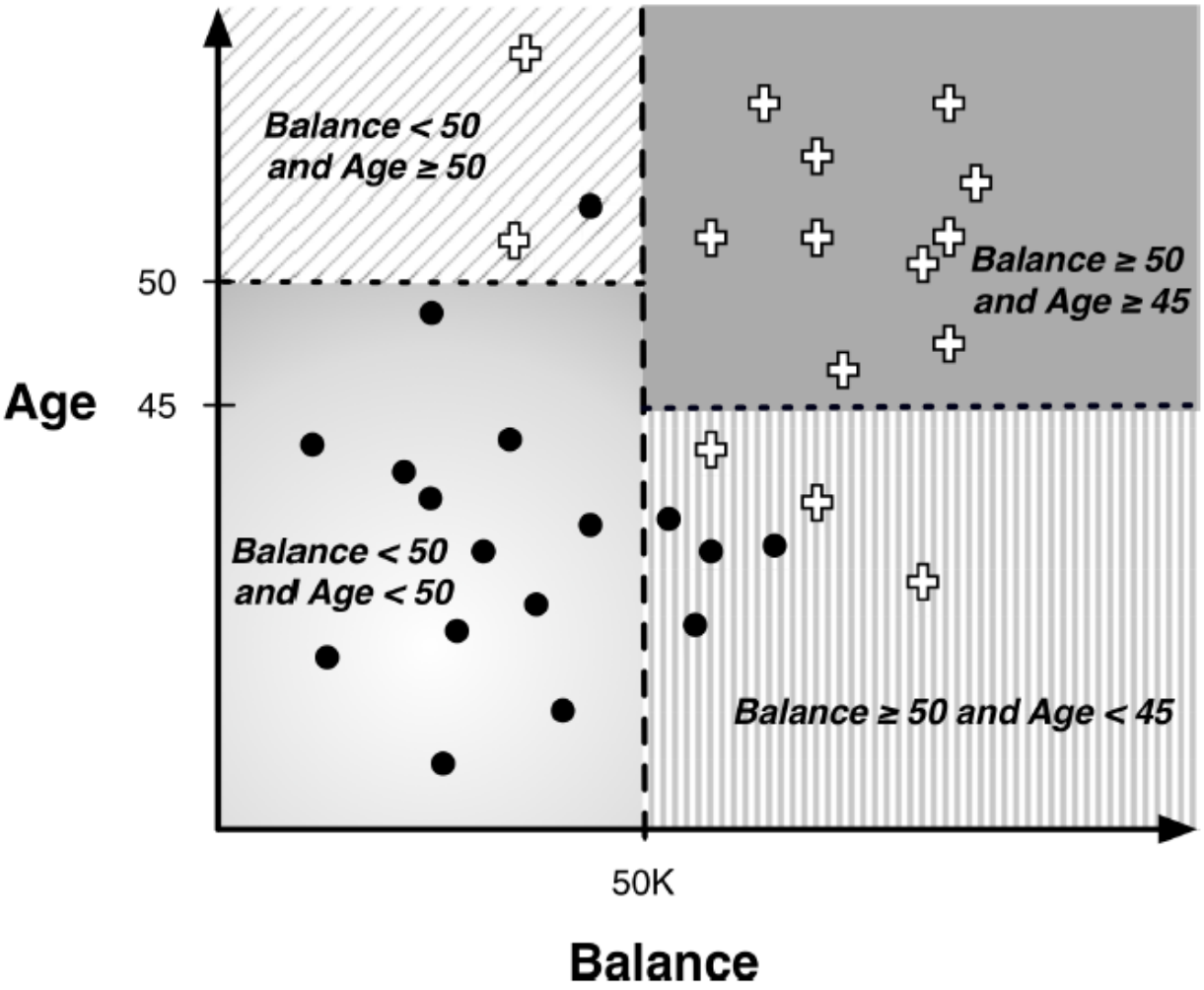
Source: Data Science for Business by Provost & Fawcett

분류분석 기법
– Logistic Regression (or SVM)



Source: Data Science for Business by Provost & Fawcett

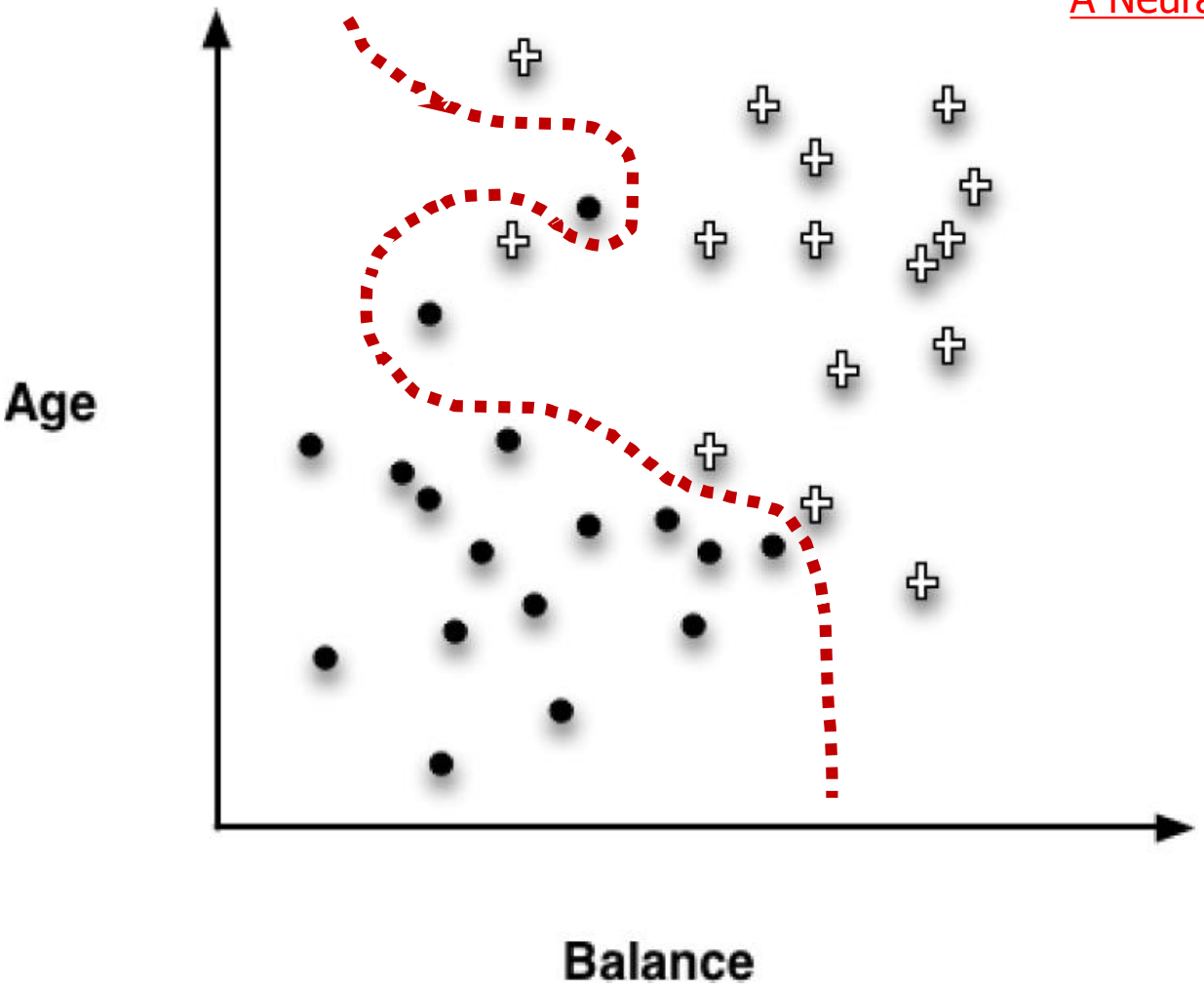
분류분석 기법
– Decision Tree based Methods



Source: Data Science for Business by Provost & Fawcett

분류분석 기법
– Neural Networks

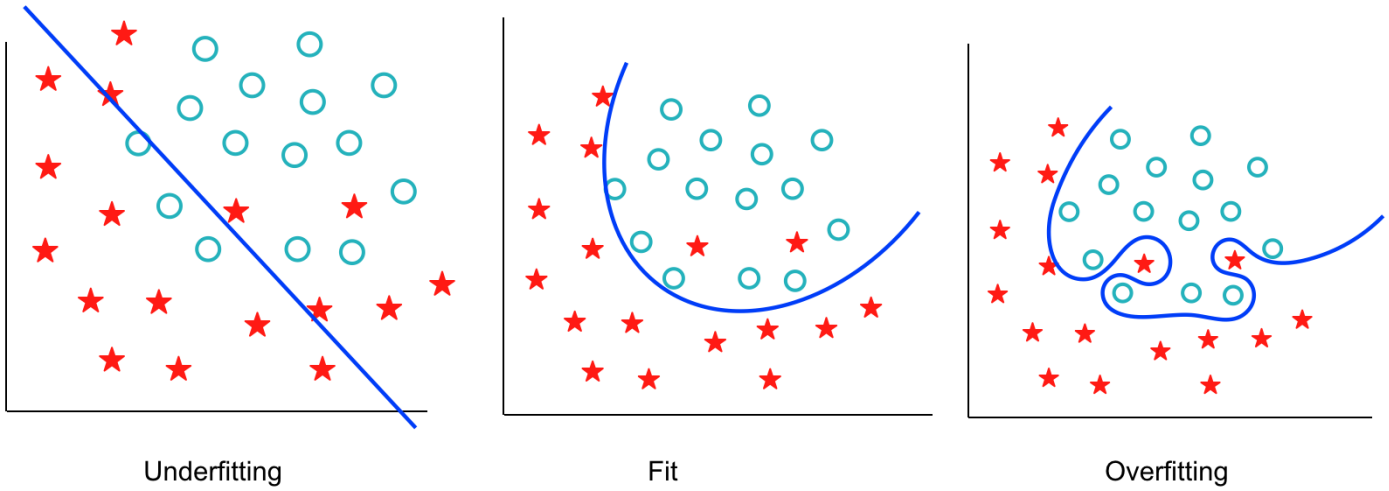
A Neural Network Playground



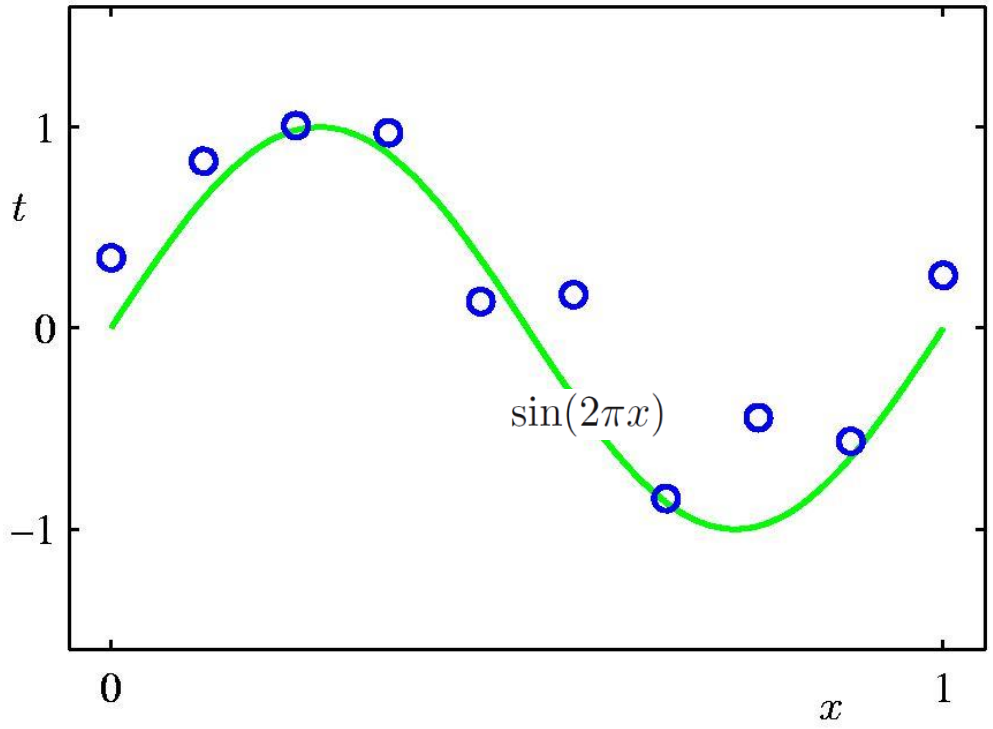
Source: Data Science for Business by Provost & Fawcett

Overfitting vs. Underfitting

- 과대적합(overfitting)
  - 모델이 데이터에 필요이상으로 적합한 모델
  - 데이터 내에 존재하는 규칙 뿐만 아니라 불완전한 레코드도 학습
- 과소적합(underfitting)
  - 모델이 데이터에 제대로 적합하지 못한 모델
  - 데이터 내에 존재하는 규칙도 제대로 학습하지 못함



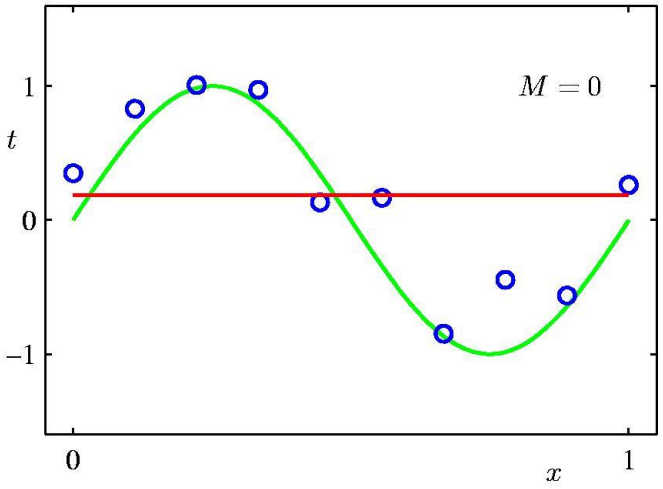
An Overfitting Example: Polynomial Curve Fitting



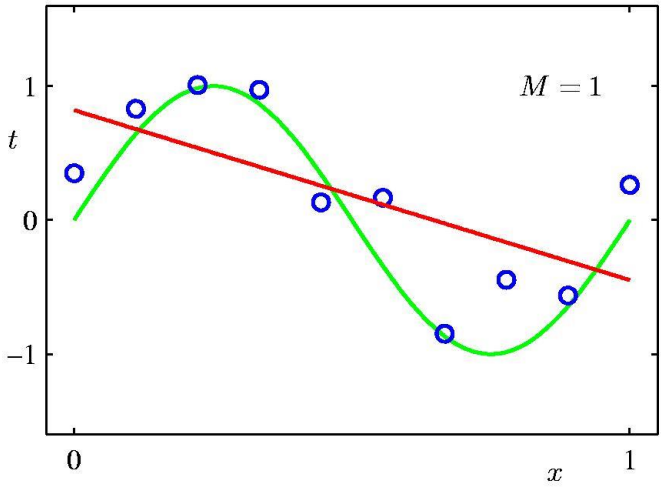
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

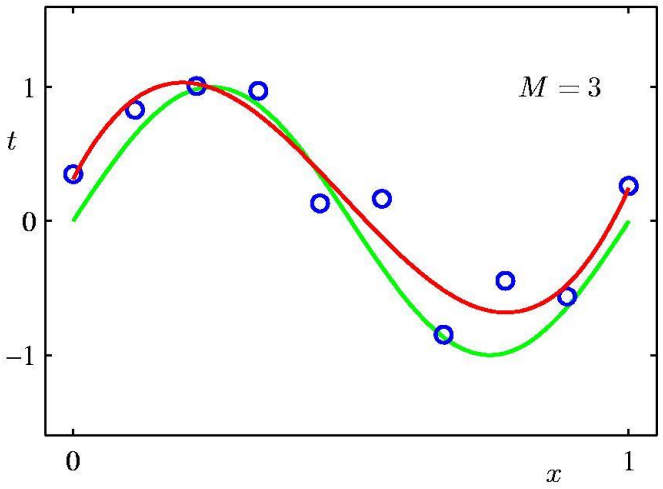
Source: "Pattern Recognition and Machine Learning" , Christopher M. Bishop

Underfit

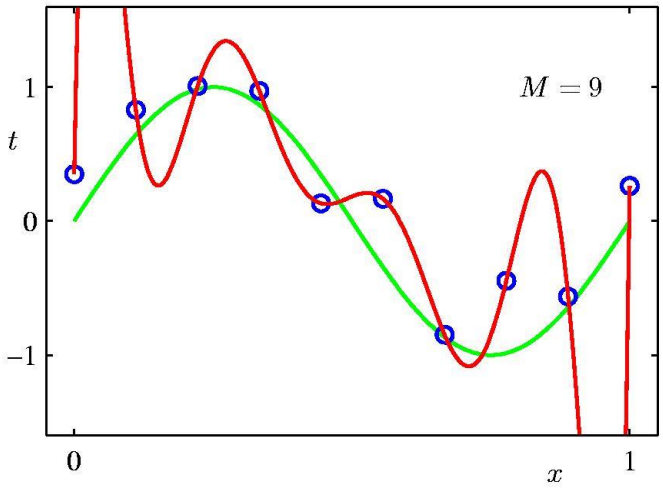


Underfit





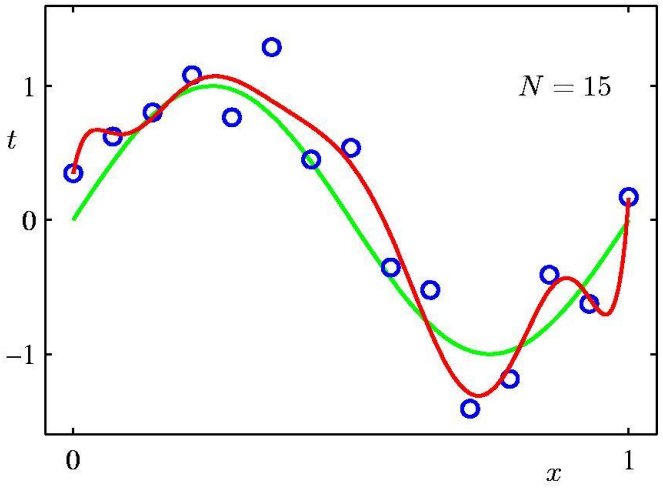
Overfit



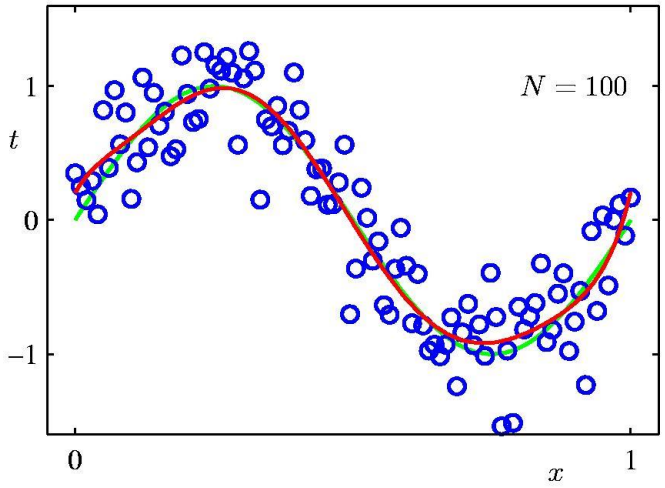
Source: "Pattern Recognition and Machine Learning" , Christopher M. Bishop

Data Set Size:

$N = 15$

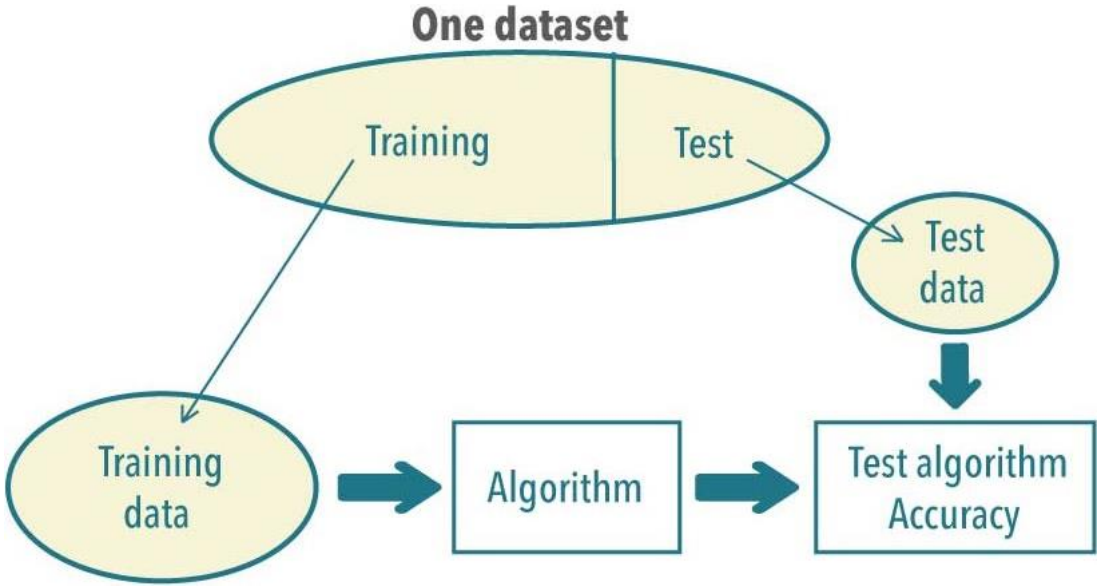


$N = 100$



Source: "Pattern Recognition and Machine Learning" , Christopher M. Bishop

데이터 분할(Data Partitioning)

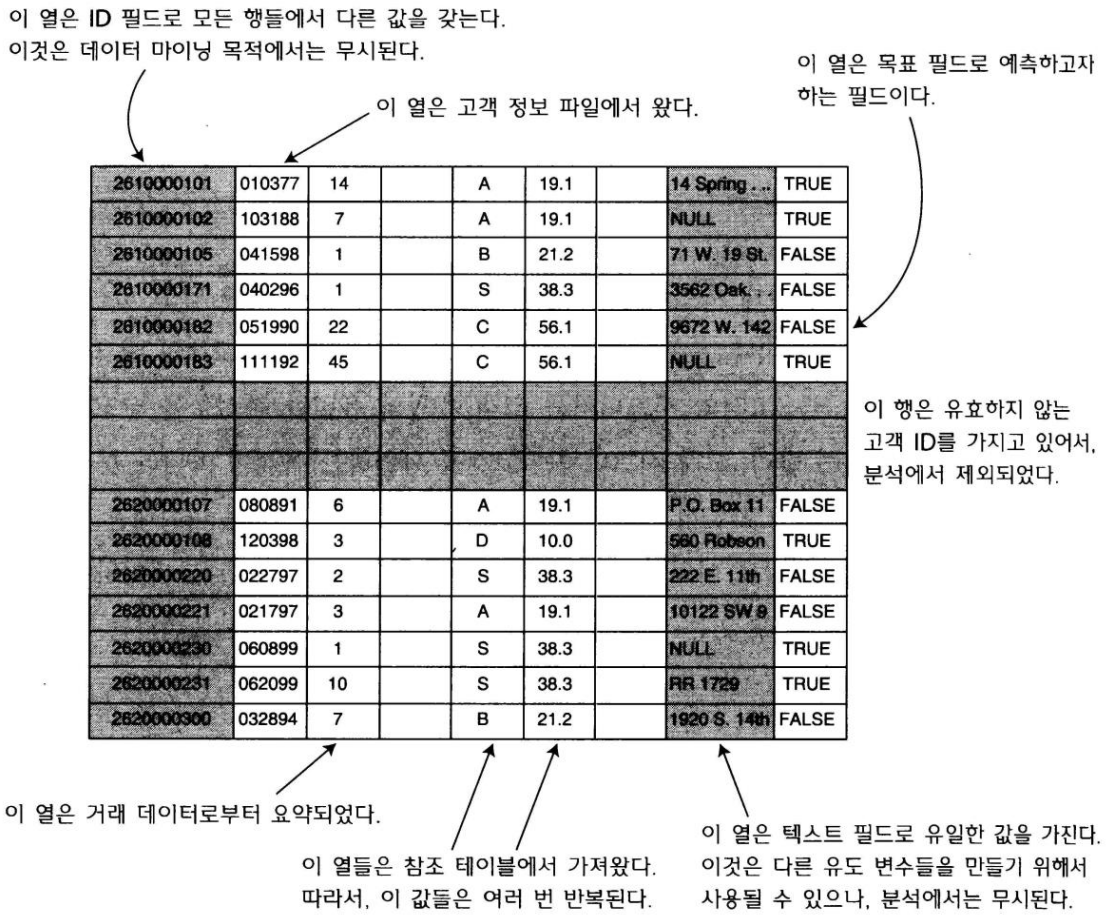


- 모형의 Overfitting 여부(일반화 가능성)를 검증하기 위해 데이터를 학습용과 평가용으로 분리
- 학습데이터 (Training data) → 모형적합(Model Fitting)
- 평가데이터 (Test data) → 모형평가(Model Evaluation)
- 60%(학습) – 40%(평가) 분할, 75% – 25% 분할을 주로 사용

분류분석 문제 예시 (금융분야)

- 특정 상품(Ex: 퇴직연금)에 가입할 가능성이 높은 고객은 누구인가?
- 추가 가입을 유도하기 위해 대상고객에게 어떤 상품을 추천해야 하는가?
- 향후 6개월 안에 이탈(Ex: 계약 만기 전에 실효 또는 중도 해약)할 가능성이 높은 고객들은 누구이며 그들의 특징은?
- 고객별 LTV(Life Time Value)를 계산해 주는 모델은?
  - 예: LTV = 현재 고객의 기여가치 + 추가구매확률에 의한 기여가치 – 이탈확률에 의한 손실 가치
- 담보대출 고객 중에서 누가 향후 90일 내에 조기 상환할 것인가?
- 각 고객별로 클릭할 가능성이 가장 높은 광고는 어떤 것인가?
- 고객별로 어떤 할인쿠폰이 다음달에 사용될 것인가?
- 추후 VIP의 고객이 될 가능성이 높은 고객들은?
- 누가 이직할 것인가?

머신러닝에서 요구되는 데이터구조



2610000101	010377	14		A	19.1		14 Spring . . .	TRUE
2610000102	103188	7		A	19.1		NULL	TRUE
2610000105	041598	1		B	21.2		71 W. 19 St.	FALSE
2610000171	040296	1		S	38.3		3562 Oak. . .	FALSE
2610000182	051990	22		C	56.1		9672 W. 142	FALSE
2610000183	111192	45		C	56.1		NULL	TRUE
2620000107	080891	6		A	19.1		P.O. Box 11	FALSE
2620000108	120398	3		D	10.0		560 Robson	TRUE
2620000220	022797	2		S	38.3		222 E. 11th	FALSE
2620000221	021797	3		A	19.1		10122 SW 9	FALSE
2620000230	060899	1		S	38.3		NULL	TRUE
2620000231	062099	10		S	38.3		RR 1729	TRUE
2620000300	032894	7		B	21.2		1920 S. 14th	FALSE

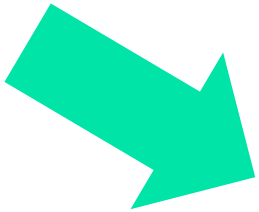
- ✓ 모든 데이터가 하나의 테이블에 존재해야 한다.
- ✓ 각 행은 기업과 관련 있는 한 개체(Ex: 고객)에 대응해야 한다.
- ✓ 하나의 값을 갖는 필드는 무시되어야 한다.
- ✓ 대부분이 한 값을 갖는 필드도 가급적 무시되어야 한다.
- ✓ 각 행마다 다른 값들을 가지는 필드는 무시되어야 한다.
- ✓ 목표필드와 지나치게 높은 상관관계를 갖는 필드는 제거되어야 한다.

Source: "경영을 위한 데이터마이닝", Michael J. A. Berry, Gordon S. Linoff

- A Clickstream Analysis Case

	CUS_ID	TIME_ID	SITE	SITE_CNT	ST_TIME	SITE_NM	BACT_NM	MACT_NM	ACT_NM
1	1	2012070905	search.naver.com	3	794	네이버 검색	인터넷/컴퓨터	검색	포털검색
2	1	2012072507	plus.google.com	1	1	구글 Plus	커뮤니티	블로그/SNS	SNS
3	1	2012081116	joongang.joinsmsn.com	2	5	중앙일보	뉴스/미디어	일간지	종합일간지
4	1	2012090304	news.naver.com	5	504	네이버 뉴스	뉴스/미디어	인터넷신문	포털뉴스
5	1	2012090506	news.nate.com	1	0	네이트 뉴스	뉴스/미디어	인터넷신문	포털뉴스
6	1	2012091004	plus.google.com	2	66	구글 Plus	커뮤니티	블로그/SNS	SNS
7	1	2012092017	plus.google.com	2	23	구글 Plus	커뮤니티	블로그/SNS	SNS
8	1	2012122801	news.naver.com	3	213	네이버 뉴스	뉴스/미디어	인터넷신문	포털뉴스
9	1	2012123114	search.naver.com	1	0	네이버 검색	인터넷/컴퓨터	검색	포털검색
10	1	2013061008	blog.naver.com	1	46	네이버 블로그	커뮤니티	블로그/SNS	포털블로그

	CUS_ID	QRY_STR	QRY_CNT
1	1	못내	1
2	1	배트맨 리부&acr=1&qdt=0&ie=utf8&query=배트맨 리부트	1
3	1	배트맨 비긴즈 명대사&sm=top_sug.pre&fbm=0&acr=9…	1
4	1	베수비&acr=1&qdt=0&ie=utf8&query=베수비오 화산	1
5	1	베이징올림픽 장미란	1
6	1	베이징올림픽 장미란 몸무게	1
7	1	베이징올림픽 장미란 코로브카	1
8	1	별이 빛나는 밤&sm=top_sug.pre&fbm=0&acr=2&acq=…	1
9	1	본 레거시 첫주 박스오피스	1
10	1	본 슈프리머시 첫주 박스오피스	1



	CUS_ID	AGE
1	1	40
2	2	20
3	3	20
4	4	30
5	5	40
6	6	40
7	7	40
8	8	30
9	9	40
10	10	30

Y

Clickstream Data로부터 만들 수 있는 Feature

필드 명	필드 개수	내용
DWELLTIME	1	총 체류시간(ST_TIME의 총합)
PAGEVIEWS	1	총 페이지뷰(SITE_CNT의 총합)
VSITES	1	접속한 서로 다른 사이트(SITE_NM)의 수
COVERAGE	1	총 22개의 사이트 카테고리(BACT_NM)에 얼마나 다양하게 접속했는지에 대한 비율("서로 다른 카테고리 수/22"로 계산. 예: 22개 카테고리에 모두 접속한 경우는 1, 11개만 접속한 경우는 0.5)
SITECOV	1	사이트 카테고리(BACT_NM) 별 체류시간 변동계수(카테고리 별 체류시간의 "표준편차/평균" 값)
VDAYS	1	총 접속 일수
DAYTIME	1	하루 평균 체류시간
DAYCOV	1	일별 변동계수(일일 체류시간의 "표준편차/평균" 값)
CT* (*: BACT_NM)	22	사이트 카테고리 별 체류시간 비율. 즉, 전체 체류시간(ST_TIME) 중 특정 카테고리(BACT_NM)에 얼마나 머물렀는가를 비율로 계산. BACT_NM은 총 22개임
DF* (*: 월요일 ~ 일요일)	7	요일 당 체류시간 비율
HF* (*: 0005, 0611, 1217, 1823)	4	시간대별(0-5시, 6-11시, 12-17시, 18-23시) 체류시간 비율

Traditional ML vs. deep learning

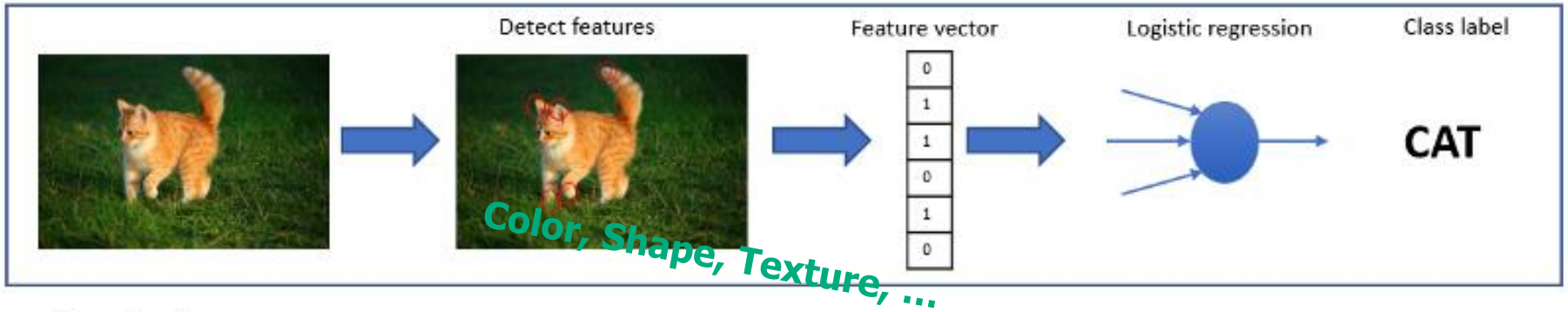
Traditional Machine Learning Flow

Deep Learning Flow

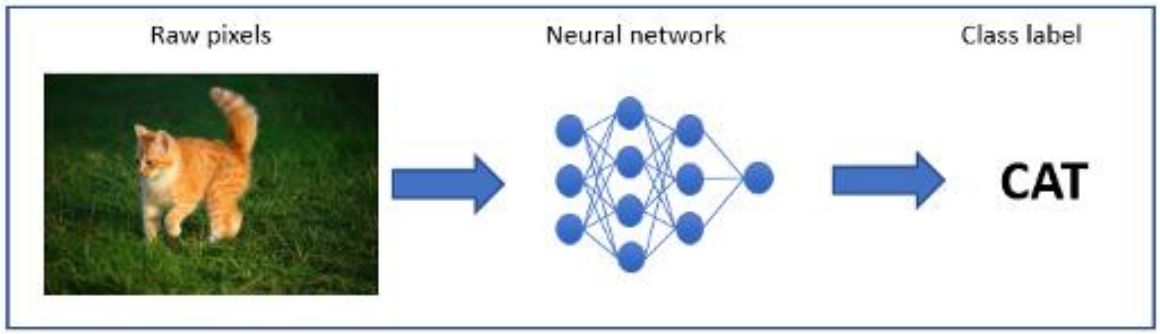
Coming up with features is difficult, time-consuming, requires expert knowledge.“

Applied machine learning" is basically feature engineering. ***- Andrew Ng***

Traditional Computer Vison

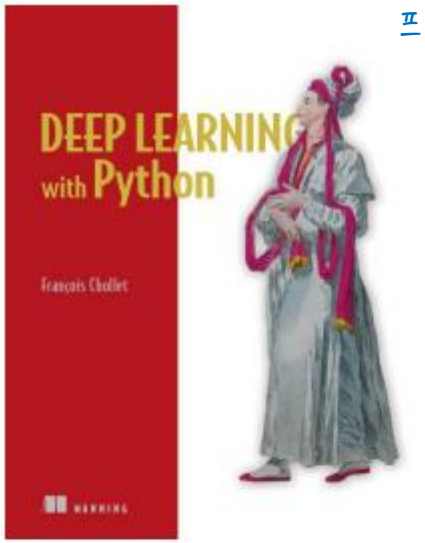


Deep learning



Source: https://blog.esciencecenter.nl/mcfly-time-series-classification-made-easy-e47de8d29838

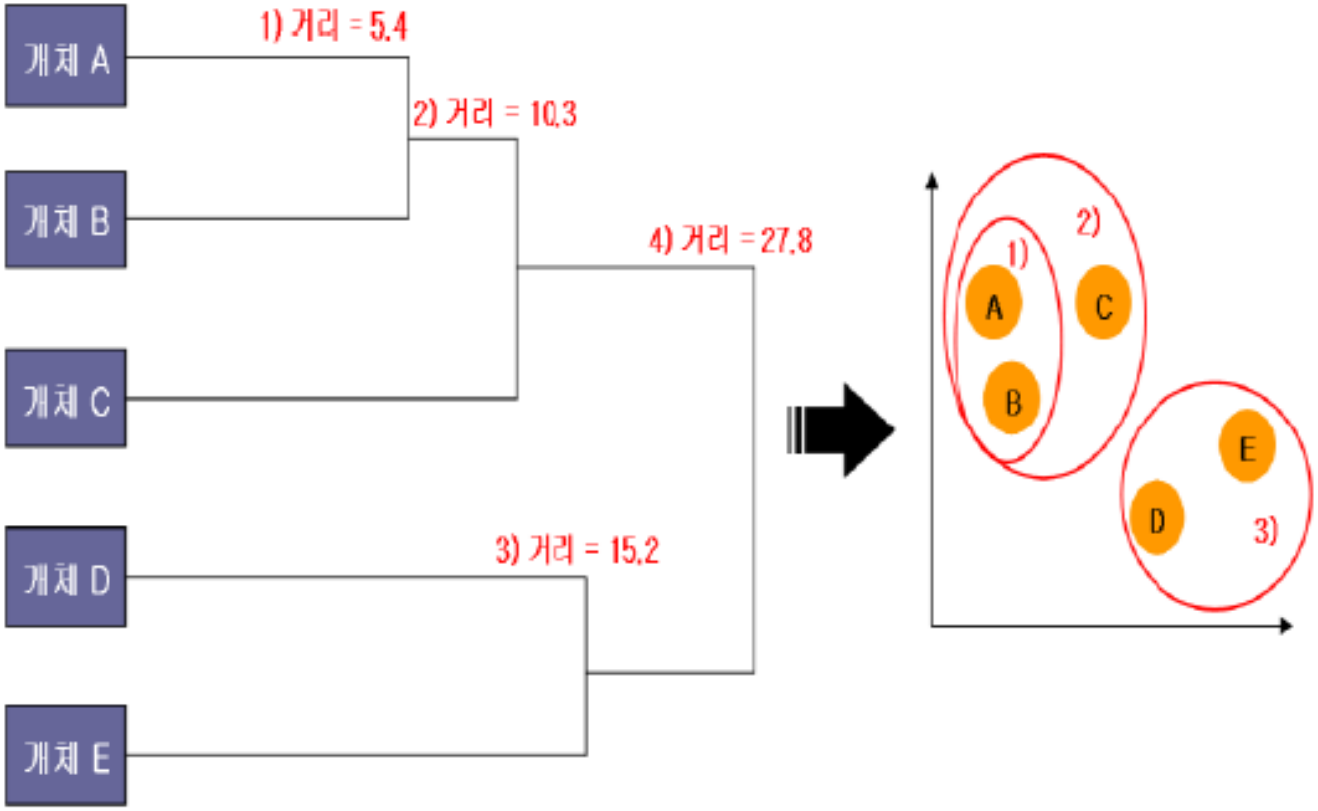
프랑스와 숄레 著



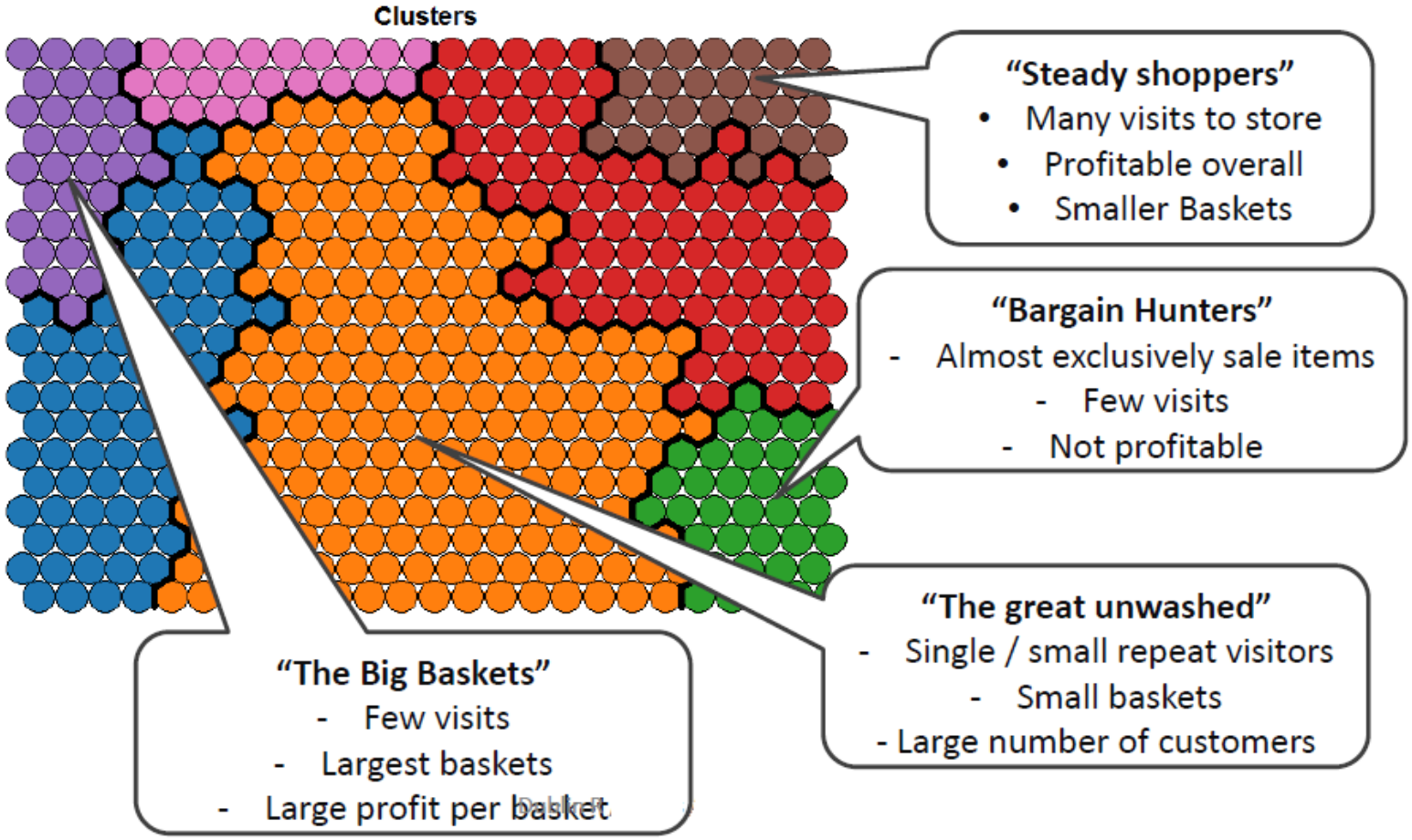
In 2016 and 2017, Kaggle was dominated by two approaches: gradient boosting machines and deep learning. Specifically, gradient boosting is used for problems where structured data is available, whereas deep learning is used for perceptual problems such as image classification. Practitioners of the former almost always use the excellent XGBoost library, which offers support for the two most popular languages of data science: Python and R. Meanwhile, most of the Kaggle entrants using deep learning use the Keras library, due to its ease of use, flexibility, and support of Python.

These are the two techniques you should be the most familiar with in order to be successful in applied machine learning today: gradient boosting machines, for shallow-learning problems; and deep learning, for perceptual problems. In technical terms, this means you'll need to be familiar with XGBoost and Keras—the two libraries that currently dominate Kaggle competitions. With this book in hand, you're already one big step closer.

군집화(Clustering)



- SOM을 이용한 식료품점 고객 세분화 사례



Source: Ta-Feng Grocery Shopping Analysis by Shane Lynn

연관규칙탐사(Association Rule Learning)

- 정의
  - 데이터 안에 존재하는 항목간의 종속 관계를 찾아내는 작업
- 장바구니 분석(market basket analysis)
  - 고객의 장바구니에 들어있는 품목 간의 관계를 발견
- 규칙의 표현
  - 항목 A와 품목 B를 구매한 고객은 품목 C를 구매한다.
  - (품목 A) & (품목 B) ⇒ (품목 C)
- 연관규칙의 활용
  - 제품이나 서비스의 교차판매
  - 매장진열, 첨부우편
  - 사기적발

고객의 구매 상품 List

ID	판매 상품
1	소주 , 콜라 ,맥주
2	소주 , 콜라 , 와인
3	소주 , 주스
4	콜라 , 맥주
5	소주 , 콜라 ,맥주 , 와인
6	주스

지지도가 **50%** 이상인 연관성 규칙

지지도 **50%** 이상인 규칙	해당 **Transaction**	신뢰도
소주 => 콜라	1,2,5	75 %
콜라 => 맥주	1,4,5	75 %
맥주 => 콜라	1,4,5	100 %

* 연관규칙 : 맥주를 구입한 사람들 모두는(100%) 콜라도 구매한다

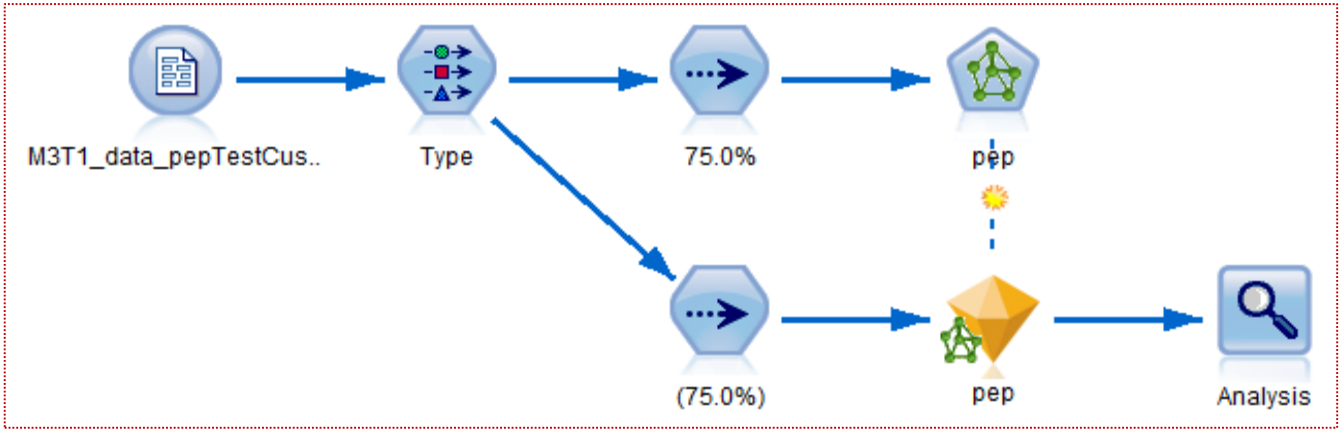
- 지지도: 그리고 이러한 경향을 가지는 사람들은 전체의 절반(50%) 정도이다.

Machine Learning Tools

구분	SAS Enterprise Miner	IBM SPSS Modeler	R ecosystem	Python Ecosystem
특징	사용이 간편한 GUI를 통해 모델 구축의 가속화가 가능하다. (비주얼 프로그래밍 기반)	Data 핸들링에 강하고 사용하기 쉬운 사용자 인터페이스를 가지고 있다. (비주얼 프로그래밍 기반)	통계학적 요소가 잘 스며들어있는 오픈 소스 데이터 프로그래밍 도구 (함수형 언어 기반)	문법이 간결하고 직관적인 오픈 소스 다목적 프로그래밍 도구 (객체지향 언어 기반)
장점	대용량 데이터분석이 가능하고 활용영역이 다양하다.	초보자가 배우기 쉽다. 체계적인 분석 프로세스를 지원한다.	코딩을 통해 효율적으로 분석할 수 있으며, 활용 가능한 패키지(특히 통계와 시각화)가 많다.	코딩을 통해 효율적으로 분석할 수 있으며, 딥러닝 등 최신의 머신러닝 라이브러리를 사용할 수 있다.
단점	사용법을 습득하는데 다소 시간이 걸린다. 데이터 처리의 유연성이 떨어지고 분석 알고리즘의수가 제한적이다.	분석을 위한 설정과 연결 과정에 대한 프로세스가 다소 많은 편이다. 제공되는 데이터 처리와 분석 기능이 제한적이다.	배우는데 상당히 많은 시간이 소요되며, 딥러닝 지원이 부족하고 속도가 매우 느리다.	R 보다는 코딩이 쉽지만 제공되는 패키지가 상대적으로 적고, 시각화 기능이 약하며 속도가 느리다.
비용	매우 고가	고가	무료	무료

Machine Learning Tools

IBM SPSS Modeler



Python ecosystem

```
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix
df = pd.read_csv('M3T1_data_pepTestCustomers.csv')
dfX = mdf.drop(['pep'], axis=1)
dfy = mdf['pep']
X_train, X_test, y_train, y_test = train_test_split(dfX, dfy, test_size=0.25, random_state=0)
tree = MLPClassifier(random_state=0)
tree.fit(X_train, y_train)
tree.score(X_test, y_test)
confusion_matrix(y_test, tree.predict(X_test))
```

Machine Learning Tools

R ecosystem

```
library(caret)
library(nnet)
df = read.csv('M3T1_data_pepTestCustomers.csv')
df$pep = factor(df$pep)
set.seed(0)
inTrain = createDataPartition(y=df$pep, p=0.75, list=FALSE)
train = df[inTrain,]
test = df[-inTrain,]
set.seed(0)
nn_model = nnet(pep ~ ., data=train)
confusionMatrix(predict(nn_model, newdata=test, type="class"), test$pep)
```

Python ecosystem

```
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix
df = pd.read_csv('M3T1_data_pepTestCustomers.csv')
dfX = mdf.drop(['pep'], axis=1)
dfy = mdf['pep']
X_train, X_test, y_train, y_test = train_test_split(dfX, dfy, test_size=0.25, random_state=0)
tree = MLPClassifier(random_state=0)
tree.fit(X_train, y_train)
tree.score(X_test, y_test)
confusion_matrix(y_test, tree.predict(X_test))
```

머신러닝의 한계

- 과적합(over-fitting) 또는 과도한 일반화(Overgeneralization) 문제
- 활용할 수 있는 정확한 데이터가 충분하지 않을 경우
- 샘플링 잡음 및 편향(Sampling Noise or Bias) 문제
- 낮은 품질의 데이터
- 모델 성능 약화(Model performance deterioration)

  Ex) 과거 현상들이 더 이상 미래에 유사하게 일어나지 않을 때 발생
- 영역 복잡성(Domain Complexity)

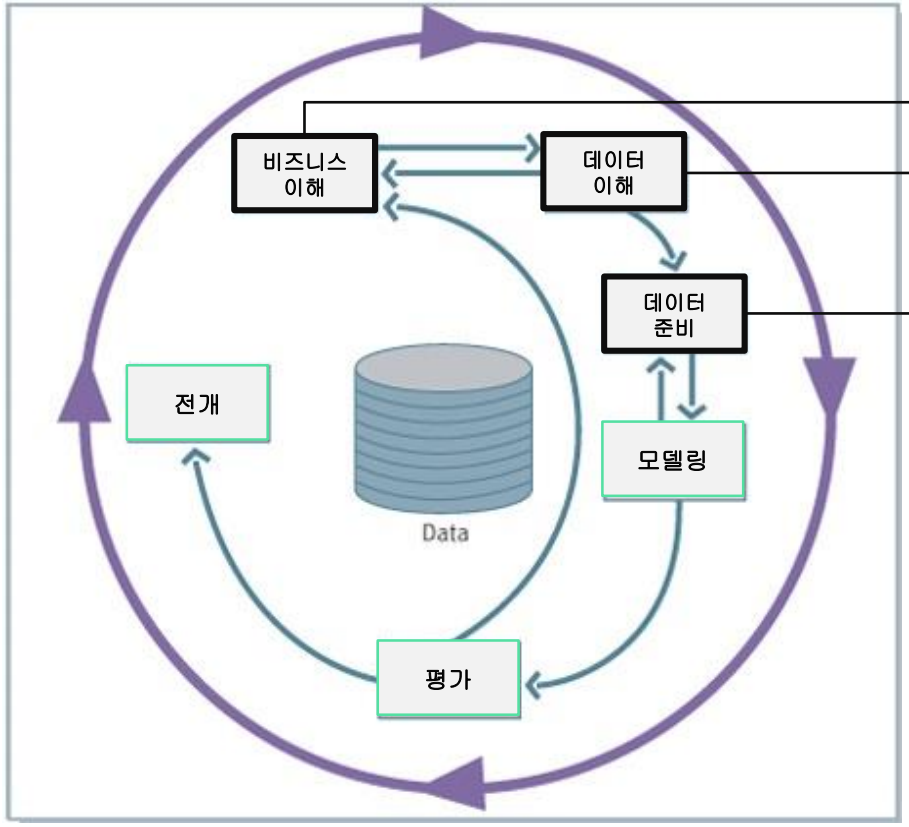
  Ex) 과거 학습된 내용(금융분야)을 다른 영역(법률분야)에 적용하지 못함
- 데이터 유출(Data Leakage)

  Ex) 부동산 가격을 예측할 때 매입 이후 지불하는 인지세 및 부동산 수수료와 같은 데이터를 입력 데이터에 포함시키는 경우

General Machine Learning Process: CRISP-DM

CRISP-DM : SPSS에서 제시하는 프로세스 (1/2)

CRISP-DM(cross-industry standard process for data mining)은 머신러닝에 관련된 광범위한 업무의 범위를 다루고 있음.



단계 1: 비즈니스 이해(Business Understanding)

- 각종 참고 자료와 현업 책임자와의 커뮤니케이션을 통해 해당 비즈니스를 이해하는 단계

단계 2: 데이터 이해(Data Understanding)

- 레코드 수, 변수종류, 자료의 질 등, 현업이 보유 관리하고 있는 데이터를 이해하는 단계

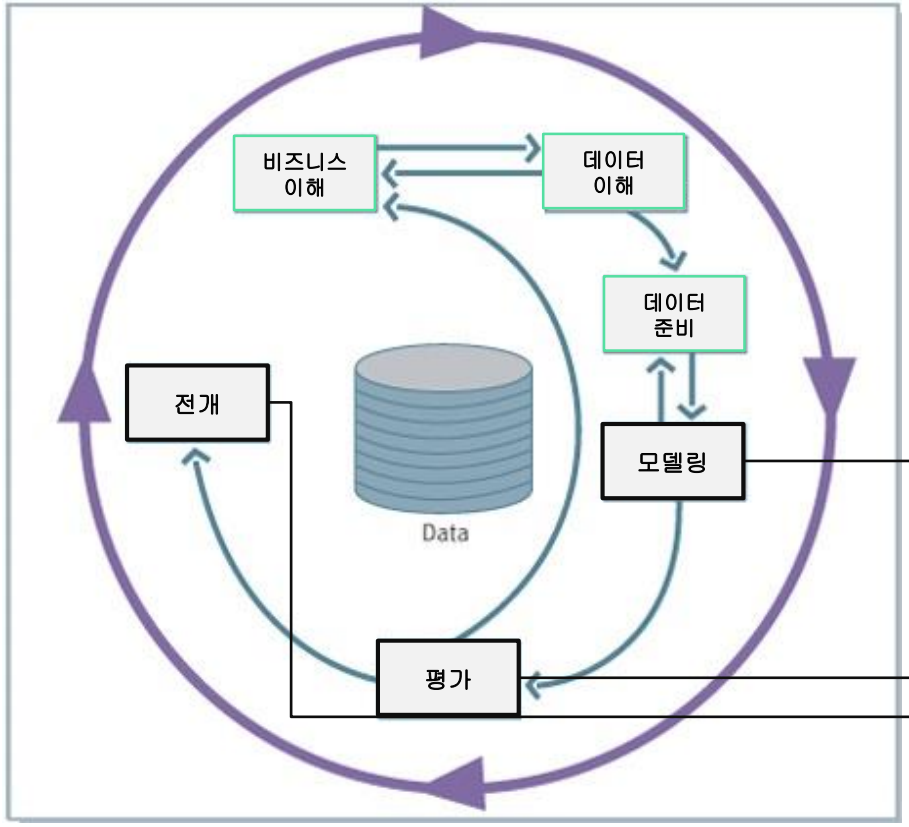
단계 3: 데이터 준비(Data Preparation)

- 데이터의 정제, 새로운 데이터 생성 등, 자료를 분석 가능한 상태로 만드는 단계

General Machine Learning Process: CRISP-DM

CRISP-DM : SPSS에서 제시하는 프로세스 (2/2)

CRISP-DM(cross-industry standard process for data mining)은 머신러닝에 관련된 광범위한 업무의 범위를 다루고 있음.



단계 4: 모델링 (Modeling)

- 자료 기술 및 탐색을 포함하여 필요한 각종 모델링을 하는 단계

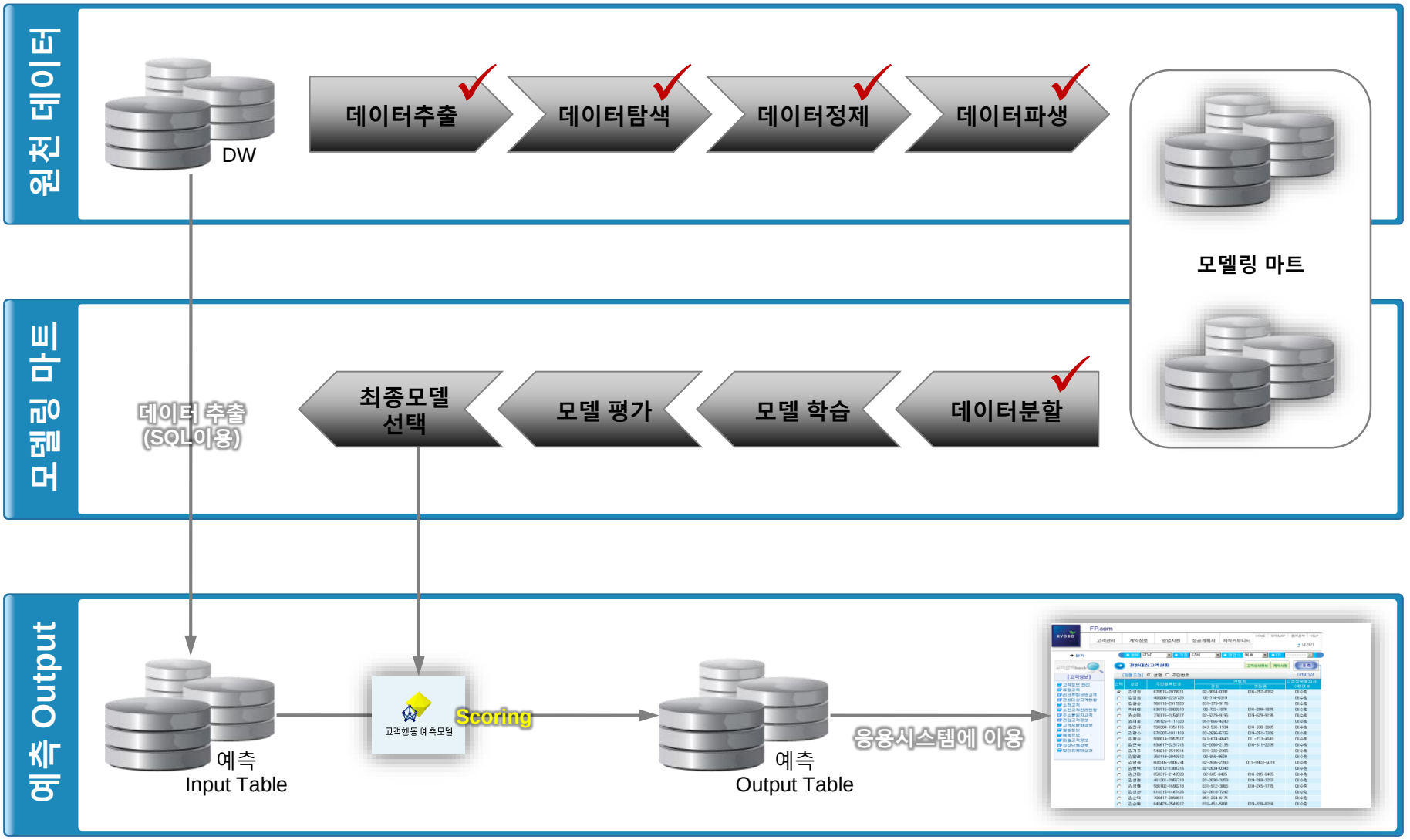
단계 5: 평가 (Evaluation)

- 모형의 해석 가능 여부, 독립적인 새 자료에 적용되는 경우에도 재현가능한지를 검토하는 단계

단계 6: 전개 (Deployment)

- 각 관리자에게 전달하여 필요한 조치를 취하는 등 검토가 끝난 모형을 실제 현업에 적용하는 단계

Predictive Analytics Process



Scikit-learn: Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Classification

데이터 준비

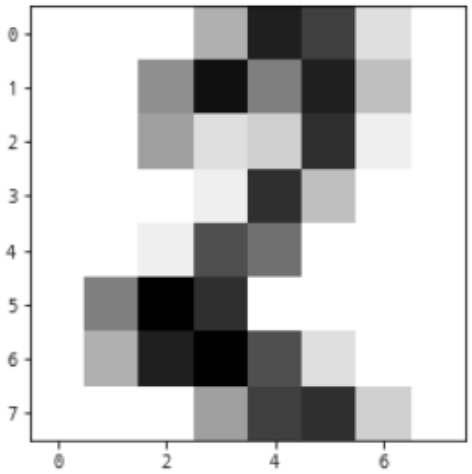
In [1]:
```
from sklearn import datasets
```

In [2]:
```
digits = datasets.load_digits()
```

In [3]:
```
import matplotlib.pyplot as plt
%matplotlib inline

plt.imshow(digits.data[50].reshape(8,8), cmap=plt.cm.gray_r)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x7f5cac0761d0>
```



In [4]:
```
digits.target[50]
```

Out[4]:
```
2
```

In [5]:
```
X, y = digits.data, digits.target
```

데이터 분할

In [6]:
```
from sklearn.model_selection import train_test_split
```

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

모형 생성 - 로지스틱 회귀분석

In [8]:
```
from sklearn.linear_model import LogisticRegression
```

In [9]:
```
model = LogisticRegression()
```

In [10]:
```
model.fit(X_train, y_train)
```

Out[10]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

모형 평가

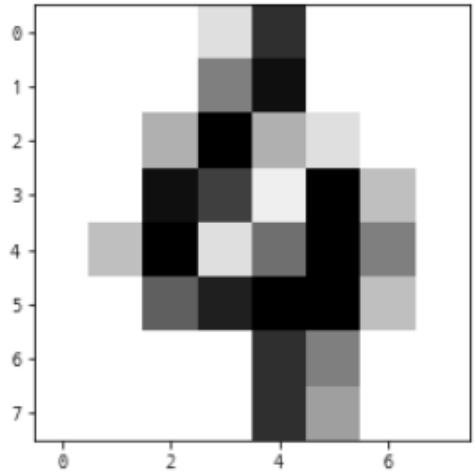
In [11]:
```
model.score(X_test, y_test)
```

Out[11]:
```
0.9518518518518518
```

모형 적용

In [12]:
```
plt.imshow(digits.data[100].reshape(8,8), cmap=plt.cm.gray_r)
digits.target[100]
```

Out[12]: 4



In [13]:
```
X_unkwon = digits.data[100].reshape(-1,64)
model.predict(X_unkwon)
```

Out[13]: array([4])

In [14]:
```
model.predict_proba(X_unkwon)
```

Out[14]:
```
array([[2.11642117e-07, 1.90068392e-03, 4.64681361e-16, 8.42195460e-15,
        9.98017776e-01, 1.19241887e-08, 8.06990618e-05, 4.94125151e-07,
        1.23669825e-07, 2.96368895e-19]])
```

Regression

데이터 준비

In [15]:
```
boston = datasets.load_boston()
```

In [16]:
```
X =  boston.data
y = boston.target
```

In [17]:
```
print(X.shape)
print(boston.feature_names)
```

```
(506, 13)
['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO'
 'B' 'LSTAT']
```

데이터 분할

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

모형 생성 - 선형회귀분석

In [19]:
```
from sklearn.linear_model import LinearRegression
```

In [20]:
```
model = LinearRegression()
```

In [21]:
```
model.fit(X_train, y_train)
```

Out[21]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

모형 평가

In [23]:
```
model.score(X_test, y_test) # R-square(r2_score)
```

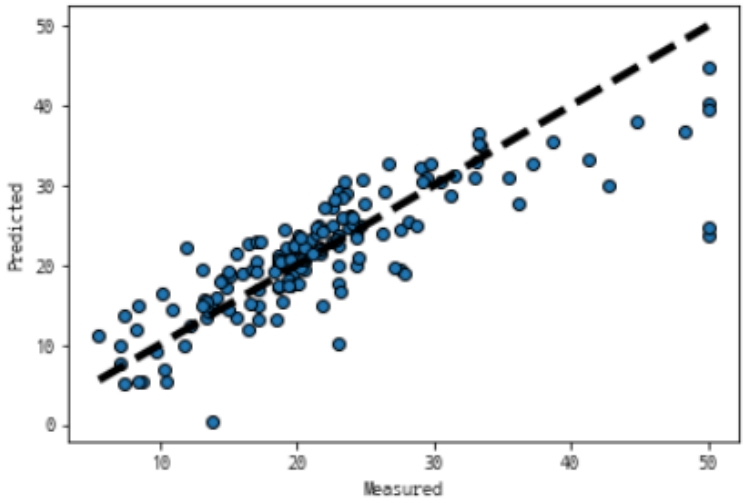
Out[23]:
```
0.6735280865347263
```

분석결과 시각화

In [24]:
```
y_pred = model.predict(X_test)
```

In [25]:
```
plt.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=4)
plt.xlabel('Measured')
plt.ylabel('Predicted')
```

Out[25]:
```
Text(0,0.5,'Predicted')
```



Clustering

데이터 준비

In [36]:
```
X = digits.data
```

군집 분석 - Kmeans

In [37]:
```
from sklearn.cluster import KMeans
```

In [38]:
```
kmeans = KMeans(n_clusters=10)
```

In [39]:
```
kmeans.fit(X)
```

Out[39]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=10, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [40]:
```
kmeans.labels_
```

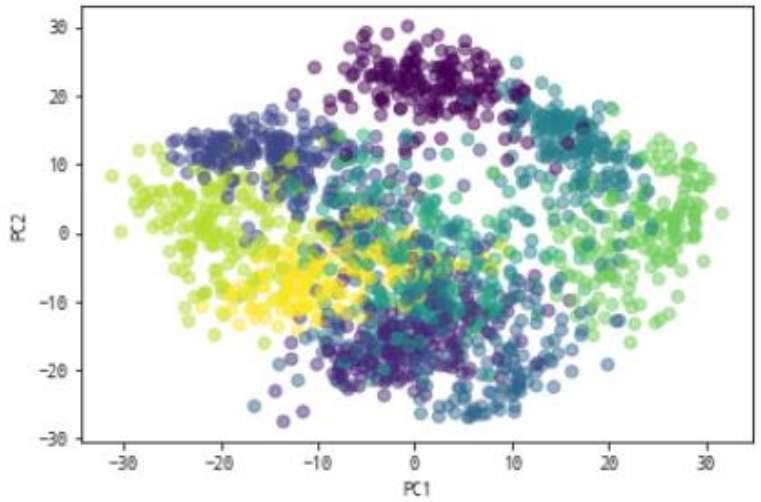
Out[40]:
```
array([0, 3, 3, ..., 3, 2, 2], dtype=int32)
```

분석결과 시각화

In [41]:
```
from sklearn.decomposition import PCA
pca_cov = PCA(n_components=2)

X_reduced = pca_cov.fit_transform(X)
plt.scatter(X_reduced[:,0], X_reduced[:,1], c=kmeans.labels_, alpha=0.5)
plt.xlabel('PC1')
plt.ylabel('PC2')
```

Out [41]:
```
Text(0,0.5,'PC2')
```



Association Rule Mining (Market Basket Analysis)

데이터 준비

In [33]:
```
import pandas as pd
```

In [34]:
```
store_data = pd.read_csv('store_data.tab', sep='₩t')
store_data.head()
```

Out [34]:

	ID	heel	tee	skirt	knit	jacket	jewelry	coat	flat	shorts	blous
0	1	1	0	0	0	0	0	0	0	1	0
1	2	1	0	0	0	0	0	0	1	0	0
2	3	1	0	0	0	0	0	0	1	1	0
3	4	1	0	0	0	1	1	0	0	0	0
4	5	1	0	0	0	0	0	0	0	0	0

In [35]:
```
transactions = store_data.iloc[:,1:]
```

빈발항목집합 추출 - Apriori

In [36]:
```
# !pip install mlxtend
from mlxtend.frequent_patterns import apriori, association_rules
```

In [37]:
```
freq_items = apriori(transactions, min_support=0.05, use_colnames=True)
freq_items.sort_values(by='support', ascending=False)
```

Out [37]:

	support	itemsets
0	0.492366	(heel)
7	0.474555	(shorts)
8	0.455471	(blous)
5	0.428753	(jewelry)
1	0.402036	(tee)

연관규칙 도출

In [38]:
```
rules = association_rules(freq_items, metric='confidence')
rules.query('confidence >= 0.85')
```

Out [38]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
13	(jacket, tee, skirt)	(jewelry)	0.083969	0.428753	0.072519	0.863636	2.014297	0.036517	4.189143
15	(jacket, tee, shorts)	(jewelry)	0.076336	0.428753	0.066158	0.866667	2.021365	0.033429	4.284351
16	(jacket, tee, blous)	(jewelry)	0.081425	0.428753	0.071247	0.875000	2.040801	0.036336	4.569975
19	(jacket, blous, skirt)	(jewelry)	0.078880	0.428753	0.067430	0.854839	1.993778	0.033610	3.935256
21	(jacket, heel, tee, skirt)	(jewelry)	0.066158	0.428753	0.057252	0.865385	2.018375	0.028887	4.243548
24	(jacket, heel, tee, shorts)	(skirt)	0.057252	0.394402	0.050891	0.888889	2.253763	0.028310	5.450382
29	(heel, tee, blous, skirt)	(jewelry)	0.090331	0.428753	0.077608	0.859155	2.003845	0.038879	4.055852
32	(jacket, tee, blous, heel)	(jewelry)	0.063613	0.428753	0.055980	0.880000	2.052463	0.028705	4.760390
40	(jacket, shorts, blous, skirt)	(heel)	0.061069	0.492366	0.052163	0.854167	1.734819	0.022095	3.480916
42	(shorts, jewelry, blous, skirt)	(heel)	0.087786	0.492366	0.077608	0.884058	1.795529	0.034385	4.378340
44	(jacket, heel, blous, shorts)	(jewelry)	0.064885	0.428753	0.055980	0.862745	2.012219	0.028160	4.161941
45	(jacket, shorts, tee, skirt)	(jewelry)	0.063613	0.428753	0.054707	0.860000	2.005816	0.027433	4.080334
47	(jacket, tee, blous, skirt)	(jewelry)	0.062341	0.428753	0.055980	0.897959	2.094350	0.029251	5.598219
50	(jacket, tee, blous, shorts)	(jewelry)	0.055980	0.428753	0.050891	0.909091	2.120313	0.026889	6.283715
51	(jacket, shorts, blous, skirt)	(jewelry)	0.061069	0.428753	0.053435	0.875000	2.040801	0.027252	4.569975
52	(blous, tee, skirt, jewelry, shorts)	(heel)	0.067430	0.492366	0.058524	0.867925	1.762761	0.025324	3.843511
53	(blous, tee, skirt, heel, shorts)	(jewelry)	0.068702	0.428753	0.058524	0.851852	1.986812	0.029068	3.855916
55	(blous, tee, jewelry, heel, shorts)	(skirt)	0.067430	0.394402	0.058524	0.867925	2.200609	0.031930	4.585242