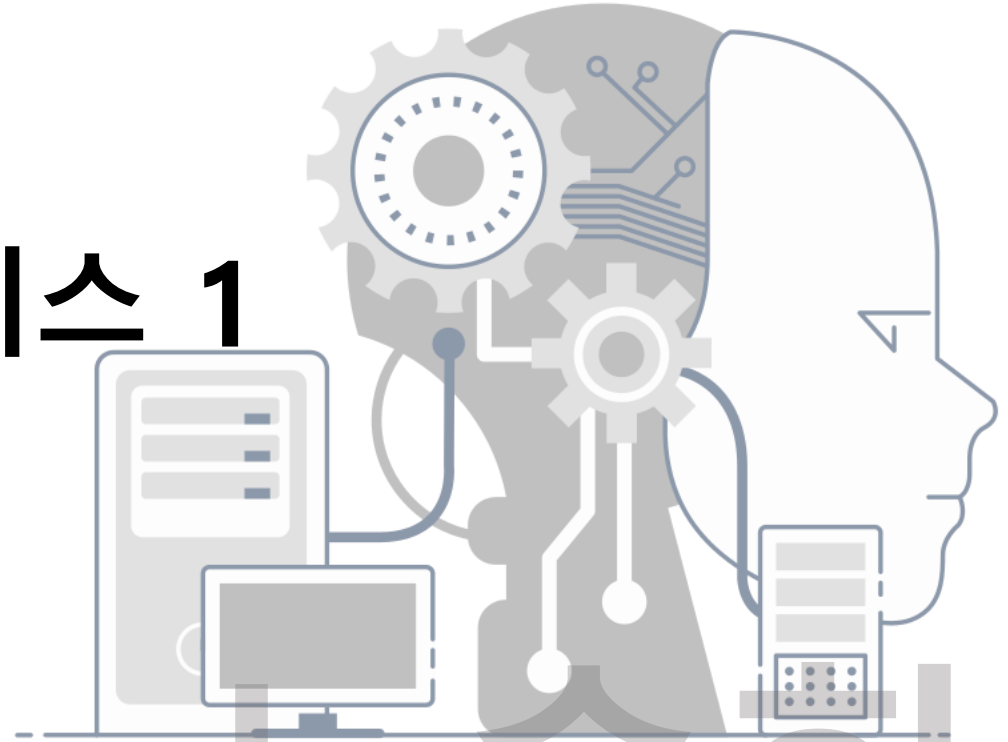
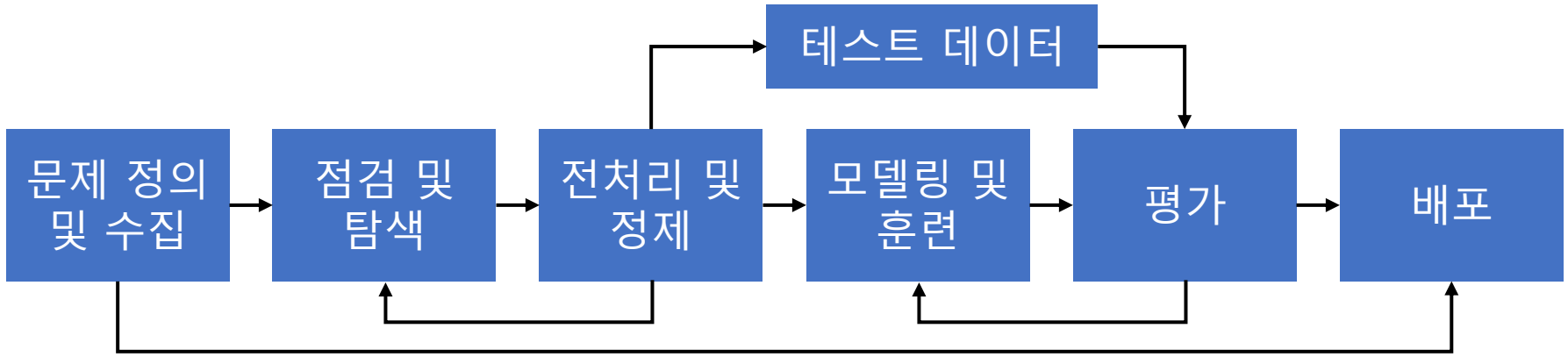
2022 DX Camp

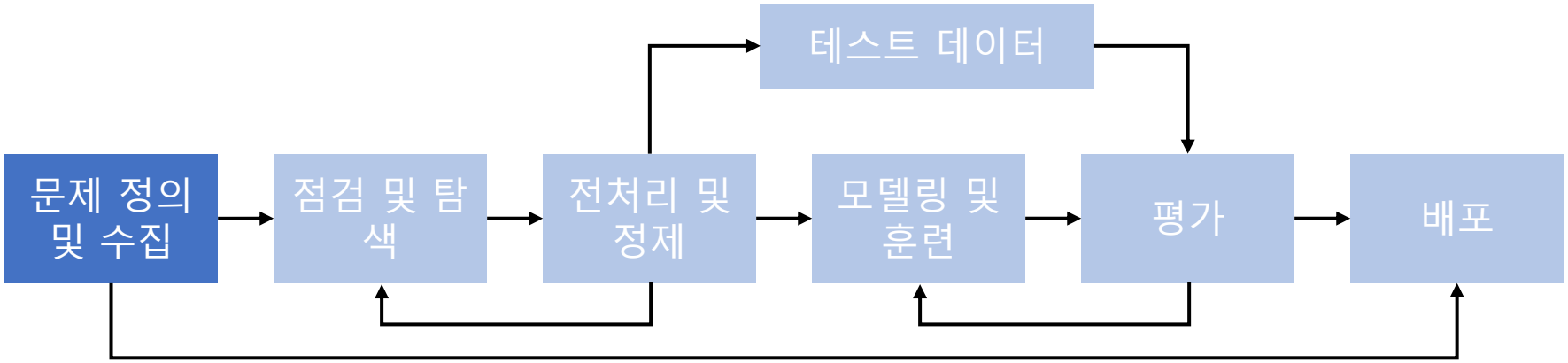
2강 머신러닝 프로세스 1



머신러닝 워크플로우



1. 문제 정의 및 데이터 수집



- 전체 머신러닝 워크플로우 중 가장 중요한 단계
- 명확한 목적 의식을 가지고 프로세스를 시작
- 모델의 종류 결정 및 탐색할 데이터의 종류를 결정
- 문제 정의 후 모델을 학습할 데이터를 수집(크롤링, 센서 활용, 구글링, 데이터 API 활용 등)

문제 정의

- 비즈니스 목적 정의
- 현재의 솔루션 파악
- 시스템 구성 결정
- 성능 측정 지표 선택
- 가정 검사

저장 방식에 따른 데이터 종류

- **정형데이터**

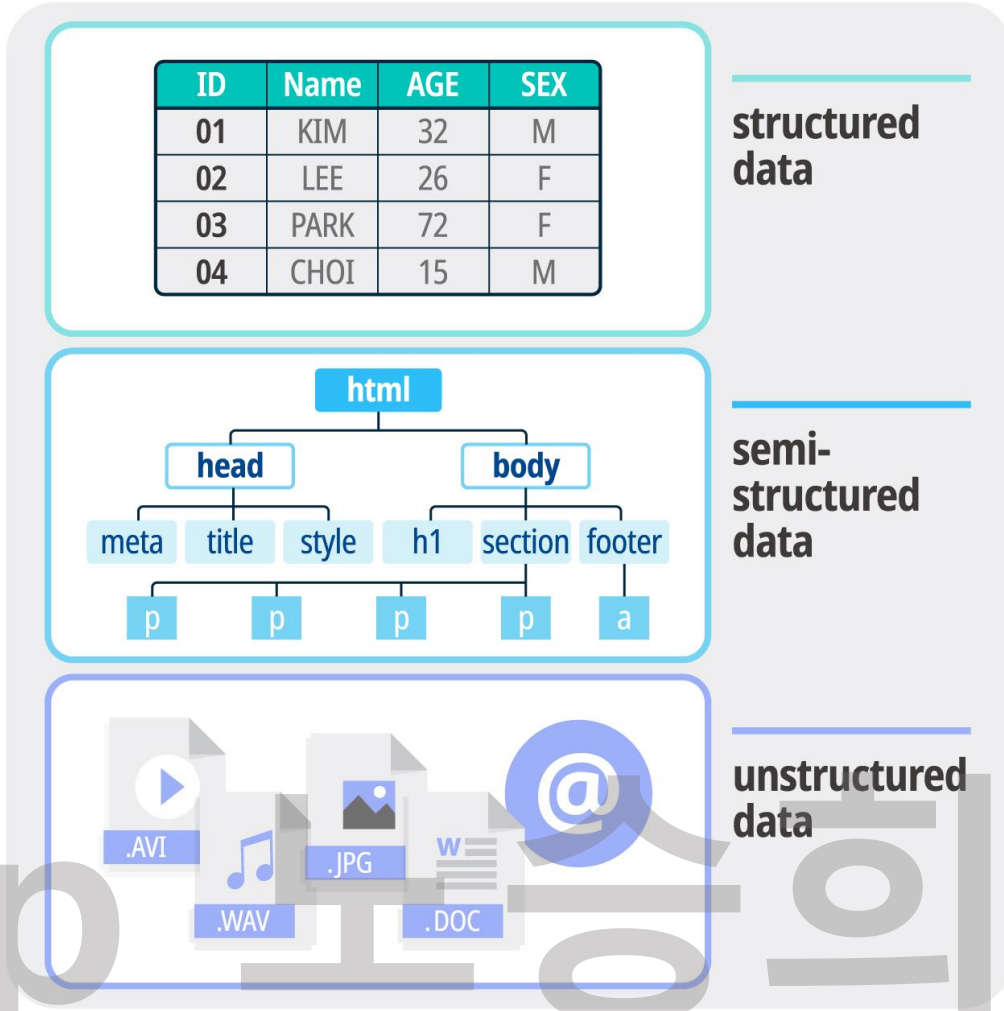
- 관계형 데이터 베이스(RDBMS), 스프레드시트, 엑셀

- **반정형데이터**

- JSON, HTML, XML, 로그

- **비정형데이터**

- 텍스트, 이미지, 음원 데이터, 빅데이터



http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=175129-2

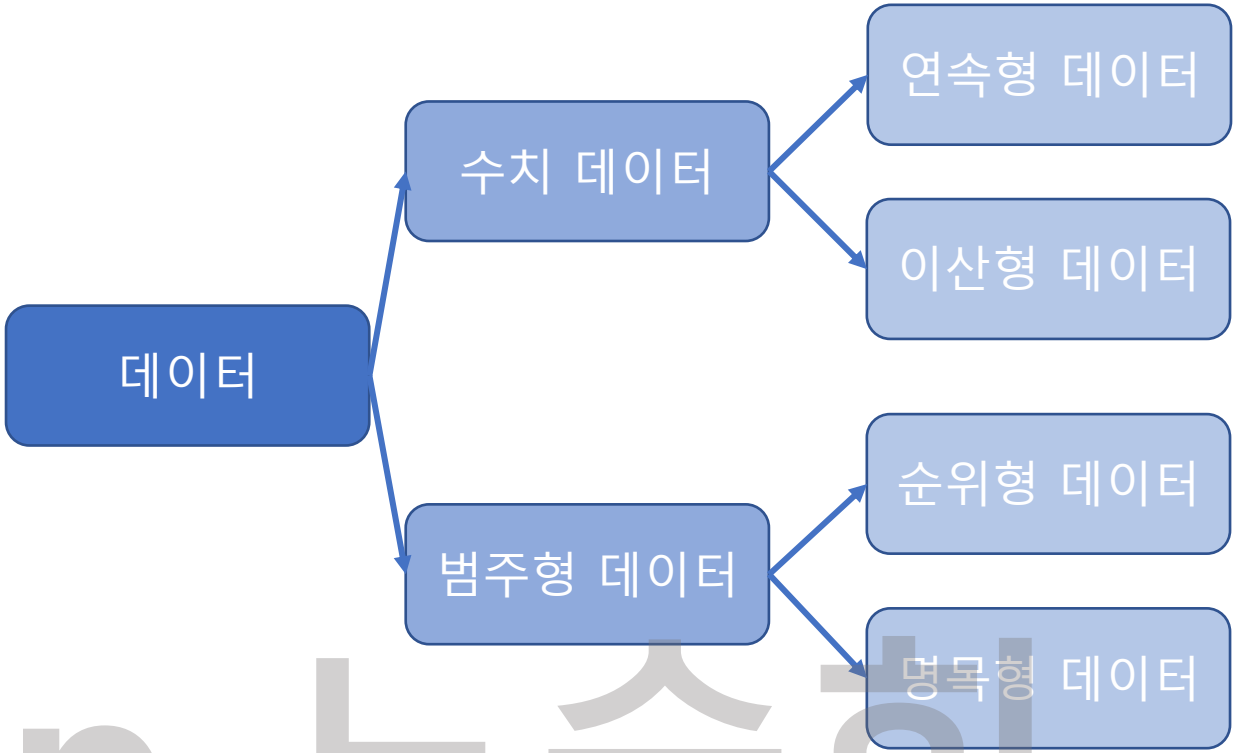
형태에 따른 데이터 종류

- **범주형 데이터**

- 정성적 데이터로써 몇 개의 범주로 나누어진 자료

- **수치형 데이터**

- 정량적 데이터로써 이산형과 연속형으로 이루어진 자료



데이터 수집 방법 정의

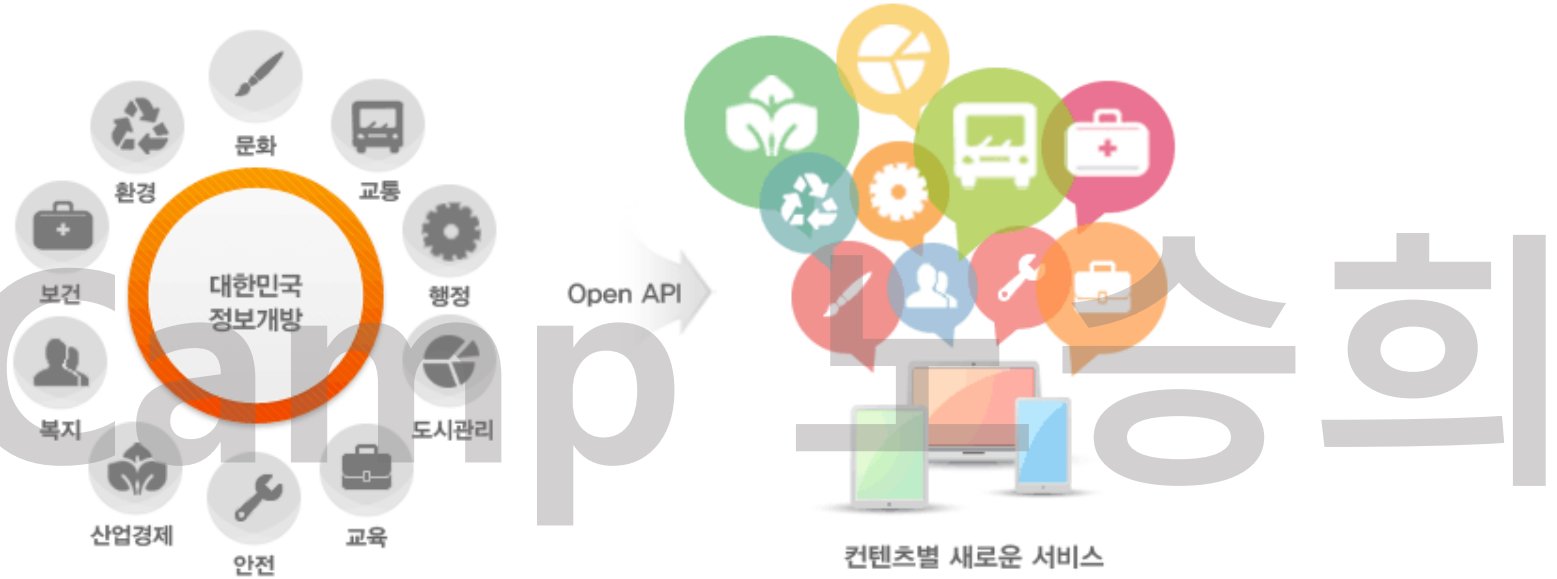
형태	특징	난이도
정형 데이터	• **내부 시스템**인 경우가 대부분이라 수집이 쉬움 • DBMS에 저장된 정형 데이터가 주를 이룸. • 파일 형태의 스프레드시트라도 내부에 형식을 가지고 있어 처리가 쉬운 편임	하
반정형 데이터	• **외부 시스템**인 경우가 많음 • 보통 **API 형태**로 제공, 데이터 처리 기술이 요구	중
비정형 데이터	• 텍스트 마이닝 혹은 파일일 경우 파일을 데이터 형태로 파싱(parsing) 필요	상

DBMS 수집

- 데이터베이스 관리 시스템으로부터 특정 테이블과 컬럼을 선택하여 정형 데이터 수집 가능.
- DBMS에서 데이터 수집을 하기 위해서는 SQL에 대한 이해 필요
- SQL을 통해 사용하고자 하는 머신러닝 프로젝트에 부합하는 정보를 추출하는 것이 핵심.
- SQL만 잘 다뤄도 데이터 분석가로서 중간 이상은 간다.

공개 데이터(Open API)

- API는 정의 및 프로토콜 집합을 사용하여 두 소프트웨어 구성 요소가 서로 통신할 수 있게 하는 메커니즘
- Open API(Open Application Programming Interface)는 누구나 사용할 수 있도록 공개된 API.
- 개발자에게 사유 응용 소프트웨어나 웹 서비스에 권한 제공.
- 네이버 지도, 구글맵, 오픈스트리트맵 등이 대표적인 예.
- 대한민국 정부에서는 공공데이터포털을 통해 오픈 API 운영 및 제공.

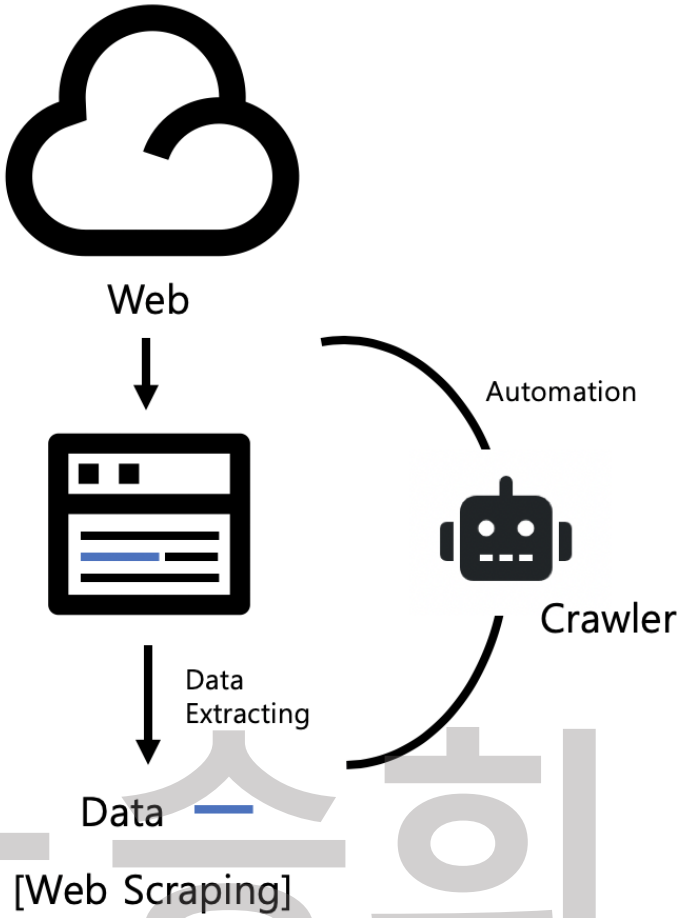


https://www.socialservice.or.kr:444/user/pubdata/open/openapi.do

웹 스크래핑(Web Scraping)

- 뉴스, 블로그, SNS, 웹 페이지 등 웹에서 공개되어 있는 데이터를 자동으로 수집하여 데이터를 추출하고 저장.
- 일반적으로 HTTP GET 요청을 보낸 다음 웹 서버가 전송하는 모든 정보를 복사하여 저장.
- 수백 또는 수천 개의 웹 페이지가 있는 대규모 사이트의 경우에도 몇 초안에 웹 사이트의 모든 콘텐츠를 다운로드 가능.
- 야놀자 vs 여기어때:

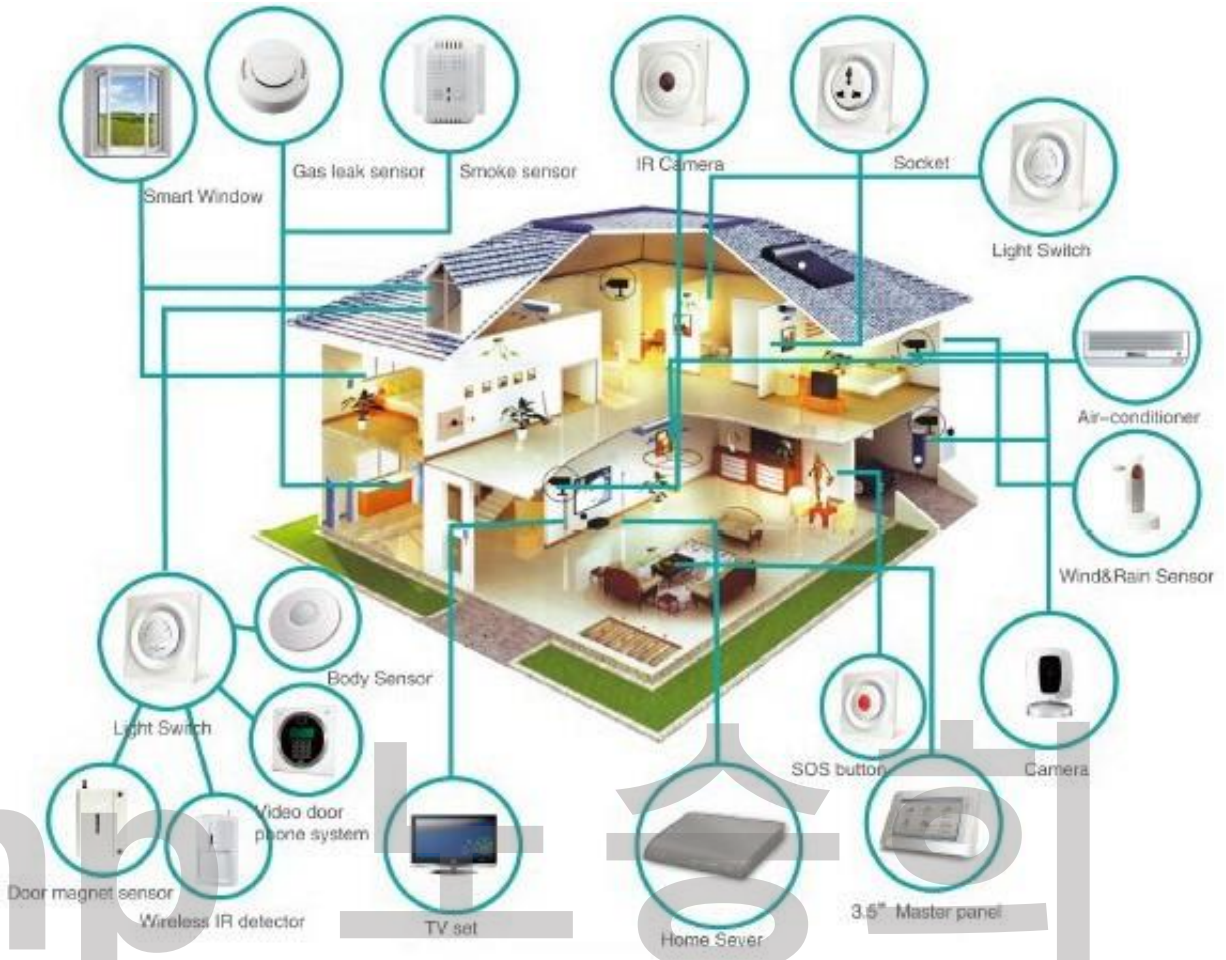
'이미 공개된 데이터는 크롤링으로 확보해도 문제가 없다 ' vs '영리를 위한 크롤링 행위는 지적재산권을 침해하는 행위'라는 주장이 맞서고 있다.
https://www.econovill.com/news/articleView.html?idxno=587292&fbclid=IwAR3BfuRoymHay85i94MTtf6iKaAhZqzd1PXEsuKS4El6OqNChIxHuGKkqd4



https://stackhoarder.com/2019/08/18/python%EB%B6%80%ED%84%B0-web-scraping-%EA%B9%8C%EC%A7%80-%EC%B5%9C%EB%8B%A8-%EC%8B%9C%EA%B0%84%EC%97%90-%EC%9D%B5%ED%98%80%EB%B3%B4%EC%9E%90/

센서 데이터 수집

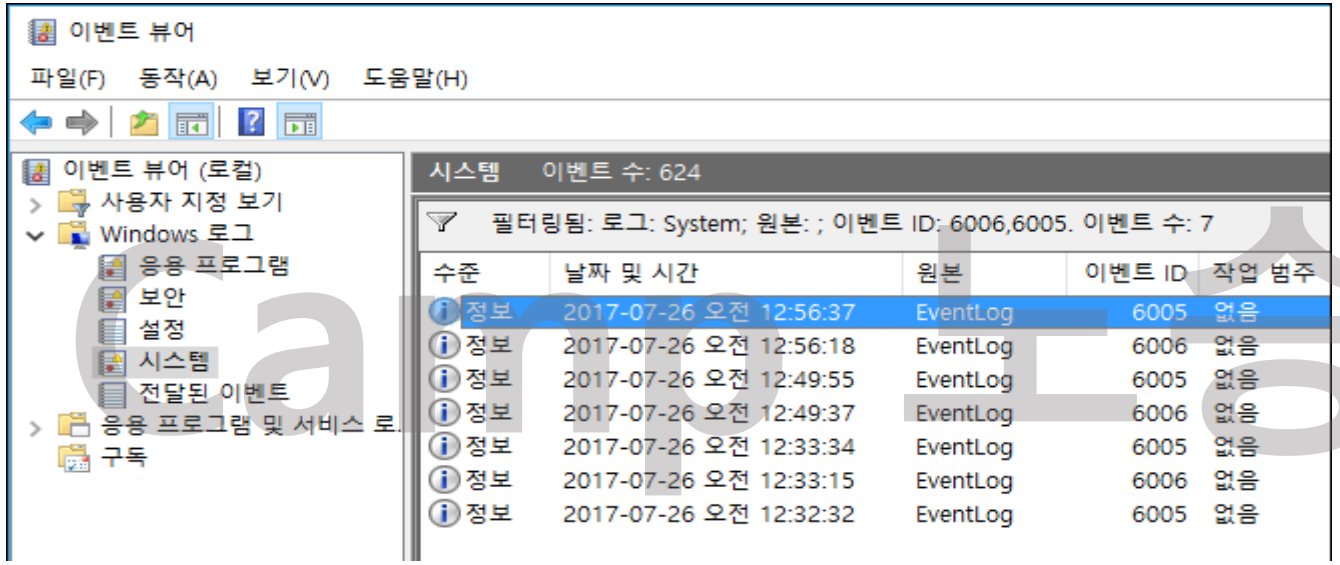
- IoT 기술의 발달로 수많은 기기로부터 각종 센서 데이터 수집 가능.
- 온도, 습도, 자이로, 압력, 속도/가속도, 가스, 초음파, 자기 센서 등 수 많은 센서 존재.
- 도시, 차량, 철도, 비행기, CCTV, 스마트 기기 등에서 센서 데이터 수집.



https://www.greened.kr/news/articleView.html?idxno=29890

로그 데이터 수집

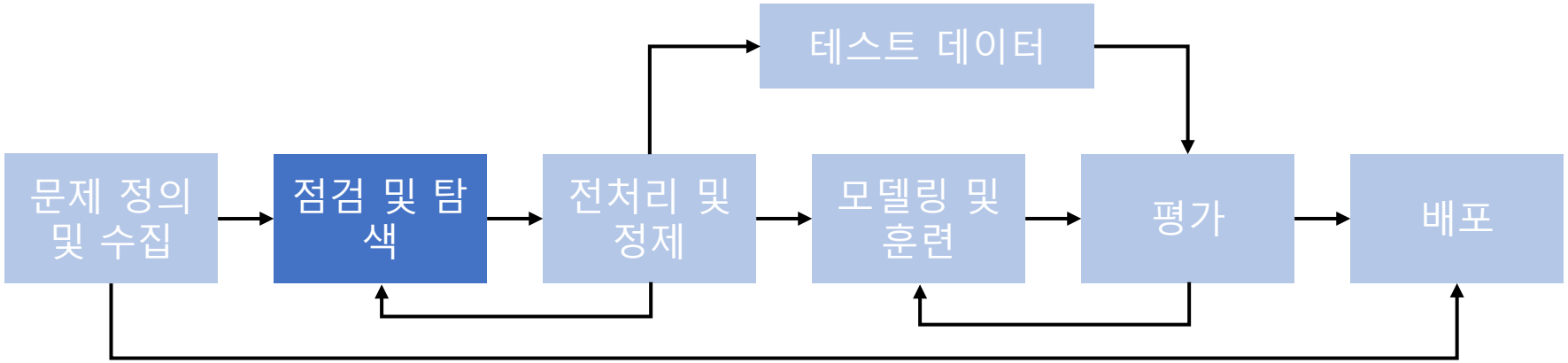
- 각종 시스템과 서버 그리고 네트워크 장비에서 수 많은 로그 데이터가 축적.
- 시스템 로그, 이벤트 로그, 웹 서버 로그, DB 로그, 트랜잭션 로그, 클릭 로그, 보안 로그 등 다양한 로그 데이터 존재.
- 시스템에서 에이전트를 통해 각종 로그를 수집하여 분석에 활용



데이터 관련 사이트

- 캐글: https://www.Kaggle.com/datasets
- 데이콘: https://dacon.io
- AI팩토리: https://aifactory.space
- 구글: https://datasetsearch.research.google.com/
- 레딧: https://www.reddit.com/r/datasets/
- UCI: https://archive.ics.uci.edu/ml
- 공공데이터포털: https://www.data.go.kr/
- AI허브: https://www.aihub.or.kr

2. 점검 및 탐색



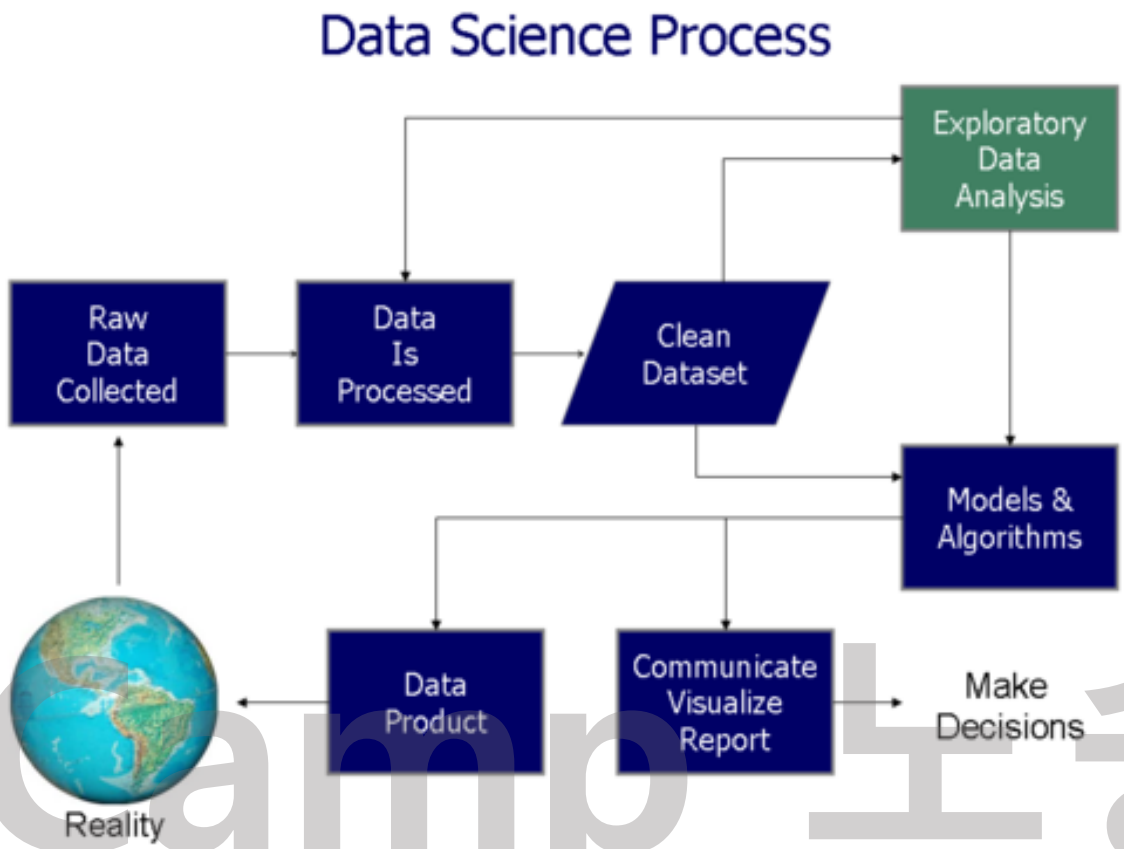
- 데이터를 점검하고 탐색하는 단계
- 데이터의 구조, 노이즈 데이터, 머신 러닝 적용을 위해서 데이터를 어떻게 정제해야 하는지 등을 파악
- 독립 변수, 종속 변수, 변수 유형, 변수의 데이터 타입 등을 점검
- 데이터의 특징과 내재하는 구조적 관계를 알아내는 과정을 의미
- 이 과정에서 시각화와 간단한 통계 테스트를 진행

https://eda-ai-lab.tistory.com/13

탐색적 데이터 분석(EDA)

- 미국의 존 튜키박사에 의해 창안
- 가설검증이나 모형 적용하기 전 데이터에 대한 정보를 사람에게 전달하도록 만드는 방법
- 시각적인 기법을 사용, 5-숫자요약(5-number summary) 등 다양한 방법을 적용
- 기존의 통계학은 정보 추출에서 가설 검정 등에 치우쳐 자료가 가지고 있는 본연의 의미를 찾는데 어려움
- 이를 보완하고자 주어진 자료만 가지고도 충분한 정보를 찾을 수 있도록 여러가지 **탐색적 데이터 분석 방법**을 개발

데이터 분석 사이클



https://www.kdnuggets.com/2016/03/data-science-process.html

탐색적 데이터 분석 = 요리 재료 파악

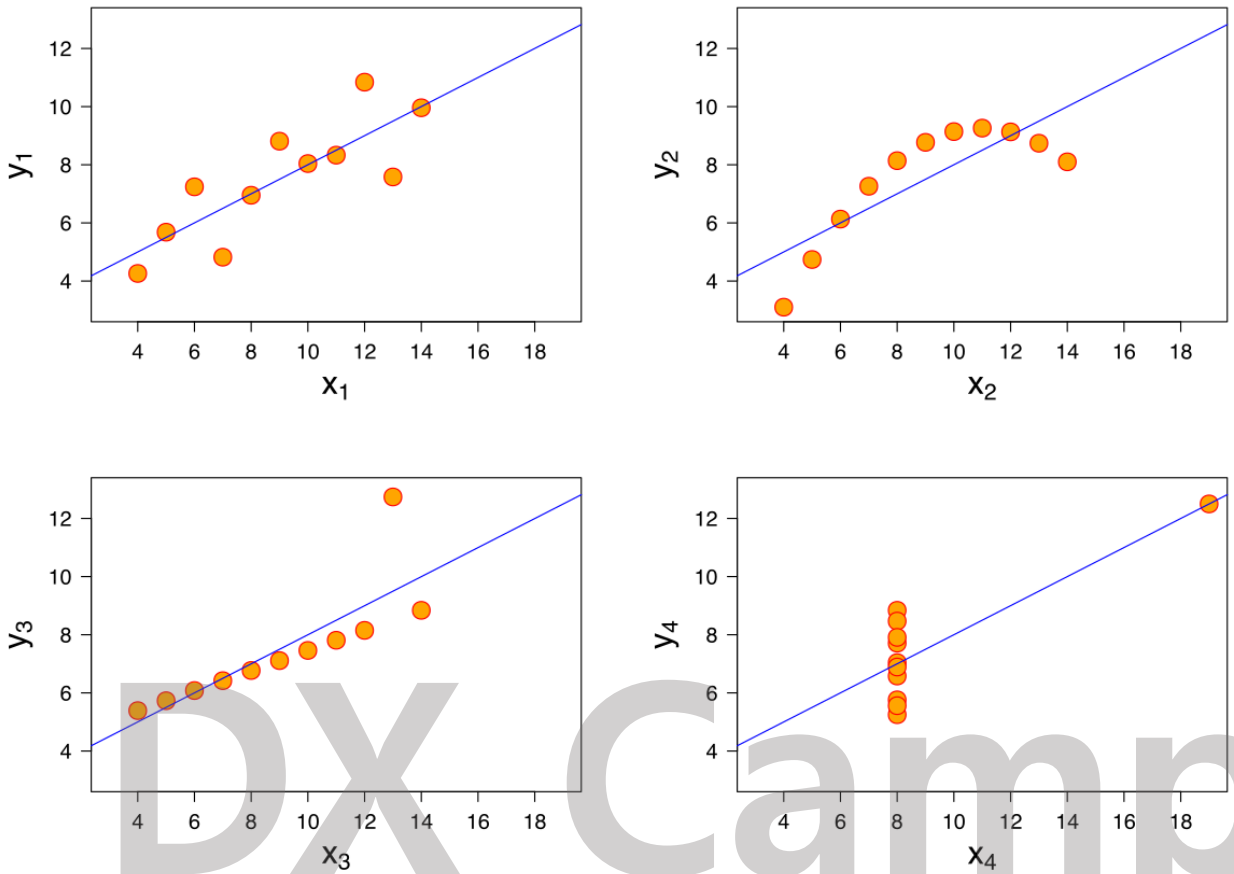
'맛있는 요리'를 만들기 위해서 가장 먼저 해야 할 일은 '맛있는 식재료'를 준비하는 것. 식재료가 맛있으면, 조리방법이 간단해도 맛있는 요리가 나오듯, 데이터 분석에 있어서 '맛있는 식재료'라고 할 수 있는 EDA를 잘 한다면 의미 있는 결과값을 도출하기 조금 더 수월해진다.

https://hub.zum.com/realfoods/26793

탐색적 데이터 분석의 필요성

- 데이터의 분포 및 값을 검토함으로써 데이터가 표현하는 현상을 더 잘 이해하고, 데이터에 대한 잠재적인 문제를 발견.
- 본격적인 분석에 들어가기에 앞서 데이터를 다시 수집하거나 추가로 수집하는 등의 결정을 내릴 수 있음.
- 데이터를 다양한 각도에서 살펴보는 과정을 통해 문제 정의 단계에서 미처 발생하지 못했을 다양한 패턴을 발견하고, 이를 바탕으로 기존의 가설을 수정하거나 새로운 가설을 새울 수 있음.
- 데이터에 대한 이런 지식은 이후에 통계적 추론을 시도하거나 예측 모델을 만들 때 유용.

앤스컴 콰르텟(Anscombe's quartet)



항목	값	정확도
x 평균	9	정확
x 표본분산	11	정확
y 평균	7.50	소수점 2자리
y 표본분산	4.125	정확
*x*와 *y*의 상관	0.816	소수점 3자리
선형회귀선	$y = 3.00 + 0.500x$	각 소수점 2자리, 소수점 3자리
선형회귀 결정계수	0.67	소수점 2자리

참고: https://www.autodesk.com/research/publications/same-stats-different-graphs

https://ko.wikipedia.org/wiki/%EC%95%A4%EC%8A%A4%EC%BB%B4_%EC%BD%B0%EB%A5%B4%ED%85%9F

탐색적 데이터 분석 과정

- 문제 정의 단계에서 세웠던 연구 질문과 가설을 바탕으로 분석 계획을 세우는 것.
- 분석 계획에는 어떤 속성 및 속성 간의 관계를 집중적으로 관찰해야 할지, 이를 위한 최적의 방법은 무엇인지가 포함되어야 함.
- 분석의 목적과 변수가 무엇이 있는지 확인하고, 개별 변수의 이름이나 설명을 가지는지 확인.
- 데이터를 전체적으로 살펴보기
- 데이터의 개별 속성값 관찰
- 속성 간의 관계에 초점을 두고, 개별 속성 관찰에서 찾지 못한 패턴 발견(상관관계, 시각화 등)

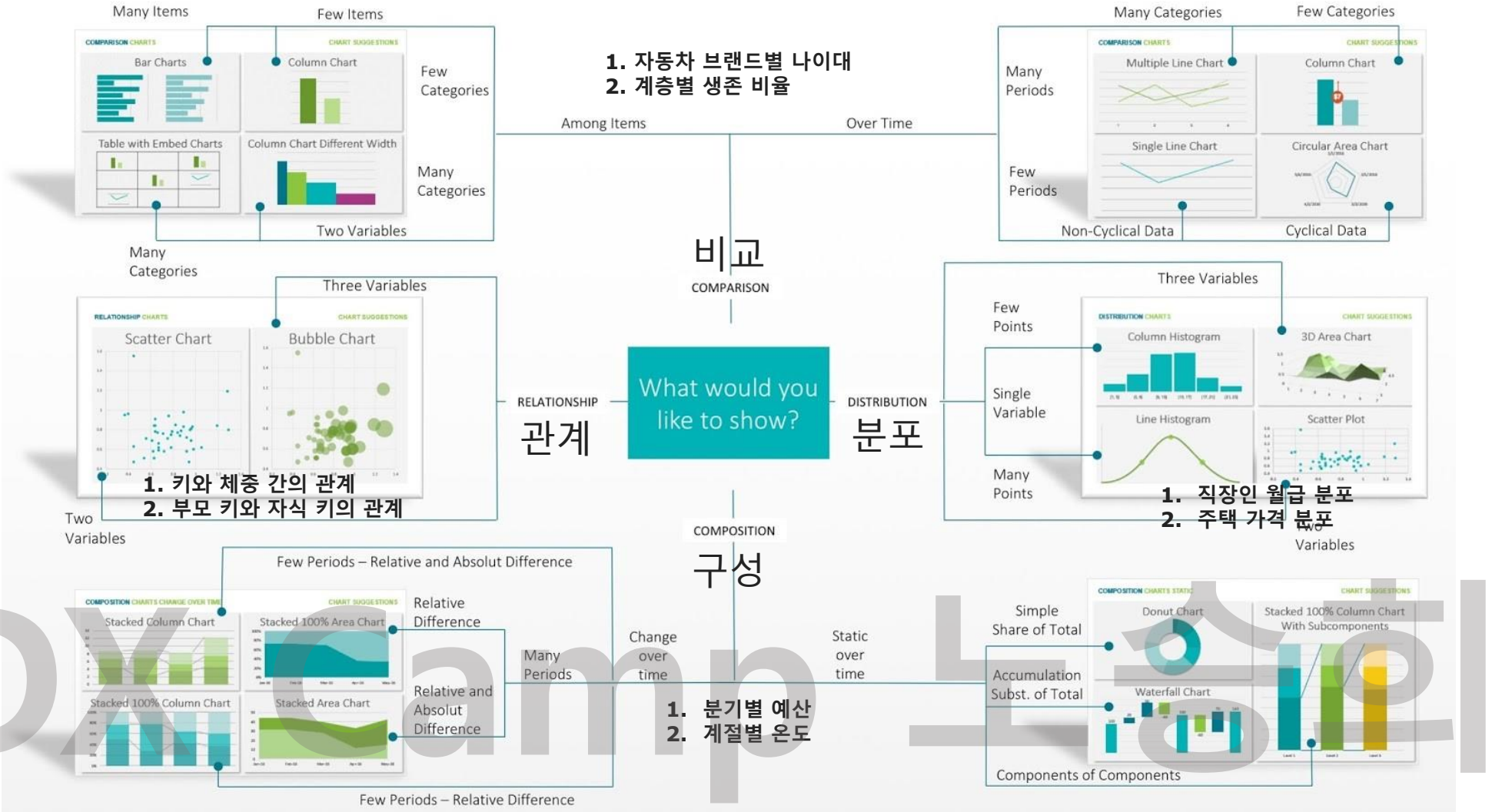
데이터 탐색

- Data.info()
- Data.describe()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

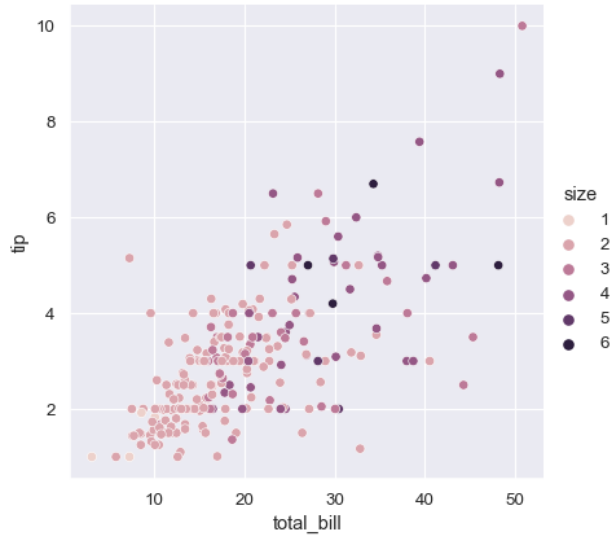
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

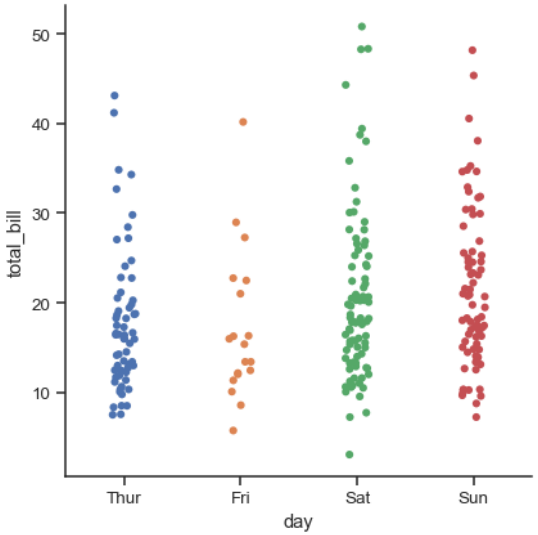
데이터 탐색을 위한 시각화

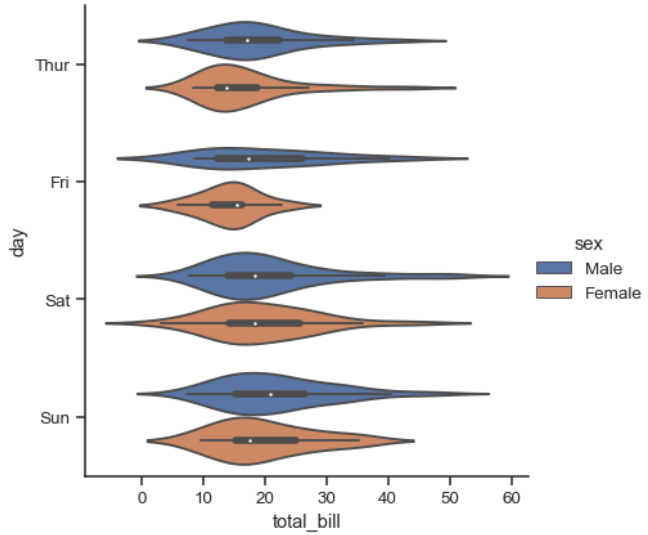


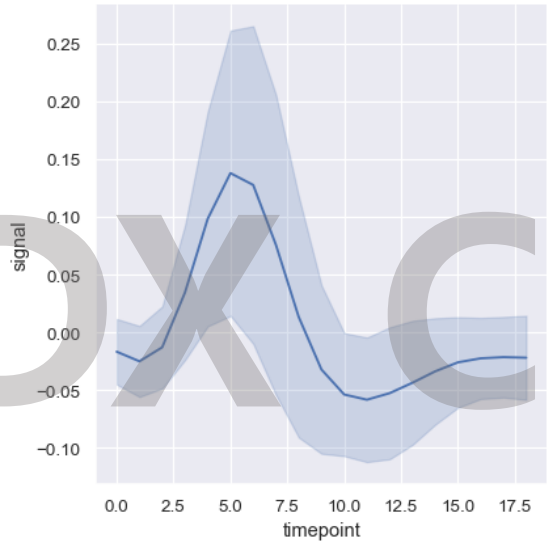
https://www.techprevue.com/decision-tree-perfect-visualisation-data/

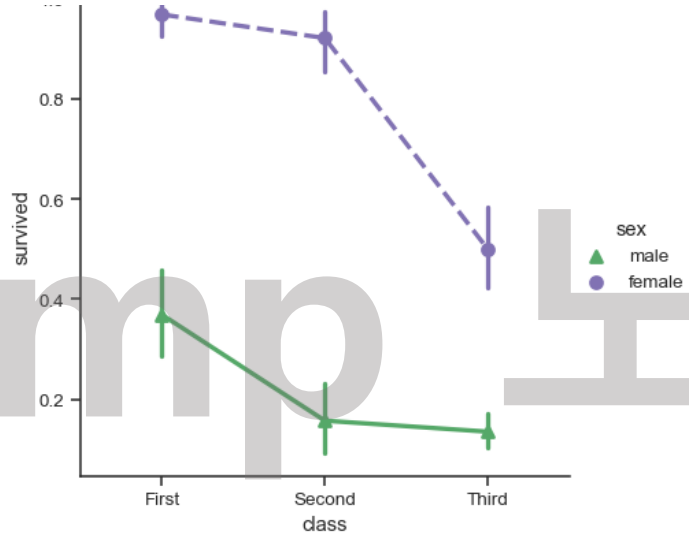
Seaborn 차트 시각화

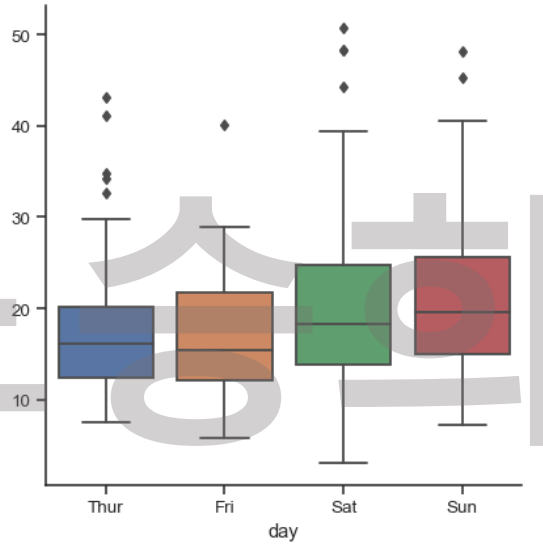






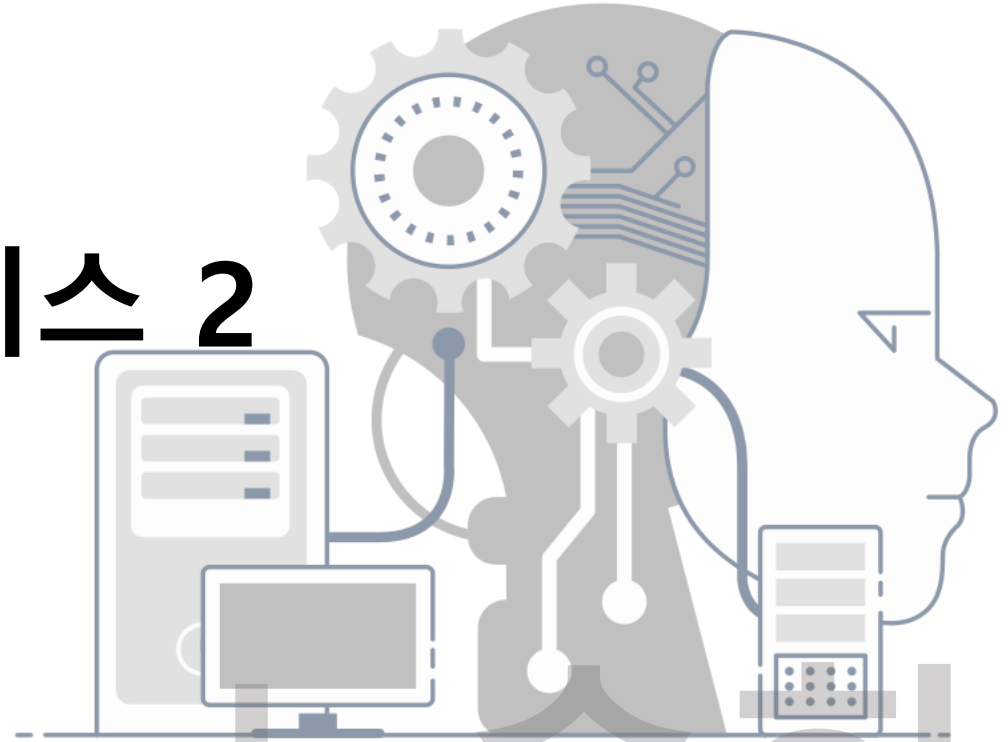




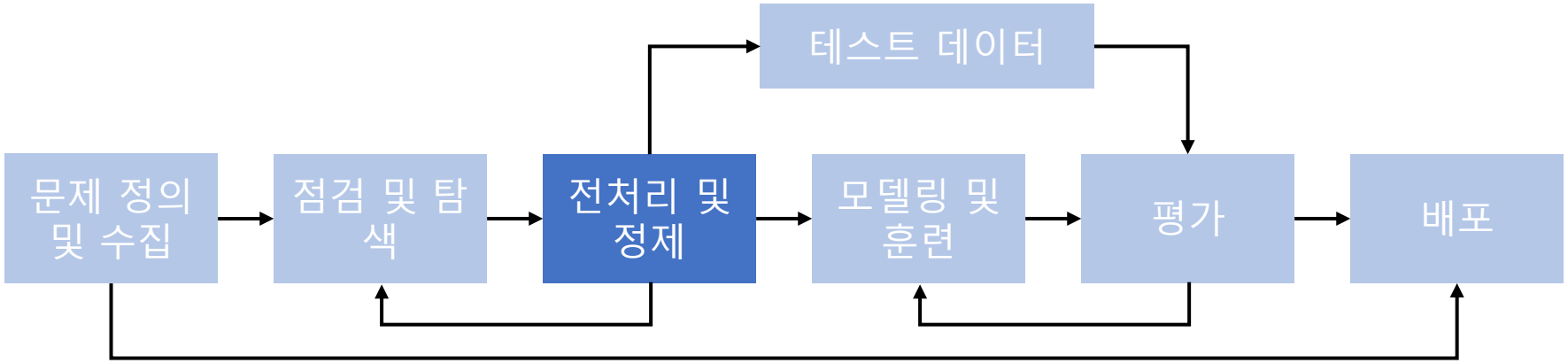


2022 DX Camp

3강 머신러닝 프로세스 2



3. 전처리 및 정제



- 머신 러닝 워크플로우에서 가장 까다로운 작업 중 하나이다
- 빠르고 정확한 데이터 전처리를 하기 위해서는 사용하고 있는 툴에 대한 다양한 라이브러리 지식이 필요
- 데이터의 결측치 및 이상치 확인, 제거, 일관성 있는 데이터의 형태로 전환하는 과정
- 전처리의 종류 : 데이터 클리닝(cleaning), 데이터 통합(integration) , 데이터 변환(transformation), 데이터 축소(reduction), 데이터 이산화(discretization) 등

데이터 전처리 단계

1) 데이터 정제

- 누락 데이터나 잡음, 모순된 데이터 등을 정합성이 맞도록 교정하는 작업

2) 데이터 통합

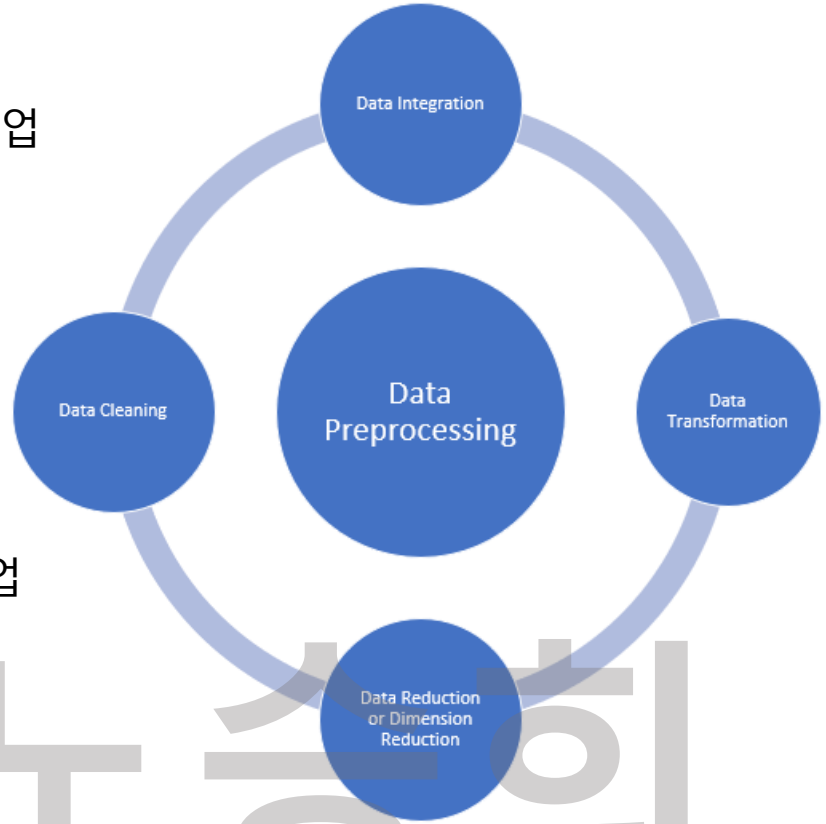
- 여러 개의 데이터베이스, 데이터집합 또는 파일을 통합하는 작업

3) 데이터 축소

- 샘플링, 차원축소, 특징 선택 및 추출을 통해 데이터 크기를 줄이는 작업

4) 데이터 변환

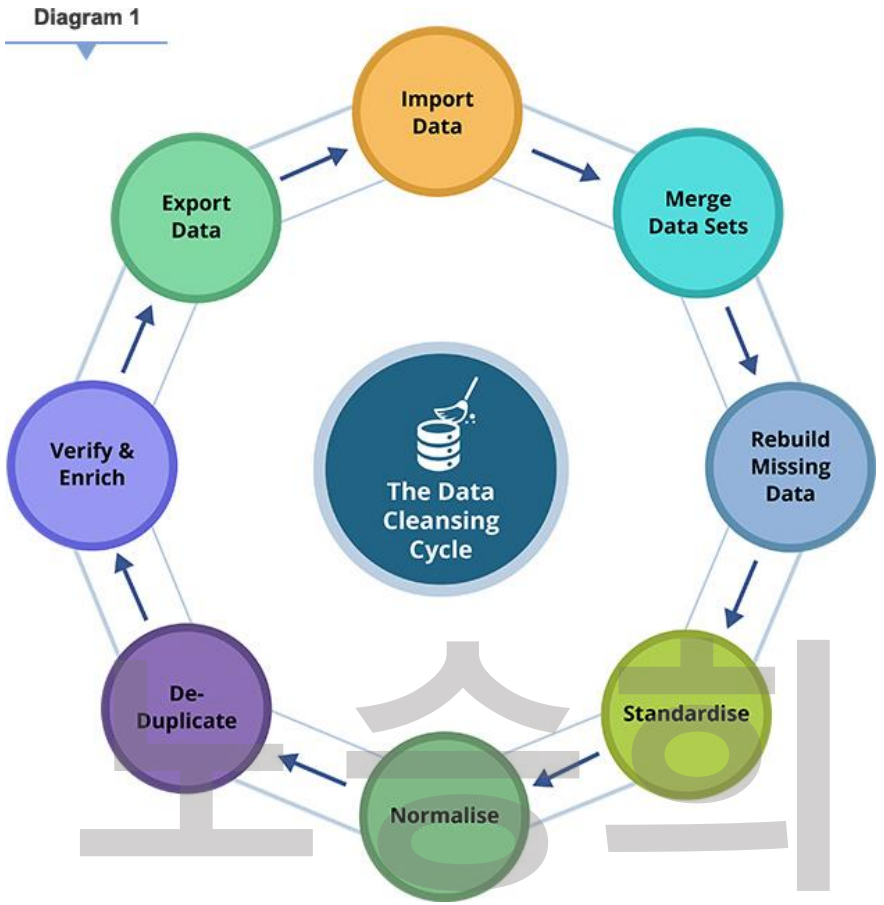
- 데이터를 정규화, 이산화 또는 집계를 통해 변환하는 작업



https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/

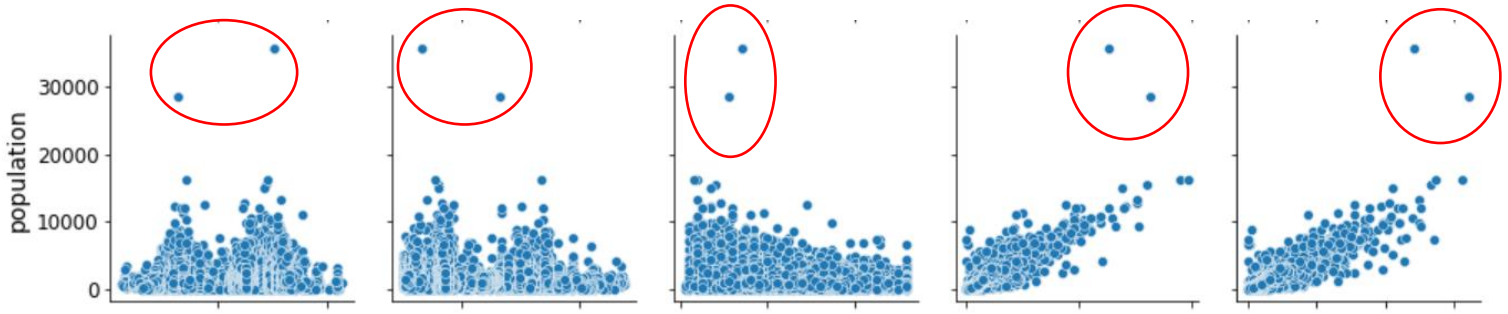
1) 데이터 정제

- 데이터를 활용할 수 있도록 만드는 과정
- 데이터의 누락, 불일치, 오류의 수정
- 컴퓨터가 읽을 수 없는 요소의 제거
- 숫자나 날짜 등의 형식에 대해 일관성 유지
- 적합한 파일 포맷으로 변환



https://www.dataentryoutsourced.com/blog/cxos-guide-to-marketing-and-sales-data-cleansing-and-enrichment/

- 이상치 처리



- 결측치 처리

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_inc
20046	-119.01	36.06	25.0	1505.0	NaN	1392.0	359.0	1.6
3024	-119.46	35.14	30.0	2943.0	NaN	1565.0	584.0	2.5
15663	-122.44	37.80	52.0	3830.0	NaN	1310.0	963.0	3.4
20484	-118.72	34.28	17.0	3051.0	NaN	1705.0	495.0	5.7
9814	-121.93	36.62	34.0	2351.0	NaN	1063.0	428.0	3.7

- 문자열 변환

ocean_proximity
INLAND
NEAR OCEAN
NEAR BAY
ISLAND

One Hot Encoding

ocean_proximity			
INLAND	NEAR OCEAN	NEAR BAY	ISLAND
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Label Encoding

ocean_proximity
1
2
3
4

- 단위 일치: 시간 단위, 거리 단위 ,화폐 단위 등

https://techblog-history-younghunjo1.tistory.com/99

2) 데이터 통합

- 서로 다른 출처의 여러 데이터를 결합
- 서로 다른 데이터 세트가 호환이 가능하도록 통합
- 같은 객체, 같은 단위나 좌표로 데이터를 통합
- 링크드 데이터의 핵심 목표 중 하나는 데이터 통합을 완전히 또는 거의 완전히 자동화하는 것

https://wadic.net/data-integration-importance/

3) 데이터 축소

- 대용량 데이터에 대한 복잡한 데이터 분석은 실행하기 어렵거나 불가능한 경우가 많음
- 데이터를 축소하면 데이터 분석 시 좀 더 효과적이고 원래 데이터와 거의 동일한 분석 결과를 얻어 낼 수 있는 장점이 있음
- 축소된 데이터도 원래 데이터와 같은 분석 결과를 얻을 수 있어야 함

  * 컴퓨팅 시간 등 고려 위해 데이터 축소가 필요, 방대한 로그 데이터의 경우 일정 시간 단위로 데이터 축소 필요

https://www.cohesity.com/blogs/cohesity-data-reduction-lock-stock-barrel/

데이터 축소 기법

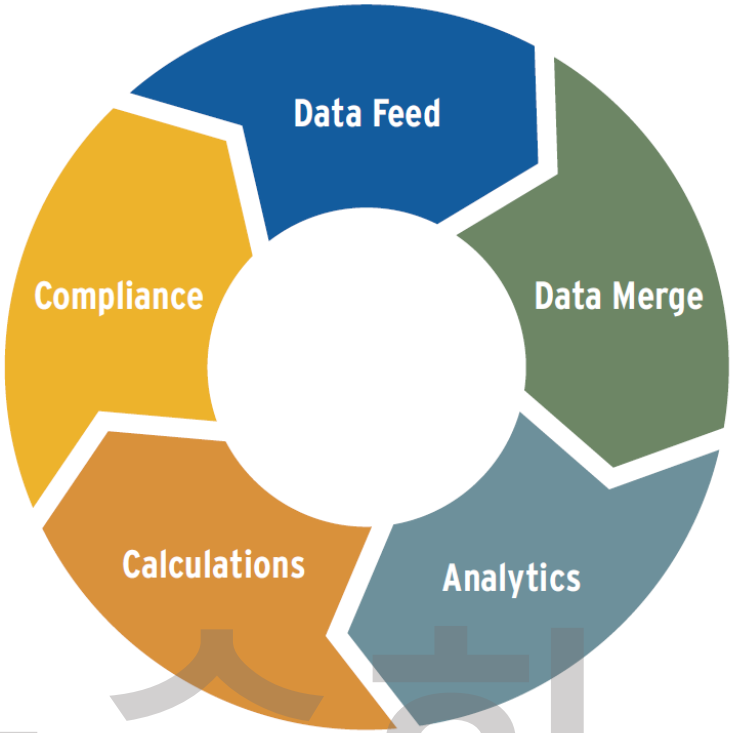
- 점진 선택법: 속성의 공집합에서 시작되며 원본 속성들 중에서 최선의 값이 결정되어 그 집합에 추가된다. 다음 단계에서는 남아 잇는 원본 속성들 중에서 최선의 속성이 집합에 추가된다.

- 후진 제거법: 속성들의 완전집합에서 시작하여 각 단계마다 그 집합에 남아있는 최악의 속성을 제거한다.

- 전진선택법과 후진제거법 결합: 위의 두 방법을 결합하며 각 단계마다 최선의 속성을 선택하고 남아 있는 속성들 중에서 최악의 속성 제거한다.

4) 데이터 변환

- 데이터를 한 형식이나 구조에서 다른 형식이나 구조로 변환
- 원본 데이터와 대상 데이터 간에 필요한 데이터 변경 내용을 기반으로 데이터 변환이 간단하거나 복잡할 수 있음
- 데이터 변환은 일반적으로 수동 및 자동 단계가 혼합되어 수행
- 데이터 변환에 사용되는 도구 및 기술은 변환되는 데이터의 형식, 구조, 복잡성 및 볼륨에 따라 크게 다를 수 있음
- 정규화, 집합화, 요약, 계층 생성



https://taxexecutive.org/data-transformation-where-to-begin/

스케일링

- **정규화(Min-Max Scaling)**

- Normalization(정규화)이라는 용어도 많이 쓰임.
- 아래 식과 같은 변환을 통해 값의 범위를 0~1로 제한 (=좁은 범위로 압축)

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- **표준화(Standardization)**

- 우리말로 표준화라고 부르며, Z-score라고도 함.
- 여기서, μ는 x의 평균이고, σ는 x의 표준편차

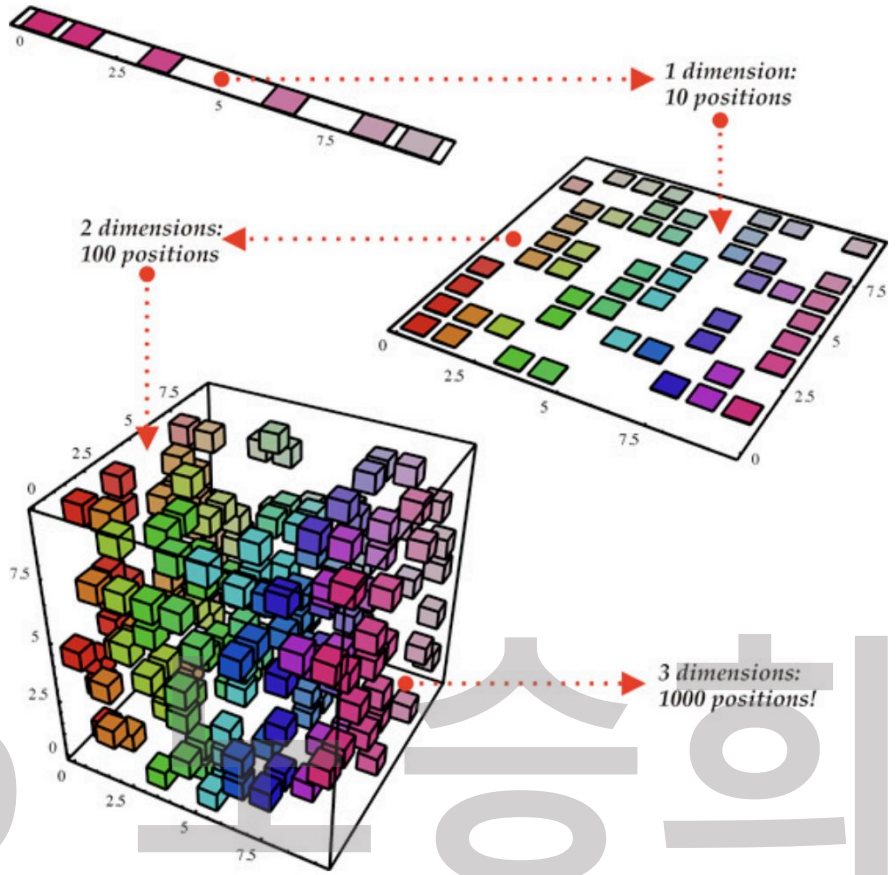
$$x_{scaled} = \frac{x - \mu}{\sigma}$$

특징 공학(Feature Engineering)

- 머신러닝 알고리즘을 작동하기 위해 데이터에 대한 도메인 지식을 활용하여 특징을 만들어내는 과정
- 머신러닝 모델을 위한 데이터 테이블의 컬럼(특징)을 생성하거나 선택하는 작업을 의미
- 모델의 성능을 높이기 위해 모델에 입력할 데이터를 만들기 위해 주어진 초기 데이터로부터 특징을 가공하고 생성하는 전체 과정을 의미
- Feature Engineering은 모델 성능에 미치는 영향이 크기 때문에 머신러닝 응용에 있어서 굉장히 중요한 단계이며, 전문성과 시간, 비용이 많이 드는 작업

차원의 저주

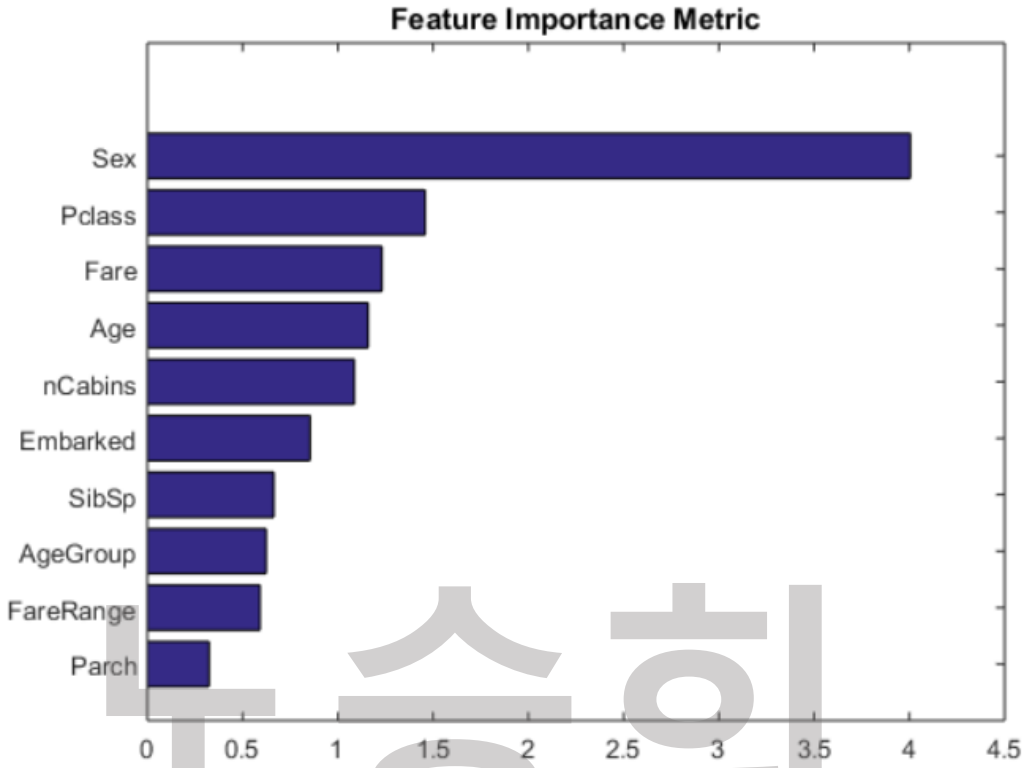
- 차원이 증가하면서 학습데이터 수가 차원 수보다 적어져서 성능이 저하되는 현상
- 차원이 증가할수록 변수가 증가하고, 개별 차원 내에서 학습할 데이터 수가 적어짐
- 변수가 증가한다고 반드시 차원의 저주가 발생하는 것은 아님
- 해결 방안: **특징 선택, 특징 추출**



KDD 2014 tutorial, Yoshua Bengio

특징 선택(Feature Selection)

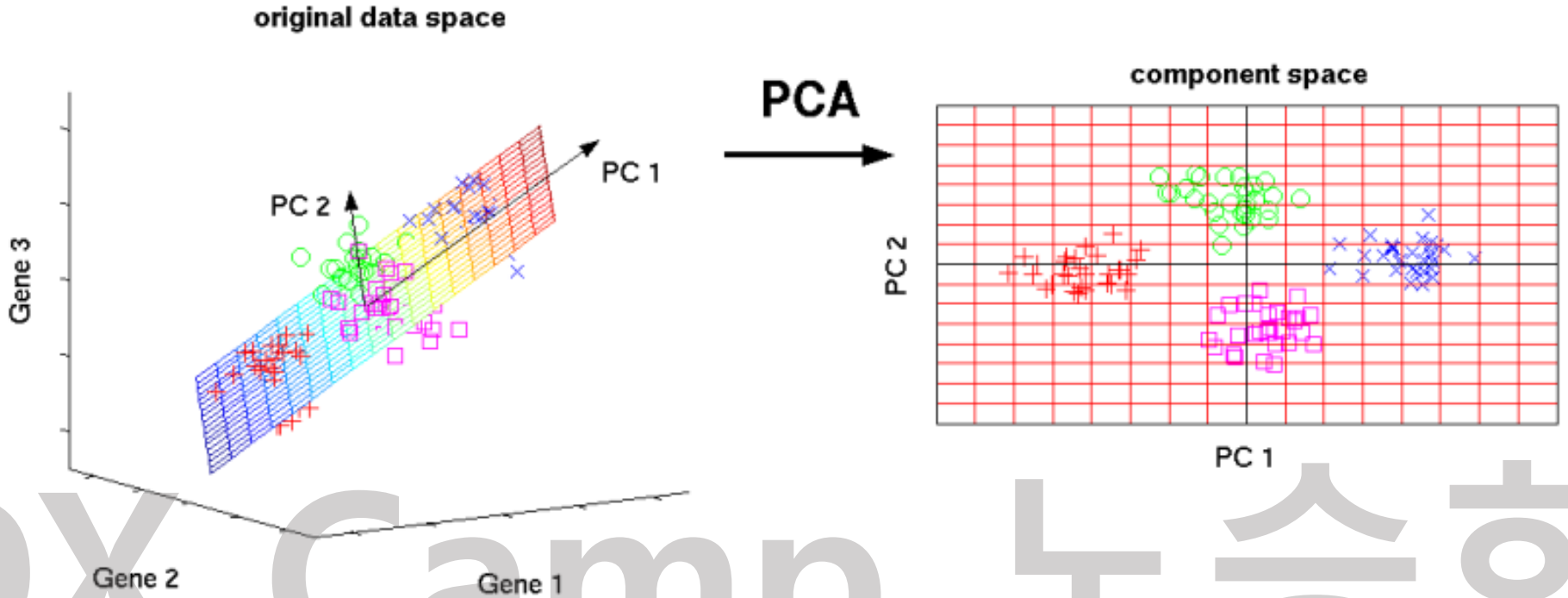
- 특징 랭킹(Feature Ranking) 또는 특징 중요도(Feature Importance)라고도 불림.
- 분류 모델 중 Decision Tree 같은 경우는 트리의 상단에 있을 수록 중요도가 높으므로 이를 반영하여 특징 별로 중요도를 매길 수 있음.
- 회귀 모델의 경우 forward selection과 backward elimination 같은 알고리즘을 통해 특징을 선택.



특징 추출(Feature Extraction)

- 특징 추출은 단순히 데이터의 압축이나 잡음을 제거하는 것이 아님
- 특징 추출을 통해 데이터의 압축이나 잡음을 제거하는 효과도 있지만, 이것의 가장 중요한 의의는 관측 데이터를 잘 설명할 수 있는 잠재 공간(latent space)을 찾는 것.
- 가장 대표적인 알고리즘으로 PCA(Principle Component Analysis)가 있음.
- PCA를 간단히 설명하면 각 변수(Feature)를 하나의 축으로 투영시켰을 때 분산이 가장 큰 축을 첫번째 주성분으로 선택하고 그 다음 큰 축을 두번째 주성분으로 선택하고 데이터를 선형 변환하여 다차원을 축소하는 방법.

주성분 분석(Principal Component Analysis)



특징 생성(Feature Generation)

- 특징 구축(Feature Construction)이라고도 하며, 이 방법을 많은 사람들이 Feature Engineering이라고 생각함
- 초기에 주어진 데이터로부터 모델링 성능을 높이는 새로운 특성을 만드는 과정
- 데이터에 대한 도메인(분야) 전문성을 바탕으로 데이터를 합치거나 쪼개는 등의 작업을 거쳐 새로운 Feature를 만들게 됨
- 간단한 예로 시간 데이터를 AM / PM 으로 나누는 것이 있음
- 이 작업은 한번해서 끝나는 것이 아니라 끊임없이 모델링 성능을 높이는 목적으로 반복해서 작업할 수 있는 부분이기 때문에 전문성과 경험에 따라 비용과 시간을 줄일 수 있는 부분

EDA vs 전처리

탐색적 데이터 분석(EDA, Exploratory Data Analysis)

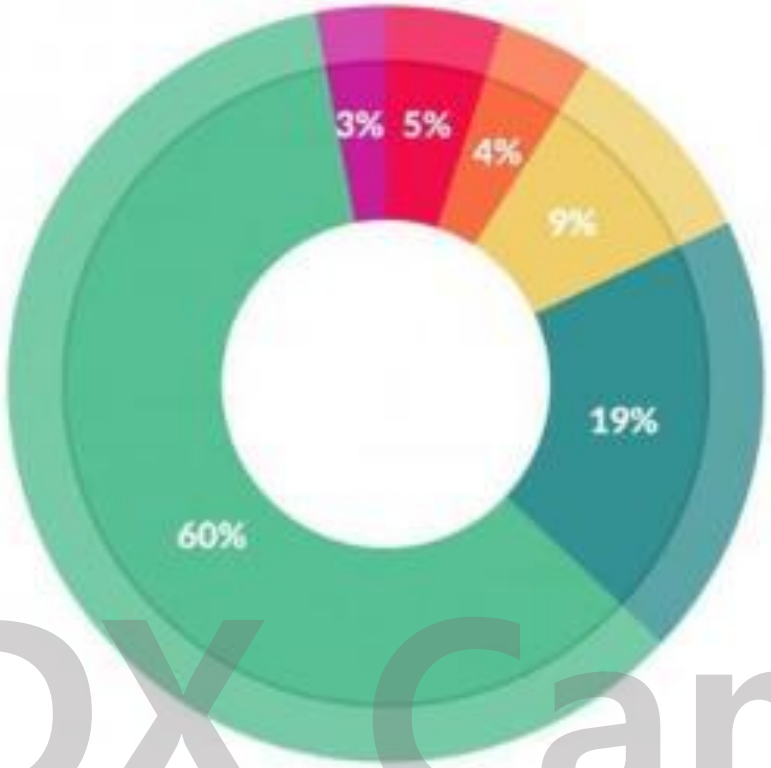
- 데이터 모델링에 들어가기 전에, 수집된 데이터의 분포, 관계를 **파악**하는 과정(히스토그램, 산포도, 기초 통계량 등으로 분포 파악)하고 모델링, 알고리즘에 사용될 수 있는 Data Set인지 파악함) 모델링 입력 데이터에 사용하기 부족하다면 다시 정제(Cleaning)

데이터 전처리(Data Preprocessing)

- 수집된 데이터에는 이상한(?), 극단적인(?), 공란의(?), 잡음(?)의 데이터가 섞여 있을 수 있음. 데이터의 가공 없이 데이터 분석 및 모델링을 하면 결과가 이상하게 나올 수 있음. 데이터를 분석에 사용할 수 있도록 **정제**(Cleaning)하고 가공하고 **변환**(Transformation)등을 거쳐서 모델링에 필요한 변수로 만드는 과정.

https://www.youtube.com/watch?v=k6OKKaBR-SU

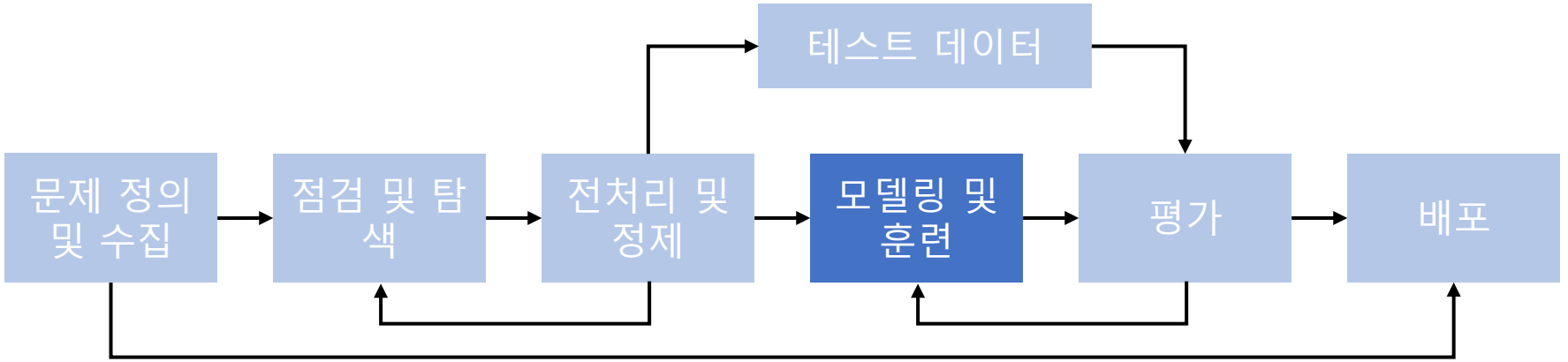
가장 시간이 많이 걸리는 작업



What data scientists spend the most time doing

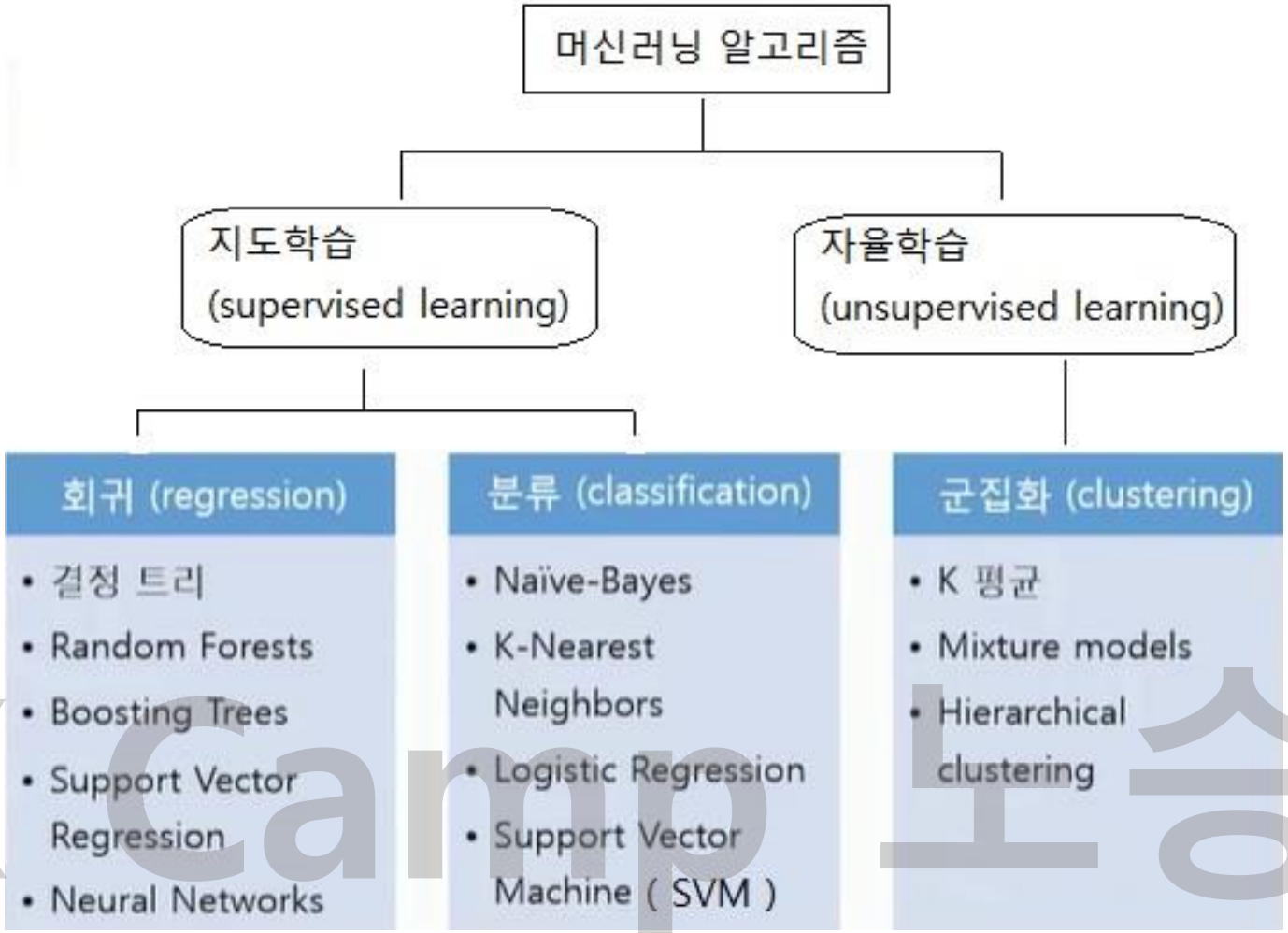
- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

4. 모델링 및 훈련



- 데이터에 적합한 머신러닝 모델을 선택 후 모델링
- 전처리가 완료된 데이터를 머신러닝 모델학습
- 학습 후 훈련이 제대로 되었다면 우리가 원하는 태스크(task)를 수행 가능
- 주의해야 할 점 : 데이터 훈련하기 전 훈련용, 테스트용 데이터를 나누어 모델 학습 (성능 테스트 필요)

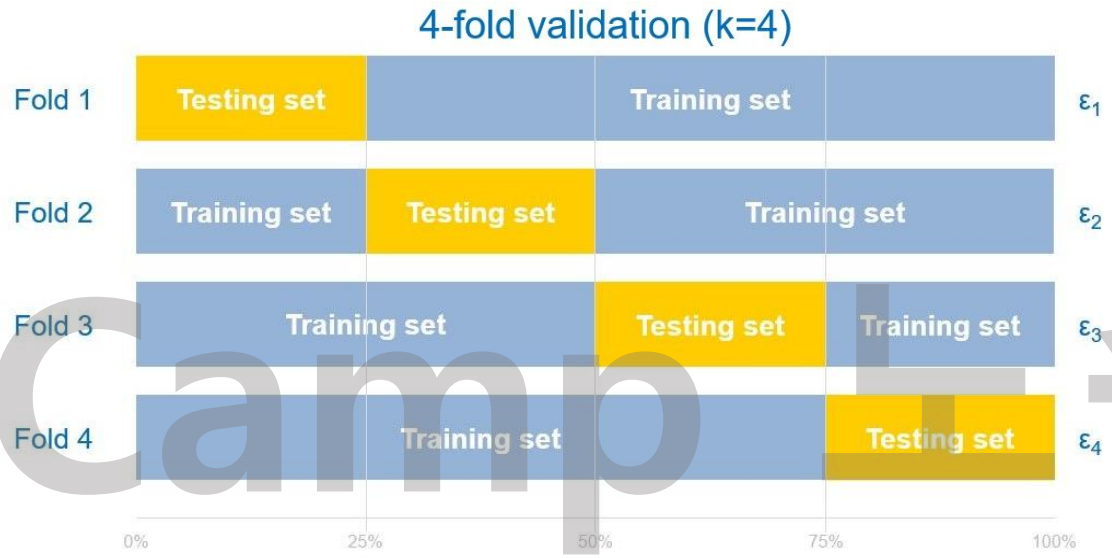
모델 선택



https://artificialnetworkforstarters.readthedocs.io/en/latest/_post/chap6.html

교차 검증

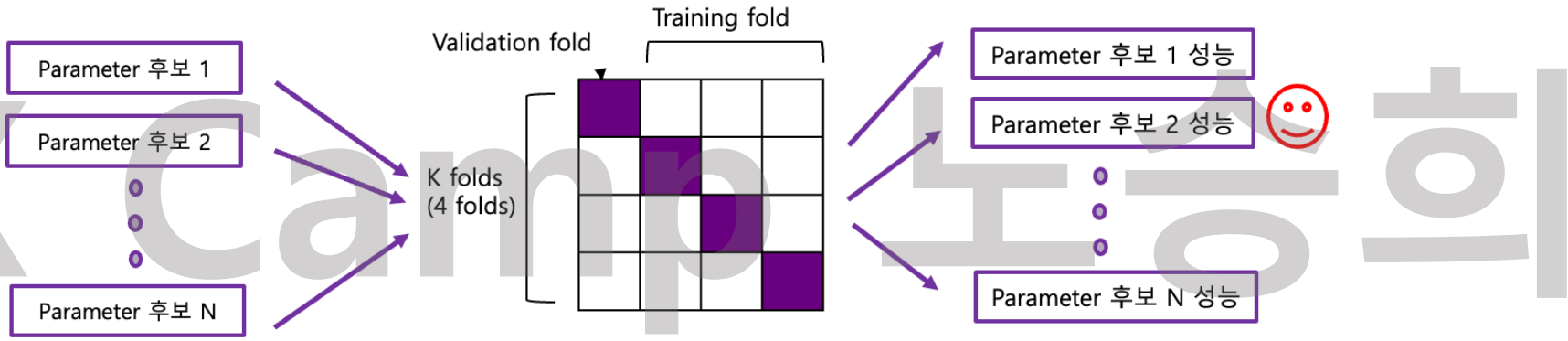
- 집합을 체계적으로 바꿔가면서 모든 데이터에 대해 모형의 성과를 측정하는 검증 방식.
- 교차 검증은 과적합을 피하면서 파라미터를 튜닝하고 일반적인 모델을 만들고 더 신뢰성 있는 모델 평가를 진행하기 위해.
- 교차 검증이란 쉽게 생각하면 본고사를 치르기 전 모의고사를 여러 번 보는 것



https://kr.mathworks.com/discovery/cross-validation.html

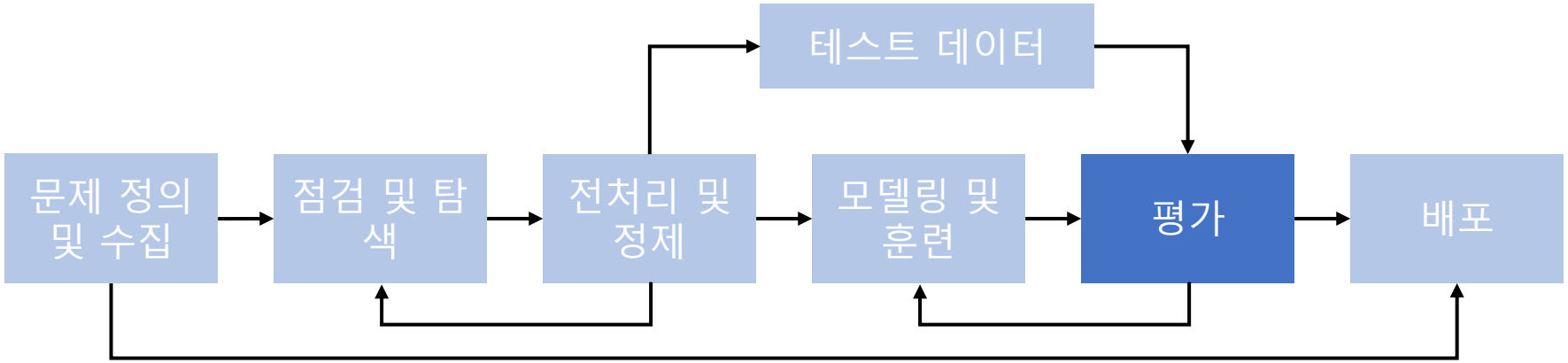
모델 최적화 - Grid Search

- 교차 검증을 기반으로 주어진 하이퍼 파라미터의 모든 조합 중 최적의 값을 찾아주는 탐색 방법.
- 모든 하이퍼 파라미터 조합을 순차적으로 적용 및 검증.



https://dacon.io/codeshare/4922

5. 평가

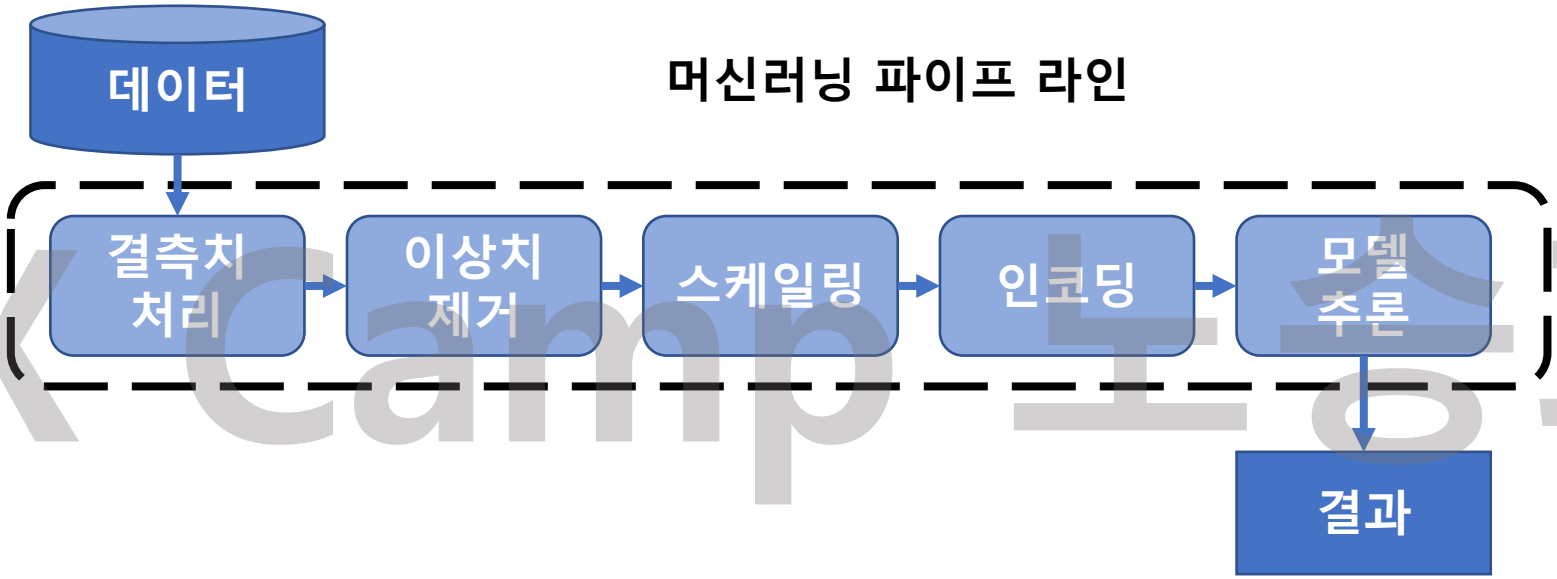


- 학습이 된 모델을 테스트용 데이터로 성능 평가하는 과정
- 기계가 예측한 데이터가 테스트용 데이터의 실제 정답과 얼마나 가까운지를 측정

파이프라인 구축

- 전처리 과정에서 인코딩, 결측치 처리, 훈련, 테스트 집합으로 분할 등 필수로 해야 하는 것을 파이프라인을 이용하여 자동화 할 수 있다.

- 파이프라인은 데이터의 변환을 순차적으로 적용한 다음 학습을 할 수 있다. 파이프라인을 사용하면 직관적으로 흐름을 읽을 수 있으며 순서가 파이프라인에 의해 정해지고 만들기 간편하다.



범주형 데이터 – 혼동 행렬(Confusion Matrix)

- 100명의 환자와 100 명의 건강한 사람에 대하여 이 검사 키트의 성능을 테스트한다고 가정
- 검사 키트가 COVID-19 환자(양성positive:P)에 대해서 5명을 음성
- COVID-19에 감염되지 않은 건강한 사람(음성negative:N)에 대해서 89명을 음성으로, 11명을 양성으로 판정

		KoKIT22의 예측값 (검사결과)					
		음성			양성		
환자의 실제 상태값		N			P		
음성 (COVID 안걸림)	N	89 TN	T 일치	N 예측	11 FP	F 불일치	P 예측
양성 (COVID 걸림)	P	5 FN	F 불일치	N 예측	95 TP	T 일치	P 예측

- 정확도: 전체 데이터(FP+FN+TP+TN)중에서 제대로 판정한 데이터(TP + TN)의 비율

$$Acc = \frac{TP+TN}{FP+FN+TP+TN} = \frac{95+89}{11+5+95+89} = 0.92$$

- 민감도: 실재 확진된 사람 중 검사 키트가 확진자로 분류한 비율

$$TPR = Rec = \frac{TP}{P} = \frac{TP}{FN+TP} = \frac{95}{100} = 0.95$$

- 정밀도: 검사 키트가 확진자로 분류한 사람들 중 실제 양성의 비율

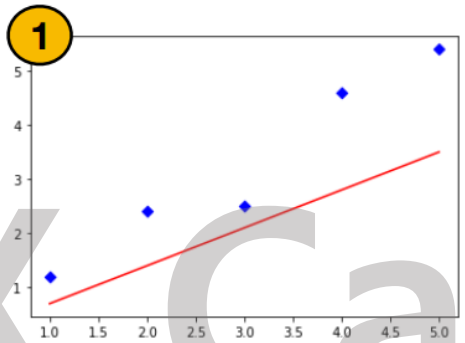
$$Pre = \frac{TP}{TP+FP} = \frac{95}{106} = 0.896$$

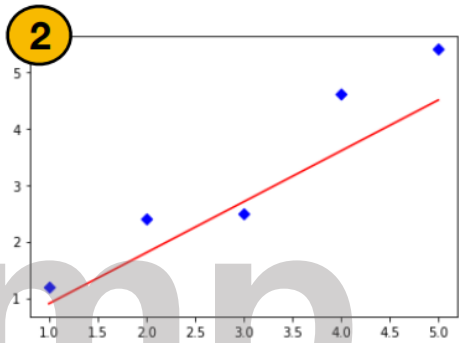
연속형 데이터 – MSE, MAE

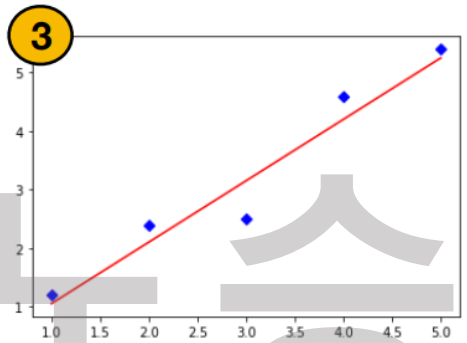
- 어떤 데이터를 추정하는 가설이 얼마나 정확한지를 평가하는 방법은 무엇일까?
- 가설이 훌륭한 모델이라면 데이터는 가설이 나타내는 직선 위에 모두 놓이게 될 것이다.
- **적당히 좋은 가설이라면 데이터가 이 직선들 근처에 있을 것**이다.
- 차트를 그려서 확인하는 것은 정확도를 시각적으로 표시할 수 있는 장점은 있지만, 서로 다른 가설을 비교할 때 정확한 척도로 사용하기는 힘들다.
- 우선 다음 그림과 같이 (1, 1.2), (2, 2.4), (3, 2.5), (4, 4.6), (5, 5.4)의 5개의 데이터가 이차원 공간에 분포하고, 이 데이터의 분포를 설명하는 선형방정식을 1) $y = 0.7x$라고 추정하도록 하자.

연속형 데이터 – MSE, MAE

- 이제 데이터의 분포를 파란색 점으로, 선형방정식을 빨간색 직선으로 그려보면 그림과 같이 나타나서 이 방정식이 데이터의 분포를 잘 설명하지 못한다는 것을 알 수 있을 것이다.
- 이제 2) y=0.9x 로 선형방정식을 사용할 경우 데이터의 분포를 이전보다 더 정확하게 설명하는 직선을 얻을 수 있을 것이며, 3) y=1.05 라는 선형방정식은 더욱 더 나은 결과를 보여주는 것을 눈으로 확인할 수 있을 것이다.







- '더 나은' 선형방정식(혹은 '더 좋은' 가설)은 너무나 주관적인 표현이므로 이를 정량화하는 것이 필요한데 이때 사용되는 것이 바로 오차함수이다.

- 오차의 합을 그대로 사용하지 않고 별도의 오차함수를 사용하는 이유는 실제값이 {1, 2, 3}이고 예측값이 {1, 4, 1}로 나타날 경우 (1-1) + (2-4) + (3-1) = 0이 되는 경우가 발생하기 때문이다.

평균 절대 오차(Mean Absolute Error)

- 머신러닝에서 사용 가능한 오차함수 중에서 비교적 단순한 오차함수로 예측값 $\hat{y}$과 관측값 y의 차이 값의 절대값을 구한 후 이 값들의 평균값을 사용한다.
- 이 오차함수는 오차값을 그대로 보여주는 특징이 있으며 다음과 같이 정의할 수 있다.

$$E_{mae} = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|$$

- 평균 절대 오차는 직관적이며 계산이 편리한 반면 다음의 문제가 있다.
- 첫째, 축적을 보정하지 않기 때문에 앞의 값의 10배에 해당하는 (10, 12), (20, 24), (30, 25), (40, 46), (50, 54) 값에 대해서 동일한 10%오차가 발생하더라도 10배의 차이가 나는 문제가 있다.
- 둘째, 절대값의 사용으로 인해 미분이 불가능한 지점이 발생한다는 문제가 있다.

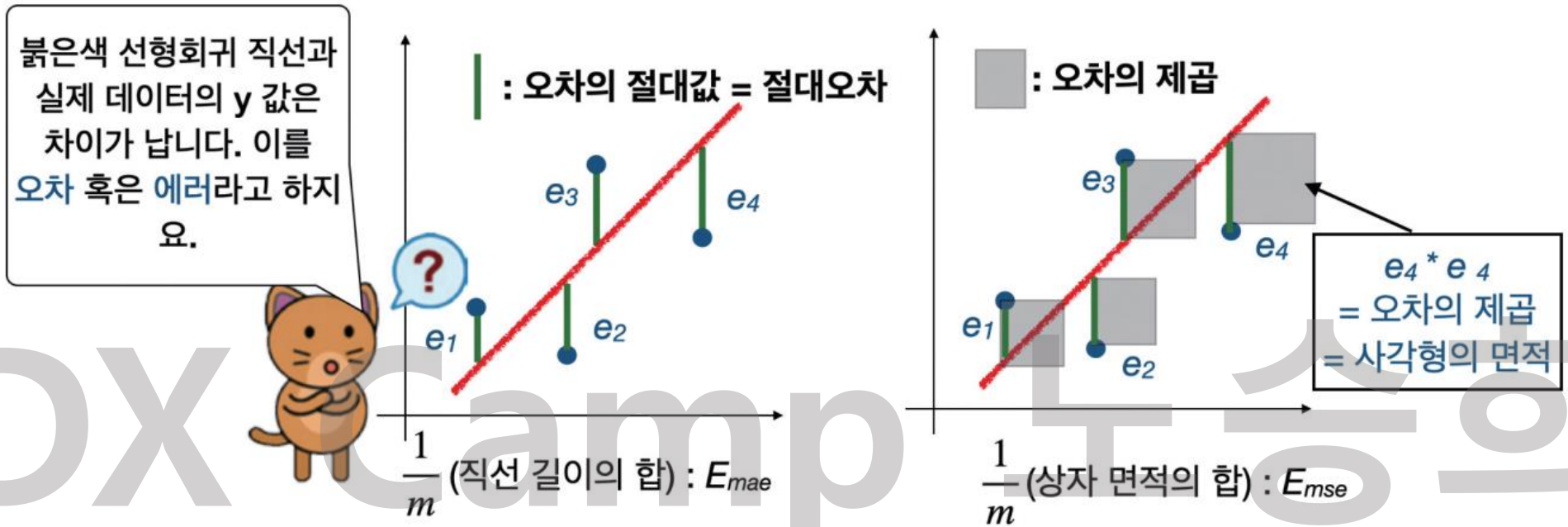
평균 제곱 오차(Mean Square Error)

- 머신러닝에서 사용하는 대표적인 오차 척도는 **평균 제곱 오차**이다.
- 이 방법은 예측치 $\hat{y}$ 와 정답 레이블 y 사이의 차이를 제곱하여 모두 더한 뒤에 전체 데이터의 개수 m 으로 나누는 것인데, 다음과 같은 식으로 표현할 수 있다.

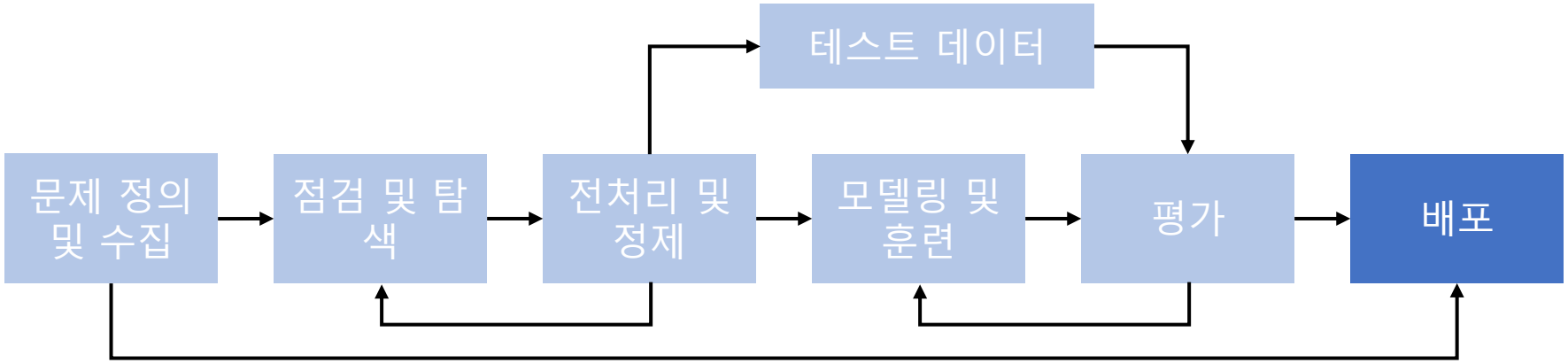
$$E_{mse} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

- 머신러닝의 문제를 해결하는 데 **주로 사용되는 오차는 평균 제곱 오차**로 우리는 이 오차 측정 방법이 왜 유용한지 집중적으로 살펴볼 것이다.

- 아래 그림을 보면, 파란색 점으로 표시된 레이블과 붉은색 가설 직선의 y 값은 차이가 난다.
- 이를 오차라고 하는데, e1에서 e4까지 전체 에러의 합이 최소가 되는 모델이 가장 바람직한 모델이 될 것이며 우리는 이 직선을 찾는 것이다.



6. 배포



- 평가 단계에서 기계가 성공적으로 훈련이 된 것으로 판단된다면 완성된 모델이 배포
- 다만, 여기서 완성된 모델에 대한 전체적인 피드백으로 인해 모델을 업데이트 해야 하는 상황이 온다면 수집 단계로 돌아감.

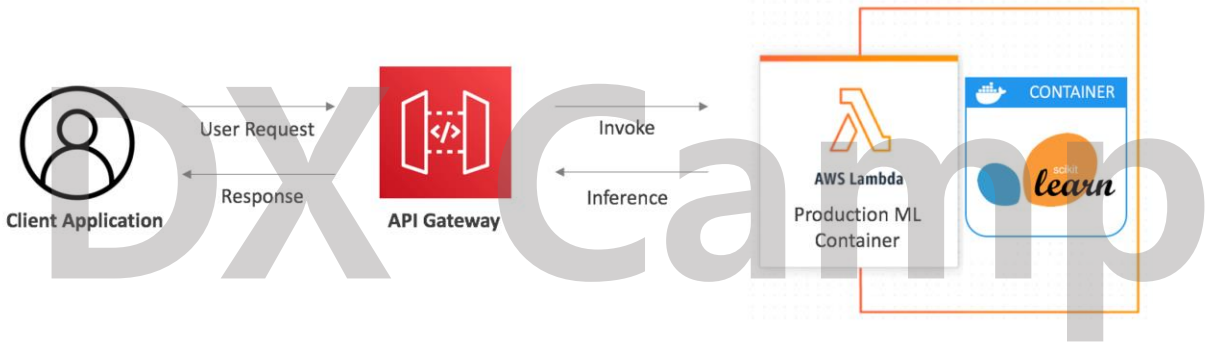
학습된 모델 배포

Server

FastAPI

docker

scikit learn



Client

TensorFlow Lite

TensorFlow.js

https://engineering.rappi.com/using-fastapi-to-deploy-machine-learning-models-cd5ed7219ea

캘리포니아 주택 가격 예측 코드 실습

- 머신러닝 프로젝트 처음부터 끝까지 실습하기(어려운 버전)