

# **DSAC Module1**

## **(Data Programming)**

**2019.5**

KPC(한국생산성본부)

Title	Contents	Labs (gg-?)
1. 데이터 사이언스 (Data Science)	• 데이터 사이언스 산업	
2. 데이터 사이언스 도구 (Data Science Tools)	• Python                      • 프로그램 개발환경	
3. Python 기초	• 기본변수                      • 랜덤숫자 • Numpy	1,2
4. Pandas	• DataFrame                      • 그래프 (Graphs) • (ex) DataFrame 연습	3,4,5
5. 데이터 처리	• 데이터처리 기본                      • 파일 다루기 • 데이터 정제                      • (ex) 항공운항 데이터 • (ex) 날짜 데이터	6,7,8
6. 웹 크롤링 (WEB Crawling)	• 웹 크롤링                      • (ex) 기상청 데이터읽기 • (ex) 스크래핑                      • (ex) 부동산정보 읽기 • (ex) 국제가격동향                      • (ex) 웹지도 데이터	9,10,11,12,13
7. 베이지스 알고리즘 (Bayes Algorithm)	• 베이지스 이론                      • (ex) 병원 예약 부도율 • (ex) 이름으로 성별 예측                      • (ex) 스팸 메시지 예측	14,15,16

# **Probability and Statistics review**

# Linear Algebra review

# 4차 산업혁명

5

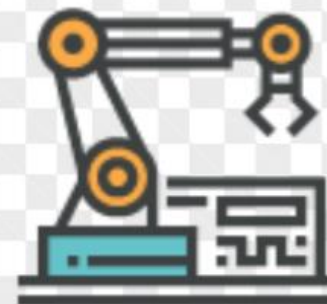
- 1, 2차 산업혁명
  - 에너지 생산과 에너지 전달의 혁명
  - 인간 **근육 노동력**의 한계
- 3, 4차 산업혁명
  - 정보(IT) 기술 혁명
  - 인간 **지적 노동력**의 한계



1st Industrial Revolution  
WATER & STEAM



2nd Industrial Revolution  
ELECTRICITY



3rd Industrial Revolution  
AUTOMATION



4th Industrial Revolution  
CYBER-PHYSICAL SYSTEMS

- 대량의 숫자 데이터 분석 – 빅데이터 분석
- 이미지 인식 – 보는 능력
- 음성인식, 텍스트 인식 – 듣는 능력
- 실시간 번역 – 말하는 능력
- 이미지 캡션, 언어 모델링 – 쓰는 능력
- 감성 능력까지 ?



# 빅데이터(AI) 도입 사례

# 자동차 보험사

8

- 보험사는 교통사고를 낼 확률이 높은 사람을 찾아 이들의 보험료를 올리고 이들이 다른 회사로 가게 한다





# 응급환자 예측

- 응급환자를 미리 찾아내어 구급차 이용을 줄인다



- 개인별 맞춤형으로 마케팅을 하기 위해서 고객의 과거 상품 구매 이력, SNS 분석
  - 고객이 매장을 둘러보는 도중에 고객의 성향과 현재의 욕구를 파악하여 실시간 추천
- 새로운 영화가 소개되면 얼마나 투자를 해야할지 또는 몇 개의 개봉관에서 이 영화를 상영하는 것이 적절할지를 예측
  - 영화 시나리오 대본, 주인공 분석, 예고편에 대한 고객 반응을 분석



- 고객 성향에 맞게 잘 대응할 적합한 직원을 배정
- 고객의 과거 이용 기록, 목소리 감성분석
  - 까다로운 고객인지, 또는 지금 무슨 용건으로 전화를 했는지를 예측
  - 예를 들어 평소에 카드 납입액을 자주 물어본 고객이라면 기다리는 동안 미리 카드 납입액을 자동으로 알려준다.
  - 고객이 평소에 자주 궁금해 하는 정보를 바로 자동응답기로 안내

- 응답 최적 루트
- 콜 요청량 예측
- 다음 행동(질의) 추정
- 실시간 고객 감성 분석
- 실시간 응대 제안 (상급 응대자 판정)
- 스크립트 최적화
- Script optimization
- 음성 인식 (상습 불법 신고자 파악)

- 인재 필요 역량 예측
- 작업량 예측
- 인재 유출 예측 (인재의 가치 평가, Talent Analytics)

- 고객 세분화 (행동 세분화, 구매 패턴 분석)
- 고객 재방문 예측
- 고객 이탈 예측
- 개인 맞춤형 광고
- 크로스셀링, 업셀링 아이템 추천
- 소셜 데이터 분석
- 고객 불만 분석
- 매출 예측
- 수요 예측

- 개인이나 기업에 대출을 위한 신용 평가
- 과거의 대출 사고 사례, 개인의 금융 활동을 분석
- 도난된 신용 카드 사용을 찾아내기 위해서
  - 평소 정상적인 거래를 분석하고 이상 현상을 발견
- 보험사는 사기성 보험 청구를 찾거나 불법자금 세탁을 찾아내는데 사용
- 자동차 보험회사에서는 차량에 블랙박스를 달면 보험료를 감해 준다.
  - 블랙박스 데이터를 보험사로 보내주면, 예를 들어 급정거를 하는지, 급커브를 하는지 등의 정보를 제공하면 보험료를 더 감해준다.
  - 고객의 안전한 운전을 유도하며 동시에 사고를 줄일 방법을 찾는다.

- 생산 라인의 최적화, 상품의 물류와 재고 관리의 정보화에 추가로
- 이 과정에서 발생하는 데이터를 상세히 분석함으로써 더 효과적인 생산, 유통, 재고관리
  - 제품 자체의 품질 향상에도 이용하고 있다.



- 에너지의 역할
  - 에너지는 경제, 안전, 산업의 기반
  - 기후변화, 재해, 가난에 대응
  - clean, affordable, reliable 에너지 필요
- AI가 에너지 생산, 전송, 분배, 사용 전분야의 지능화 수단으로 도입
  - 에너지 생산 방법의 다양화
  - 에너지 사용의 효율화

- 전력 수요 예측
- 신재생 에너지의 불규칙한 공급 예측
- 도전 감시
- 정전 예측 (시점 및 기간)
- 전기자동차의 충전 시간, 장소, 용량 예측
- AMI 분석을 통한 에너지 이용 패턴 분석
- 에너지 다변화 대응 (Peaker plant 운영)

- 피크 타임은 시기, 장소에 따라 다르다
  - 일반적으로 저녁시간(가전제품 사용),
  - 더운날은 오후에(에어컨),
  - 추운날은 아침에(히터),
- Peaker로는 개스터빈, 화력발전이 주로 사용된다.
  - 기반 전력생산비보다 높다.

- 양방향 통신, 제어, 지능형 컴퓨팅으로 구현
- 주요 구성 장치
  - Phasor Measurement Units (PMUs) – 그리드 안정성 센서
  - Digital Meters – 정밀한 전력사용 측정
  - Relays – 장애 우회 릴레이
  - Automated feeder switches
  - Battery – 에너지 저장 장치
- 애니메이션  
<https://goo.gl/c6zEq9>

- 에너지 센서 데이터를 분석하여 에너지 자원의 적절한 배분
- 딥러닝 기술 도입
  - 기존의 모델 중심의 머신러닝에서, 빅데이터 중심의 머신러닝으로 진화
- 정교한 예측 및 관리
  - ▶ 소지역 중심의 관리, 축전 장치의 활용
- 에너지 절감
  - ▶ 구글 데이터 센터의 에너지 절약 (cooling)



- “돌이 다 떨어져서 석기시대가 끝난 것이 아니다 (yamani)”



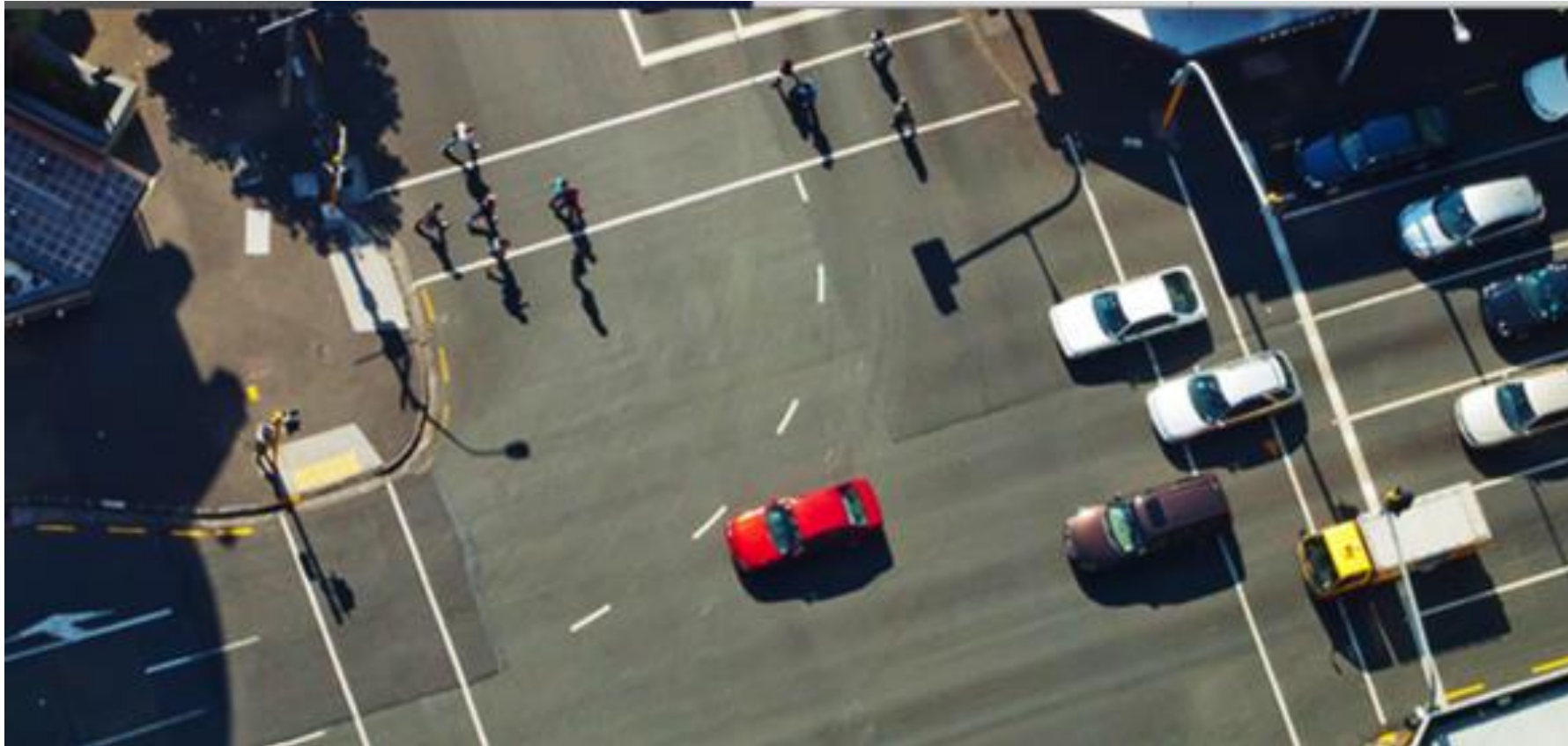
- 자율주행차의 상용화 대비 (기술적으로는 이미 완비)
- 부정적인 면
  - 새로운 물체나 상황 인식이 어렵다
  - 공사현장, 수신호 등을 파악하기 어렵다
  - 돌발상황에 실시간으로 대응
- 긍정적인 면
  - 버스, 트럭부터 운행한다 (길을 잘 아는 상황)
  - 360도 관찰이 가능하다
  - 기술 발전이 빠르다



# 적응형 신호등

24

- 대기시간 10~50% 단축





# Self-Driving Car Timeline (18/7)

25

**Ford – True Self-Driving by 2021**

**Volvo – Self-Driving on the Highway by 2021**

**BMW – Fully self-driving possible by 2021**

**Honda – Self-Driving on the Highway by 2020**

**Toyota – Self-Driving on the Highway by 2020**

**Hyundai – Highway 2020, Urban Driving 2030**

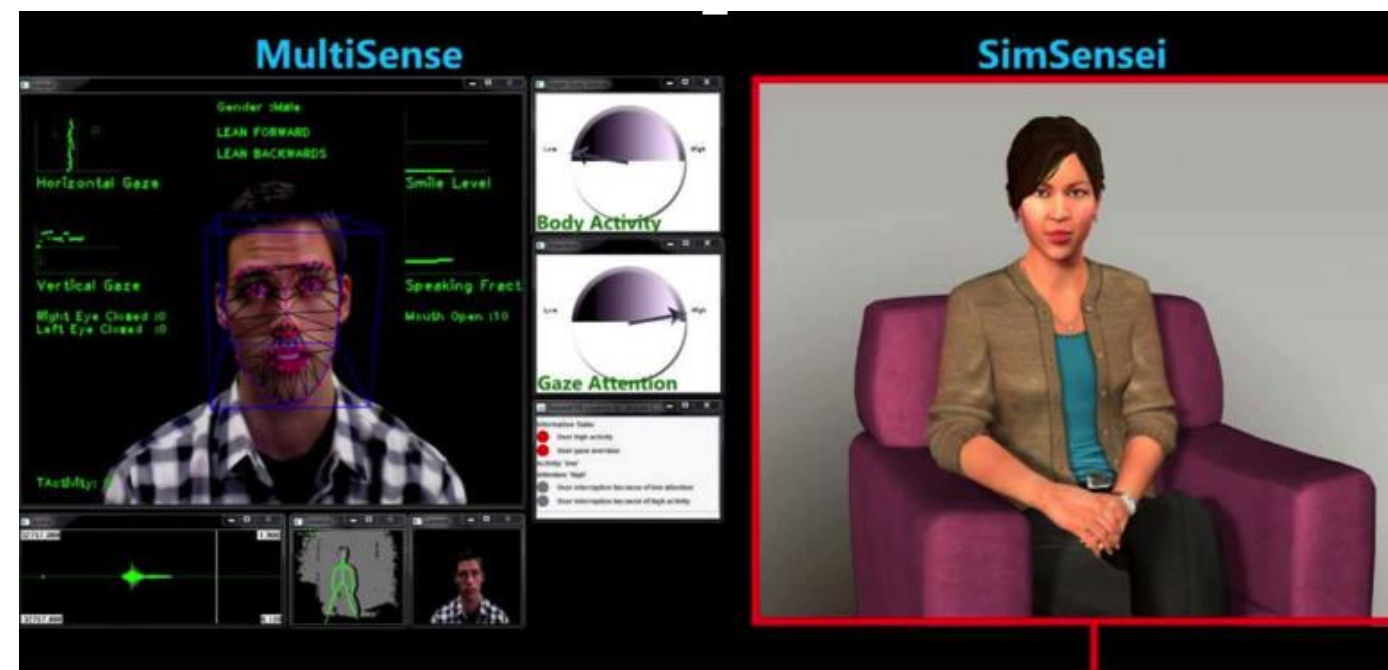
- 혈압, 심장박동수, 수면 패턴, 운동 패턴과 같은 개인 데이터를 종합적으로 분석
- 개인의 DNA 염기서열, 유전자 발현 분석을 통해서 개인별 질병 예측과 치료방법 제시도 상당히 정확해질 것
- 개인간의 약 복용에 대한 효과 차이를 구분할 수 있다면 맞춤형 처방 또는 맞춤형 신약개발이 가능

- 현재 가이드라인
  - 나이, 콜레스테롤 수치, 혈압, 흡연, 당뇨 등에 기반
- 영국 노팅햄대학교 연구
  - 37만명의 EMR 분석
- 실제로는
  - 인종차이, 정신질환, 경구용 스테로이드 복용 등이 큰 영향

- 존슨앤존슨의 자동 수면 마취기
  - 마취약 자동 주사
  - 심박수, 산소포화도, 심전도, 혈압을 보고 투약량 조절
  - 수면 내시경 의료비를 1/10로 낮춤
  - 더 빠르게 회복,
  - 저산소증문제도 적게 겪음
- 2016년 시장에서 철수
  - 의사들의 반발



- 심리 상담:
  - 인간 의사보다 AI의사에게 더 솔직하게 상담
- 서던 캘리포니아대에서 개발
  - 가상의 여성 상담사
  - 환자의 시선, 미소, 머리 움직임, 표정 분석



- 목소리로 감성 분석 (Beyond Verbal)
- 앱 Moodies
  - 언어적 요소 linguistics 뿐 아니라 비언어적 요소 acoudtics도 분석
- 조현병, 자살위험군 예측
- 우울증: 6초간 목소리 분석으로 AUC 0.93 (PureTech Health)
- 조울증 90% 진단

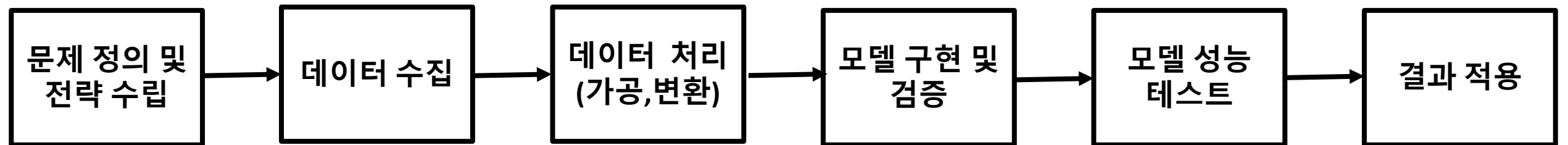
# AI 우선 도입 분야 예 (의료)

31

- 판독에 많은 시간과 노력이 드는 분야
- 문제의 난이도와 위험도가 적은 분야
- 의사들이 하기 싫어 하는 분야
- 보험 수가가 낮은 분야
- 비용감소와 효율이 향상되는 분야

# AI 프로젝트 전략



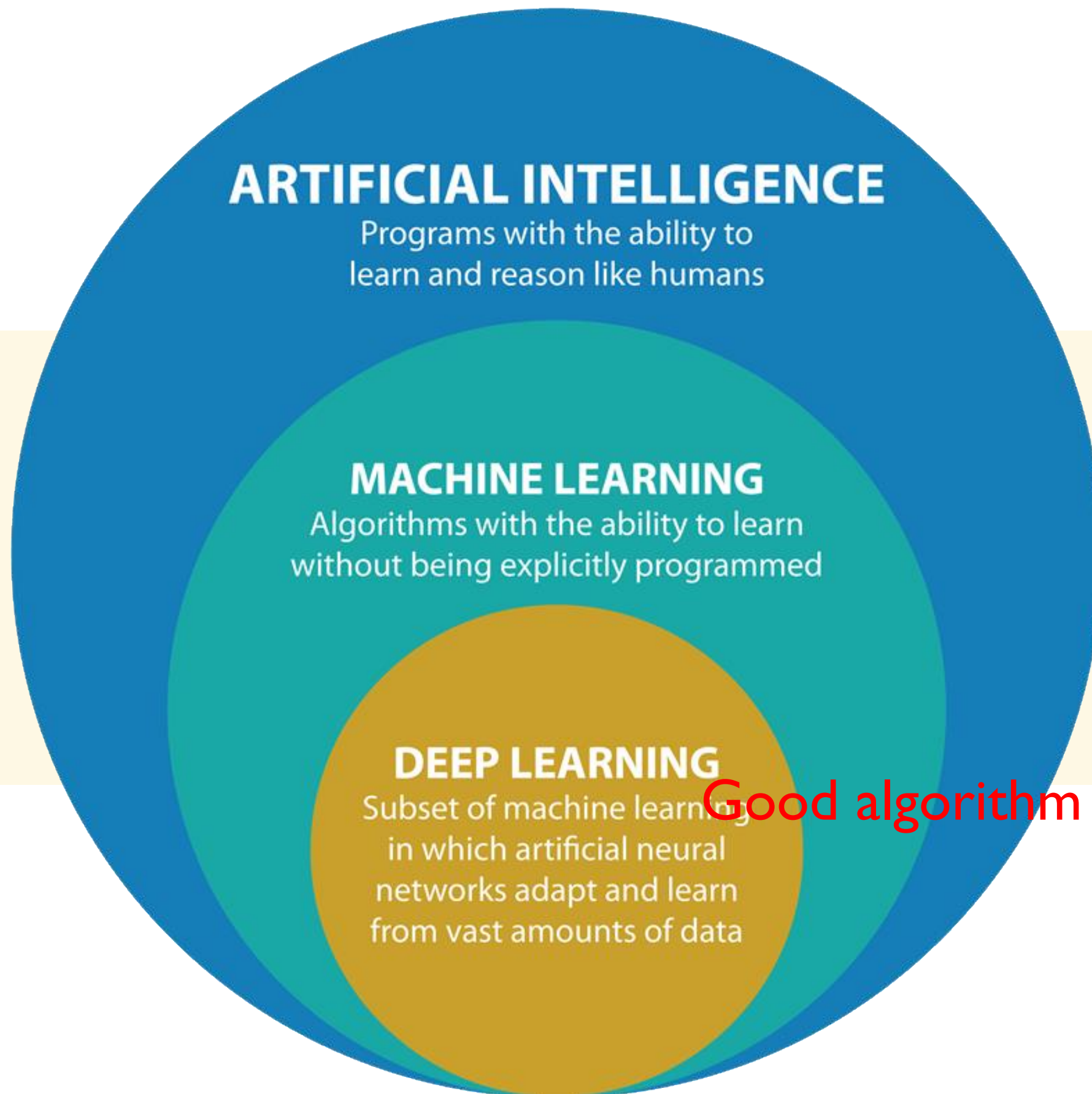


- 문제 정의 및 전략 수립 – 해결하려는 문제를 명확히 정의하고, 문제 해결을 위해 어떤 데이터를 어떻게 사용할지를 정함
- 데이터 수집 – 머신러닝에 필요한 데이터를 수집함
- 데이터 처리 – 머신 러닝 알고리즘에 사용될 수 있도록 데이터를 변환하고, 필요한 데이터를 가공함
- 모델 구현 및 검증 – 분류, 회귀, 설명, 추천 등을 위한 머신러닝 모델을 구현하고 검증을 통해 모델을 개선함
- 모델 성능 테스트 – 구현된 머신 러닝 모델의 성능을 평가함
- 결과 적용 – 성능이 만족스러울 경우 머신러닝 모델을 실제 상황에 적용

- 문제를 명확하게 구체적으로 정의하는 것
- 가장 중요한 단계
- 문제가 아닌 것을 해결하려고 시도하는 경우가 의외로 많음
- 문제는 조건, 조직, 시기, 장소 등에 따라 다름

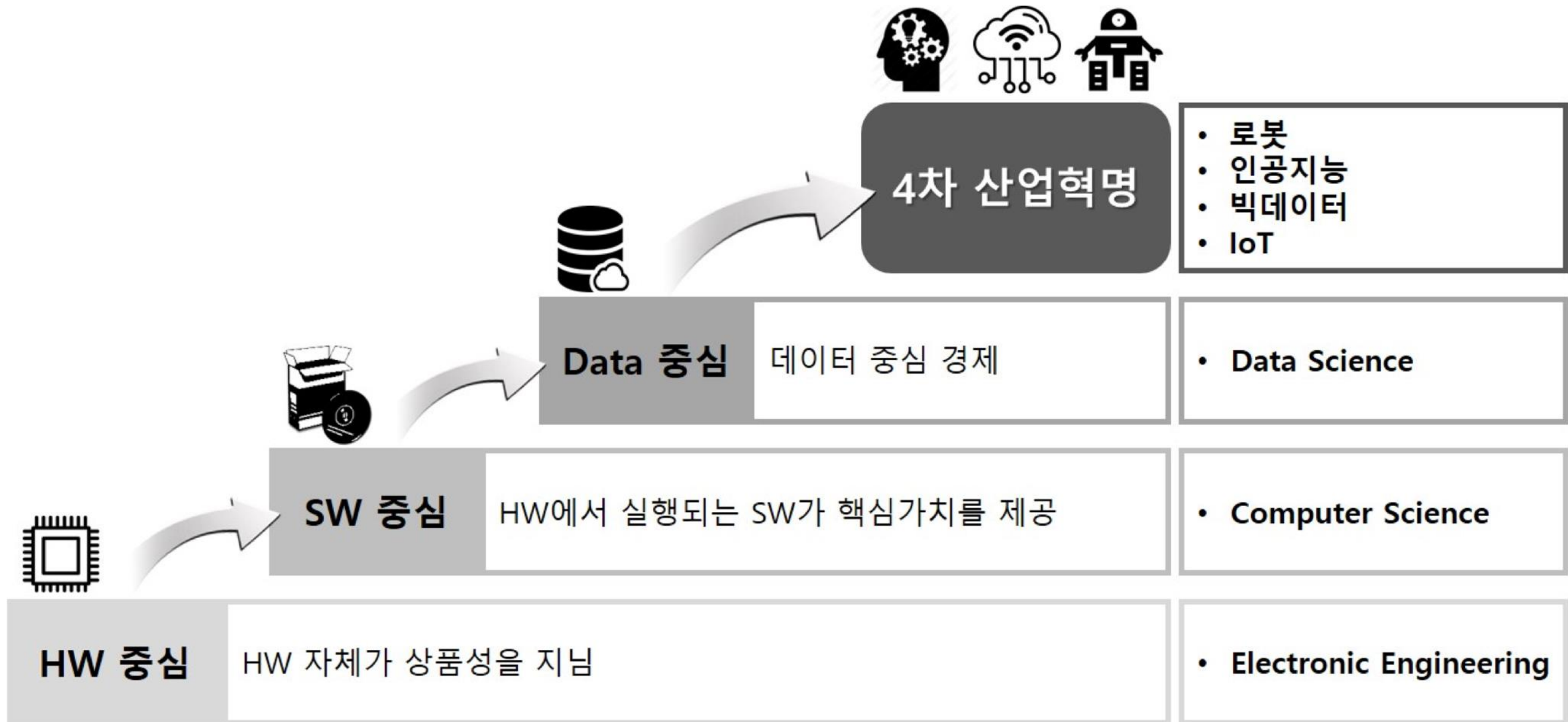
- 큰 문제를 한번에 해결하기 어려움
- 해결할 수 있는 크기의 작은 문제로 나누어 해결하는 전략
- 예) 큰 문제: 기업의 수익 감소
  - 배송이 늦어지는 이유 분석
  - 고객 불만이 발생하는 원인 분석
  - 반품이 많은 이유 분석

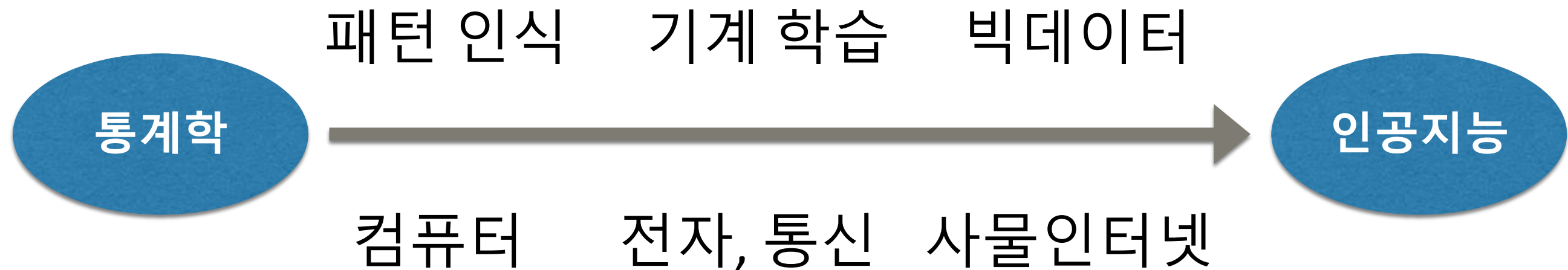
# 관련 기술



Good algorithm < Big data

# 데이터 사이언스





- 인공지능:
  - 지능이 있는 것처럼 컴퓨터가 **똑똑하게** 일을 처리하는 것
- 데이터마이닝:
  - 이미 보유하고 있는 데이터에서 **새로운** 지식을 얻는 것
  - 평범한 데이터에 숨겨져 있던 가치 있는 새로운 지식을 얻는 과정
- 비즈니스 인텔리전스:
  - 데이터 분석을 통해 새로운 **비즈니스** 전략을 얻는 것
- 통계분석:
  - **샘플 데이터**로부터 전체 데이터의 속성을 파악하는 것
- 빅데이터 분석:
  - 대량의 데이터를 분석하여 일반적인 분석으로는 찾지 못하던 **새로운 가치**를 얻는 것
- 머신러닝:
  - 데이터로부터 새로운 지식을 얻는 모델을 만들고 학습으로 **모델의 성능을 개선**
- 딥러닝:
  - 머신러닝의 한 방법으로 신경망을 사용



- 사물 인터넷 (IOT: Internet of Things)
  - TV, 냉장고, 세탁기, 보안장치, 난방장치
  - 가속 센서를 부착하여 가속의 이동, 소화, 건강상태도 모니터링
  - 데이터의 실시간 분석, 다양한 센서 데이터를 융합 처리
- 지금까지 가치 있는 정보는 주로 사람이 만들어냈다.
  - 뉴스, SNS 데이터, 블로그, 통계자료, 음악, 영화 등
- 앞으로는 사람이 직접 만드는 데이터보다 센서와 정보기기들이 생산하는 데이터가 급격히 늘어날 것
  - 온도, 습도와 같은 과학적 측정 데이터
  - 소음, 카메라, 오염도, 교통상황, 인구밀도 변화, 약물사용 통계 등
- 전혀 새로운 서비스가 나타날 것
  - 기계 장치의 고장을 미리 알아내고, 이용자의 패턴을 쉽게 찾아내며, 위험을 조기에 예측하게 될 것이다.

- 데이터 사이언스는 통계적 분석, 데이터 마이닝, 빅데이터 분석, 머신러닝, 딥러닝과 모두 관련된 기술을 다루는 연구 분야이다.
- 지금까지는 데이터를 다루는 기술이 각 영역별로 발전하였으나 최근 빅데이터라는 키워드로 데이터 분석 분야가 주목을 받았고 이들을 전체적으로 아우르는 학문 영역을 데이터 사이언스라고 한다.
- 데이터 사이언스에 대한 지식과 프로그래밍 능력을 갖추고 데이터 분석과 머신 러닝 문제를 해결하는 전문가를 데이터 사이언티스트(데이터 과학자)라고 부른다.
  - 프로그래밍 능력 (computer)
  - 수학적 지식 (math)
  - 도메인 지식 (domain)

- 생활
- 스마트홈
- 마케팅
- 로봇
- 금융
- 핀테크
- 안전
- 산업
- 에너지관리
- 자동차
- 자율주행차
- 건강
- 스포츠
- Almost all areas in our life

# 심슨 패러독스(Simpson's Paradox)

44

- 각 그룹 데이터에서 나타나는 특징이 그룹들이 결합되었을 때 달라지는 현상 (같은 데이터가 분석 방법에 따라 해석이 달라지는 예 -> 즉, 결과 수치만으로는 정확한 판단 어려움)

도시	A사	B사
서울	정상품 90 불량품 10 (불량률 10%)	정상품 920 불량품 80 (불량률 8%)
춘천	정상품 980 불량품 20 (불량률 2%)	정상품 99 불량품 1 (불량률 1%)
전체	A사 총 불량률 30/1,100 = 3%	B사 총 불량률 81/1,100 = 8%

# 데이터를 보는 눈 (또다른 예)

45

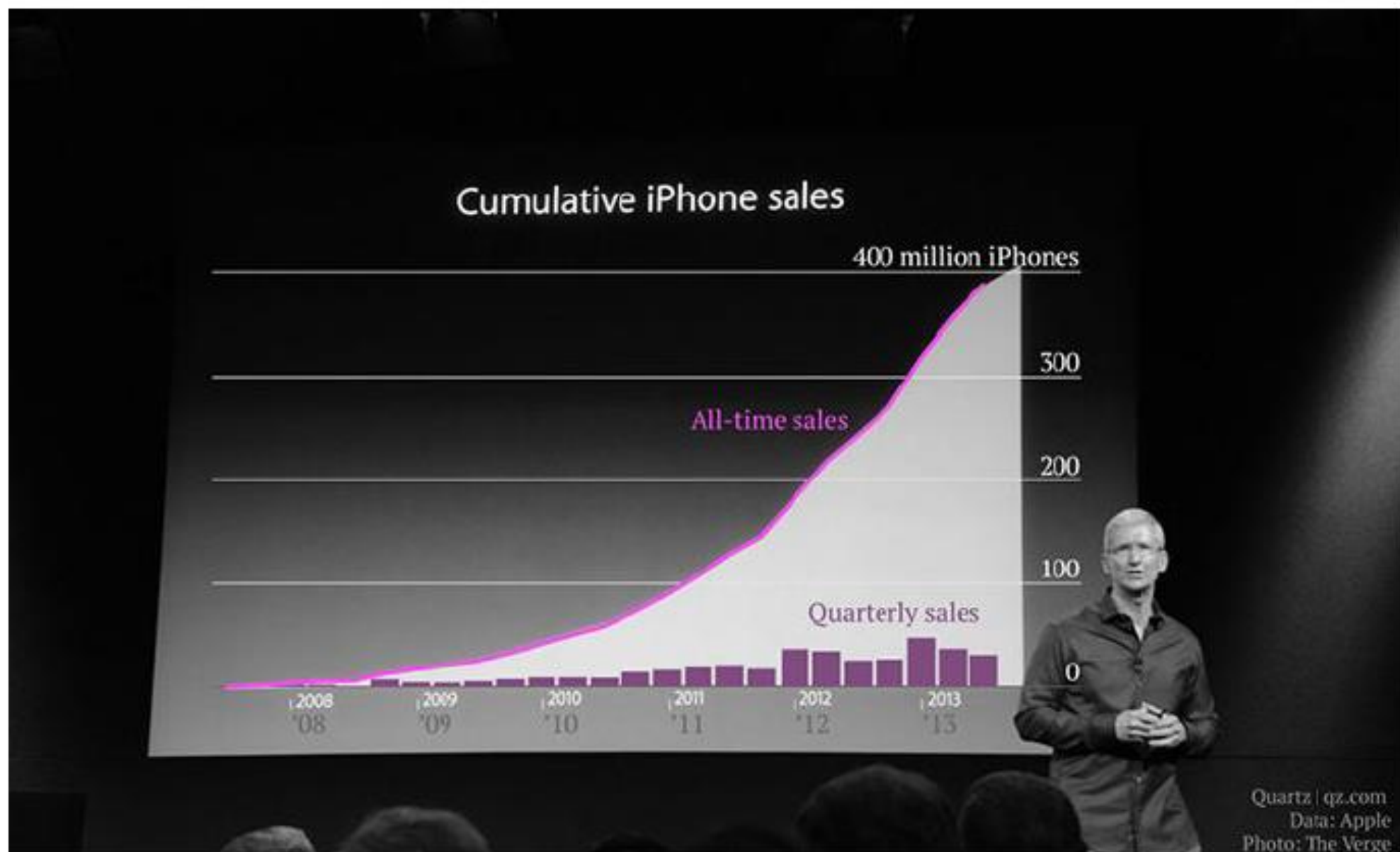
- Apple iPhone sales have been exploding, right?



# How to show and how to analyze data?

46

- Cumulative distributions present a misleading view of growth rate.



# 딥러닝 예

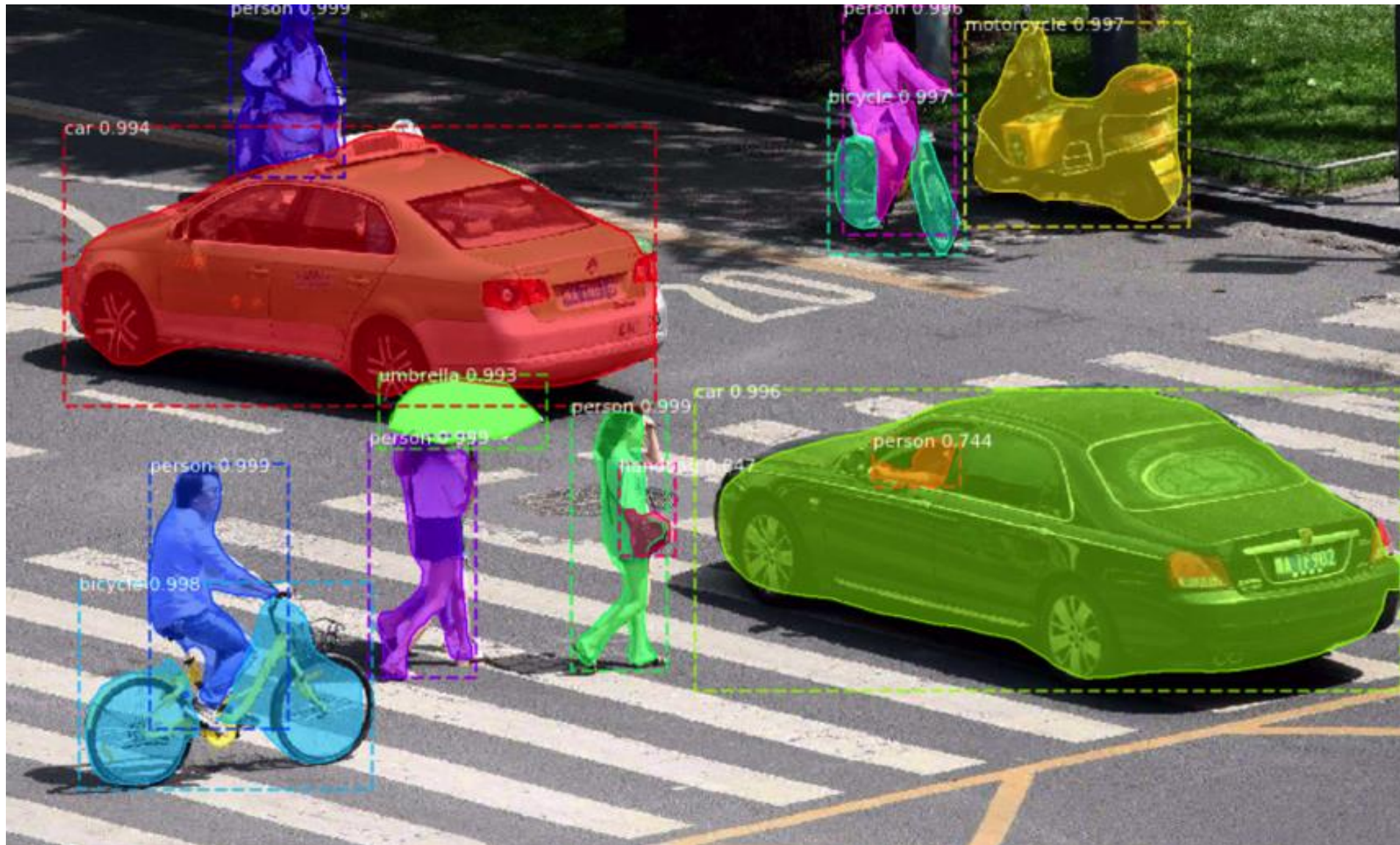
47

- 동영상을 보고 요리법을 스스로 배우는 로봇





# 딥러닝 예





형식	내용
정형 (structured)	<ul style="list-style-type: none"><li>• 데이터의 포맷이 정해져 있는 데이터</li><li>• 서식이 정해진 데이터(엑셀의 표 등)</li><li>• CSV(comma separated value) 파일과 같이 포맷이 일정</li></ul>
비정형 (unstructured)	<ul style="list-style-type: none"><li>• 미리 정해진 포맷을 가지지 않는 데이터</li><li>• 블로그, 트위터 데이터 등 임의의 문장 등으로 구성</li><li>• 오디오나 비디오 데이터</li></ul>
반정형 (semi-structured)	<ul style="list-style-type: none"><li>• 데이터 내부에는 논리적인 형식을 가지고 있으나 외형상으로는 데이터 포맷이 정형 데이터처럼 완전하게 정의되어 있지 않은 데이터</li><li>• 센서 데이터, 웹 사용 기록 등</li></ul>

- 데이터는 네가지 타입이 있다.
  - 문자형: "Hello World", "대한민국", ... (ex) string
  - 수치형: 1, 5, 10, 3.14, 0.9, ... (ex) int, float
  - 바이너리형: 0100100101010101... (ex) array or list
  - 논리형: True, True, False, True, ... (ex) boolean
- 수치형 데이터는 다시 범주형(categorical), 순서형(ordinal), 연속형(continuous)으로 나눌 수 있다.

- 범주형은 클래스를 구분하는 데이터이다.
  - 성별, 국가명, 요일, 사람 이름 등은 범주형 데이터이다.
  - 범주형은 대부분 문자로 표현되지만 편의상 숫자로 대체하여 표현하기도 한다. 예를 들어 월요일=1, 화요일=2, 수요일=3 등
- 순서가 의미를 가지는 데이터를 순서형 데이터라고 한다.
  - 여성의 옷 사이즈를 나타내는 44, 55, 66 같은 숫자, 달력의 1일, 2일, 3일 등이 순서형 데이터이다.
  - 순서형 데이터에서는 덧셈이나 뺄셈이 아무런 의미가 없다.
- 연속형 데이터는 숫자의 양이 어떤 의미를 가지는 데이터
  - 무게, 길이, 온도, 압력, 속도, 화폐 단위
  - 덧셈과 뺄셈의 결과가 계속 같은 연속형 데이터로서 의미를 갖는다.

# 파이썬기초

- gg-1 파이썬 기초
- gg-2 numpy


# Pandas

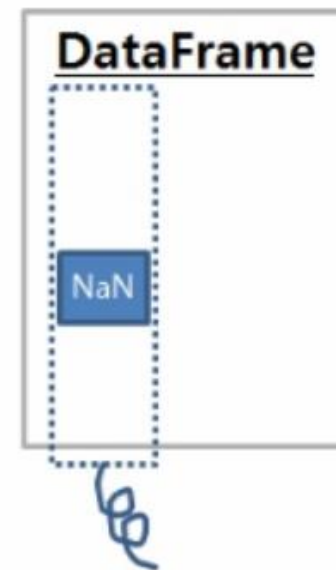
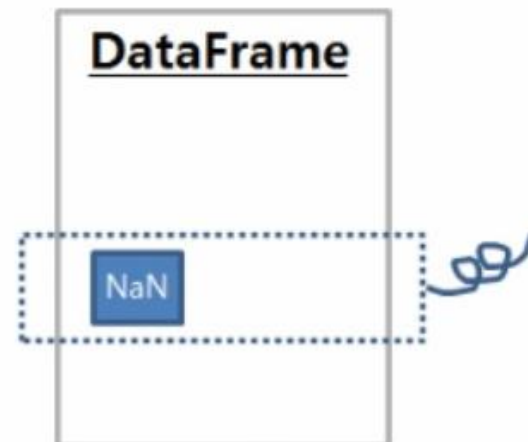
- DataFrame
- 그래프

# Missing value & outliers

55

[Python pandas] 결측값 있는 행, 열 제거하기

 Delete *row* with NaN      Delete *column* with NaN  
df.**dropna(axis=0)**      df.**dropna(axis=1)**



- gg-3 pandas
- gg-4 데이터 시각화
- gg-5 데이터프레임 연습



# 데이터 처리

- 파일 다루기

- File open/close:

```
f = open("file_name", "rwa"); f.close()
```

- File read/write:

```
data=f.read(); line=f.readline(); f.write(data)
```

- Line split:

```
string.strip()
```

- With 문과 함께 사용하기

- 자동으로 file close 됨.

```
f = open("foo.txt", 'w')  
f.write("Life is too short, you need python")  
f.close()
```

```
with open("foo.txt", "w") as f:  
    f.write("Life is too short, you need python")
```

- **폴더 관리**

- `import os`
- `os.getcwd()` # get current working directory
- `os.chdir('my_dir')` # change directory
- `os.listdir` # show file names
- `os.mkdir('test')`
- `os.rename('test', 'new_one')`
- `os.remove('out.csv')`
- `os.rmdir('new_one')`

- Pandas 함수 이용
  - `pd.read_csv('test.csv', nrows=2)` # 상위 2개의 행만
  - `df.to_csv` # csv file 로 출력
- csv 패키지 이용
  - Example:

```
import csv
f = open("./data/coffee.csv")
for row in csv.reader(f):
    print(row)
```

# JSON(JavaScript Object Notation)

61

- JSON

- JavaScript Object Notation)은 속성-값 쌍( attribute-value pairs and array data types (or any other serializable value)) 또는 "키-값 쌍"으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽고 쓰기 쉽게 **텍스트**를 사용하는 개방형 표준 포맷이다.
- 비동기 브라우저/서버 통신 (AJAX)을 위해, 넓게는 XML(AJAX가 사용)을 대체하는 주요 데이터 포맷이다. 특히, 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법으로 알려져 있다. 자료의 종류에 큰 제한은 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합하다.

- Example

```
1 {  
2   "이름": "홍길동",  
3   "나이": 25,  
4   "성별": "여",  
5   "주소": "서울특별시 양천구 목동",  
6   "특기": ["농구", "도술"],  
7   "가족관계": {"#": 2, "아버지": "홍판서", "어머니": "춘섬"},  
8   "회사": "경기 수원시 팔달구 우만동"  
9 }
```

# JSON(JavaScript Object Notation)

62

- JSON package
  - **JSON Encoding:** Python Object (dict, list, tuple 등) 를 JSON 문자열로 변경 (ex) `json.dumps(result)`
  - **JSON Decoding:** JSON 문자열 -> Python type (dict, list, tuple, 등)  
(ex) `json.loads(obj)`
- JSON format normalize
  - `pandas.io.json_normalize()`: normalize semi-structured JSON data into a flat table (for 문을 사용하지 않고도 JSON 데이터를 손쉽게 DataFrame으로 전환할 수 있음)

- gg-6 데이터처리
- gg-7 파일다루기
- gg-8 날짜데이터

# 웹 크롤링



- **스크레이핑(Scraping)**: 웹사이트의 특정 정보를 추출하는 것으로 웹데이터의 구조 분석이 필요하다. 로그인에 필요한 경우가 많다.
- **크롤링(Crawling)**: 프로그램으로 자동으로 (보통 정기적으로) 웹사이트를 돌며 정보를 추출하는 것을 말한다. 이러한 작업을 수행하는 프로그램을 크롤러, 스파이더라고한다.

- Regular Expression
  - 특정한 규칙의 문자열의 집합을 표현하기 위한 형식언어
- Meta characters

표현식	의미
$\wedge x$	문자열의 시작을 표현하며 x 문자로 시작됨을 의미한다.
$x\$$	문자열의 종료를 표현하며 x 문자로 종료됨을 의미한다.
$.x$	임의의 한 문자의 자리수를 표현하며 문자열이 x 로 끝난다는 것을 의미한다.
$x+$	반복을 표현하며 x 문자가 한번 이상 반복됨을 의미한다.
$x?$	존재여부를 표현하며 x 문자가 존재할 수도, 존재하지 않을 수도 있음을 의미한다.
$x^*$	반복여부를 표현하며 x 문자가 0번 또는 그 이상 반복됨을 의미한다.
$x y$	or 를 표현하며 x 또는 y 문자가 존재함을 의미한다.
$(x)$	그룹을 표현하며 x 를 그룹으로 처리함을 의미한다.
$(x)(y)$	그룹들의 집합을 표현하며 앞에서 부터 순서대로 번호를 부여하여 관리하고 x, y 는 각 그룹의 데이터로 관리된다.
$(x)(?:y)$	그룹들의 집합에 대한 예외를 표현하며 그룹 집합으로 관리되지 않음을 의미한다.
$x\{n\}$	반복을 표현하며 x 문자가 n번 반복됨을 의미한다.
$x\{n,\}$	반복을 표현하며 x 문자가 n번 이상 반복됨을 의미한다.
$x\{n,m\}$	반복을 표현하며 x 문자가 최소 n번 이상 최대 m 번 이하로 반복됨을 의미한다.

# Regular Expression (RE)

67

표현식	의미
[xy]	문자 선택을 표현하며 x 와 y 중에 하나를 의미한다.
[^xy]	not 을 표현하며 x 및 y 를 제외한 문자를 의미한다.
[x-z]	range를 표현하며 x ~ z 사이의 문자를 의미한다.
\w^	escape 를 표현하며 ^ 를 문자로 사용함을 의미한다.
\wb	word boundary를 표현하며 문자와 공백사이의 문자를 의미한다.
\WB	non word boundary를 표현하며 문자와 공백사이가 아닌 문자를 의미한다.
\wd	digit 를 표현하며 숫자를 의미한다.
\WD	non digit 를 표현하며 숫자가 아닌 것을 의미한다.
\ws	space 를 표현하며 공백 문자를 의미한다.
\WS	non space를 표현하며 공백 문자가 아닌 것을 의미한다.
\wt	tab 을 표현하며 탭 문자를 의미한다.
\wv	vertical tab을 표현하며 수직 탭(?) 문자를 의미한다.
\ww	word 를 표현하며 알파벳 + 숫자 + _ 중의 한 문자임을 의미한다.
\WW	non word를 표현하며 알파벳 + 숫자 + _ 가 아닌 문자를 의미한다.

Flag	의미
g	Global 의 표현하며 대상 문자열내에 모든 패턴들을 검색하는 것을 의미한다.
i	Ignore case 를 표현하며 대상 문자열에 대해서 대/소문자를 식별하지 않는 것을 의미한다.
m	Multi line을 표현하며 대상 문자열이 다중 라인의 문자열인 경우에도 검색하는 것을 의미한다.

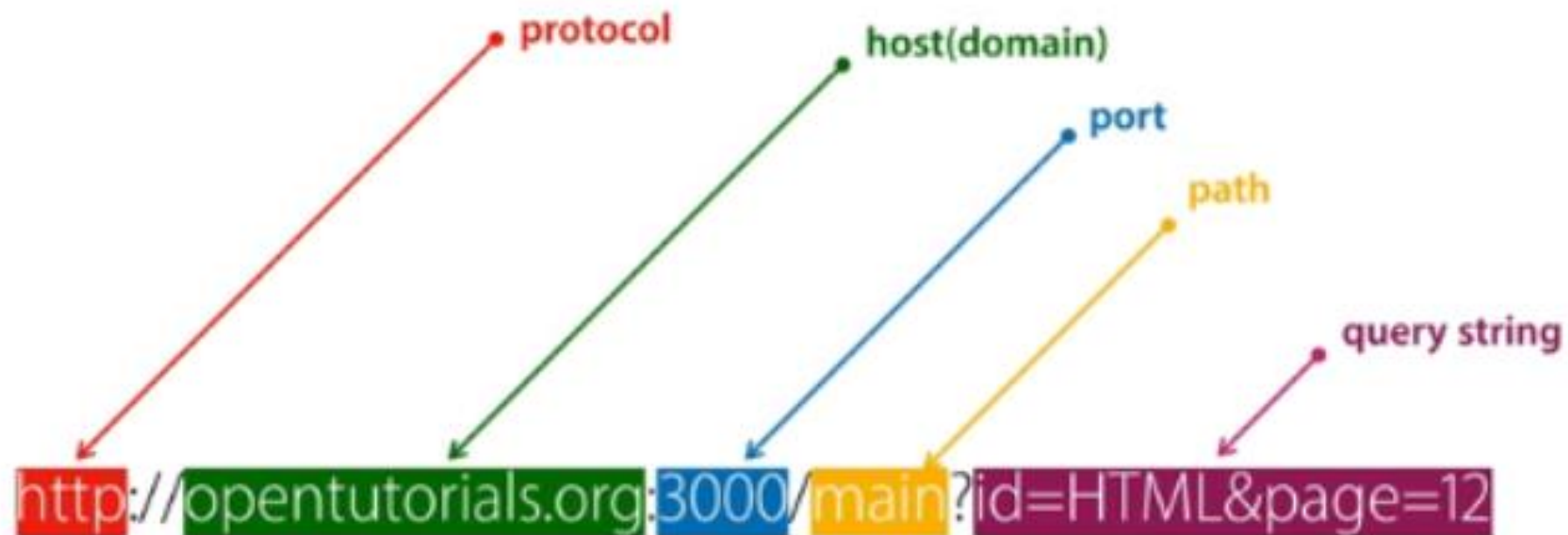
## ❖ Examples

- `^http` : 문자열의 맨 처음에 `http`가 온 경우에 매치
- `them$` : 문자열이 `them`으로 끝난 경우에 `them`에 매치
- `\bplay\b` : `play` 의 양 끝에 단어 경계가 오는 경우에만 `play`에 매치
  - (“playground”의 `play`에는 매치하지 않음)
- `\bplay\B` : `play`뒤에 단어 경계가 아닌 것이 왔을 때 `play`에 매치
  - (`play`는 No, “playground”, “playball” 은 Yes)
- `/[0-9]/g` : 전체에서 0~9 사이의 아무 숫자 ‘한 개’ 찾음
- `/[to]/g` : 전체에서 `t` 혹은 `o` 를 찾음
- `/filter/g` : 전체에서 ‘filter’ 라는 단어에 매칭되는 것을 찾음
- `\b(?!\bto\b)\w+\b` : ‘to’라는 단어 빼고 다른 단어 매칭
  - (`\w+` 는 match one or more word characters: ‘[a-zA-Z0-9\_]’)
- `/^\d{3}-\d{3,4}-\d{4}$/` : 전화번호
- `/^01([0|1|6|7|8|9]?)-?([0-9]{3,4})-?([0-9]{4})$/` 휴대폰번호

# URL(uniform resource locator)

69

- URL (uniform resource locator)
  - 네트워크 상에서 자원이 어디 있는지 알려주기 위한 규약



## ■ HTML

- Standard Markup language for creating web pages and web applications
- With CSS(Cascading STYPE sheets) and Javascript, it forms a triad of cornerstone technologies for WWW.
- HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
- The World Wide Web Consortium (W3C), has encouraged the use of CSS over explicit presentational HTML since 1997.

## ❖ Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

```
<!-- HTML5 -->
<!-- Describes web page -->

<!-- Brower page title -->

<!-- Visible page content -->
```

## ❖ Elements

- Building blocks of HTML pages
- (ex) `<head></head>`: 시작을 알리는 태그, `<title>` 은 `<head>` 안에서 제목 부여.
- `<body></body>`: 실제 웹검색기 화면에 출력되는 부분
- `<h1></h1>` : 제목 텍스트의 크기 (`<h1>` ~ `<h6>`)
- `<p></p>` : paragraphs (단락)
- Tags such as `<img />` and `<input />` directly introduce content into the page.
- `<a href="https://www.wikipedia.org/">A link to Wikipedia!</a>` : create a link

## ❖ Attributes of an element

- id: document-wide unique identifier for an element
- class: classifying similar elements
- style: assign presentational properties
- title: attach subtextual explanation to an element
- lang: identifies a natural language to an element

## ■ Some of the Basic Tags

- Headings: `<h1>` `<h2>` `<h3>`
- Paragraphs: `<p>`
- Images: `` ; alternative text, size (width and height)
- Links: `<a href="http://www.w3schools.com/tags/tag_a.asp">Here</a>`
- Tables: `<table border="1" cellpadding="5" cellspacing="5">`  
`<tr>` `<td>`One`</td>` `<td>`Two`</td>` `</tr>` ; row and data `</table>`
- Divisions: `<div>` This is a DIV container`</div>` ; define a section in a document (you can think of it like a container or a building block.)
- Lists: `<ul>` `<ol>` `<li>` ; defines unordered list, ordered list, a list item
- Line breaks: `<br>`
- Bold text: `<b>`...`</b>`
- Italic text: `<i>`...`</i>`



## ❖ CSS

- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- Enable the separation of presentation and content, including layout, colors, and fonts. It can improve content accessibility, provide more flexibility and control, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file.

## ❖ Three ways to add CSS in HTML

- Inline – by using the style attribute in HTML elements  
(ex) `<h1 style="color:blue;">This is a Blue Heading</h1>`
- Internal – by using a `<style>` element in the `<head>` section  
(ex) next page
- External – by using an external CSS file (the most common way)  
(ex) next page

# Internal CSS

74

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.

- An external style sheet is used to define the style for many HTML pages.
- With an external style sheet, you can change the look of an entire web site, by changing one file!
- To use an external style sheet, add a link to it in the `<body>` section of the HTML page:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

In the “styles.css” file:

# Example (HTML only)

76

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>
11    안녕하세요.
12  </h1>
13  <h2>
14    LKT Programmer 입니다.
15  </h2>
16 </body>
17 </html>
```

Colored by Color Scripter

---

안녕하세요.

LKT Programmer 입니다.

# Example (CSS)

77

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8
9   <style>
10     h1 {
11       background-color : red;
12     }
13     h2 {
14       background-color : green;
15     }
16   </style>
17
18 </head>
19 <body>
20   <h1>
21     안녕하세요.
22   </h1>
23   <h2>
24     LKT Programmer 입니다.
25   </h2>
26 </body>
27 </html>
```

Colored by Color Scripter

안녕하세요.

LKT Programmer 입니다.

## ❖ Some attributes

- CSS Fonts: **color**, **font-family**, **font-size**
- CSS Border: **border**
- CSS Padding: **padding** ; defines a padding(space) between the text and the border
- CSS Margin: **margin** ; defines a margin(space) outside the border
- id attribute: `<p id="p01">I am different</p>` ; defines a style with the specific id
- Class attribute: `<p class="error">I am different</p>` ; a style for the specific class
- External reference:  
`<link rel="stylesheet" href=https://www.w3schools.com/html/styles.css>`

## ❖ Example of Web page containing JavaScript (HTML 5 syntax) and the DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <button id="hellobutton">Hello</button>
    <script>
      document.getElementById('hellobutton').onclick = function() {
        alert('Hello world!');           // Show a dialog
        var myTextNode = document.createTextNode('Some new words. ');
        document.body.appendChild(myTextNode); // Append "Some new words" to the page
      };
    </script>
  </body>
</html>
```

Hello Some new words.

이 페이지 내용:

Hello World!

확인

# JS example(2)

80

```
<!DOCTYPE html>
<html>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <body>
    <script>
      var sum=0;
      n=parseInt(prompt("1 부터 n 까지 합을 구하려 합니다. n을 입력해주세요", ""));
      for(var i=1; i<=n; i++){
        sum+=i;
      }
      document.write("1부터 " + n + "까지의 합은 " + sum + "입니다.");
    </script>
  </body>
</html>
```

이 페이지 내용:

1 부터 n 까지 합을 구하려 합니다. n을 입력해주세요

확인

취소

1부터 100까지의 합은 5050입니다.



# BeautifulSoup parser

81

## ❖ Python 내장 html.parser or lxml's HTML parser

- BeautifulSoup(html, "html.parser") or BeautifulSoup(html, "lxml")
- 일반적으로 lxml 이 빠르고 flexible 하다고 알려짐 (정확하게 html 로 마크업이 안되어 있을 경우에는 lxml 사용)
- (ex) Google Finance 의 예 - </tr>, </td> 가 없음. 이때 "lxml" parser 사용

```
1 <div id=prices class="gf-tablewrapper sfe-break-bottom-16">
2 <table class="gf-table historical_price">
3   <tr class=bb>
4     <th class="bb lm lft">Date
5     <th class="rgt bb">Open
6     <th class="rgt bb">High
7     <th class="rgt bb">Low
8     <th class="rgt bb">Close
9     <th class="rgt bb">Volume
10  <tr>
11    <td class="lm">Feb 28, 2014
12    <td class="rgt">100.71
13    <td class="rgt">100.71
14    <td class="rgt">100.71
15    <td class="rgt rm">0
16 </table>
```

# Requests module

82

```
don@don-Lenovo-Ideapad-500S-14ISK: ~  
  
In [1]: import requests  
  
In [2]: response = requests.get('http://www.tistory.com')  
  
In [3]: response.status_code  
Out[3]: 200  
  
In [4]: response.text  
Out[4]: u'<!DOCTYPE html>\n<html lang="ko">\n<head>\n\t<meta charset="utf-8">\n\t<meta property="og:url" content="http://www.tistory.com">\n\t<meta property="og:site_name" content="TISTORY">\n\t<meta property="og:title" content="TISTORY">\n\t<meta property="og:description" content="\ub098\ub97c \ud45c\ud604\ud558\ub294 \ube14\ub85c\udadf8\ub97c \ub9cc\ub4e4\uc5b4\ubcf4\uc138\uc694.">\n\t<meta property="og:image" content="https://t1.daumcdn.net/cssjs/icon/557567EA016E200001">\n\t<title>TISTORY</title>\n\t<link rel="shortcut icon" href="//i1.daumcdn.net/cfs.tistory/static/top/favicon.ico">\n\t<link rel="stylesheet" href="//s1.daumcdn.net/svc/attach/U0301/cssjs/tistory-web-top/1470361988/static/css/pc/T.p.top.css">\n\t<link rel="apple-touch-icon" href="http://i1.daumcdn.net/thumb/C180x180/?fname=http://cfile5.uf.tistory.com/image/241F093D5701E7380371B5">\n\t<link rel="apple-touch-icon" sizes="76x76" href="http://i1.daumcdn.net/thumb/C76x76/?fname=http://cfile5.uf.tistory.com/image/214AF9425701E76D0ACB4B">\n\t<link rel="apple-touch-icon" sizes="120x120" href="http://i1.daumcdn.net/thumb/C120x120/?fname=http://cfile5.uf.tistory.com/image/241F093D5701E7380371B5">\n\t<link rel="app
```

# requests module

83

- ❖ Useful and good to use to access url
  - `request.get()` ; try to get or retrieve data from a specified source
  - `request.post()`
  - `request.put()`
  - `request.delete()`
- ❖ Regardless of whether GET / POST, you never have to encode parameters again, it simply takes a dictionary as an argument and is good to go:
  - `userdata = {firstname:"John", "lastname":"Doe", "passwd":"jdoe123"}`
  - `resp = request.post('http://www.mywebsite.com/user', data=userdata)`
- ❖ Plus, it even has a built-in JSON decoder.
  - `resp.json()`
  - `resp.text` ; if response data is just text

- ❖ BeautifulSoup ; parsing html
  - find()
  - find\_all()
  - prettify()
  - select\_one() ; query by CSS
  - select(id or tags or class or tag.class or id>tag.class ..)
- ❖ urllib.request
  - urlopen() ; open the url
- ❖ Requests
  - get()

- gg-9 데이터크롤링
- gg-10 스크레이핑
- gg-11 부동산정보
- gg-12 국제상품가격
- gg-13 스타벅스매장지도

# 베이지스 알고리즘



# 베이스(Bayes) 이론

- 독립적인 두 사건 A와 B가 있을 때
- 사건 A가 발생했다는 조건에서 사건 B가 발생할 확률

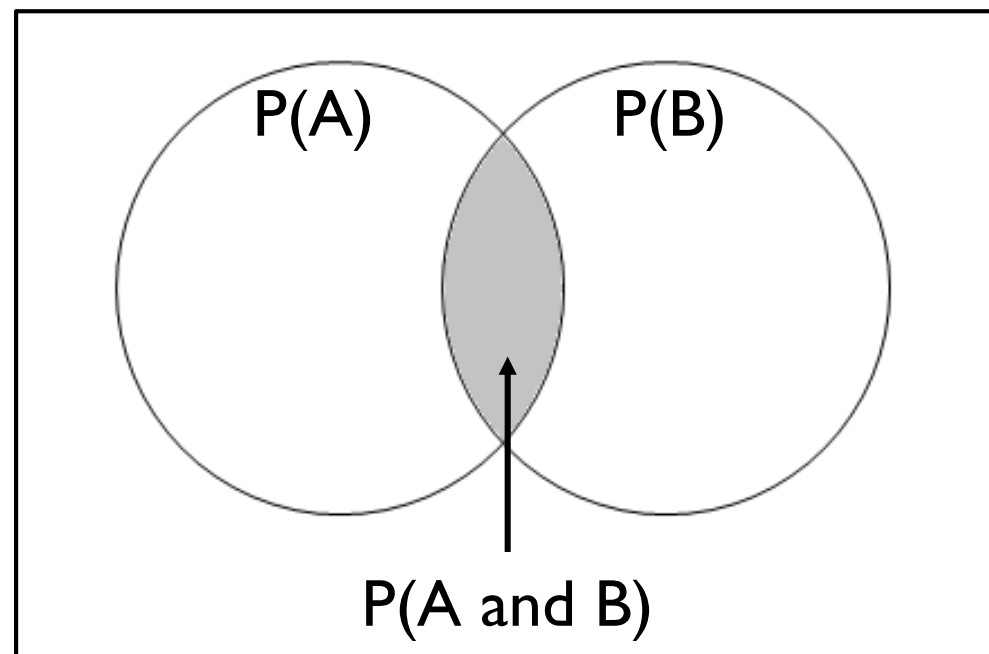
$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A,B)}{P(A)}$$

- 사건 B가 발생했다는 조건에서 사건 A가 발생할 확률

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(A,B)}{P(B)}$$

- $P(A,B)$ 로 정리하면  $P(B|A)P(A) = P(A|B)P(B)$ 
  - 수식에서 3개의 확률을 알면 나머지 하나는 구할 수 있음

- 은조건부 확률 이론을 이용하여 새로운 사건의 조건부 확률을 예측하는 방법
  - 베이즈 이론은 단순하지만 매우 강력한 이론이다.



$$P(B|A) = \frac{P(A,B)}{P(A)}$$

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

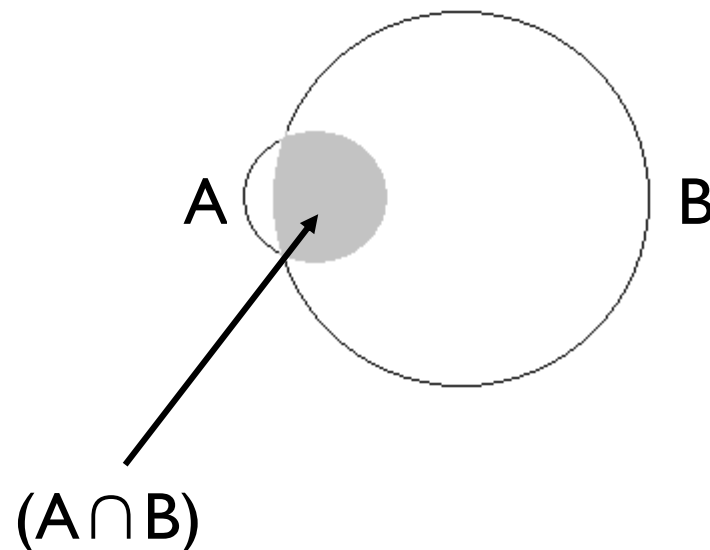
$$P(B|A)P(A) = P(A|B)P(B)$$

- 세 개의 확률을 알면 나머지 한 가지 확률을 구할 수 있다.



# 베이스 이론의 예

- 사건 A는 피부암에 걸릴 확률을 나타내고 사건 B는 붉은 반점이 생길 확률이라고 하자. 그리고 피부암에 걸리면 붉은 반점이 생길 확률이 0.99라고 하자.
  - 피부암 발생 확률  $P(A) = 0.0001$  (1만명 중에 1명)
  - 붉은 반점이 나타난 확률  $P(B) = 0.01$  (1만명 중에 100명)
- 붉은 반점이 나타났다는 조건하에 내가 피부암일 확률
$$P(A|B) = P(B|A)P(A)/P(B) = (0.99)*(0.0001)/0.01 = 0.01$$



- 베이지스 알고리즘은 이와 같이 미리 파악한 사전 확률들을 기초로 어떤 새로운 사건의 조건부 확률을 예측하게 해준다
  - 주는 간결하지만 명확한 동작을 한다
- 베이지스 알고리즘은 의학분야에서 널리 사용되며 (여러 증상을 보고 병을 진단할 때 등),
- 스팸메일 검출에도 사용된다

- 수신한 메일이 스팸인 사건을 A라고 하자. 메일이 스팸일 확률이 0.01이라고 하면 즉 100통의 메일 중에 하나가 스팸이면,
  - $P(A) = 0.01$  이다
- “쿠폰”이라는 단어 하나만 보고 스팸인지 아닌지를 판단하는 간단한 알고리즘을 생각해 보자. 메일에 “쿠폰”이 들어 있을 확률이 과거 통계로부터 평균 0.003이라고 하자.
- 전체 메일중에 ‘쿠폰’ 단어가 들어있는 사건을 B라고 하면
  - $P(B) = 0.003$ 이다.
- 전체 스팸 중에 쿠폰을 포함한 메일이 10%였다고 하면 이는 다음과 같이 표현된다.

$$P(B|A) = 0.1$$

- 이제 새로운 메일이 하나 도착했는데 여기에 “쿠폰” 단어가 포함되어 있다면 이 메일이 스팸일 확률은 얼마일까?
- 베이지 이론에 따라 이 값을 다음과 같이 구할 수 있다.

$$\begin{aligned}P(A|B) &= P(B|A)P(A)/P(B) \\ &= (0.1)(0.01)/(0.003) \\ &= 0.333\end{aligned}$$

	“쿠폰”		
	있음	없음	합
스팸	1	9	10
정상 메일	2	988	990
합	3	997	1000

- 분석 대상인 1000 개의 메일 중에 스팸의 총 수가 10개 이고 따라서 스팸 메일이 발생할 확률은  $P(A)=0.01$ 이다.
- 모든 메일 중에 “쿠폰” 단어가 들어간 메일은 총 3개이므로  $P(B)=0.003$ 이다.
- 스팸 메일 10개만을 놓고 볼 때 쿠폰 단어가 들어간 것은 1개이고 9개에는 쿠폰 단어가 없었다. 즉, 10%의 스팸 메일에 쿠폰이 들어 있었다.
- 새로운 메일에 쿠폰이라는 단어가 들어 있었다. 이 메일이 스팸일 확률은
 
$$P(A|B) = (1)/(1+2) = 1/3 = 0.3333$$

- 과거에 스팸에 자주 들어 있었던 단어들, 예를 들면 할인, 급매, 비아그라, 판매, 당첨 등과 같은 단어들을 동시에 고려한다면 스팸을 찾아낼 확률이 높아질 것이다.
- 그런데 여러 개의 사건이 결합된 경우 조건부 확률 식은 매우 복잡해진다. 왜냐하면 각 단어의 발생들 간에도 서로 조건부 확률이 있을 것이고 이들을 고려하는 알고리즘은 구현하기가 매우 복잡해진다.
- 예를 들어 '쿠폰'과 '할인'이라는 단어는 서로 독립적인 사건이 아니며 같이 발생할 확률이 높다.
- 이렇게 복잡해지는 문제를 단순화 한 알고리즘으로 베이즈 이론을 단순화하며 확장한 나이브 베이즈 (Naive Bayes, NB) 알고리즘이 널리 사용된다.

- 나이브(naive)라는 단어를 사용한 이유는, 분류에 고려한 여러 특성변수들의 서로 독립적이라는 "순진한" 가정을 하기 때문이다.
- 이들 단어의 발생이 독립적이지 않지만 서로 독립이라고 가정하여도 상대적인 확률을 구하는 것은 가능하다.
- 나이브 베이즈로 추정한 값은 0~1 사이의 값을 갖지만 이는 수학적으로 정확한 확률값을 구한 것이 아니라 상대적인 점수(score)를 구한 것이다.
- 일정한 점수 이상의 메일을 스팸으로 처리하고 오차가 발생하면 조정하는 방식으로 학습하면 되기 때문이다.
- NB 알고리즘은 동작이 매우 단순하고 처리속도도 빠르며 평소의 통계를 기반으로 확률 값만 구해두면 된다.
- NB 알고리즘은 특정 단어가 들어 있는지를 파악해야 하는 블로그 분석이나 트위터 분석 등 텍스트 분석에서 널리 사용되며 거의 모든 데이터 분석에서 사용할 수 있다.

# 텍스트 분석



- 텍스트 분석의 목적은 텍스트의 의미를 알아내는 것
  - 글의 목적
  - 글쓴이의 성향(찬성/반대)
  - 기분(기쁨/슬픔/우울함 등)
  - 제품 피드백 등
- 텍스트 자체는 대표적인 비정형 데이터
- 의미를 추출하려면 비정형 데이터에서 정형화된 정보를 먼저 얻어야 함
- 텍스트 구문을 분석하여 의미를 파악하고 이것을 정량적으로 측정함

- 트위터, 블로그, 페이스북 북 등 SNS의 글을 분석하여 소비자들의 반응, 감성, 트렌드를 파악하거나 개인별 마케팅이나 상품 피드백을 분석하는데 사용된다.
- 이메일, 웹사이트 댓글, 신문기사, 콜센터 상담기록, 도서 등을 분석하여 글의 주요 내용을 파악하거나, 문서의 특징을 추출하거나, 유사한 글이나 저자를 찾는 작업 등을 수행한다.
- 참고문헌이나 본문 인용의 관계를 통해서 문서간의 연계성, 전문가들의 인적 네트워크 등을 파악하는데도 사용한다.
- 인공지능 스피커, 챗봇 등에서도 기본적으로 텍스트 분석이 필요하다

- 텍스트는 대표적인 비정형 데이터이다. 먼저 비정형 데이터인 글자로부터 정형화된 데이터인 수치 데이터를 얻어야 한다.
- 사람이 단어나 문장의 의미를 인식하듯이 컴퓨터가 단어 자체 의미를 직접 파악할 수는 없다
- 컴퓨터가 다루는 텍스트의 단위를 토큰이라고 하고 주어진 텍스트를 토큰으로 나누는 작업을 토큰화(tokenize)라고 한다.

- 데이터 분석에 주어진 전체 문서 집합을 말뭉치(corpus)라고 한다.
- 문서(document)란 코퍼스 내의 한 단위의 텍스트를 말한다. 예를 들어 하나의 블로그는 문서이고 분석할 대상 블로그가 1천개이면 이 1천개 블로그 집합이 말뭉치이다.
- 코퍼스에서 의미 있는 단어를 추출하는 작업을 파싱(parsing)이라고 한다.

- 먼저 토큰화의 단위를 단어(word)로 할지 아니면 글자(character) 단위로 할지를 정해야 한다.
- 단어를 어근(stem)으로 변환하면 어미 변화를 무시하거나 조사를 무시하게 되어 텍스트에 들어 있던 정보를 잃게 된다.
- 글자 단위로 토큰화를 하면 어근으로 변환할 때 정보를 잃는 문제를 피할 수 있다.
- 영어는 알파벳이 26글자이므로 음절단위의 토큰의 수가 매우 적다. 한글은 음절단위로 나누면 음절의 수가 수천 가지가 될 것이다.

- 토큰화 단위에는 크게 세 가지가 있다.
  - 단어(word) 단위
  - 글자(character) 단위
  - n-gram 단위
- n-gram이란 n개의 연속된 단어를 하나로 취급하는 방법이다.
- 예를 들어 "러시아 월드컵"이라는 표현을 "러시아"와 "월드컵" 두 개의 독립된 단어로만 취급하지 않고 두 단어로 구성된 하나의 토큰으로 취급한다.
  - n=2 경우를 bi-gram이라고도 부른다
  - 단어의 갯수가 늘어난 효과를 얻는다

# 토큰화 – (n-gram)

103

텍스트: “어제 러시아에 갔다가 러시아 월드컵을 관람했다”

단어토큰: {“어제”, “러시아”, “갔다”, “러시아”, “월드컵”, “관람”}

2-gram 토큰: {“어제 러시아”, “러시아 갔다”, “갔다 러시아”, “러시아 월드컵”, “월드컵 관람”}

- n-gram을 허용하면 토큰화 대상의 수가 매우 크게 증가한다. 이론적으로는 10만개의 단어를 두 개 붙여서 나올 수 있는 경우의 수는 10만의 자승이 된다.
- 실제로는 빈도수가 최소한 몇 개 이상인 것만을 다룬다.
- 토큰화 한 결과를 수치로 만드는 방법
  - 원핫(one-hot) 엔코딩
  - BOW(단어모음)
  - 단어벡터(Word Vector) 방법



- 토큰에 고유 번호를 배정하고 모든 고유번호 위치의 한 컬럼만 1, 나머지 컬럼은 0인 벡터로 표시하는 방법

텍스트: "어제 러시아에 갔다가 러시아 월드컵을 관람했다"

토큰 사전: {"어제":0, "러시아":1, "갔다":2, "월드컵":3, "관람":4}

원핫 코딩:

어제 = [1, 0, 0, 0, 0]

러시아 = [0, 1, 0, 0, 0]

갔다 = [0, 0, 1, 0, 0]

월드컵 = [0, 0, 0, 1, 0]

관람 = [0, 0, 0, 0, 1]

- 원핫 코딩 방식으로 단어(토큰)을 표현하면, 단어의 수가 적을 때에는 문제가 안되지만 예를 들어 단어가 모두 10만개이면 모든 단어가 항목이 10만개인 (0과 1로 구성된) 벡터로 표시된다.
- 만일 주어진 텍스트가 20개의 단어로 구성되어 있다면,  $20 \times 100,000$ 개 크기의 벡터가 필요하다.
- 텍스트 분석은 “문장”을 단위로 하는 경우가 많으므로 한 문장을 하나의 벡터로 만드는 방법이 단어모음(BOW) 방식이다.
  - 한 문장을 단어 사전 크기의 벡터로 표현하고 그 문장에 들어 있는 단어의 컬럼만 1로, 단어가 없는 컬럼은 모두 0으로 표현한다.
- 먼저 단어 사전을 만들고 각 문장에 어떤 단어가 들어 있는지 조사하여 해당 컬럼만 1로, 나머지는 0으로 코딩한다

- 단어 사전: {"어제":0, "오늘":1, "미국":2, "러시아":3, "갔다":4, "축구":5, "월드컵":6, "올림픽":7, "관람":8, "나는":9, ..., "중국":4999 }
- Text\_1: "어제 러시아에 갔다가 러시아 월드컵을 관람했다" 를 BOW로 표현하면

문장번호	0	1	2	3	4	5	6	7	8	9	10	...	4998	4999
Text_1	1	0	0	1	1	0	1	0	1	0	0	0	0	0
Text_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Text_3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Text_4	0	0	0	0	0	0	0	1	0	0	0	0	1	0
...														
Text_50	0	0	0	0	0	0	1	0	0	0	0	0	0	0

- 앞에서는 문장 단위로 어떤 단어들이 있는지를 BOW로 만드는 방법을 소개했다. 이를 확장하여 문서(document) 단위로 어떤 단어들이 있는지를 표현하는 것을 문서-단어 (document-term) 행렬이라고 한다.
  - 같은 단어가 여러 번 등장하면 1 이상의 값을 갖는다

문서번호	0	1	2	3	4	5	6	7	8	9	10	...	4998	4999
Doc_1	1	2	3	1	4	0	2	0	1	3	0	0	0	0
Doc_2	0	0	0	0	2	0	0	0	0	0	0	0	2	0
Doc_3	0	0	0	1	0	0	0	0	3	0	0	0	0	1
Doc_4	4	0	0	0	0	0	0	1	0	0	4	0	1	0
...														
Doc_100	0	2	0	0	0	0	1	4	0	1	0	0	0	0

- Tfidf(Term Frequency-Inverse Document Frequency)
- Tf(term frequency)란 단어가 각 문서에서 발생한 빈도이다
- 그 단어가 등장한 '문서'의 빈도를 document frequency (df)라고 한다
- 적은 문서에서 발견될수록 가치 있는 정보라고 할 수 있다
- 많은 문서에 등장하는 단어일수록 일반적인 단어이며 이러한 공통적인 단어는 tf가 크다고 하여도 비중을 낮추어야 분석이 제대로 이루어질 수 있다.
- 따라서 단어가 특정 문서에만 나타나는 희소성을 반영하기 위해서 idf(df의 역수)를 tf에 곱한 값을 tf 대신 사용한다

# tf-idf (example)

110

- From [http://www.datasciencecourse.org/notes/free\\_text/](http://www.datasciencecourse.org/notes/free_text/)
  - Doc1 = "The goal of this lecture is to explain the basics of free text processing"
  - Doc2 = "The bag of words model is one such approach"
  - Doc3 = "Text processing via bag of words"

$$X = \begin{matrix} & \begin{matrix} \text{the} & \text{is} & \text{of} & \text{goal} & \text{lecture} & \text{bag} & \text{words} & \text{via} & \text{text} & \text{approach} \end{matrix} \\ \begin{bmatrix} 2 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} \text{Document 1} \\ \text{Document 2} \\ \text{Document 3} \end{matrix} \end{matrix}$$

# tf-idf (example)

- Term frequency
  - Counts of each word in a document
  - $tf_{ij}$  = frequency of word  $j$  in document  $i$
- Inverse document frequency
  - Term frequencies tend to be “overloaded” with very common words (“the”, “is”, “of”, etc)
  - Idea if inverse document frequency weight words negatively in proportion to how often they occur in the entire set of documents

$$idf_j = \log \left( \frac{\# \text{ documents}}{\# \text{ documents with word } j} \right)$$

# tf-idf (example)

112

$$X = \begin{matrix} & \begin{matrix} \text{the} & \text{is} & \text{of} & \text{goal} & \text{lecture} & \text{bag} & \text{words} & \text{via} & \text{text} & \text{approach} \end{matrix} \\ \begin{bmatrix} 2 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} \text{Document 1} \\ \text{Document 2} \\ \text{Document 3} \end{matrix} \end{matrix}$$

$$\text{idf}_{\text{of}} = \log \left( \frac{3}{3} \right) = 0$$

$$\text{idf}_{\text{is}} = \log \left( \frac{3}{2} \right) = 0.405$$

$$\text{idf}_{\text{goal}} = \log \left( \frac{3}{1} \right) = 1.098$$



# tf-idf (example)

- Term frequency inverse document frequency =  $tf_{ij} \cdot idf_j$
- Just replace the entries in the  $X$  matrix with their TFIDF score.

$$X = \begin{matrix} & \text{the} & \text{is} & \text{of} & \text{goal} \\ \begin{bmatrix} 0.8 & 0.4 & 0 & 1.1 \\ 0.4 & 0.4 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

- gg-14 나이브베이지
- gg-15 이름성별예측
- gg-16 스팸메시지필터링

# 머신러닝 개요

- 머신러닝:
  - 정답과 오답을 계속 가르쳐 주면 모델의 정확도가 높아짐
  - 많은 데이터를 사용하여 컴퓨터 성능이 점차 발전(학습)하는 것

- 예전에는 컴퓨터는 프로그래머가 코딩한 대로만 동작
  - 계산을 빨리 하든지,
  - 이미지를 처리하든지,
  - 정해진 알고리즘대로 빠르고 정확하게 동작하는 일
- 머신러닝에서는 컴퓨터가 데이터를 보면서 점차 성능을 향상시킨다.
  - 컴퓨터가 데이터를 보고 스스로 기능을 향상시키는 방법을 찾아낸 것

- 머신러닝을 사용하는 목적 즉, 머신러닝으로 문제를 해결하는 유형
  - 설명(description)
    - ▶ 클러스터링 (Clustering, 군집)
    - ▶ 비지도 학습 (Unsupervised Learning) – based on input only
  - 예측
    - ▶ 회귀(regression)
    - ▶ 분류(classification)
    - ▶ 지도학습 (Supervised Learning) – based on input/output
  - 추천(recommendation)
  - 연관분석
  - 강화학습

- 지도학습은 정답을 예측하는데 사용된다.
- 정답은 목적(target) 변수, 레이블 이라고도 한다
- 예측은 분류와 회귀로 나누어진다.
- 분류
  - 분류(classification)란 어떤 항목(item)이 어느 그룹에 속하는지를 판별하는 기능을 말한다.
  - 두 가지 카테고리를 나누는 작업을 이진 분류(binary classification)라고 하고 세 개 이상의 클래스를 나누는 작업을 다중 분류(multiclass classification)라고 한다.
- 회귀
  - 수치를 예측하는 것을 회귀라고 한다.

- 비지도 학습이란 정답이 없이 데이터로부터 중요한 의미를 찾아내는 머신러닝 기법이다.
  - 군집화: 유사한 항목들을 같은 그룹으로 묶는다.
  - 데이터 변환: 데이터를 분석하기 좋게 다른 형태로 변환한다
  - 주성분분석(PCA): 머신러닝에 사용할 특성의 수를 줄인다.
  - 시각화: 데이터의 속성을 명확하게 시각화하기 위해서 고차원의 특성 값들을 2차원이나 3차원으로 차원을 축소하는 작업



- 어떤 사건이 다른 사건과 얼마나 자주 동시에 발생하는지 파악
- 자주 발생하는 패턴 찾기(상품의 연관성, 취향의 연관성 등 분석)
- 같이 구매한 상품 분석(market basket analysis, 장바구니 분석)
- 상품의 진열 배치 및 상품 프로모션(쿠폰 발행 등)에 활용

- 강화학습(reinforcement learning)은 머신러닝 모델이 어느 방향으로 만들어져야 하는지 방향성만 알려주는 학습 방법
  - 입력 샘플마다 정답을 있어 답을 알려주는 것이 아니지만 시간이 흐르면서 모델이 바람직한 방향으로 가고 있는지를 알려줄 수 있고 이를 통해서 학습하는 방법
- 강화학습에서는 일정 기간동안의 행동(action)에 대해 보상(reward)을 해줌으로써 잘 하고 있는지, 잘 못하고 있는지를 알려주며 학습을 시킨다.
  - 예를 들어 로봇이 혼자 그네를 타는 방법, 전자 게임을 하는 방법, 바둑을 두는 방법의 학습에 사용된다.
  - 2017년에 우리나라 이세돌을 이긴 알파고(Alpha Go) 바둑 프로그램

	머신러닝 유형	알고리즘
지도학습	분류	kNN, 베이지스, 결정 트리, 랜덤포레스트, 로지스틱회귀, 그라디언트부스팅, 신경망
	회귀	선형회귀분석, SVM, 신경망
비지도학습	<u>군집화</u>	K-Means, DBSCAN
	데이터 변환	스케일링, 정규화, 로그변환
	<u>차원축소</u>	PCA, 시각화

- 해결할 문제에 적합한 머신러닝 모델 선택
  - 선형모델, 결정트리, 신경망, SVM, 랜덤포레스트 등
- 훈련 데이터로 모델을 학습
- 모델이 과대적합 (Over fitting)되었는지 또는 과소적합 (Under fitting) 인지를 검증
  - 과대적합이면 모델을 더 일반화 해야 하고, 과소적합이면 모델을 더 상세하게 설계해야 한다.
- 모델을 실제 테스트 데이터에 적용하고 성능을 평가

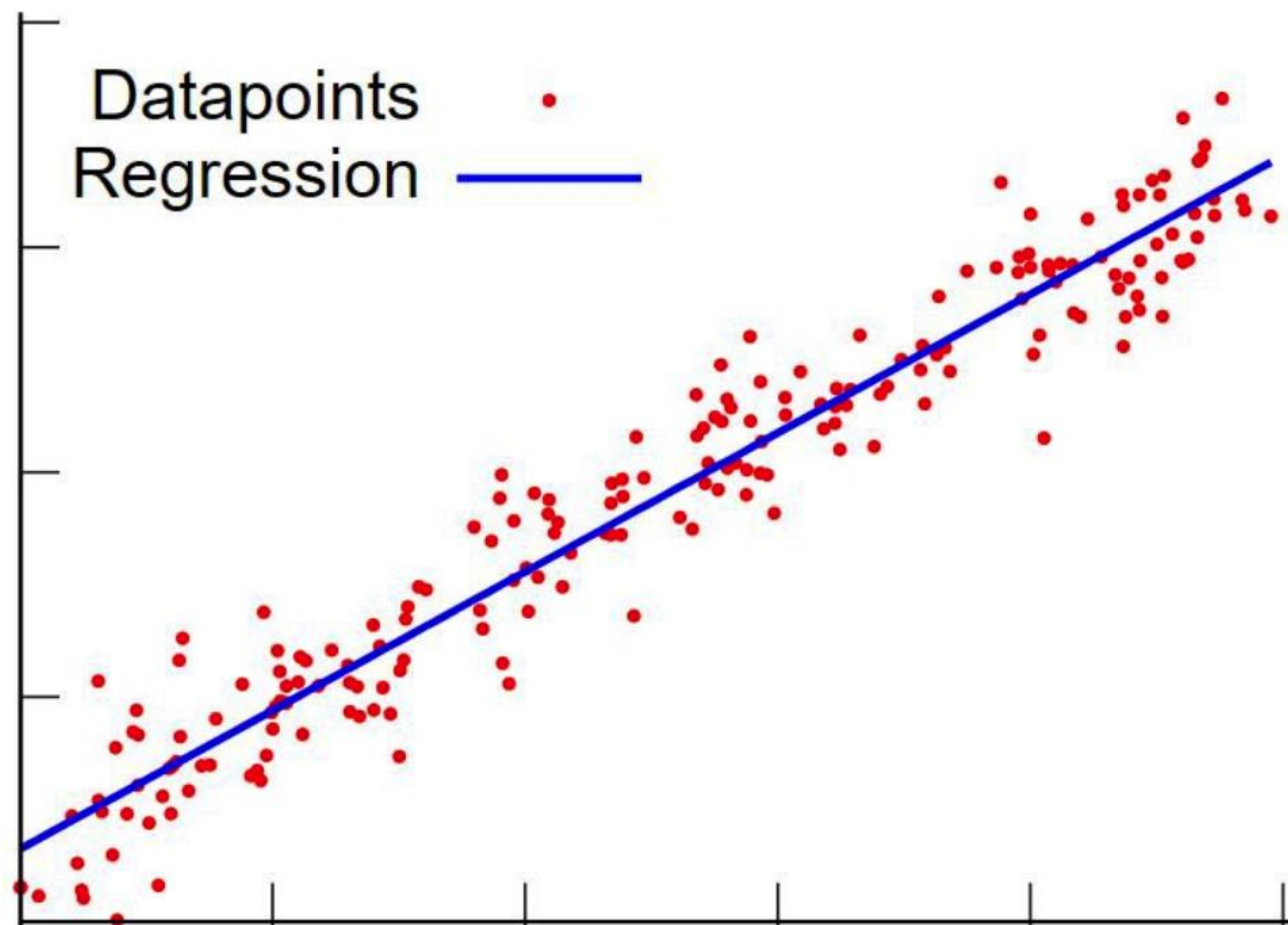
# 머신러닝 동작

- 머신러닝은 모델(model)을 사용한다
  - 스팸 메일을 찾아내는 모델,
  - 누가 게임에서 이길지 예측하는 모델,
  - 내일 날씨를 예측하는 모델
- 과학에서는 어떤 현상을 설명하는 모델로 수식을 주로 사용
  - 모든 질량을 가진 모든 물체는 서로 끌어당긴다는 만유인력 법칙은 두 물체의 질량에 각각 비례하고, 두 물체의 거리의 자승에 반비례하는 수식으로 표현된
- 머신러닝, AI 모델은 데이터 기반의 모델을 사용한다

- 와인 품질 =  $12.145 + (0.00117 \times \text{겨울철 강수량})$   
+  $(0.064 \times \text{재배철 평균기온}) - (0.00386 \times \text{수확기 강수량})$

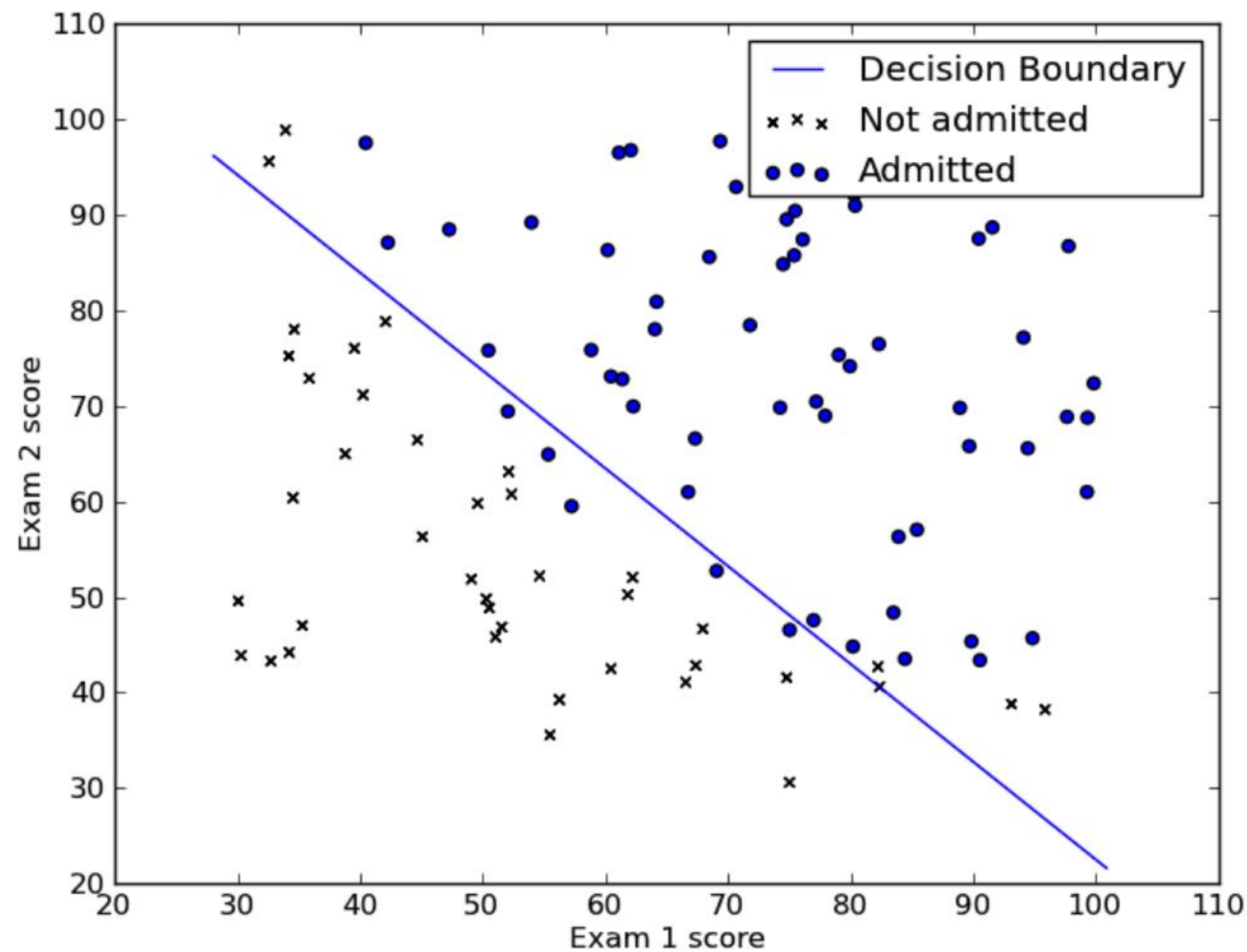


- 선형 회귀(regression)  $y = wX + b$





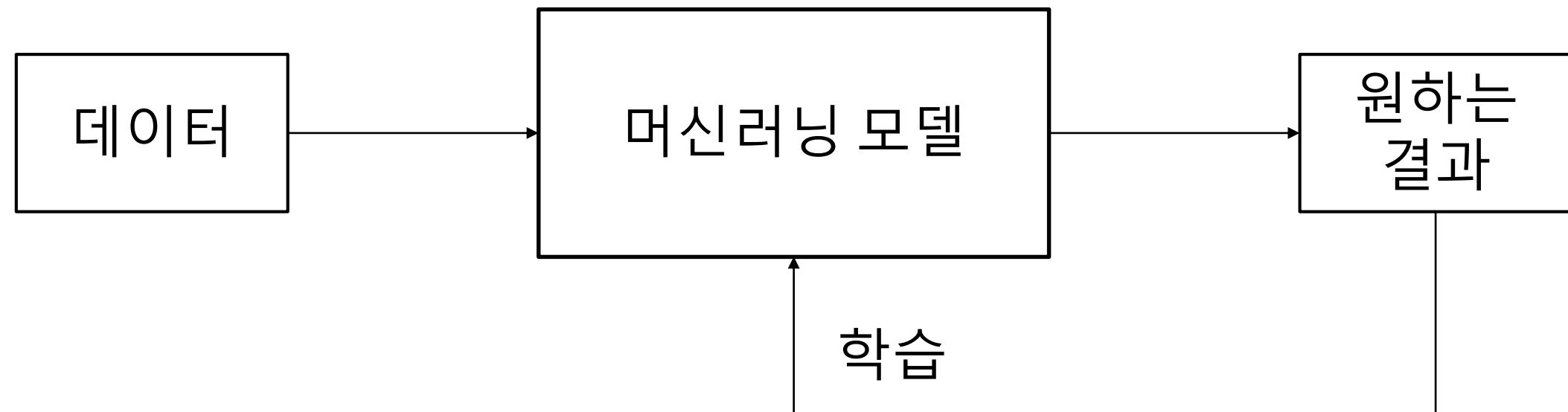
- 선형 분류(classification)  $ay + bx > c$



- 머신러닝에서는 데이터에 기반한 모델을 사용 (학습)
- 현실 세계의 많은 현상은 수식으로 간단히 모델링하기 어렵고 과학적으로 증명할 수는 없다.
- 그러나 성능이 꽤 유용



- 모델 구조: 모델의 동작을 규정하는 방법
- 모델 파라미터: 모델이 잘 동작하도록 정한 가중치 등 계수
  - 예: 머리카락 길이
  - 모델의 구조는 프로그래머가 선택
  - 적절한 파라미터를 찾는 것은 머신러닝 프로그램이 학습하여 찾는다



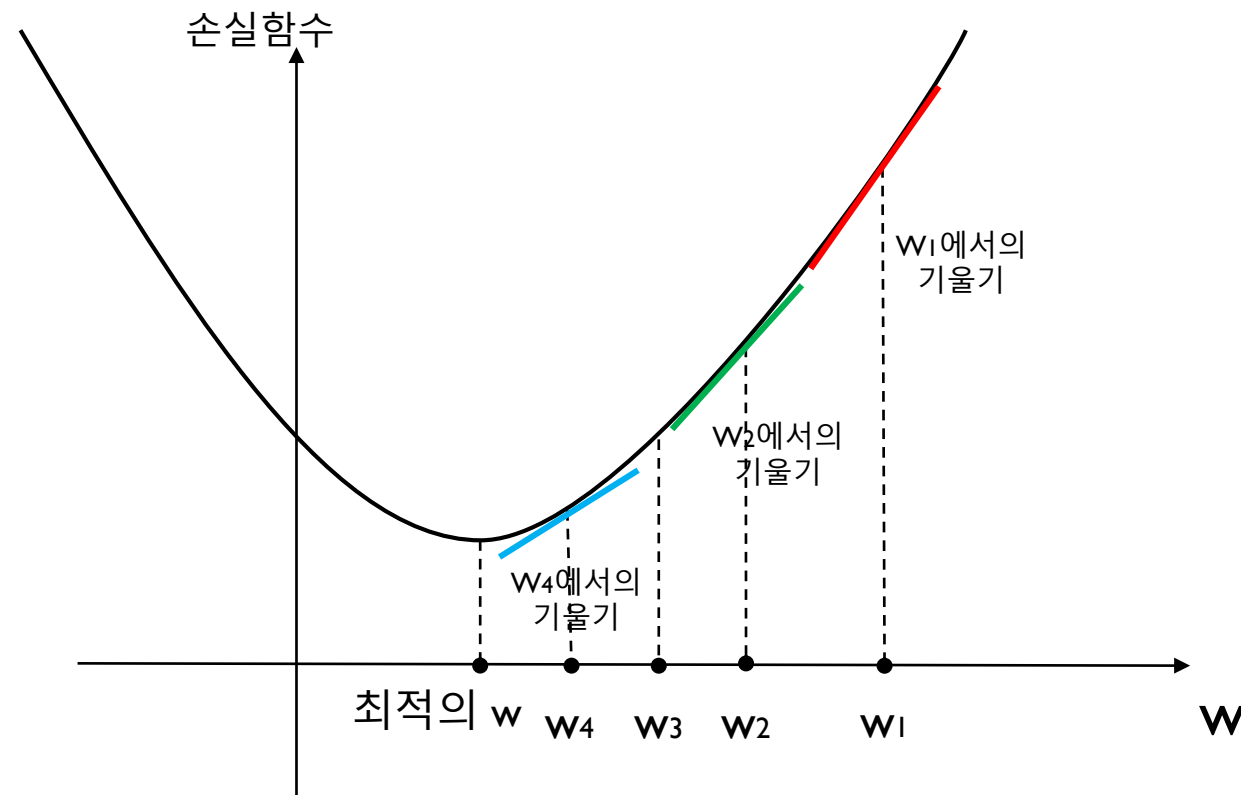
- 모델의 예측값과 실제 값과의 차이, 즉 오차로부터 손실함수 (loss function)을 계산한다
- 이 손실함수를 줄이는 방향으로 모델을 최적화 (학습) 한다
- 회귀분석에서 많이 사용하는 손실함수로는 오차 자승의 합의 평균치(**MSE: mean square error**)

$$MSE = \sum_{k=1}^N (y - \hat{y})^2$$

- N: 배치 크기
- 배치 크기 같은 설정 환경 변수를 *하0/퍼0/라미터*라고 한다.
  - **하이터파라미터**는 사람이 선택하는 변수이며, 기계 학습으로 자동으로 갱신되는 변수는 "**파라미터**"라고 한다.

- 가장 일반적인 최적화 알고리즘: (Gradient Descent)
- 손실함수를 계수에 관한 그래프로 그렸을 때 최소값으로 빨리 도달하기 위해서는 현재 위치에서의 기울기(미분값)에 비례하여 반대방향으로 이동하는 방식

$$W_i = W_{i-1} - \eta \text{Grad}(i)$$



- 학습률: 계수를 업데이트 하는 속도를 조정하는 변수
  - 학습률이 너무 작으면 수렴하는데 시간이 오래 걸리지만 최저점에 도달했을 때 흔들림 없이 안정적인 값을 얻게 되고,
  - 학습률을 너무 크게 정하면 학습하는 속도는 빠르나 자칫하면 최저점으로 수렴하지 못하고 발산하거나 수렴하더라도 흔들리는 오차가 남아있을 수 있다.
- 학습 스케줄(learning schedule) 기법
  - 초기에는 학습률을 크게 정하고 (학습률을 빠르게 학고) 오차가 줄어들면 학습률을 줄여서 안정상태(steady state)의 오차를 줄이는 방법

- 경사하강법을 적용하려면 특성 변수들을 모두 동일한 방식으로 **스케일링**해야 한다.
- 특성 값마다 크기의 편차가 크면 특정 변수에 너무 종속되어 동작할 수 있고 이로 인해 수렴속도가 직선이 되지 않고 오래 걸릴 수가 있다.
- Local minimum 에 머무를 수 있음.
- 배치 사이즈가 커지면 시간이 오래 걸림.



- 배치(Batch) GD (or Mini Batch GD)
  - 일반적으로 배치 GD방식을 많이 사용하는데, 적절한 크기 (10~ 1,000)의 배치단위로 입력 신호를 나누어 경사하강법을 적용하는 방식이다.
- SGD (확률적 경사하강법: Stochastic GD)
  - 한 번에 한 샘플씩 랜덤하게 골라서 훈련에 사용하는 방법이다.
  - 즉 샘플을 하나만 보고 계수를 조정한다. 계산량이 적어 동작속도가 빠르고, 랜덤한 방향으로 학습을 하므로 전역 최소치 (global minimum)를 찾을 가능성이 높아진다.
  - 매 샘플이 너무 랜덤하여 방향성을 잃고 수렴하는데 시간이 오래 걸릴 가능성도 있다. 노이즈가 심함.
  - In Python, **use different Loss function and penalty**
    - ▶ SGDClassifier() for classification
    - ▶ SGDRegressor() for regression