

# 강의 소개

---

Kyungsik Han

# 강의 목표

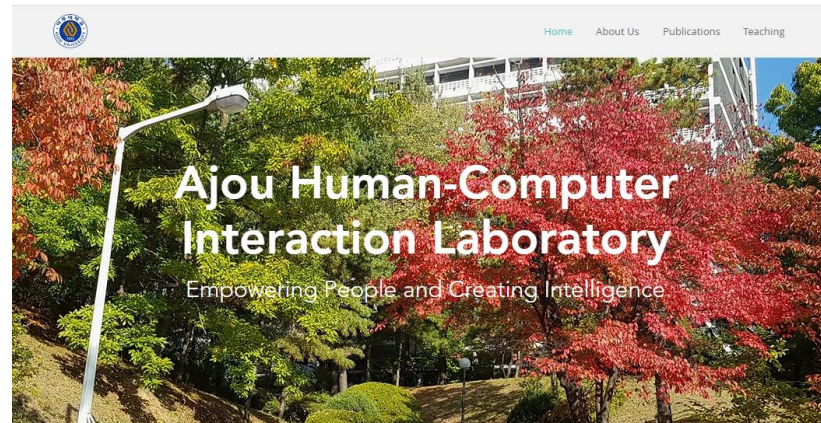
- 머신러닝 알고리즘에 대한 기초적인 내용을 이해하고, 실습을 통해서 머신러닝 작동 원리에 대해서 학습
- 다음과 같은 학습 결과 기대
  - 머신러닝 학습을 위한 python언어 핵심 요소 이해
  - 필수 지도 및 비지도 머신러닝 알고리즘 이해
  - 실습을 통한 머신러닝 모델 구축 과정 및 프로젝트 진행에 대한 이해
  - 중급 이상의 머신러닝 및 딥러닝 학습을 위한 기반 마련

# 강의 특징

- 머신러닝 학습에 필요한 핵심 내용 파악
- 챕터별 이론과 실습의 유기적인 연결을 통한 기초 필수 머신러닝 개념 이해
- 추가 미니 프로젝트 실습을 통한 머신러닝 활용에 대한 이해

# 강의자 소개

- 아주대학교 컴퓨터공학 및 인공지능학과 한경식 교수
- 이메일: kyungsikhan@ajou.ac.kr



We are looking for PhD and MS students!  
We are especially interested in students who can build systems and want to do HCI, VR and data science research ([email us](#)).



아주대학교 소프트웨어학과(컴퓨터공학) 및 데이터사이언스 학과의 인간-컴퓨터 상호작용 연구실 (Human-Computer Interaction Laboratory)에서는 컴퓨터 기술, 인간 활동 및 사회 간의 관계를 연구하고, 결과를 해석하며, 사람에게 도움이 되는 기술 개발 및 디자인 관점을 제시하는 연구를 활발히 진행하고 있습니다.

우리 연구실에서는 디자인, 컴퓨터과학, 사회과학, 심리학 등의 다양한 방법론의 활용, 학제 간 연구 및 교육을 통해 인간의 능력, 목표 및 사회적 환경과 조화되고 향상되는 기술을 이해하고 개발하는 것을 목표로 하고 있습니다.

다수의 국내외 대학과 공동 연구를 진행하고 있으며, 현재 진행 중인 과제는



**Social Computing Research:** social media big data 및 machine/deep learning 기반 사용자 모델링(user modeling) 및 사회 현상 (social phenomena)을 파악하는 연구

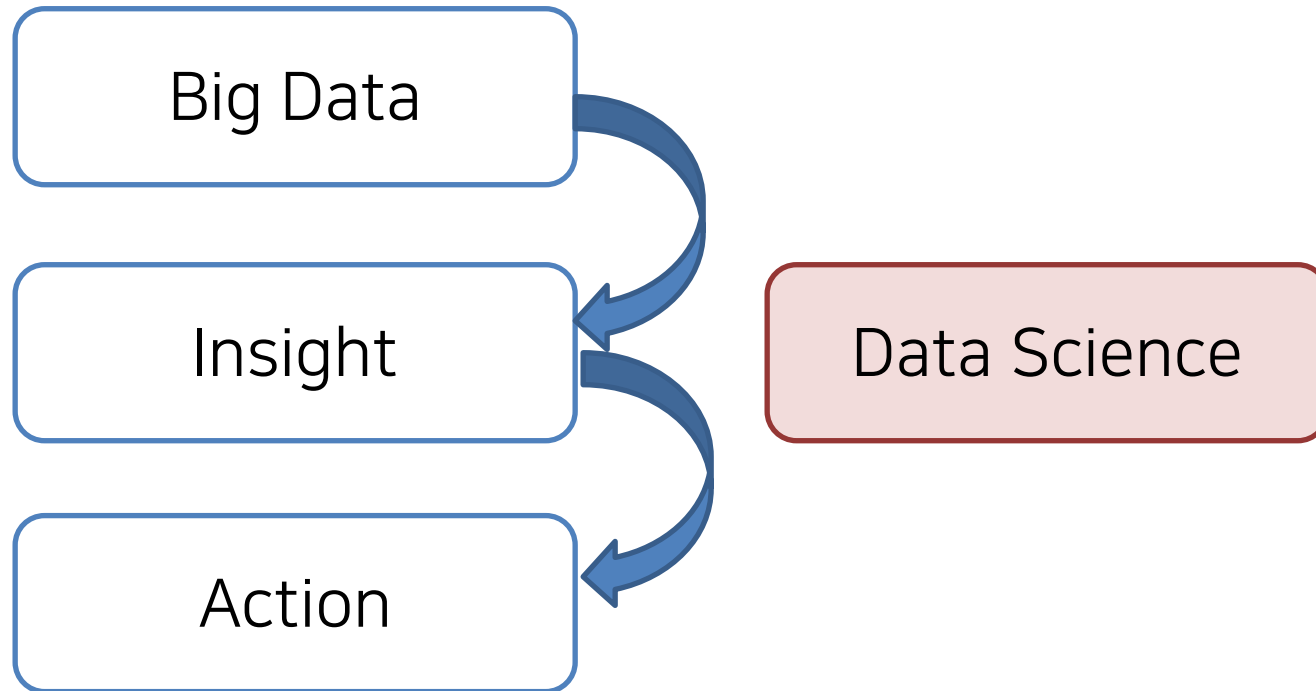


**User-Centered Research:** 도메인과 사용자에 따른 다양한 인간요소를 심리학, 사회학을 기반으로 정량적, 정성적 방법으로 연구

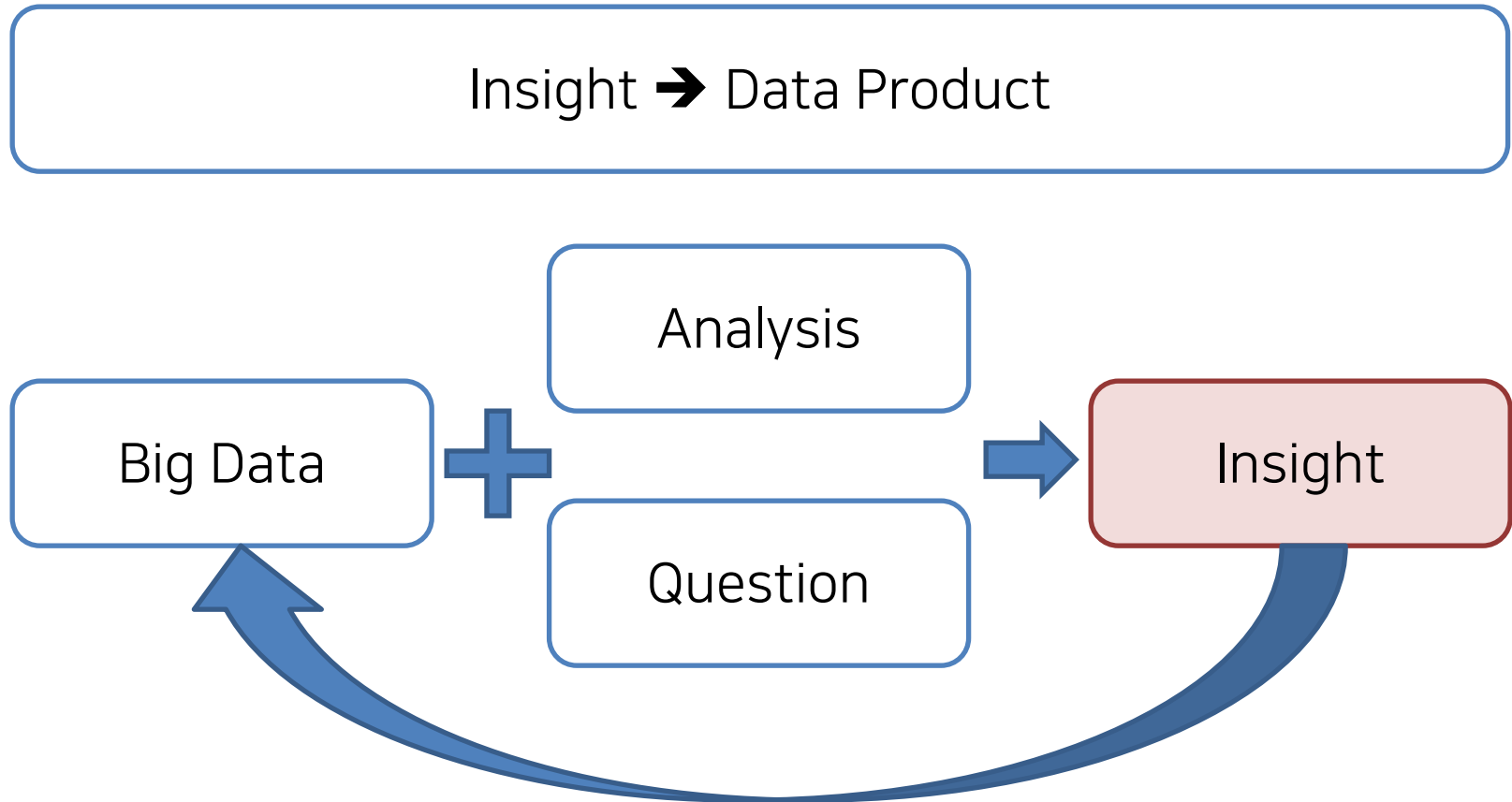
연구실 홈페이지: <https://www.ajouhcil.com/>

# 데이터 과학 소개

# What is Machine Learning?

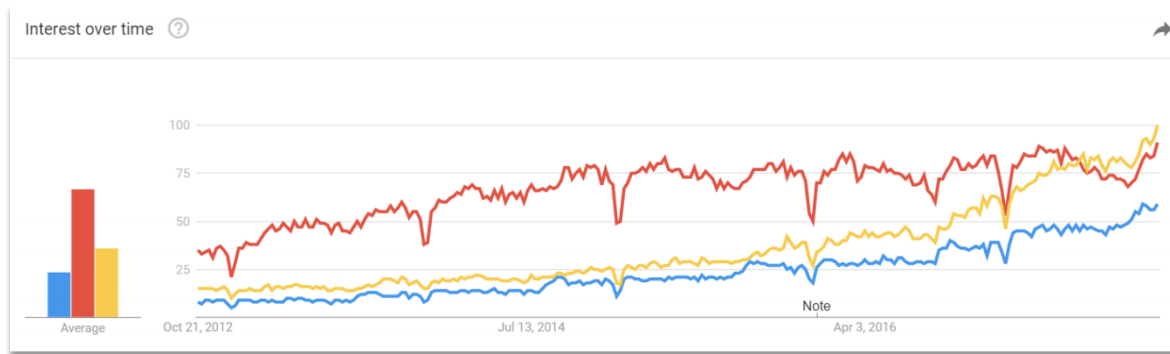


# What is Machine Learning?



# Trend

## Google Trend: Data Science & Big Data & Machine Learning



Related queries ?

T

- 1 computer science
- 2 what is data science
- 3 what is science
- 4 what is data
- 5 data analytics

100

70

70

70

55

■

■

■

■

■

Related queries ?

T

- 1 big data analytics
- 2 analytics
- 3 data analytics
- 4 hadoop
- 5 big data hadoop

100

100

100

65

65

■

■

■

■

■

Related queries ?

Top ▼ ↗

- 1 python
- 2 python machine learning
- 3 machine learning pdf
- 4 machine learning google
- 5 machine learning algorithms

100

100

80

70

60

■

■

■

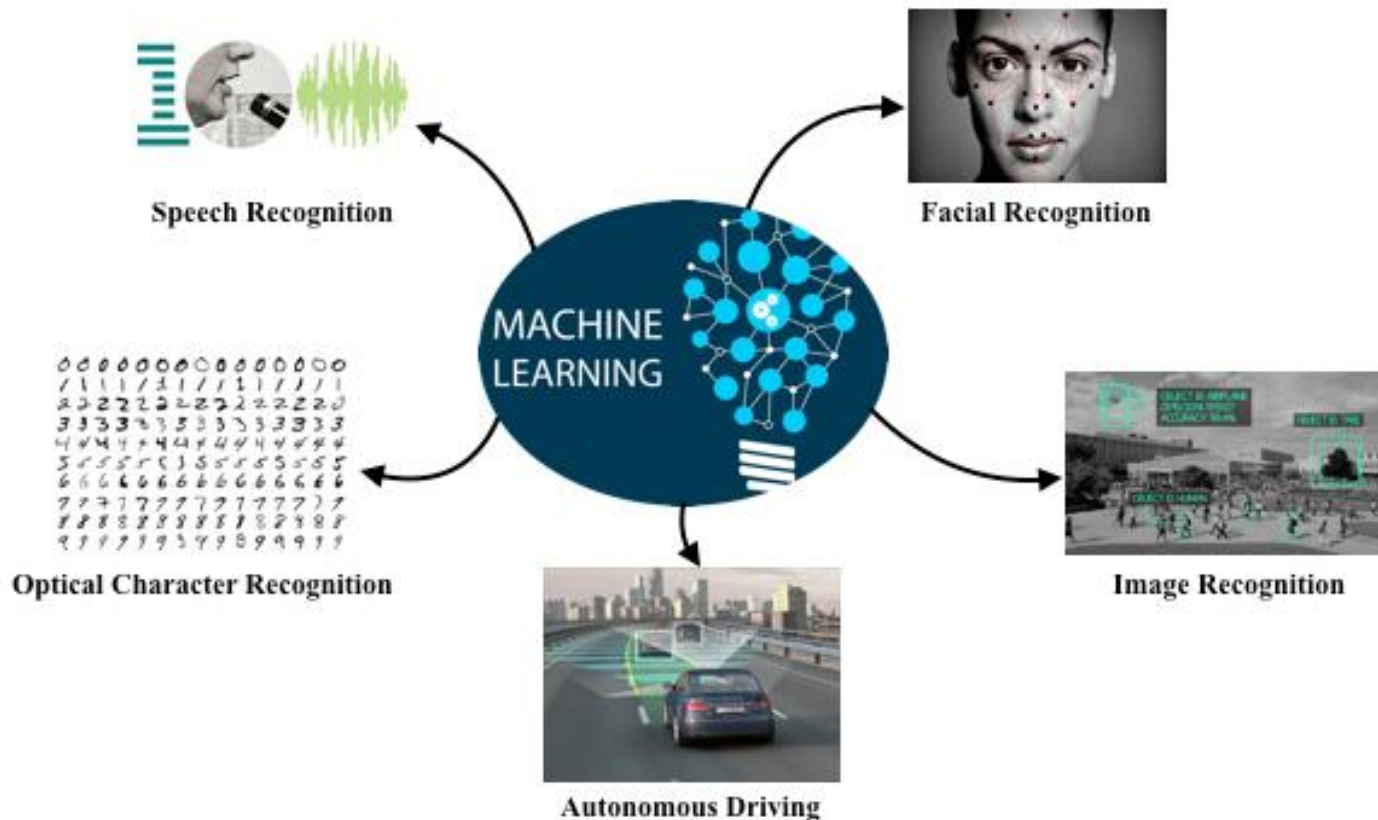
■

■



# 적용 사례

- Web search, Computational biology, Finance, E-commerce, Space exploration, Robotics, Information extraction, Social networks, Debugging
- Your own area!



# 도메인 예제

Insight → Data Product

소비자  
인적정보

이전  
소비 상품

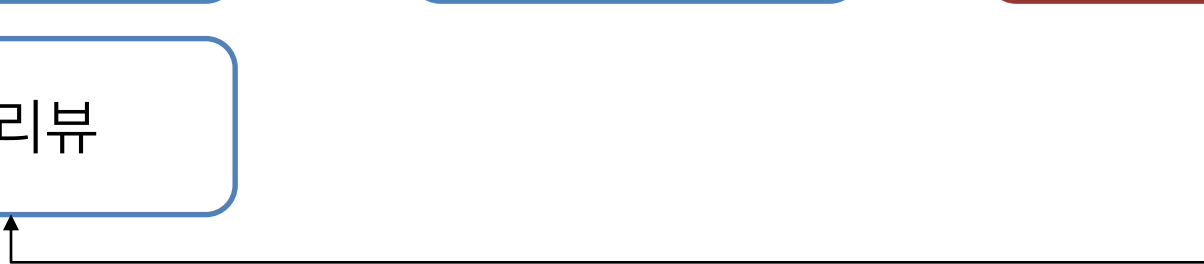
책 리뷰



소비자가  
좋아할만한 책은?



책 추천



amazon.com<sup>®</sup>

# 도메인 예제

책 판매를 위한 잠재적 고객 찾기

새로운 책 광고

고객 책  
선호 모델



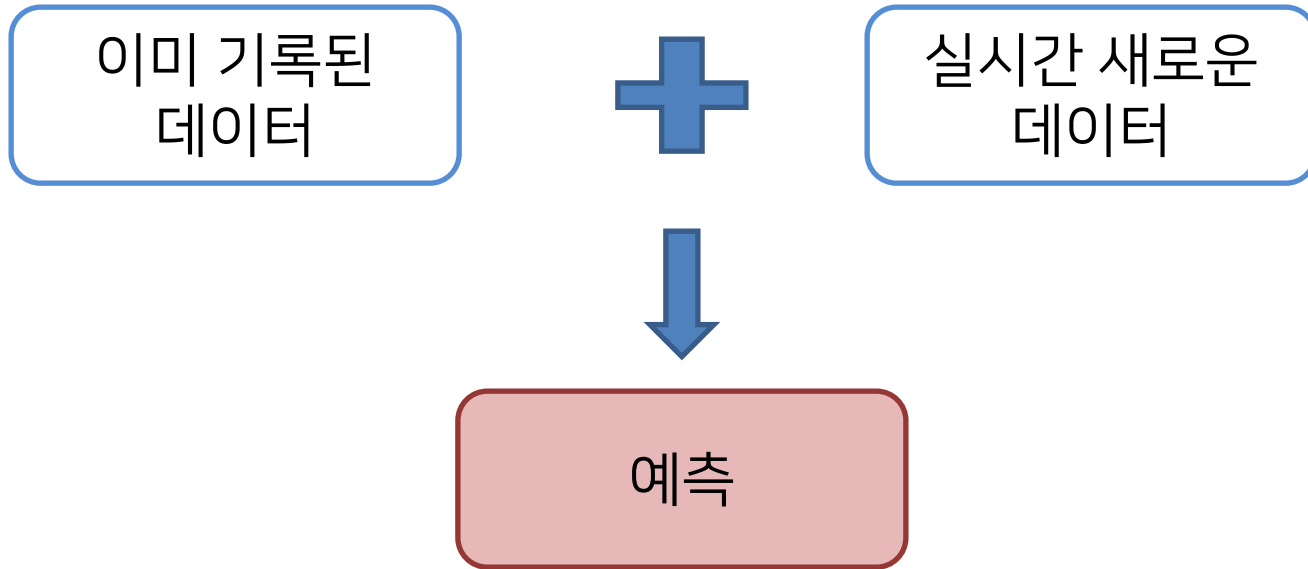
새로운 책  
정보

이 책을 좋아할  
고객은?

추천할만  
고객에게 추천

amazon.com<sup>®</sup>

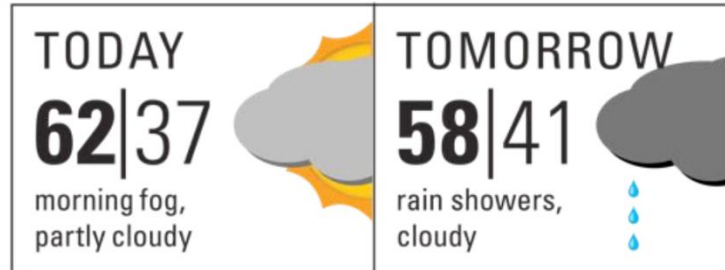
# 실행할 수 있는 정보



amazon.com<sup>®</sup>

# 실행할 수 있는 정보

예측



실행



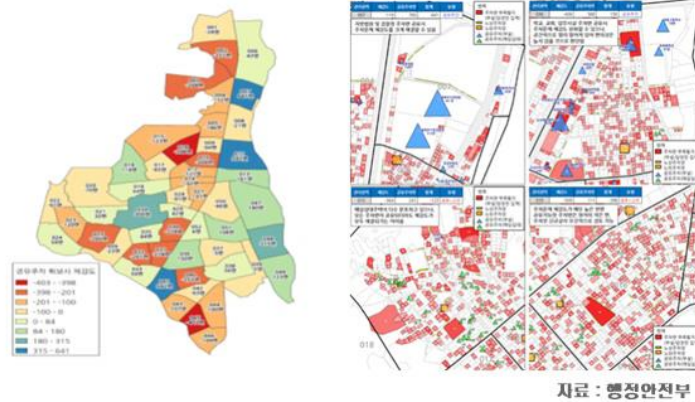
# Data Everywhere

Smartphone, Tablets, PC, SNS,  
Game, ...



공공 빅데이터

데이터 기반 블록별 주차공간 운영 현황 및 주차 지도



의료 데이터

보건 의료 빅데이터 활용 방안

보건산업 분야에서 빅데이터 활용을 통한 다양한 비즈니스 모델



금융 데이터

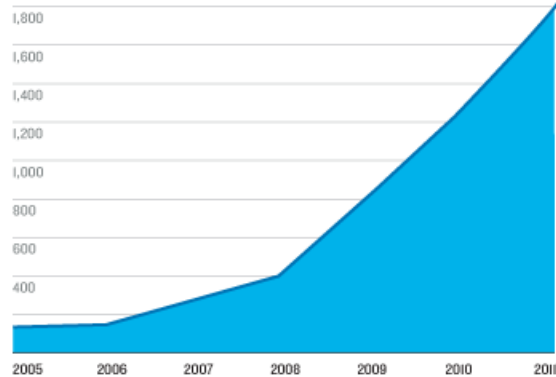


# 기하급수적으로 증가하는 데이터량



## Digital Information Created Each Year, Globally

2,000 BILLION GIGABYTES



Sources: IDC, Radicati Group, Facebook, TR research, Pew Internet

**2,000%**

Expected increase in global data by 2020

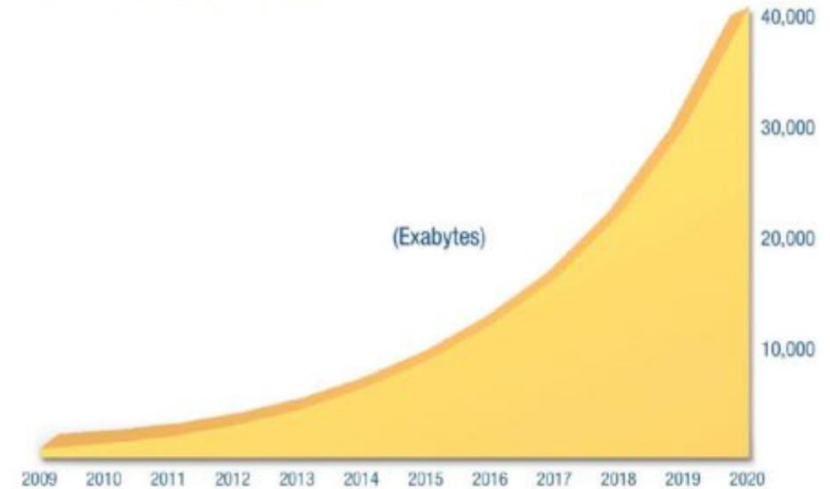
**III Megabytes**

Video and photos stored by Facebook, per user

**75%**

Percentage of all digital data created by consumers

## The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020



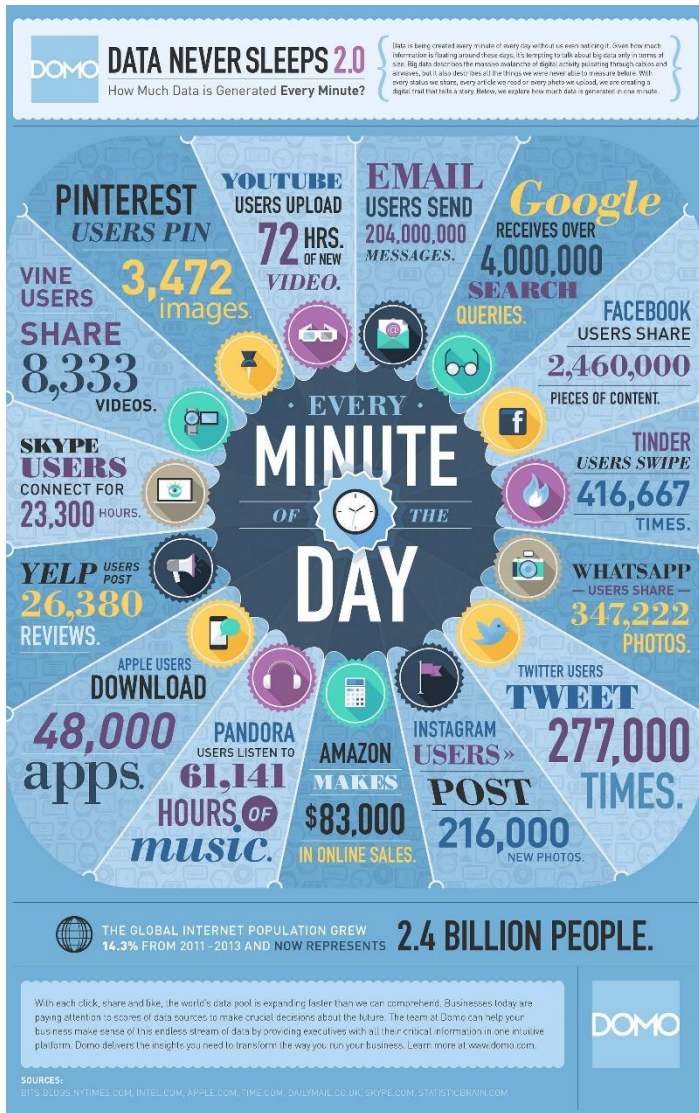
This IDC graph predicts exponential growth of data from around 3 zettabytes in 2013 to approximately 40 zettabytes by 2020. An exabyte equals 1,000,000,000,000,000 bytes and 1,000 exabytes equals one zettabyte. Source: IDC's Digital Universe Study, December 2012, <http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf>.

Source: IDC's Digital Universe Study

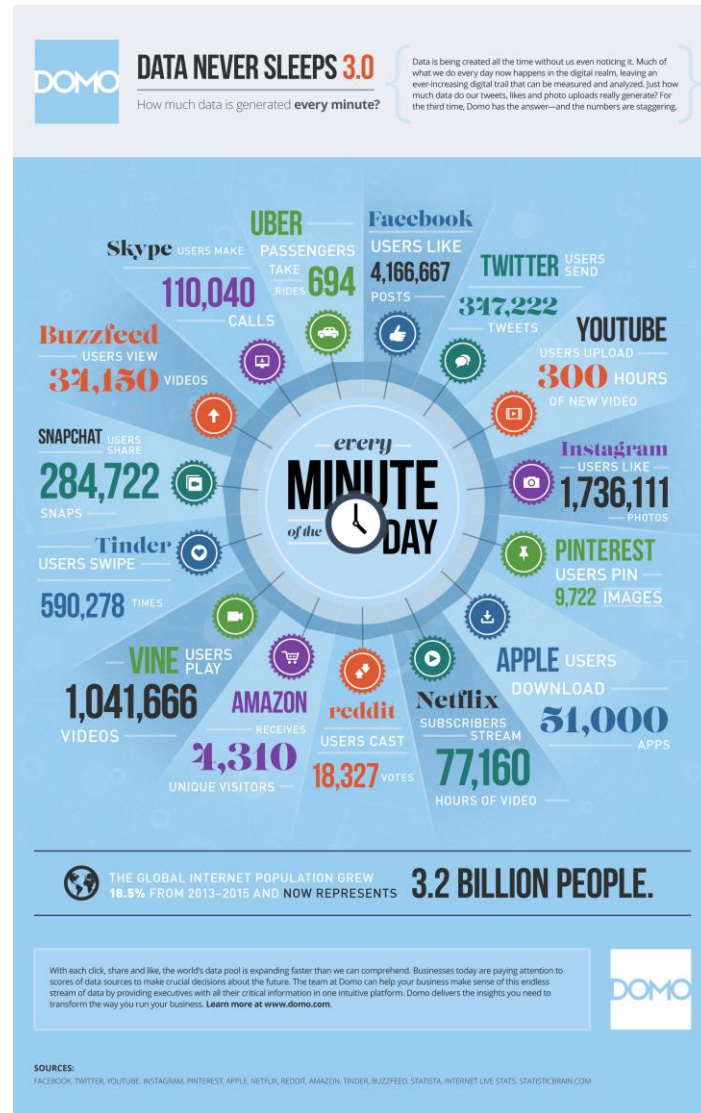
Source: IDC's Digital Universe Study



# 생산되는 데이터 량 (분당)



Source: DOMO

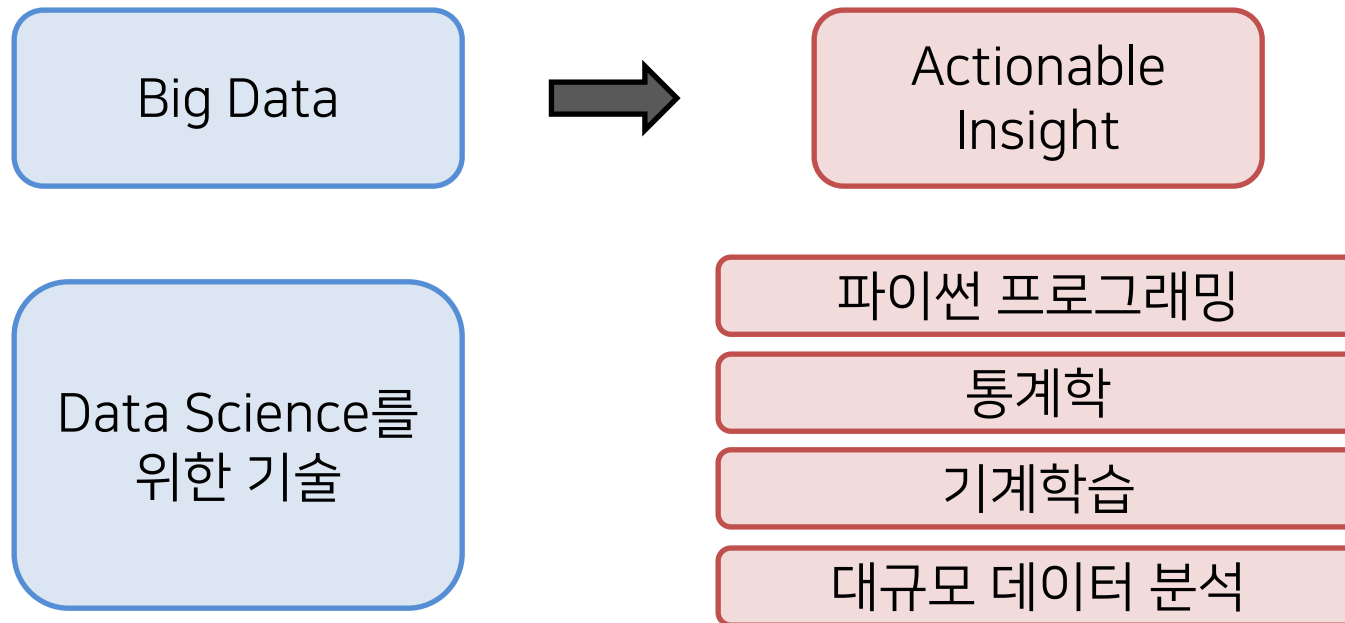


Source: DOMO



# 데이터 과학을 위한 기술

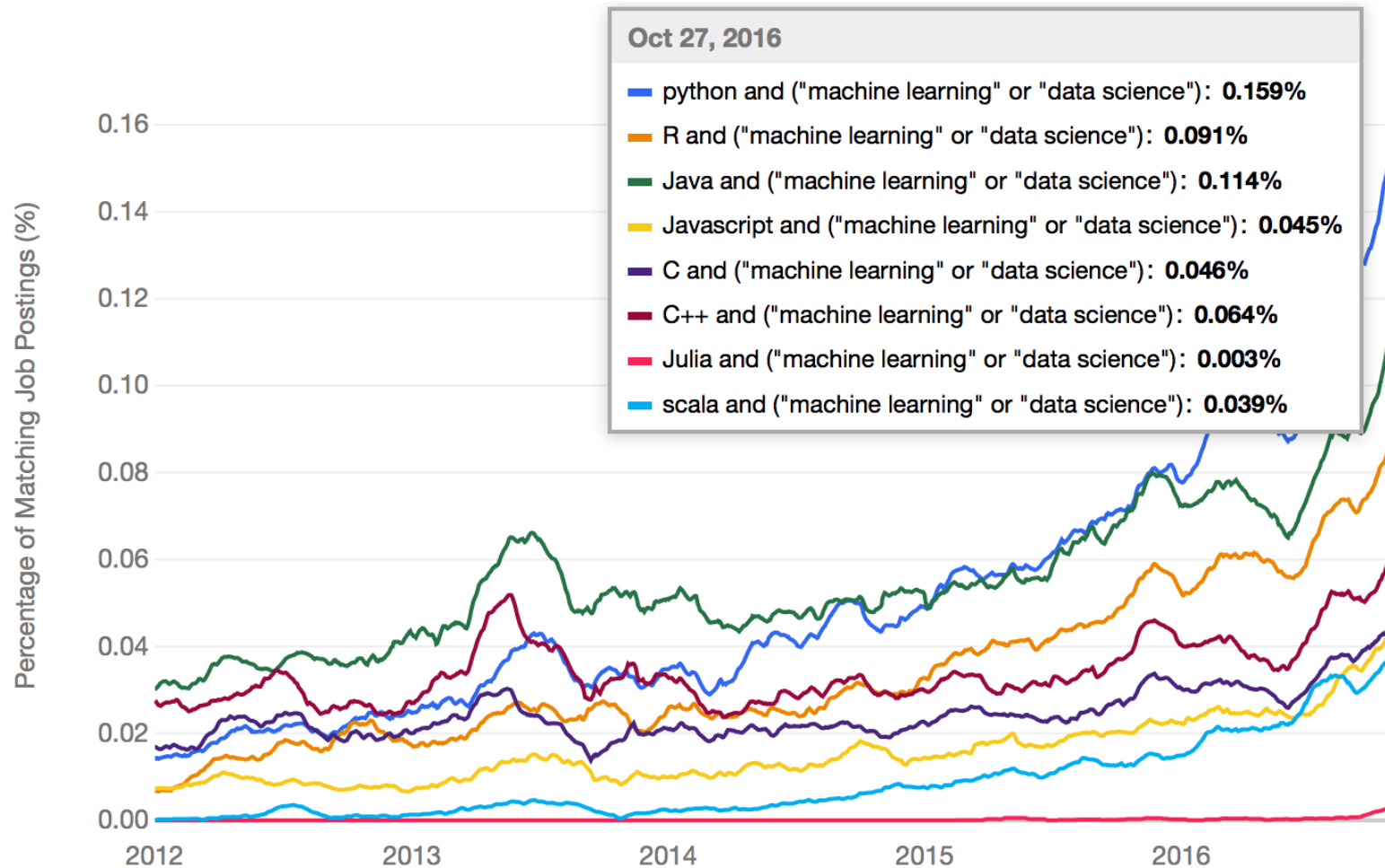
- Python (파이썬) 프로그래밍
- 통계학
- 기계학습
- 대규모 데이터 분석



# 데이터 과학

- 데이터에 대한 열정
- 문제를 분석으로 연결
- 공학적인 솔루션
- 호기심을 보여줌
- 팀원들과의 커뮤니케이션

# 데이터 과학: 언어

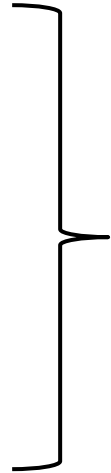


<https://www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html>

# 데이터 분석 모델

- 통계 분석

- 군집 분석
- 연관 분석
- 분류 분석
- 회귀 분석



데이터마이닝, 머신러닝, 딥러닝

- 시각화

# 파이썬을 쓰는 이유

- 읽고 배우기가 쉬움
- 활발한 온라인 커뮤니티
- 지속적인 라이브러리 업데이트
- 새로운 라이브러리 제공
  - 데이터 관리, 분석
  - 시각화
- 데이터사이언스를 위한 모든 단계에 최적화
- Notebooks



# Example

## (축구 데이터 분석)

# 케이스 스터디 1: 축구 데이터

- 의미있는 축구 그룹 생성
- 내가 선호하는 선수와 비슷한 성향을 가진 선수 찾기
- 분석을 통한 팀 구성



# 데이터사이언스 프로젝트에서의 중요 단계





# 다양한 데이터 수집

- 데이터베이스
  - 관계형 데이터베이스
  - 비관계형 데이터베이스 (NoSQL)
- 텍스트 파일
  - CSV, Text
- 실시간 피드
  - 센서 데이터
  - 온라인 플랫폼
    - 트위터
    - 실시간 날씨 정보



# 데이터 준비: 통계를 활용

```
In [8]: df.describe().transpose()
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
id	183978.0	91989.500000	53110.018250	1.0	45995.25	91989.5	137983.75	183978.0
player_fifa_api_id	183978.0	165671.524291	53851.094769	2.0	155798.00	183488.0	199848.00	234141.0
player_api_id	183978.0	135900.617324	136927.840510	2625.0	34763.00	77741.0	191080.00	750584.0
overall_rating	183142.0	68.600015	7.041139	33.0	64.00	69.0	73.00	94.0
potential	183142.0	73.460353	6.592271	39.0	69.00	74.0	78.00	97.0
crossing	183142.0	55.086883	17.242135	1.0	45.00	59.0	68.00	95.0
finishing	183142.0	49.921078	19.038705	1.0	34.00	53.0	65.00	97.0
heading_accuracy	183142.0	57.266023	16.488905	1.0	49.00	60.0	68.00	98.0
short_passing	183142.0	62.429672	14.194068	3.0	57.00	65.0	72.00	97.0
volleys	181265.0	49.468436	18.256618	1.0	35.00	52.0	64.00	93.0
dribbling	183142.0	59.175154	17.744688	1.0	52.00	64.0	72.00	97.0

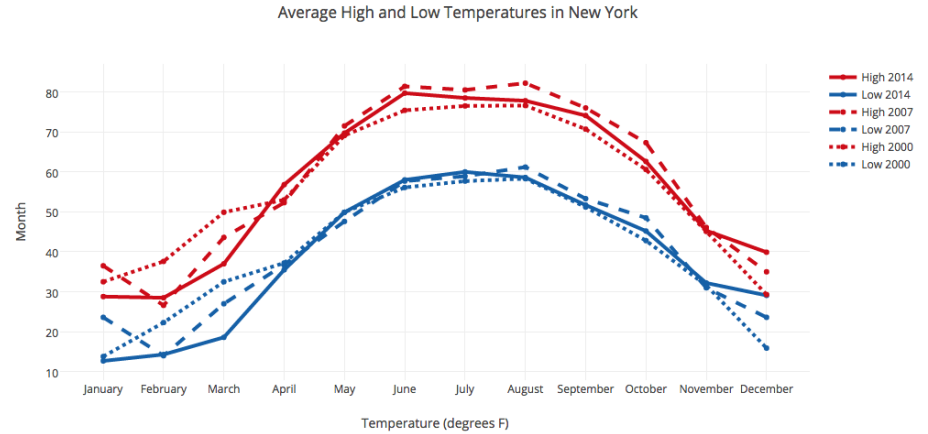
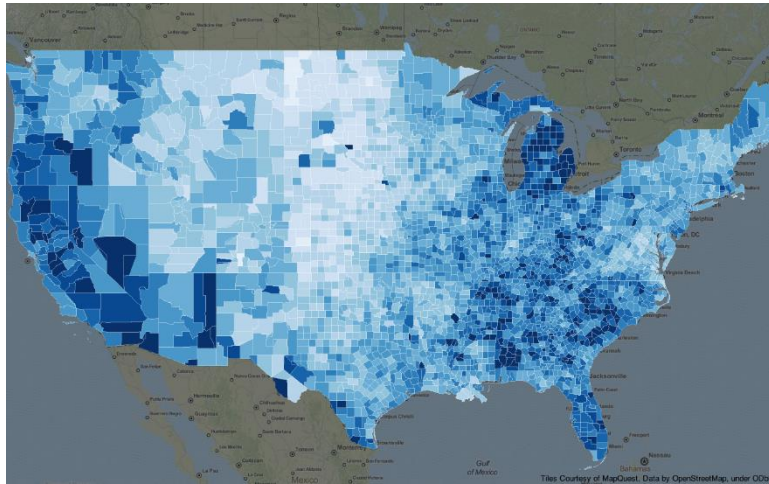
# 데이터 정리

- 데이터 정리가 필요한 이유?
  - 누락된 데이터
  - 필요없는 데이터
  - NULL 데이터
- 데이터 정리 방법
  - 정리하고자하는 데이터 삭제
  - Impute these entries with a counterpart
    - 평균값
    - 0, -1 과 같은 값

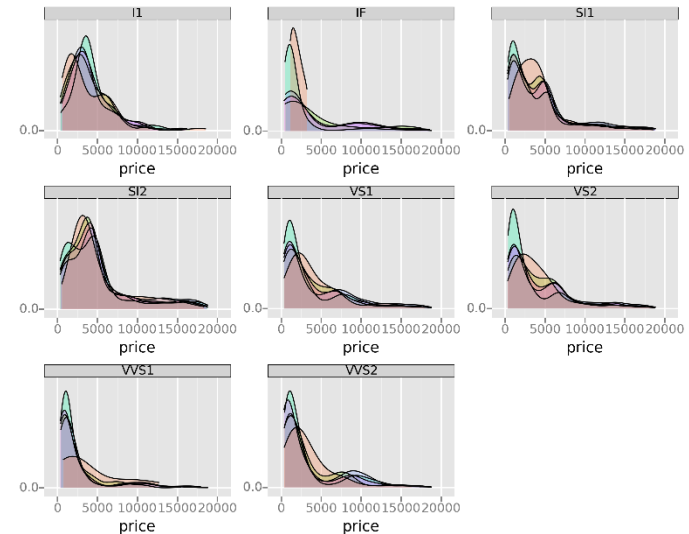
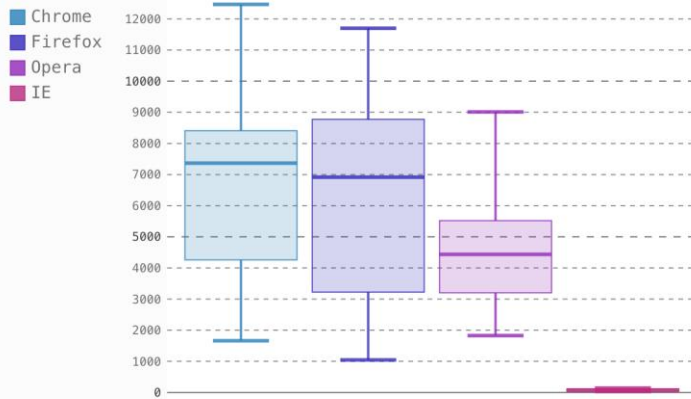
```
#is any row NULL ?  
  
rows = df.shape[0]  
df.isnull().any().any(), df.shape
```

```
# Fix it  
  
df = df.dropna()
```

# 데이터 시각화



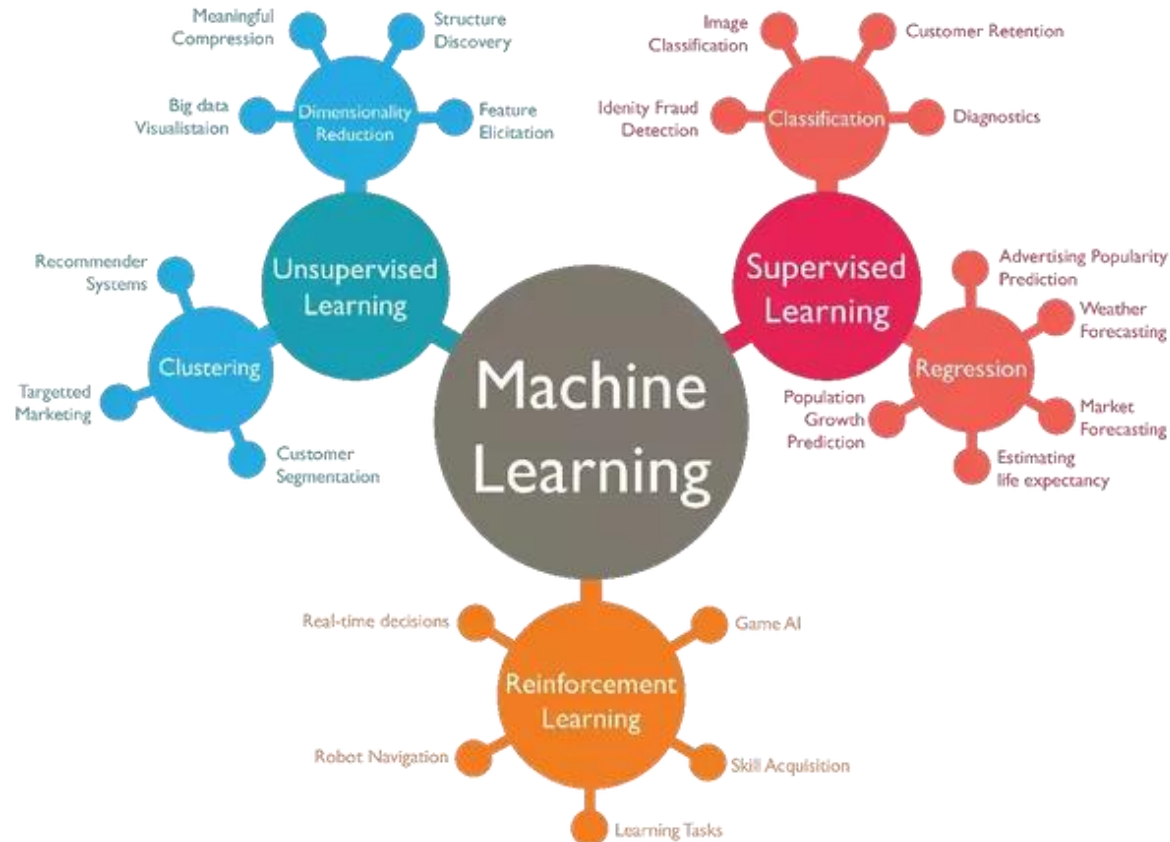
V8 benchmark results



시각적인 분석 가능

# 모델링을 위한 알고리즘 선택

- 지도 학습(Supervised Learning)
- 비지도 학습(Unsupervised Learning)



# 라이브러리 활용

## ■ scikit-learn (<http://scikit-learn.org/stable/>)



**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

# 축구 데이터 분석

- 어떤 내제적인 요소를 바탕으로 축구 선수들을 그룹핑할 것 인가?
  - Agility
  - Reaction Time
  - Shot Power
  - Sprint Speed
  - Hair Style?
  - Movies the player likes?
- 기존 요소를 바탕으로 새로운 요소를 만들 수 있음
  - $f(\text{shot power, reaction time})$



# 축구 데이터 분석: 군집화 (Clustering)

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large <code>n_samples</code> , medium <code>n_clusters</code> with <code>MiniBatch</code> code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium <code>n_samples</code> , small <code>n_clusters</code>	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large <code>n_samples</code> , medium <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large <code>n_clusters</code> and <code>n_samples</code>	Large dataset, outlier removal, data reduction.	Euclidean distance between points

<http://scikit-learn.org/stable/modules/clustering.html>



# 파이썬에서 k-Means clustering

```
from sklearn.cluster import Kmeans
```

```
...
```

```
Y = KMeans(n_clusters=3, random_state=random_state).fit_predict(X)
```

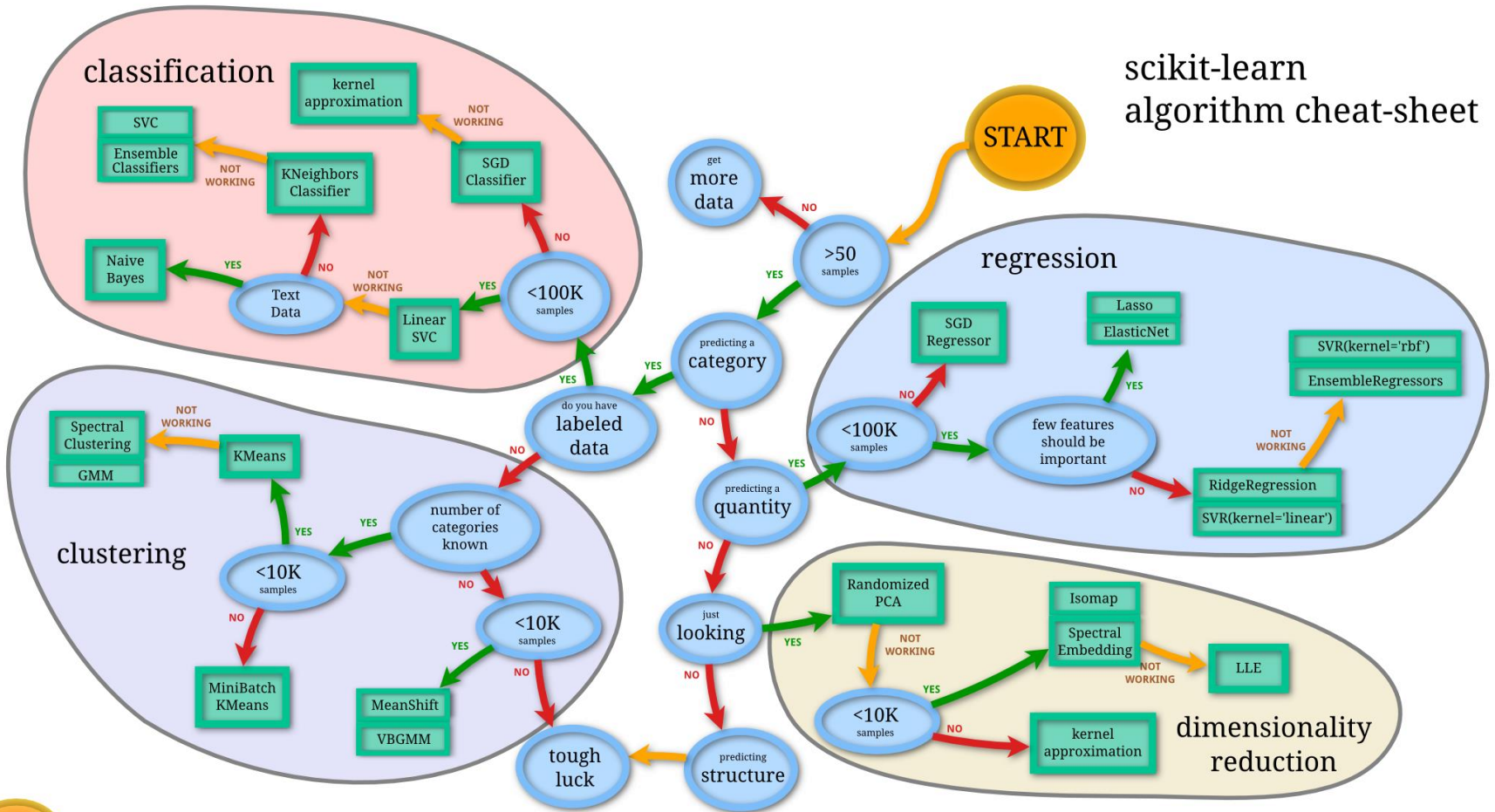
```
...
```

# 군집화 결과 해석

- 한 그룹당 선수는 몇 명인가?
  - 그룹이 너무 적은가? 너무 많은가?
  - 그룹을 나누는 기준은 무엇인가?
- 군집화에 사용했던 요소를 변경해주어 새로운 요소를 도출
  - 새로운 요소는 군집화 결과에 어떤 영향을 미치는가?
  - 추가적인 데이터 수집이 필요한가?

# 좋은 알고리즘을 찾는 방법

scikit-learn  
algorithm cheat-sheet



# 다음 영상에서 배울 내용

- 데이터 관련 내용 학습
  - 수집
  - 종류
  - 형식
  - 타입
  - 행렬
  - 품질

수고하셨습니다.