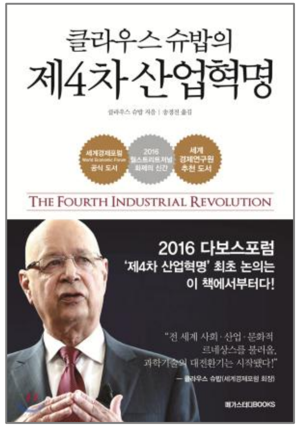
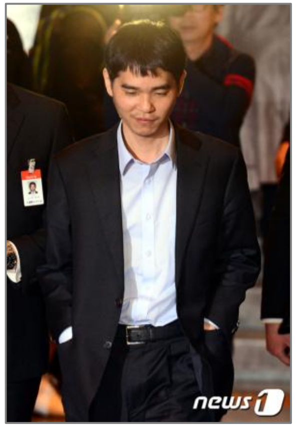
내가 해온 연구에 데이터 분석 끼얹기

Attaching Data Analysis on the Research I’ve Working On

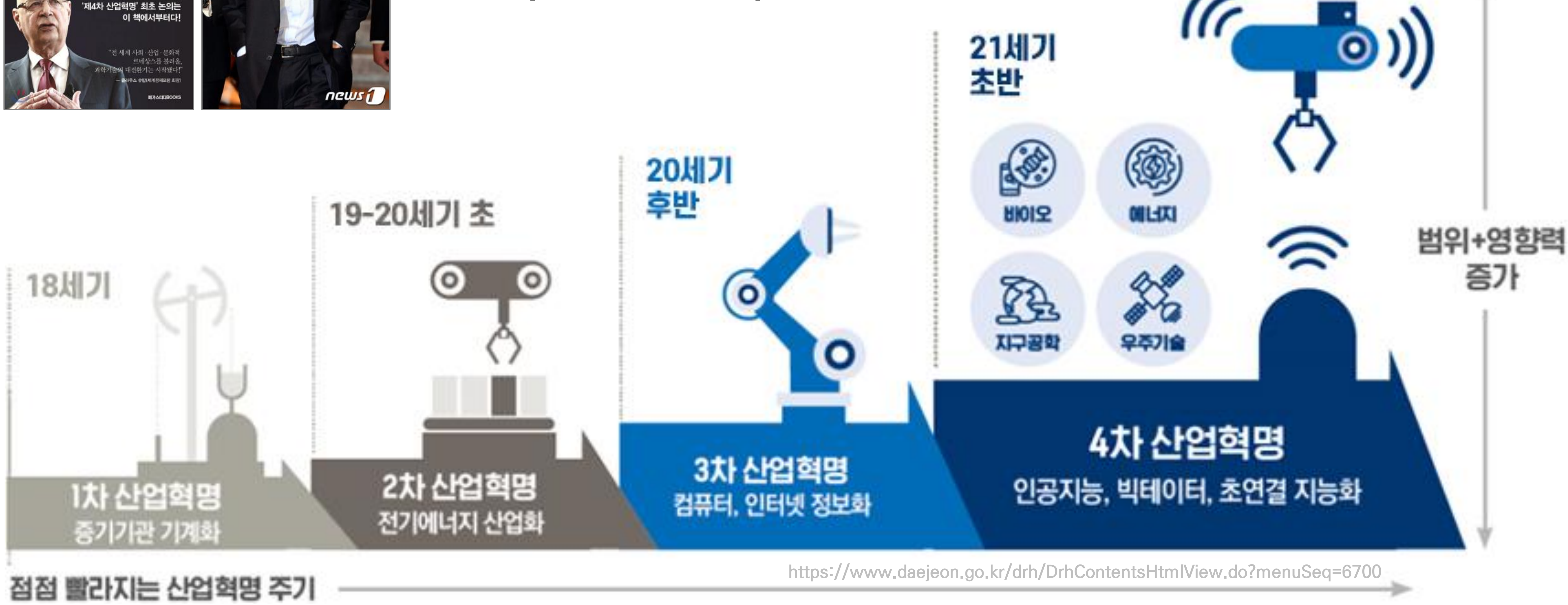
KIER 이제현 선임연구원

N차 산업혁명





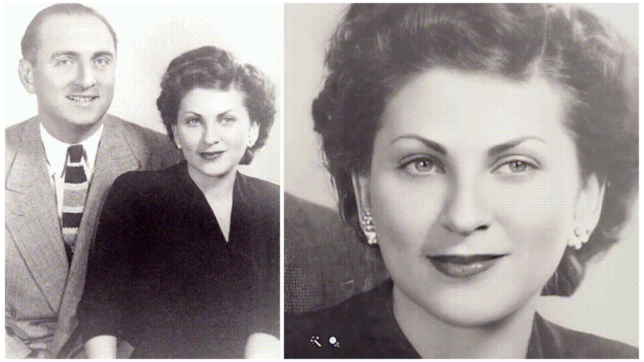
- 2016
  - 4차 산업혁명 (Claus Schwab, Davos Forum)
  - AlphaGo (DeepMind)



N차 산업혁명

- 같은 시대 다른 풍경

사진으로 동영상 제작 : Deep Nostalgia



https://www.myheritage.co.kr/

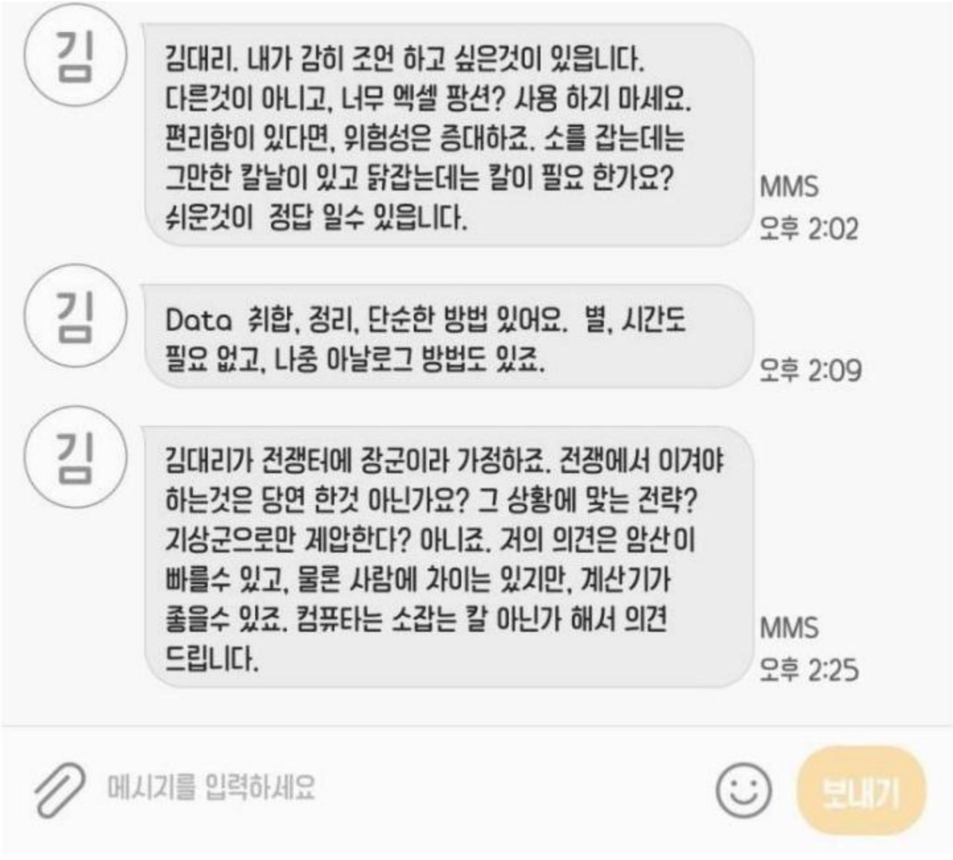
자율주행차 : Tesla

https://www.tesla.com/

문장으로 디자인 제작 : “아보카도 닮은 의자” DALL.E

https://openai.com/blog/dall-e/

엑셀 대신 계산기 사용 권유 : 한국의 모 상사



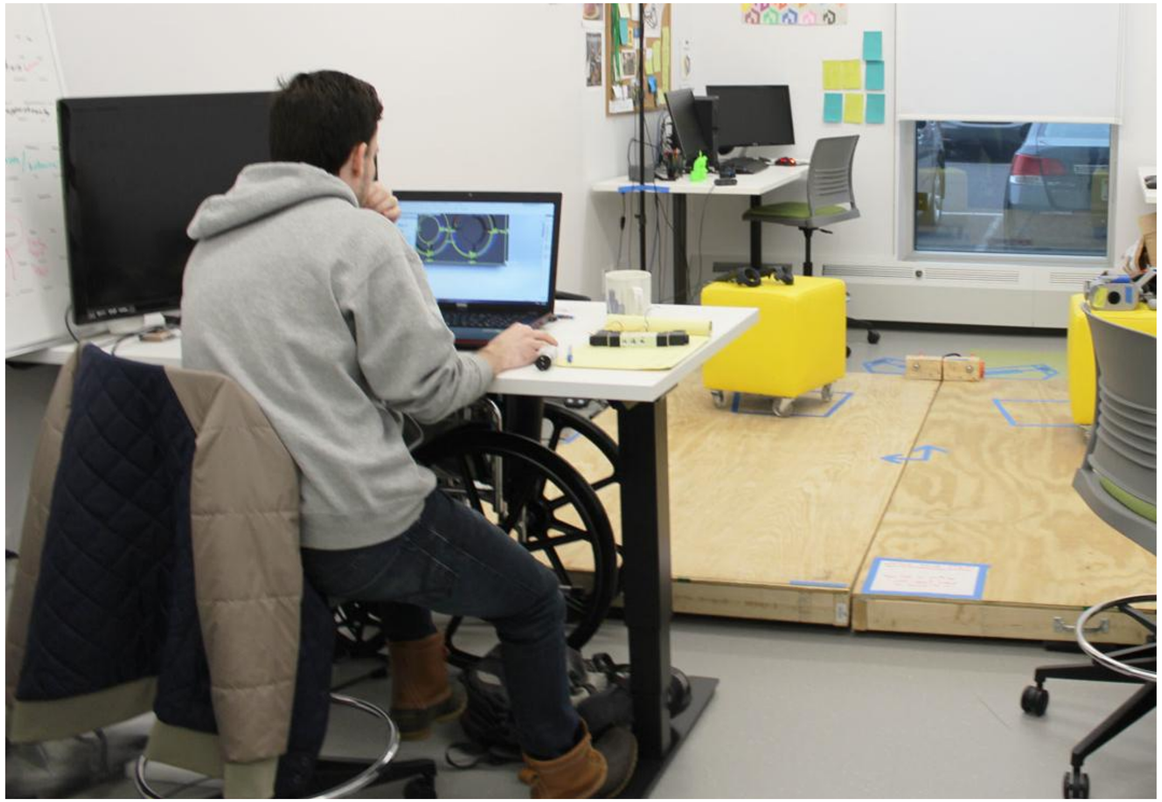
인터넷 커뮤니티 캡처

Two Labs

WET LAB vs DRY LAB

A wet lab is one where drugs chemicals, and other types of biological matter can be analyzed and tested by using various liquids.

A dry lab environment focuses more on applied or computational mathematical analyses via the creation of computer-generated models or simulations.



https://www.crbgroup.com/insights/science-technology/lab-design, https://americangardener.net/difference-between-dry-lab-and-wet-lab/, https://southerntierincubator.com/dry-lab/

Snapshots of Wet Laboratory

https://phdkim.net/, https://www.chosun.com/site/data/html_dir/2019/07/26/2019072600329.html, https://www.ibric.org/, https://www.hankookilbo.com/News/Read/201810241544359365

Robot Chemist (Univ. of Liverpool, 2020)

- 8일간 688회 실험 : 휴식 없이 풀타임 근무. 인간보다 1000배 빠름
- H/W 비용 15만달러 + S/W 3년 〈 정직원 2명 (평균 연봉 95,091천원)

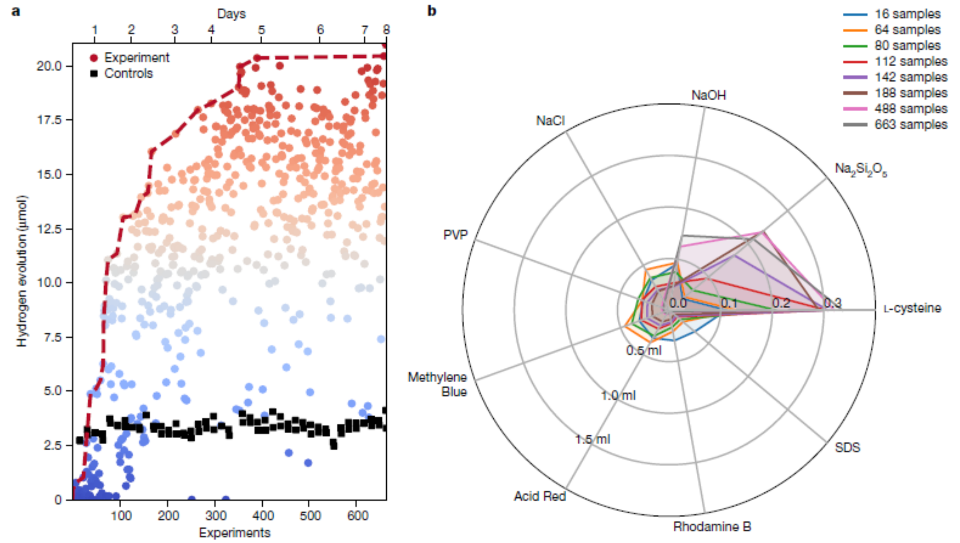


Fig. 3 | Output from the autonomous robotic search. a, Plot showing hydrogen evolution achieved per experiment in an autonomous search that extended over 8 days. Sixteen experiments were performed per batch, along with two baseline controls. The baseline hydrogen evolution was 3.36 ± 0.30 μmol (black squares). The maximum rate attained after 688 experiments was 21.05 μmol h^{-1}. The robot made 319 moves between stations and travelled a total distance of 2.17 km during this 8-day experiment. **b**, Radar plot showing the evolution of the average sampling of the search space in millilitres; the scale denotes the fraction of maximum solution volume dispensed. The starting conditions (Batch 1) were chosen randomly. The best catalyst formulation found after 43 batches contained P10 (5 mg), NaOH (6 mg), L-cysteine (200 mg) and $Na_2Si_2O_5$ (7.5 mg) in water (5 ml).

Burger, B., Maffettone, P.M., Gusev, V.V. et al. A mobile robotic chemist. Nature 583, 237–241 (2020). https://doi.org/10.1038/s41586-020-2442-2

* 평균 연봉: 공공기관경영정보 공개시스템 2020년 기준

신소재 개발 자율 실험

- 삼성전자 종합기술원

AI set for material breakthroughs

The institute is also working on autonomous R&D, concentrating on autonomous material development (AMD). SAIT is working on an AI algorithm that after learning from experimental data sets, will be capable of driving robot synthesis tools.

Autonomous materials development (AMD) at SAIT. © SAMSUNG

"In about 10 to 20 years, SAIT hopes to reach the point where all the synthesis can be done by machine and human researchers can instead focus on things that require superior creativity," Hwang says.

"It's a promising way to sift through the many possible combinations of elements in different configurations for new materials, a task that would present a huge analysis challenge for humans," says InTaek Han, SAIT senior vice president, who is leading the AMD project.

"AI can utilize enormous datasets in a very short time and search unexplored experimental spaces that humans would have never considered," Han says.

The boost from the AI platform will add to an already impressive output of materials research successes — centred around quantum dots, 2D materials, and other nanostructures — from Han's team.

https://www.nature.com/articles/d42473-020-00239-0

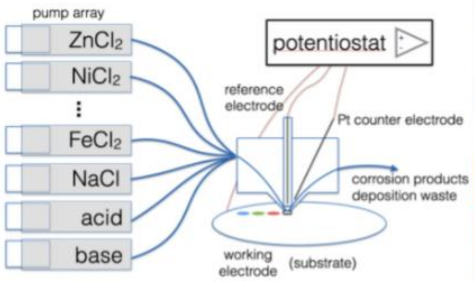
- NIST

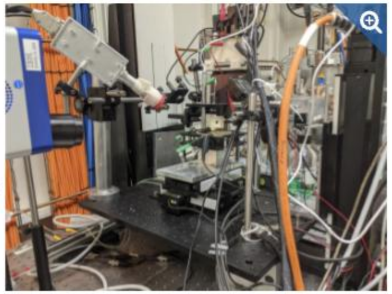
Autonomous Materials Science

Summary

We use autonomous experimentation (the merger of automated synthesis, characterization, AI-driven decision-making) to elucidate the role of composition, processing, and microstructure on the aqueous corrosion of complex metal alloys. To do this we are building a one-of-a-kind combined synthesis-characterization platform that is capable of making and measuring samples on demand. Our autonomous platform is meant to serve as a substrate for answering questions about trust and interpretability of modern AI systems, as well as bias and differences in human interpretation of data in materials science. Through this project we hope to lay the groundwork for autonomous experiments that answer scientific questions, which will be enabled by the future development of scientific AI imbued with materials science knowledge.

DESCRIPTION

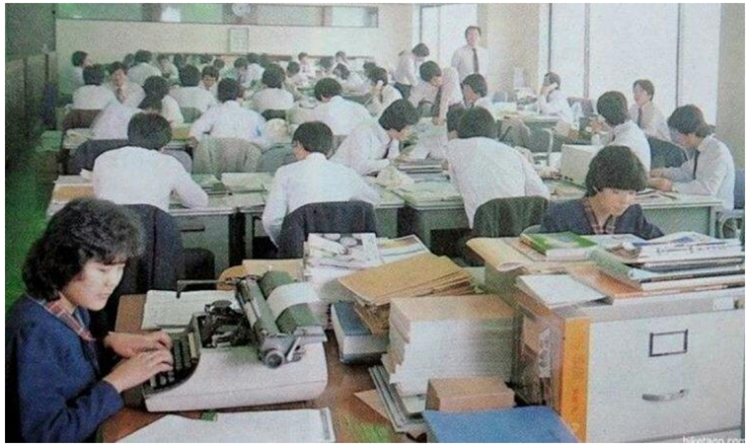
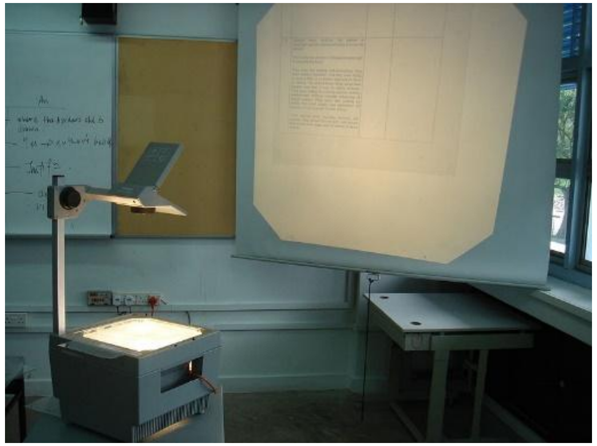




https://www.nist.gov/programs-projects/autonomous-materials-science

2차 (전기) → 3차 (컴퓨터, 인터넷) → 4차 (데이터, AI)

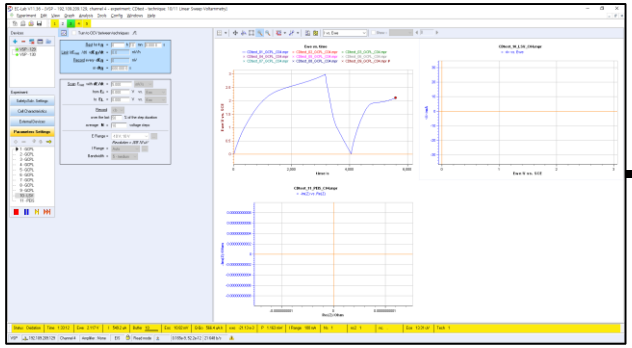
- 1990년대, 2000년대, 그리고.



2차 (전기) → 3차 (컴퓨터, 인터넷) → 4차 (데이터, AI)

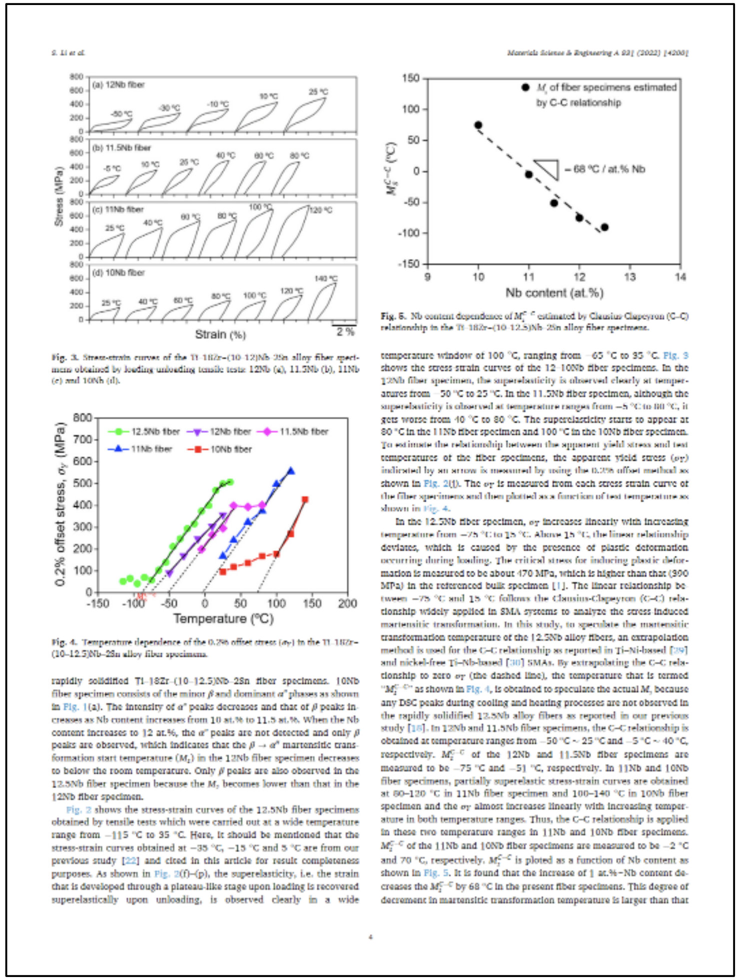
- 1990년대, 2000년대, 그리고.



→

“우리 때는 이런 걸로 논문 쓰고 연구했어!”
1차 산업혁명 시대 도구

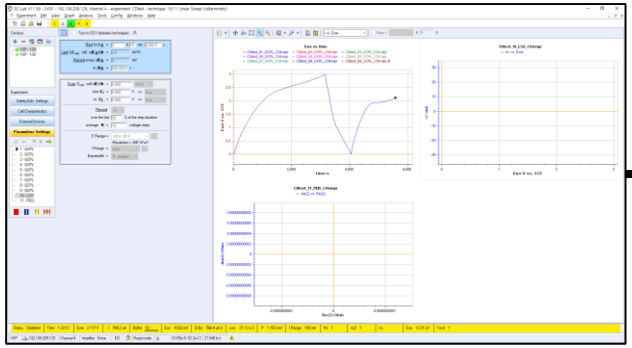
→



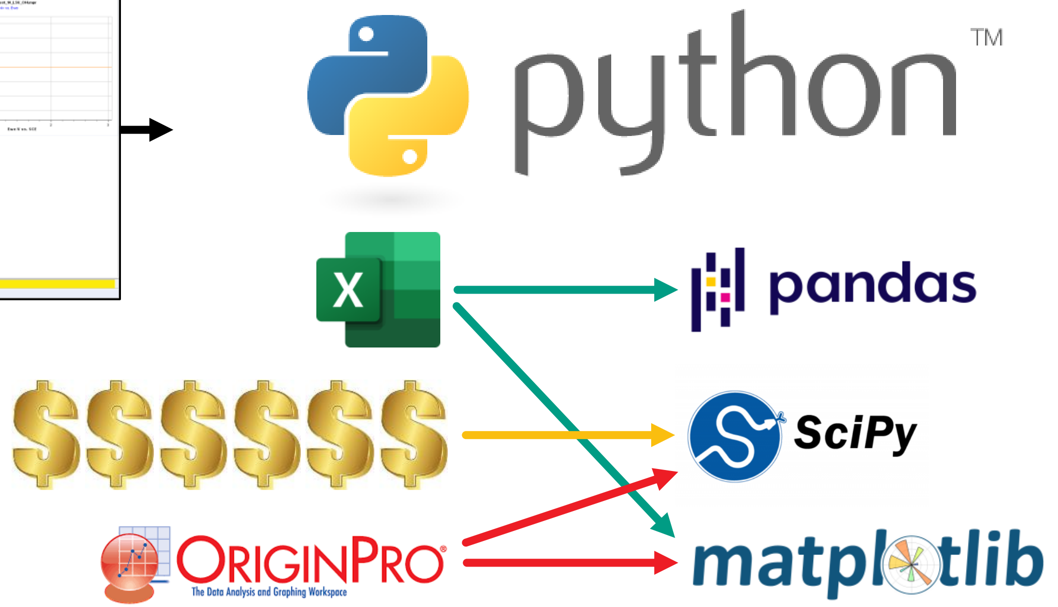
https://www.originlab.com/origin, https://ko.wikipedia.org/
Lee et al, https://doi.org/10.1016/j.msea.2021.142001

2차 (전기) → 3차 (컴퓨터, 인터넷) → 4차 (데이터, AI)

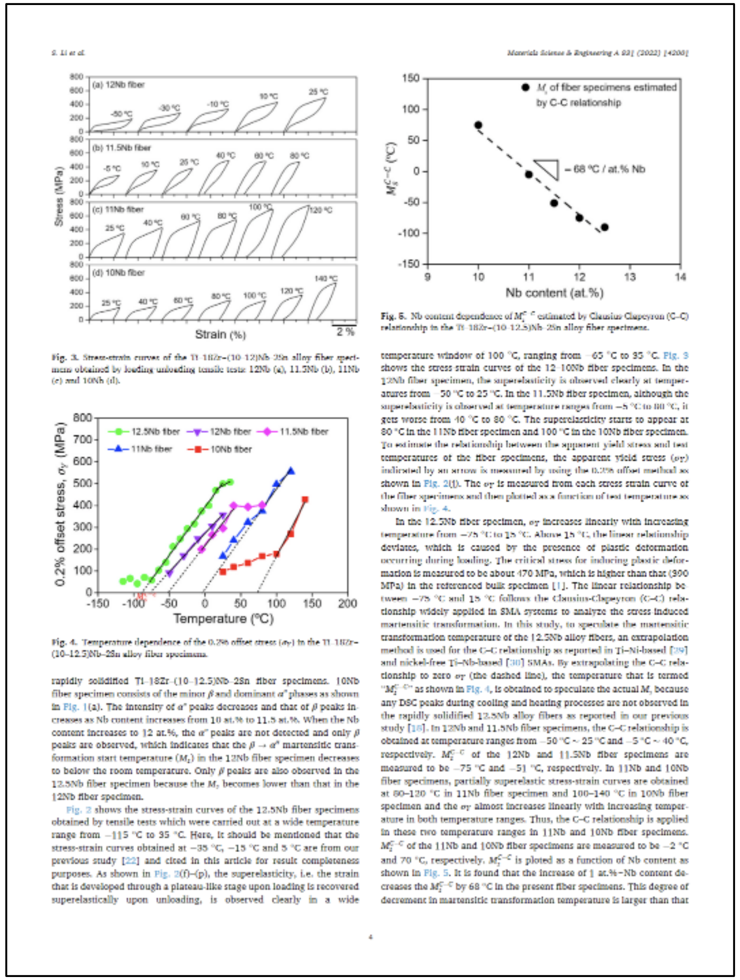
- 20년 전과는 달라져야 하지 않겠습니까?



→



→



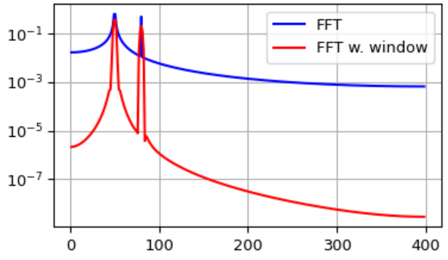
https://www.originlab.com/origin

Lee et al, https://doi.org/10.1016/j.msea.2021.142001

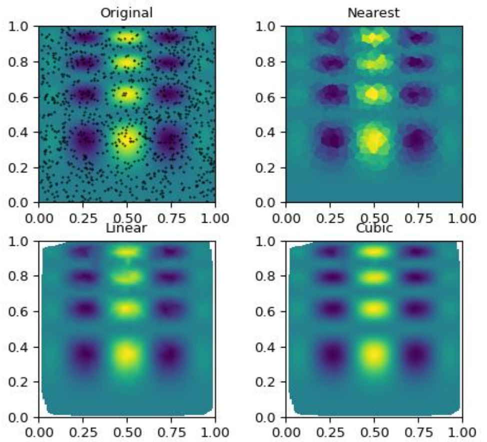
SciPy로 할 수 있는 일

- 거의 모든 신호처리, 영상처리

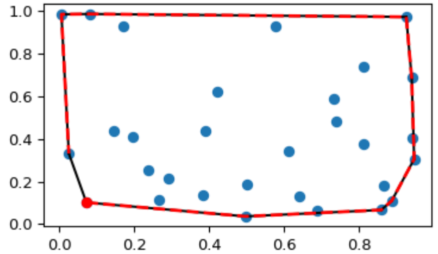
Fast Fourier Transformation



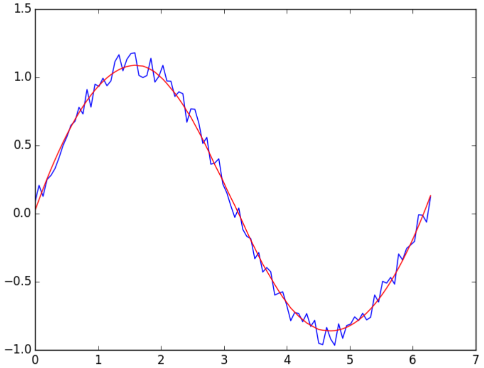
N-dimensional Interpolation



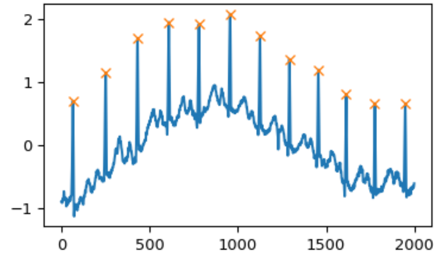
Convex Hull



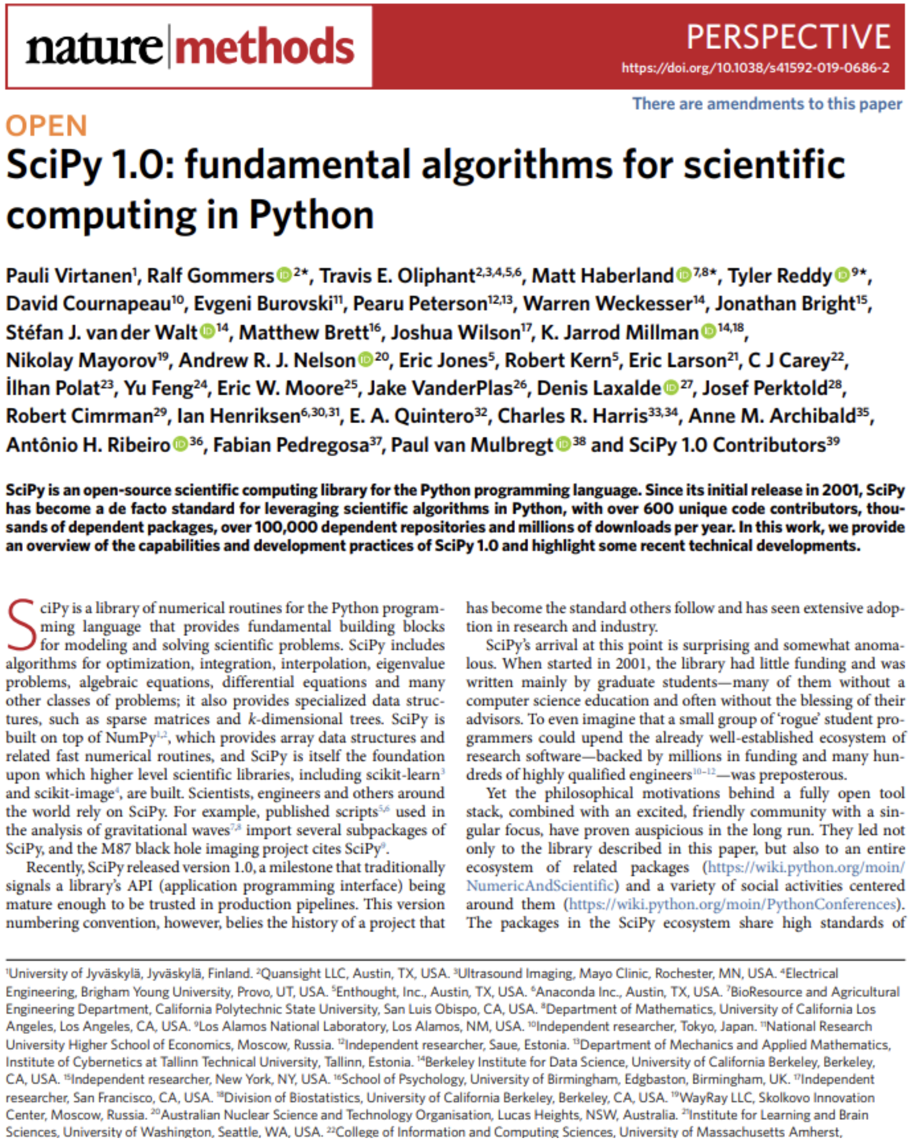
Smoothing



Peak Detection



https://scipy.org/



nature methods

PERSPECTIVE

https://doi.org/10.1038/s41592-019-0686-2

There are amendments to this paper

OPEN

SciPy 1.0: fundamental algorithms for scientific computing in Python

Pauli Virtanen[1], Ralf Gommers[2*], Travis E. Oliphant[2,3,4,5,6], Matt Haberland[7,8*], Tyler Reddy[9*], David Cournapeau[10], Evgeni Burovski[11], Pearu Peterson[12,13], Warren Weckesser[14], Jonathan Bright[15], Stéfan J. van der Walt[14], Matthew Brett[16], Joshua Wilson[17], K. Jarrod Millman[14,18], Nikolay Mayorov[19], Andrew R. J. Nelson[20], Eric Jones[5], Robert Kern[5], Eric Larson[21], C J Carey[22], İlhan Polat[23], Yu Feng[24], Eric W. Moore[25], Jake VanderPlas[26], Denis Laxalde[27], Josef Perktold[28], Robert Cimrman[29], Ian Henriksen[6,30,31], E. A. Quintero[32], Charles R. Harris[33,34], Anne M. Archibald[35], Antônio H. Ribeiro[36], Fabian Pedregosa[37], Paul van Mulbregt[38] and SciPy 1.0 Contributors[39]

SciPy is an open-source scientific computing library for the Python programming language. Since its initial release in 2001, SciPy has become a de facto standard for leveraging scientific algorithms in Python, with over 600 unique code contributors, thousands of dependent packages, over 100,000 dependent repositories and millions of downloads per year. In this work, we provide an overview of the capabilities and development practices of SciPy 1.0 and highlight some recent technical developments.

SciPy is a library of numerical routines for the Python programming language that provides fundamental building blocks for modeling and solving scientific problems. SciPy includes algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations and many other classes of problems; it also provides specialized data structures, such as sparse matrices and *k*-dimensional trees. SciPy is built on top of NumPy[1,2], which provides array data structures and related fast numerical routines, and SciPy is itself the foundation upon which higher level scientific libraries, including scikit-learn[3] and scikit-image[4], are built. Scientists, engineers and others around the world rely on SciPy. For example, published scripts[5,6] used in the analysis of gravitational waves[7,8] import several subpackages of SciPy, and the M87 black hole imaging project cites SciPy[9].

Recently, SciPy released version 1.0, a milestone that traditionally signals a library's API (application programming interface) being mature enough to be trusted in production pipelines. This version numbering convention, however, belies the history of a project that has become the standard others follow and has seen extensive adoption in research and industry.

SciPy's arrival at this point is surprising and somewhat anomalous. When started in 2001, the library had little funding and was written mainly by graduate students—many of them without a computer science education and often without the blessing of their advisors. To even imagine that a small group of 'rogue' student programmers could upend the already well-established ecosystem of research software—backed by millions in funding and many hundreds of highly qualified engineers[10–12]—was preposterous.

Yet the philosophical motivations behind a fully open tool stack, combined with an excited, friendly community with a singular focus, have proven auspicious in the long run. They led not only to the library described in this paper, but also to an entire ecosystem of related packages (https://wiki.python.org/moin/NumericAndScientific) and a variety of social activities centered around them (https://wiki.python.org/moin/PythonConferences). The packages in the SciPy ecosystem share high standards of

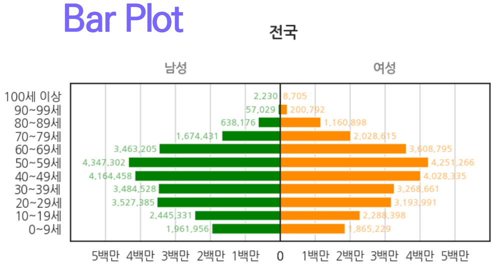
[1]University of Jyväskylä, Jyväskylä, Finland. [2]Quansight LLC, Austin, TX, USA. [3]Ultrasound Imaging, Mayo Clinic, Rochester, MN, USA. [4]Electrical Engineering, Brigham Young University, Provo, UT, USA. [5]Enthought, Inc., Austin, TX, USA. [6]Anaconda Inc., Austin, TX, USA. [7]BioResource and Agricultural Engineering Department, California Polytechnic State University, San Luis Obispo, CA, USA. [8]Department of Mathematics, University of California Los Angeles, Los Angeles, CA, USA. [9]Los Alamos National Laboratory, Los Alamos, NM, USA. [10]Independent researcher, Tokyo, Japan. [11]National Research University Higher School of Economics, Moscow, Russia. [12]Independent researcher, Saue, Estonia. [13]Department of Mechanics and Applied Mathematics, Institute of Cybernetics at Tallinn Technical University, Tallinn, Estonia. [14]Berkeley Institute for Data Science, University of California Berkeley, Berkeley, CA, USA. [15]Independent researcher, New York, NY, USA. [16]School of Psychology, University of Birmingham, Edgbaston, Birmingham, UK. [17]Independent researcher, San Francisco, CA, USA. [18]Division of Biostatistics, University of California Berkeley, Berkeley, CA, USA. [19]WayRay LLC, Skolkovo Innovation Center, Moscow, Russia. [20]Australian Nuclear Science and Technology Organisation, Lucas Heights, NSW, Australia. [21]Institute for Learning and Brain Sciences, University of Washington, Seattle, WA, USA. [22]College of Information and Computing Sciences, University of Massachusetts Amherst,

https://www.nature.com/articles/s41592-019-0686-2

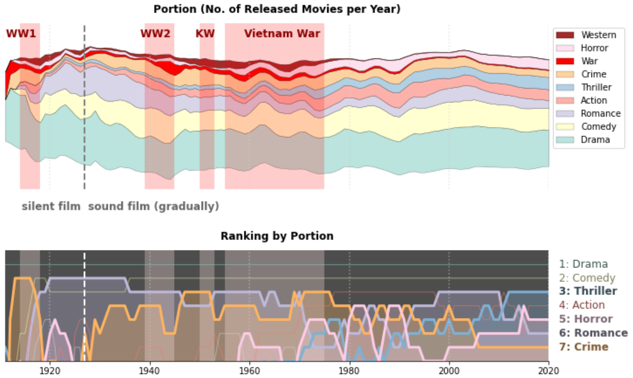
Matplotlib으로 그릴 수 있는 것

- 거의 모든 형태의 데이터 시각화

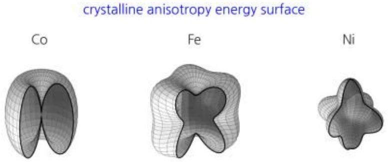
Bar Plot



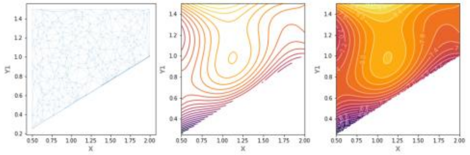
Stream Graph + Line Plot



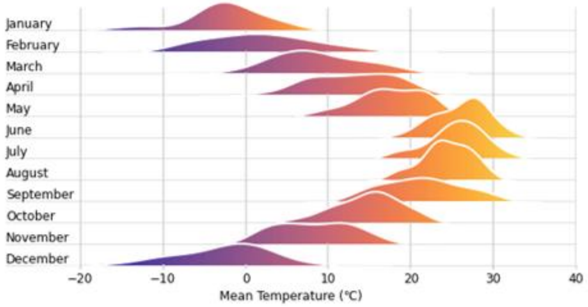
3D Surface

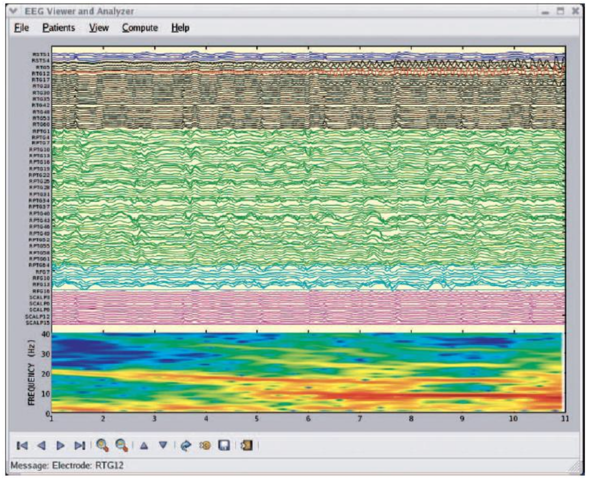


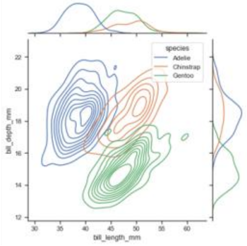
2D Data Distribution

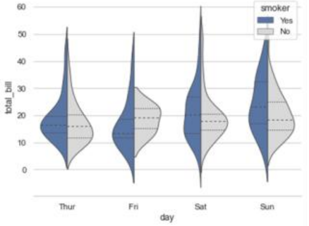


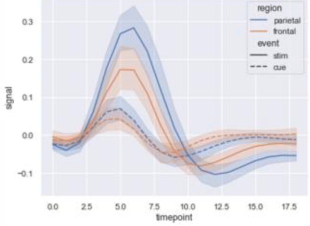
Ridge plot (feat. seaborn)











https://jehyunlee.github.io/, https://matplotlib.org/, https://seaborn.pydata.org/

Python을 쓰기 어려운 분야: 거의 없음

Quantum Computing	Statistical Computing	Signal Processing	Image Processing	Graphs and Networks	Astronomy Processes	Cognitive Psychology
QuTiP PyQuil Qiskit	Pandas statsmodels Xarray Seaborn	SciPy PyWavelets python-control	Scikit-image OpenCV Mahotas	NetworkX graph-tool igraph PyGSP	AstroPy SunPy SpacePy	PsychoPy
Bioinformatics	**Bayesian Inference**	**Mathematical Analysis**	**Chemistry**	**Geoscience**	**Geographic Processing**	**Architecture & Engineering**
BioPython Scikit-Bio PyEnsembl ETE	PyStan PyMC3 ArviZ emcee	SciPy SymPy cvxpy FEniCS	Cantera MDAnalysis RDKit	Pangeo Simpeg ObsPy Fatiando a Terra	Shapely GeoPandas Folium	COMPAS City Energy Analyst Sverchok

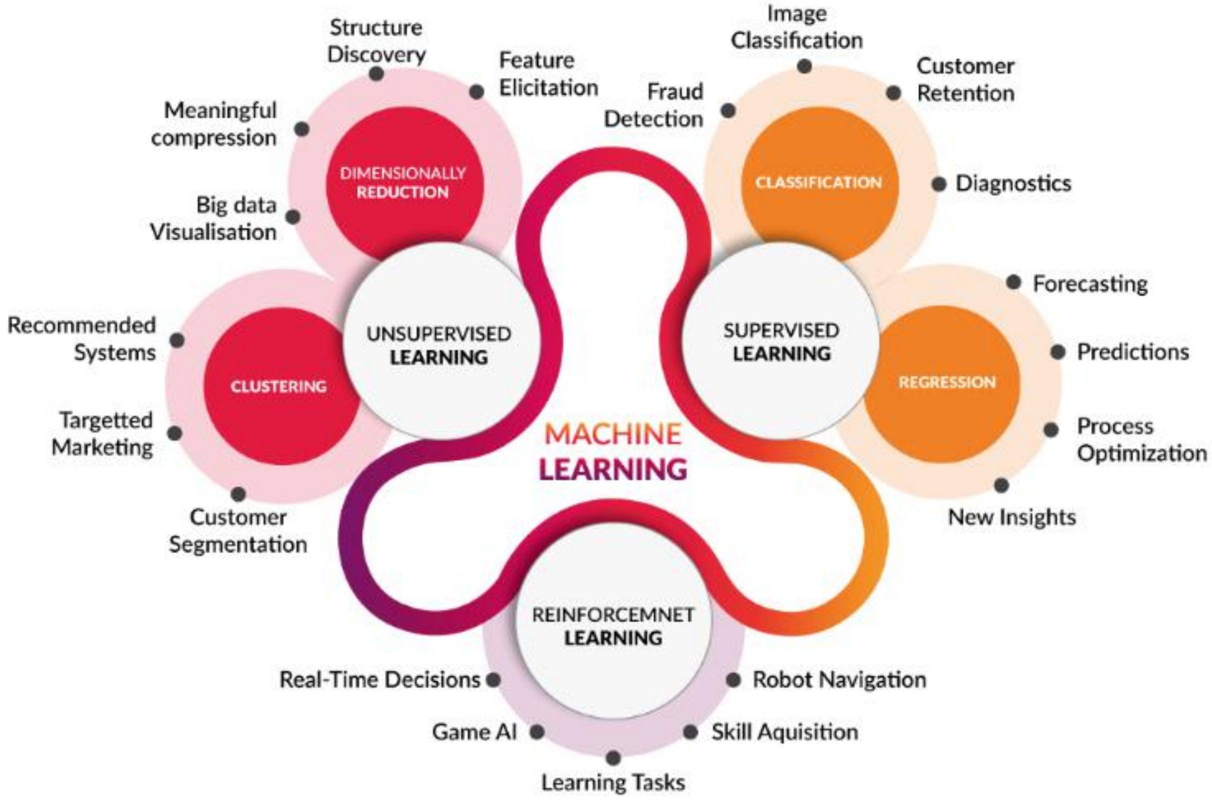
https://numpy.org/

Python을 쓰면 가능해지는 일: 머신러닝 포함 AI

python™

pandas SciPy matplotlib

NumPy



scikit learn

TensorFlow

PyTorch

https://medium.com/@ODSC/how-to-start-a-machine-learning-project-in-a-company-12af6981ec52

감사합니다

내가 해온 연구에 데이터 분석 끼얹기