生物資訊與數據分析
Machine Learning and Deep Learning in the Analysis of Biomedical Data

Yi-Ju Lee

PM of Smart Health Project, Academia Sinica

nior Research Associate, Institute of Statistical Science, Academia Sinica

14:00-16:00 ; R6005 RCEC

yijulee@stat.sinica.edu.tw

August 5, 2025,

Something about myself…

Yi-Ju Lee (李易儒)

- Postdoc Research Fellow, Institute of Statistical Science, Academia Sinica
- Project Manager, Smart Health Project of Academia Sinica
- Medical Information Manager, certificated by Taiwan Association for Medical Informatics (#1091MIM011)
- Medical Record Information Manager, certificated by TMRA (#1103236)
- GPU F&A/ web master/ consulting service / OHBM Conference Abstract Reviewer

Education

2022, 2025 OXCEP Programme, Oxford University

2020 PhD, Taiwan International Graduate Program in Interdisciplinary Neuroscience,
National Yang-Ming University and Academia Sinica
Applied Artificial Intelligence in Bio-Medicine Program, NYMU
Program of Smart Medicine, Taiwan AI Academy

2017 MS, Institute of Learning Sciences, National Tsing Hua University

Invited Talks/ Lecture

National Health Insurance Administration of Taiwan (NHI)
Institute of Applied Mathematical Sciences (NTU)
Department of Biomedical Science and Engineering (NCU)
Institute of Brain Sciences (NYMU), Department of Applied Mathematical Sciences (NSYSU)
Immunwork (National Biotechnology Research Park), International College of Innovation(NCCU)

Research Interest

Deep learning and statistical methods in biomedical data analysis,
AI application in precision medicine, bigdata in healthcare, complexity science

Acknowledgement

Dr. Hsin-Chou Yang, Dr. Chun-Houh Chen, and team mates.

SPECIAL REPORTS

Countries With The Best Health Care Systems, 2024

Despina Wilson April 2, 2024 Special Reports

Rank	Country	Medical Infrastructure and Professionals	Medicine Availability and Cost	Government Readiness	Health Care Index (Overall)
1	Taiwan	87.16	83.59	82.3	78.72
2	South Korea	79.05	78.39	78.99	77.7
3	Australia	90.75	82.59	92.06	74.11
4	Canada	86.18	78.99	88.23	71.32
5	Sweden	78.77	74.88	74.18	70.73
6	Ireland	92.58	96.22	67.51	67.99
7	Netherlands	77.86	71.82	55.1	65.38
8	Germany	86.28	75.81	83.82	64.66
9	Norway	72.48	68.68	64.78	64.63
10	Israel	88.63	75.61	90.25	61.73

Taiwan's healthcare system has been ranked number one in the world, according to the 2024 edition of the CEOWORLD Magazine Health Care Index. The result is evaluated based on various factors that contribute to overall health, including **medical infrastructure and professionals**, **medicine availability and cost**, and **government readiness**.

https://ceoworld.biz/2024/04/02/countries-with-the-best-health-care-systems-2024/


健康台灣・樂齡幸福社會
A Healthier and Happier Taiwan for All

中華民國總統府
OFFICE OF THE PRESIDENT
REPUBLIC OF CHINA (TAIWAN)

健康台灣
推動委員會

President Lai presides over the meetings of **Healthy Taiwan Promotion Committee**
(2024/08/22)

On the afternoon of August 22, President Lai Ching-te presided over the first meeting of the Healthy Taiwan Promotion Committee. As the committee's convener, the president presented committee members with their letters of appointment, and explained that the Healthy Taiwan Promotion Committee is not just about promoting a Healthy Taiwan, but also achieving a Balanced Taiwan. The president stated that the committee spans various areas of expertise, and also considers the balance of Taiwan's northern, central, southern, and eastern regions. The president expressed confidence that by soliciting a wide range of suggestions, engaging in diverse dialogue, and forging a consensus, the committee can help to realize health equality and further elevate the standard of medical care in Taiwan.

President Lai indicated that next year, the Ministry of Health and Welfare's total budget will be increased, along with expanded investment in medical treatment and care. In addition, he reported that the central government budget has also added a National Health Insurance (NHI) financial assistance program, which will help to enhance the work environments of healthcare professionals. The president stated that we will also launch the Healthy Taiwan Cultivation Plan to help rear talent and develop smart medicine. These budgets and programs, President Lai stated, reflect the government's determination to create a Healthy Taiwan, and prove that "Healthy Taiwan" is not just a slogan, and has already been turned into concrete action.

國家發展委員會
NATIONAL DEVELOPMENT COUNCIL

衛生福利部
Ministry of Health and Welfare
促進全民健康與福祉

The 3916th Cabinet meeting

National Development Plan

(2025-2028)

(Draft)

National Development Council

August 15, 2024

2024/Oct/07
Establishment of
"**Responsible AI Execution Center**"
(負責任AI執行中心),

"**Clinical AI Validation and Verification Center**"
(臨床AI取證驗證中心)

"**AI Impact Research Center**"
(AI影響性研究中心)

衛生福利部中央健康保險署
National Health Insurance Administration, Ministry of Health and Welfare

National Development Strategy(4/8)

Expanding medical investment for a healthier Taiwan

Healthy Taiwan Cultivation Plan

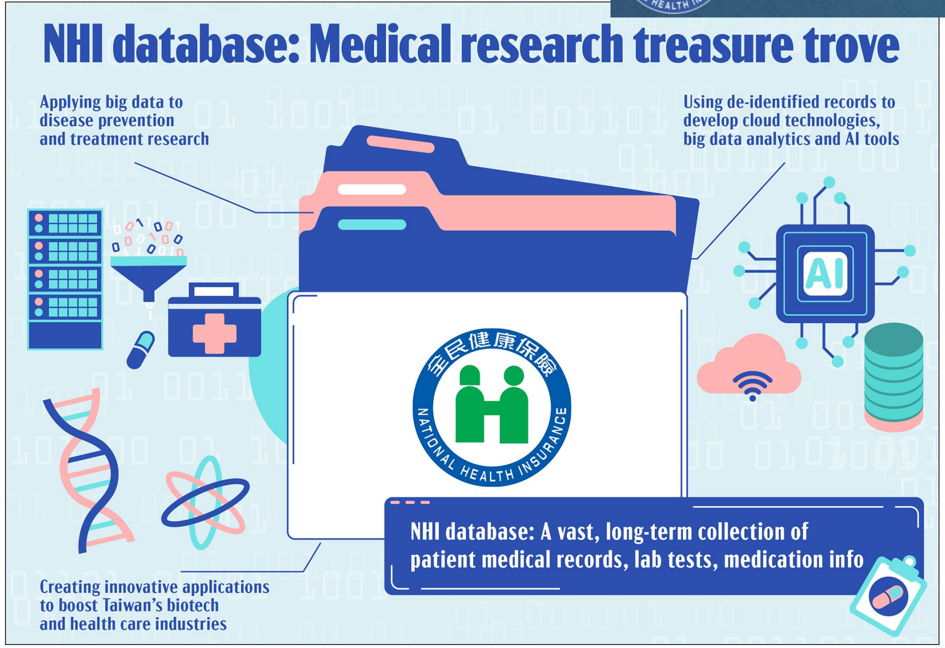
- Optimize working conditions, talent cultivation, smart medical care and social responsibility with 5-year 50 billion scale plan
- Improve the quality of medical services and optimize medical working conditions

Continuously improving the quality of medical services and improve people's health - 8-year 888 Plan

✓ 80% of patients with Triple H (hypertension, hyperlipidemia, hyperglycemia) join the care network
✓ 80% of participants receive life counseling
✓ The control rate of Triple H reaches 80%

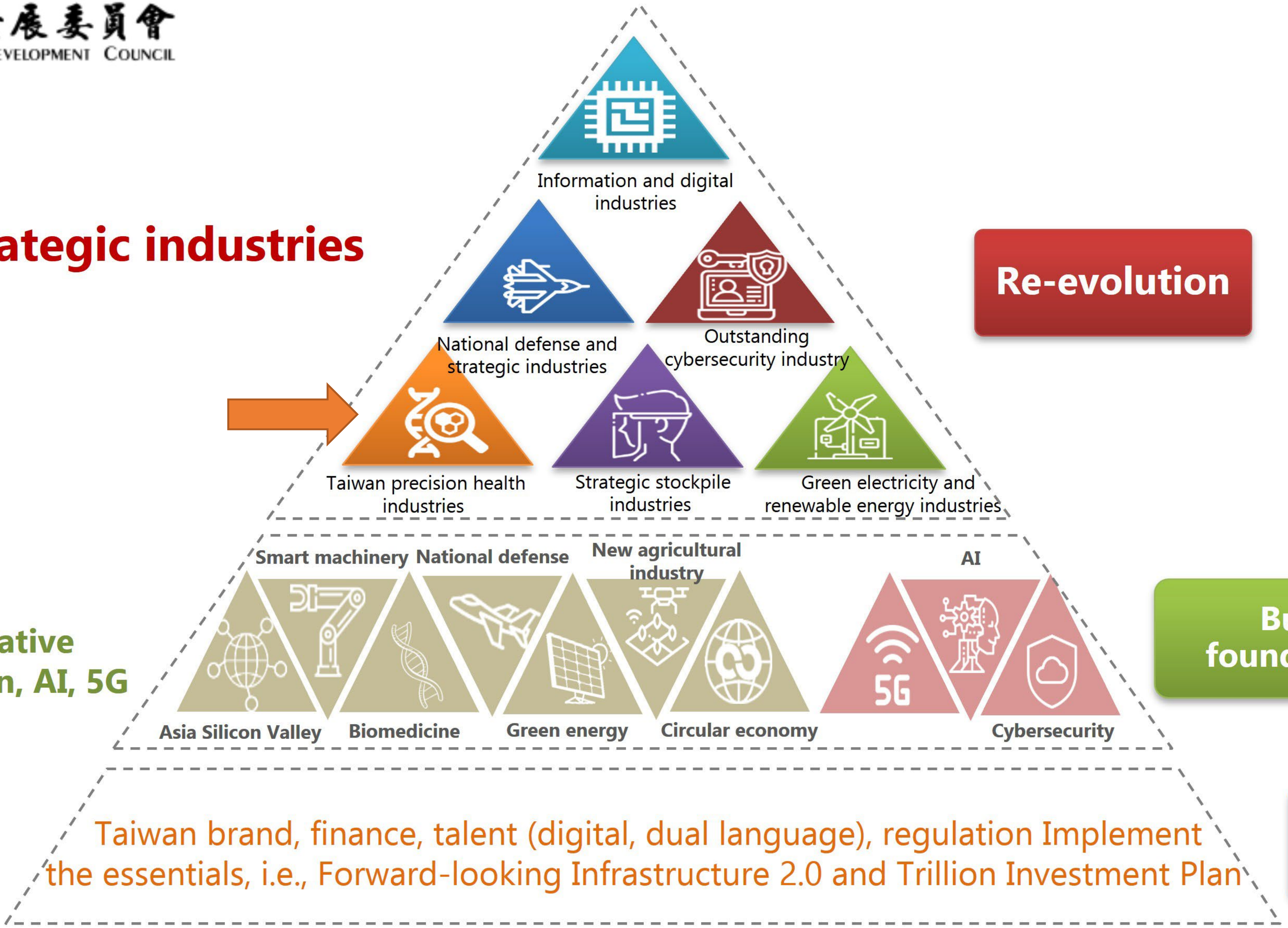
Enhancing the National Cancer Control Plan

✓ Expand cancer screening and early cancer detection services
✓ Establish a "Special Fund for Temporary Payment of New Cancer Drugs", reducing the heavy burden on cancer patients



國家發展委員會
NATIONAL DEVELOPMENT COUNCIL

6 core strategic industries



5+2 Innovative Industries Plan, AI, 5G

Common infrastructure environment

Re-evolution

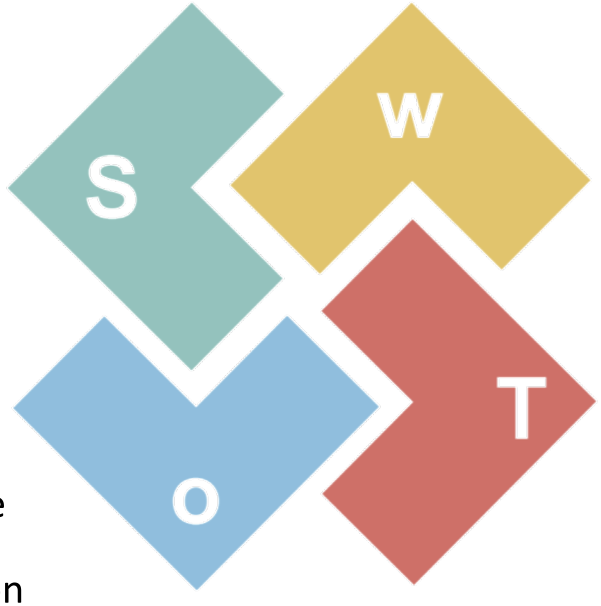
Build foundations

Superior environment

SWOT Analysis of Our Nation's Precision Health Development

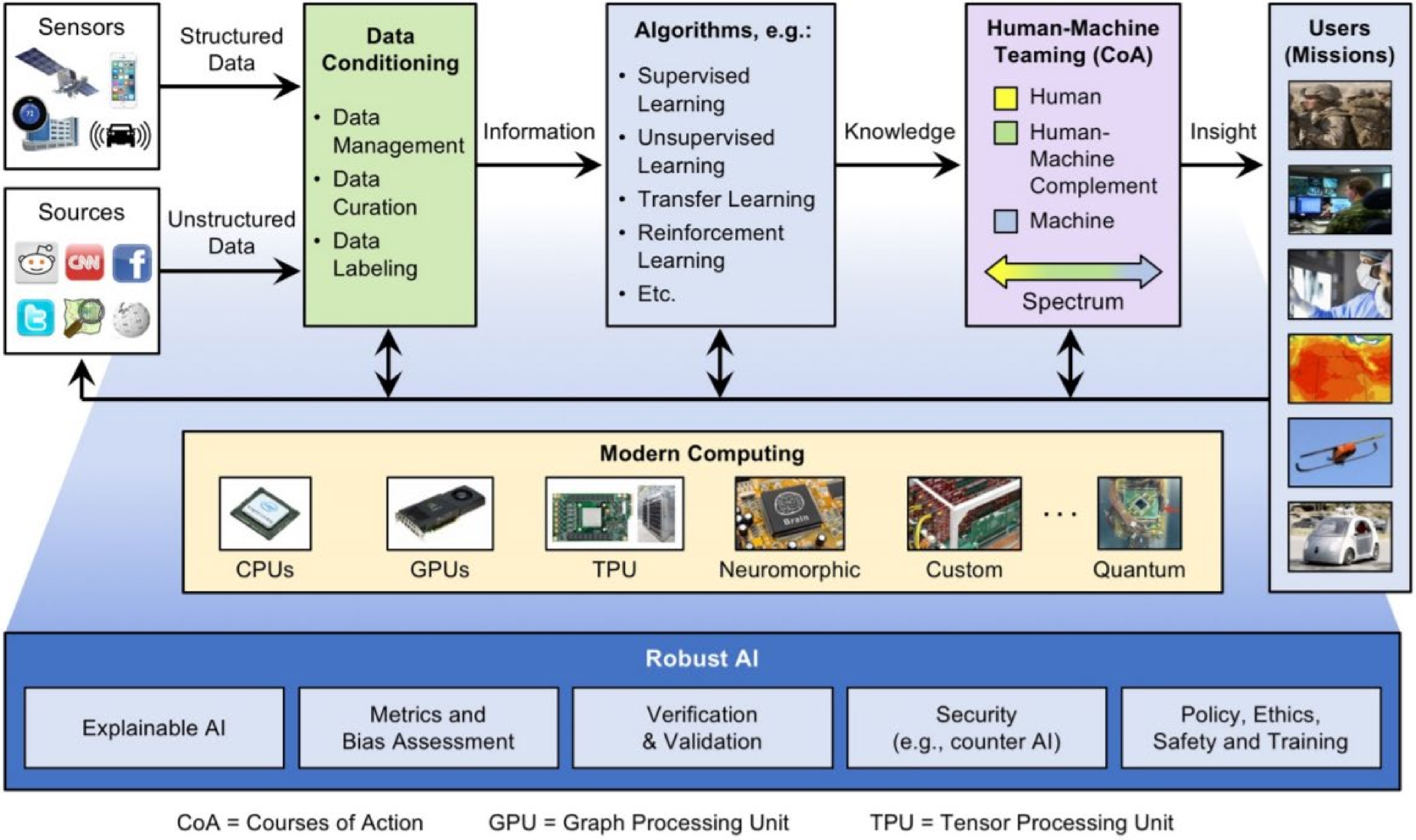
Strengths
- Superior medical system
- Complete **5G, ICT industry** chain and advanced materials
- Rich biomedical data and human genetic **data accumulated**
- Rich experience in epidemic prevention and deployment
- Abundant biomedical research and talent reserve

Opportunities
- Global trend toward precision medicine development
- Increased demand for precision healthcare due to **aging**
- Global trends in medical and ICT integration
- Business opportunities from epidemic normalization
- Taiwan's promotion of precision healthcare initiatives
- **Manufacturing and service market** new opportunities



Weaknesses
- Medical institution data has not yet been integrated and **shared**
- Lack of integrated solutions **comparable** to large international companies
- New investment environmental risk concerns
- Medical and ICT industries need cross-domain integration and **regulatory framework establishment**
- Economic, energy, technology, and security risks need to be integrated

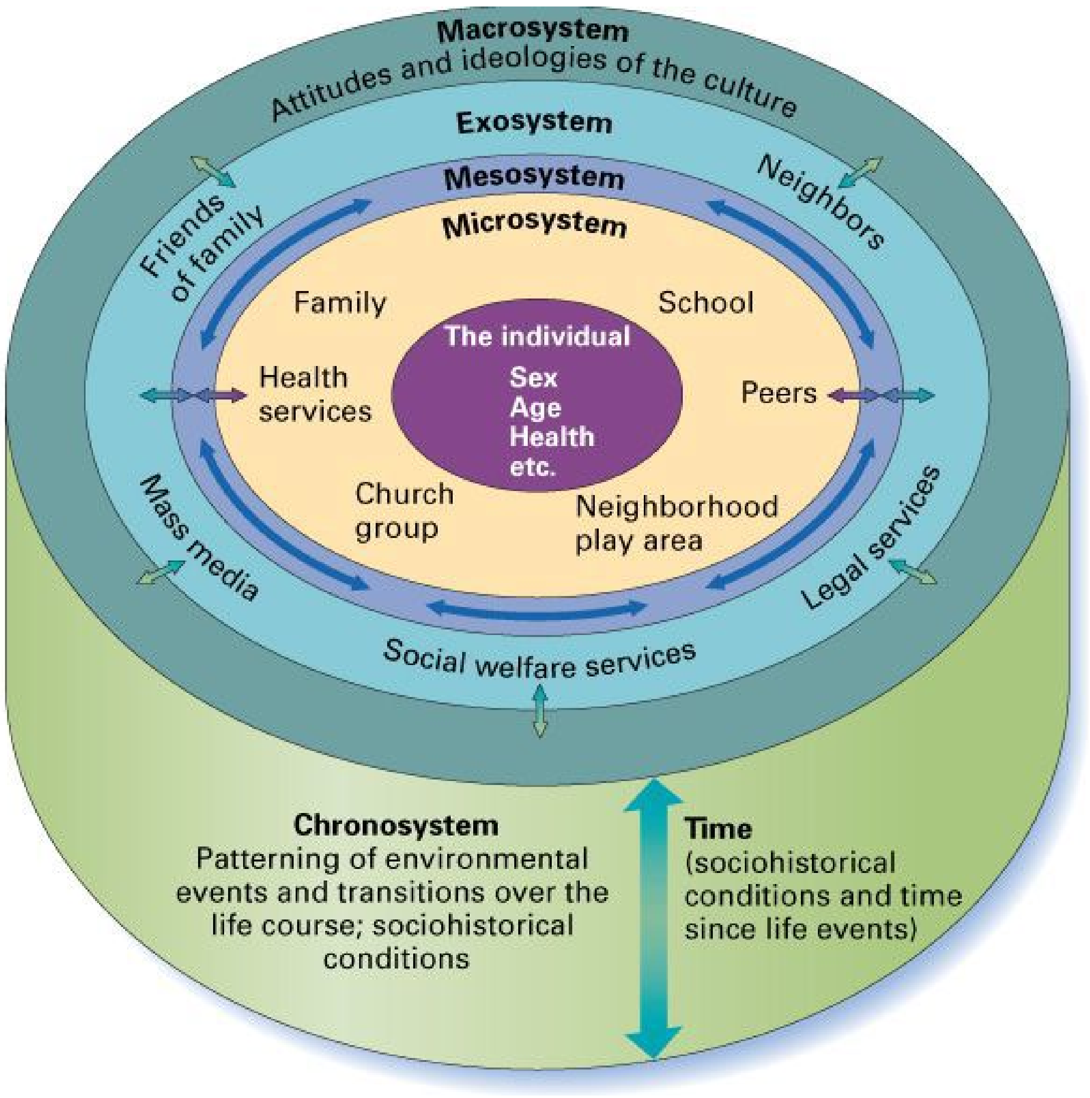
Threats
- **US-China competition** in precision healthcare development
- International precision healthcare standards recognition and certification barriers
- Global pandemic disruption of healthcare supply chain and market uncertainty



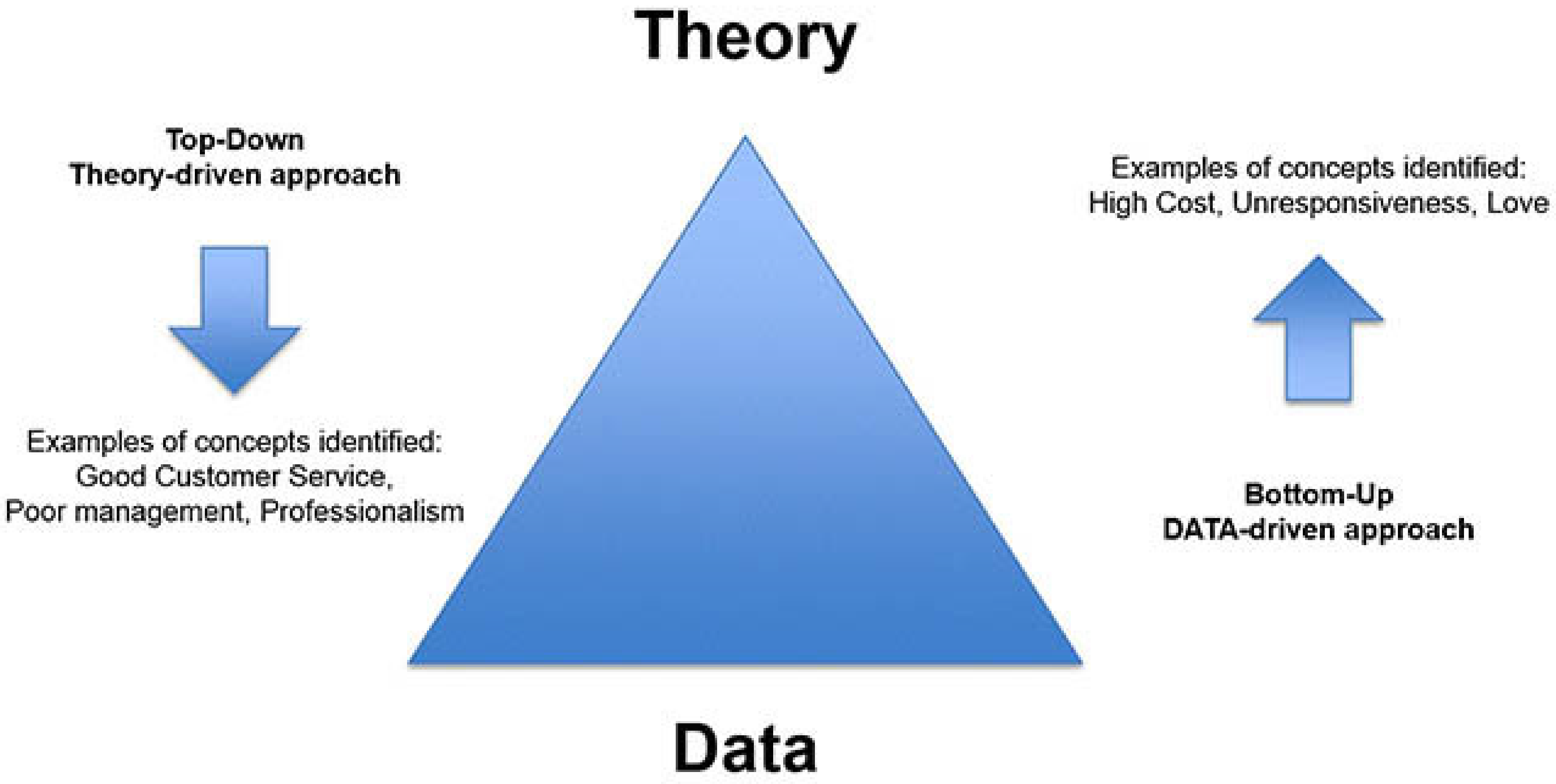
Gadepally, Vijay, Justin Goodwin, Jeremy Kepner, Albert Reuther, Hayley Reynolds, Siddharth Samsi, Jonathan Su, and David Martinez. "AI Enabling Technologies: A Survey." arXiv preprint arXiv:1905.03592 (2019).

To consider comprehensively …

Brofenbrenner's Ecological Theory

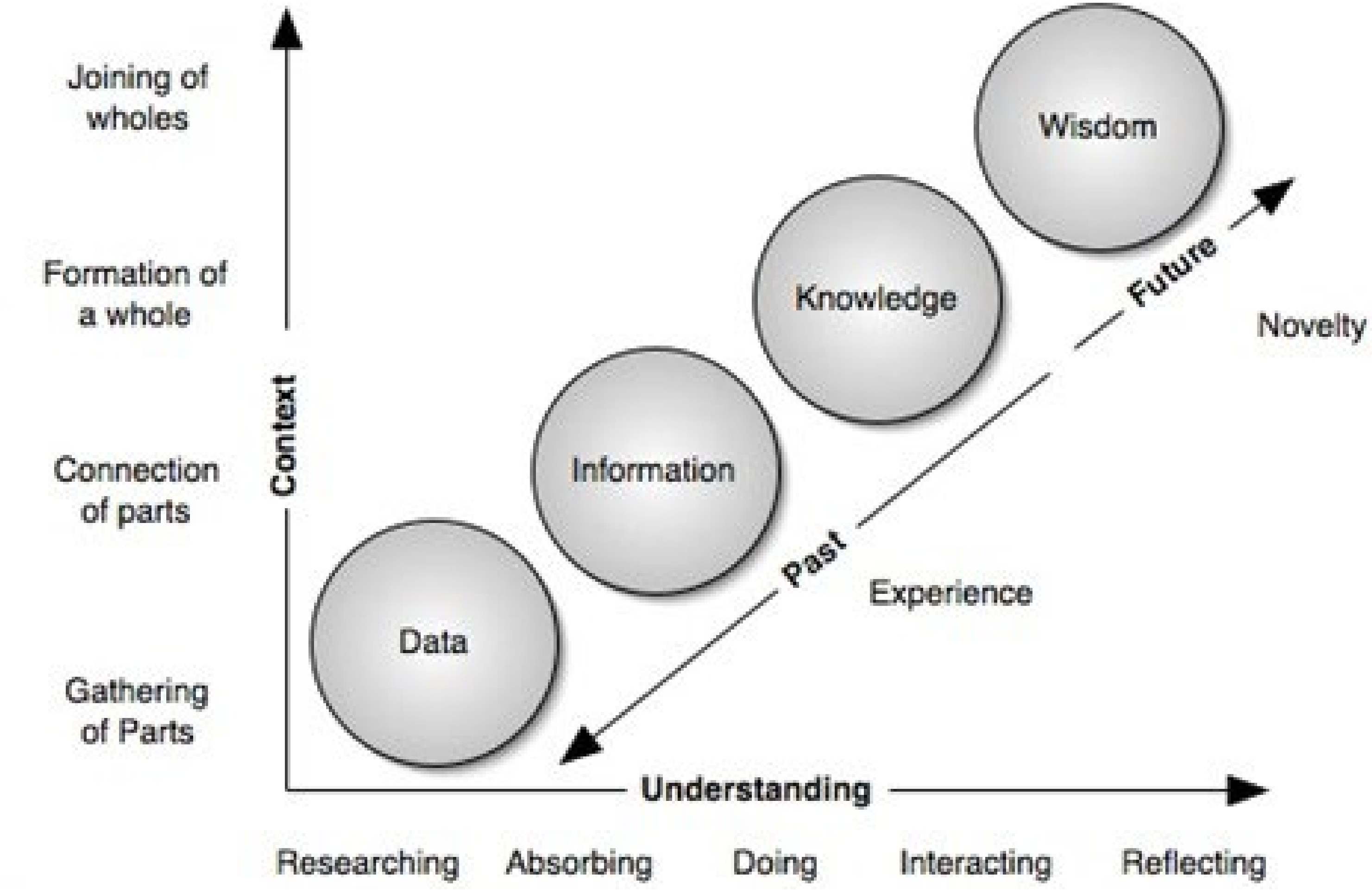


Two Ways to Understand the World: Data and Theory



https://speechling.com/blog/start-at-the-top-or-start-at-the-bottom-with-chinese/

The Hierarchal Structure of "What We Know"

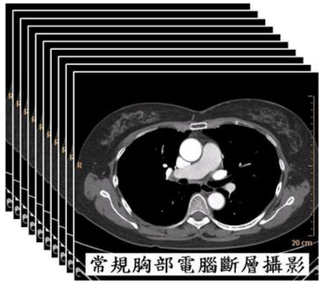


Flood, Mark & Lemieux, Victoria & Varga, Margaret & Wong, B.L.. (2014). The Application of Visual Analytics to Financial Stability Monitoring. SSRN Electronic Journal. 2438194. 10.2139/ssrn.2438194.

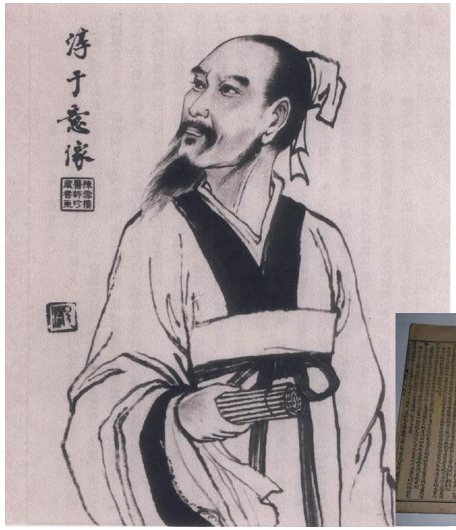
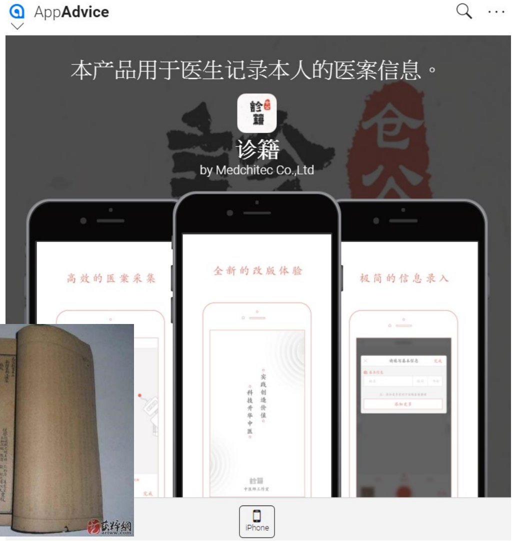
What is medical data?

- Information with Clinical Significance
- For Supporting Treatment, Diagnosis, and Care Purposes






The Earliest Medical Records in China - “Zhenji” 診籍 (Diagnostic Records)

《診籍》著於西漢時期，內容包括了患者姓名、年齡、性別、職業、籍貫、病例、病名、診斷、病因、治療、療效、預後等信息。涉及現代醫學中的消化、泌尿、呼吸、心血管、內分泌、腦血管、傳染病、外科、中毒、以及婦產科，兒科等科目。《診籍》中的治病方法有針灸，藥物，食療等，涉及的方藥有下氣湯，火劑湯，苦參湯，消石，芫花，米汁，藥酒，柔湯，竄藥，丸藥，半夏丸等，法理自通且有創新。

1500 BC Egyptian Medical Papyri
Ebers Papyrus treatment for cancer: recounting a "tumor against the god Xenus", it recommends "do thou nothing there against"

1800 BC
Sumerian Medical Clay Tablet
This tablet contains medical prescriptions of medicine and incantation against poisoning.
“Mustard, Pistacia, nuts, sweet mixed drink, meal of roast grain, thyme, bariratu-plant into wine in a small cup, you shall pour and smear on the skin, he will live.”

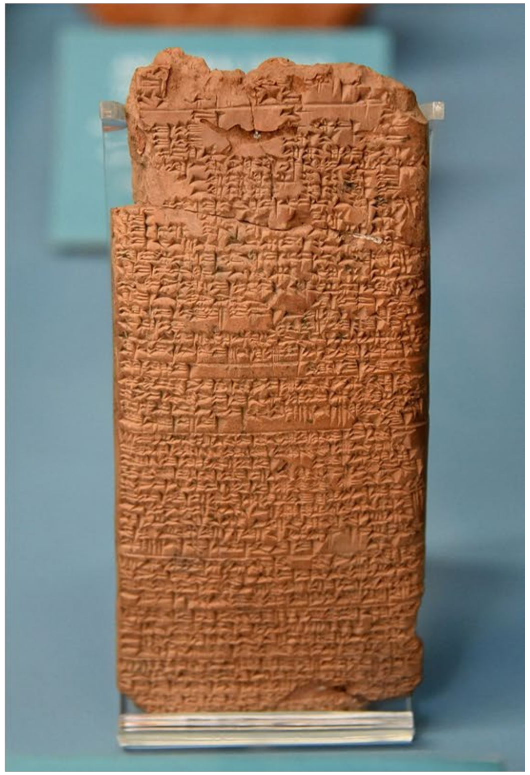


Image source:台中榮總放射醫學部, https://www.seattools.com.tw/ecommerce/%E9%A0%AD%E6%89%8B%E5%B7%A5%E5%85%B7

Data from a Single Person

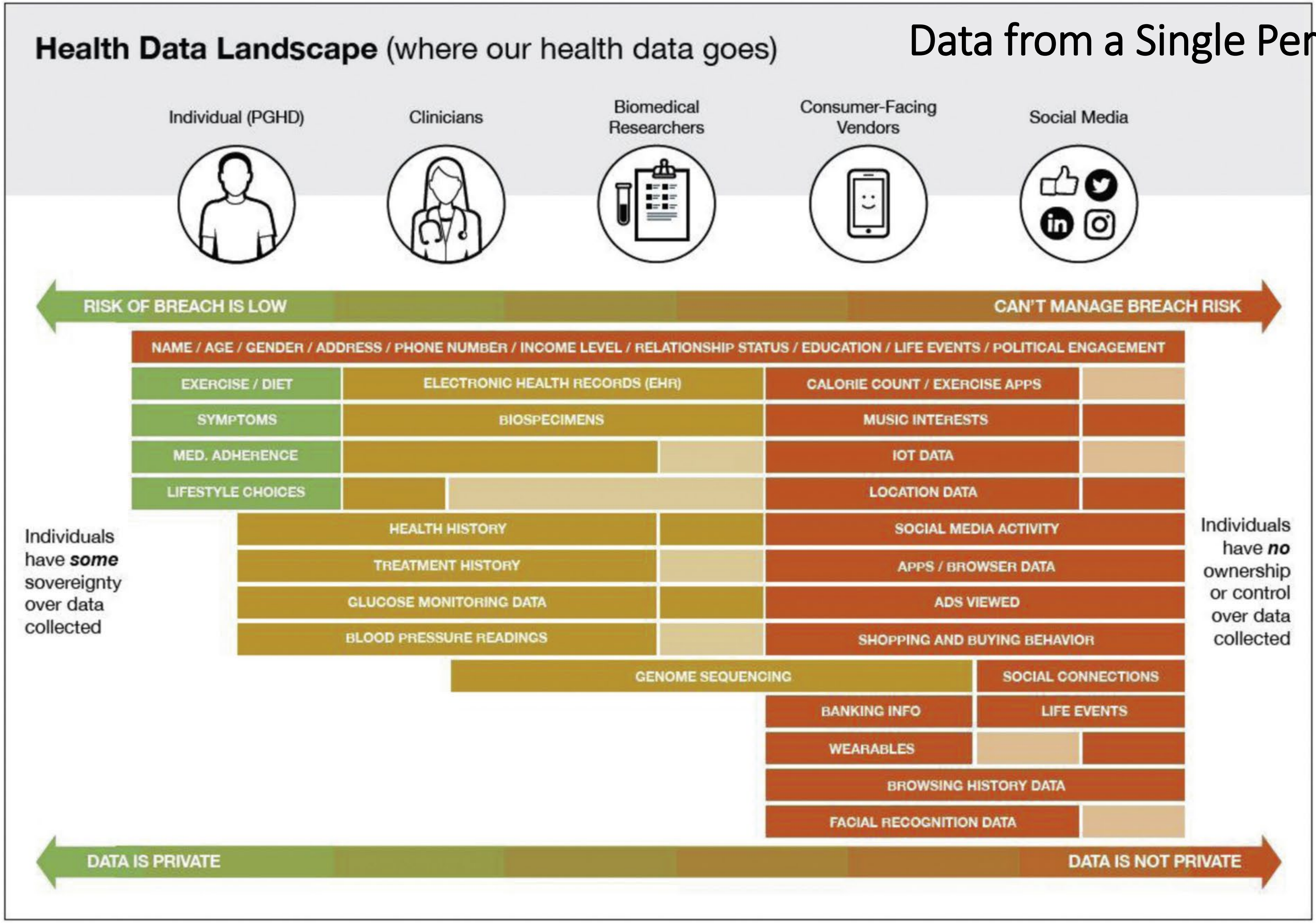
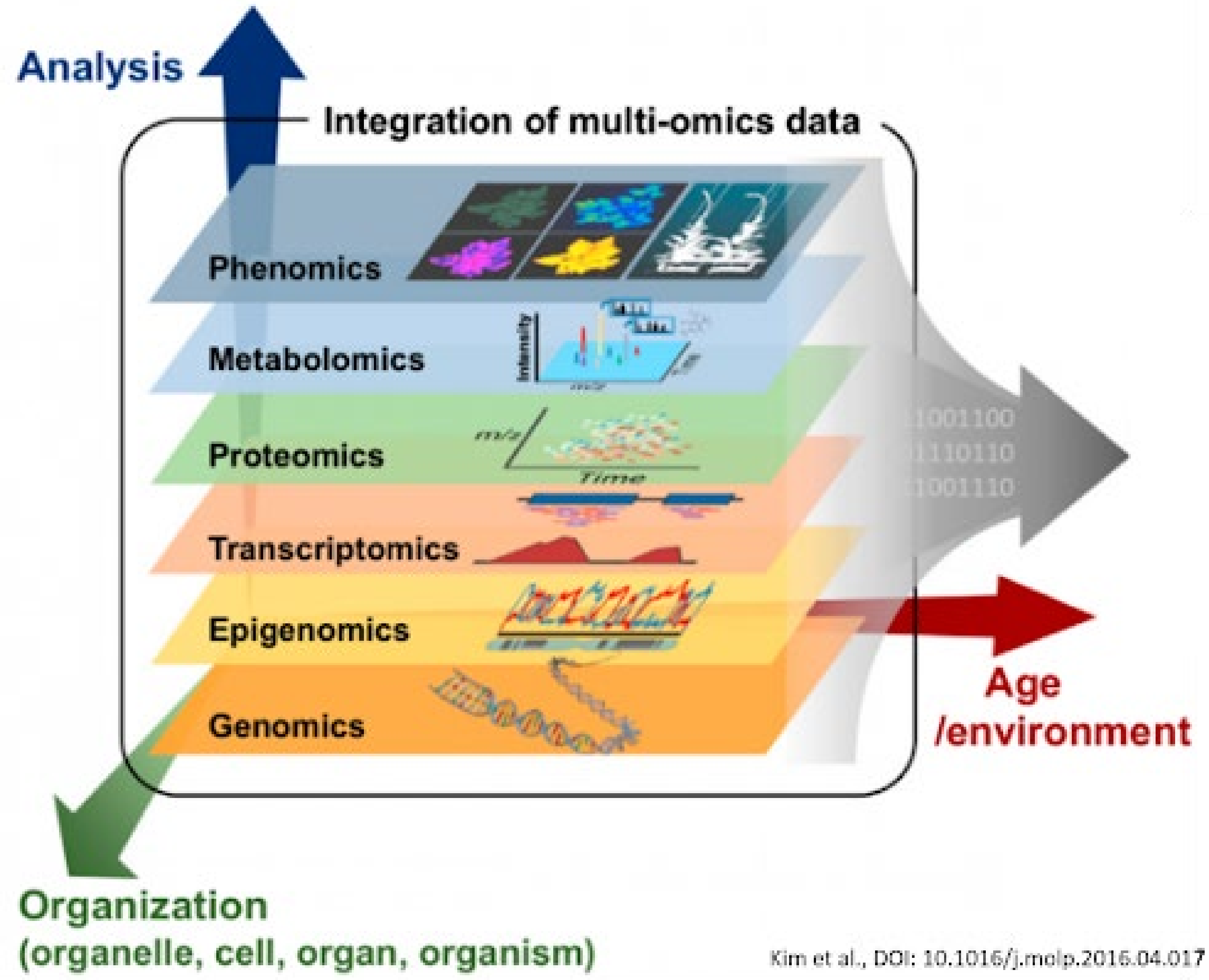


Fig. 1 *The continuum of risk for person-generated health data.* The Health Data Landscape illustrates the relative likelihood that various types of personal information (e.g., demographic, socioeconomic, health-related, financial) will be collected and/or shared without individuals' knowledge and/or permission. Figure designed by Hugo Campos for *Improving the Care Experience: A Collaborative Consensus Project.*

Biological Data from a Single Person



Kim et al., DOI: 10.1016/j.molp.2016.04.017


意匠

WHO USES MY HEALTH DATA?

1 PRIMARY DATA SOURCES

SCENARIO

At an appointment with my doctor, who...
1. reviews my blood test results
2. diagnoses IBS, and
3. prescribes Bentyl

THE PROVIDER GROUP

medical encounter note including name, dob, diagnoses, prescription, doctors name, when and where I saw my doctor, etc.
Many provider groups sell de-identified patient data.

THE PHARMACY

my prescription includes my name, dob, my doctor's name, medication, dose, etc.
75% of all retail pharmacies "send some portion of their electronic records" to at least one data miner.[11]

THE LAB

my blood sample and identification including my name, dob, sex, ordering physician, etc.
In 2015, nearly **1/2** of all labs send data to Iqvia (was IMS), labs send data to other data miners as well.[13]

MY INSURER

medical claim from my provider to my insurer including the coded services provided during the encounter
More than 60 health plans sell data to at least one data broker. This accounts for about **60%** of all US medical claims transactions.[4]

2 SECONDARY DATA SOURCES

THE EHR COMPANY

Electronic Health Record (EHR) companies have access to and sometimes ownership of the data in their EHRS.[13]
Many will de-identify and sell my healthcare data. The Practice Fusion model was one of the first to sell data to pharma and advertise drugs directly to providers.

PHARMACY BENEFIT MANAGERS

PBMs collect pharmacy data from claims.
They sell data to pharma companies who are interested to learn where their drugs are doing well vs poorly. **85%** of PBMs sell to ExamOne who sells 7 years of an individual's prescription history to life and health insurers.[13]

MY BANK

Throughout the process, my bank tracks copays with my doctors office and pharmacy. It also has record of my monthly premiums with my insurer.
Many banks sell customer data.

3 GROUPS WITH ACCESS

HEALTH IT MIDDLE-MEN

Health IT middle men offer services such as data warehousing, analytics, performance management solutions, claims processing, transition support to value-based payment models, and revenue optimization. They are used by provider groups, pharmacies, insurers, and more.

What PHI or de-identified health information they have access to and sell has not been measured to date. The total number of middlemen companies who can access, use, and/or sell my data is unknown.

Examples of Health IT middle men who work with health data:

McKesson, Celanese, NantHealth, nextgen, RWS, Optum, Harris Health System, navicure, Cotiviti

THE GOVERNMENT

Federal and State health departments maintain Public Use Files (PUF), de-identified and limited datasets to support researchers (ex: utilization and spending data aggregated at the prescriber, drug name, and generic name levels).[2]

Federal or State data sets with Patient Health Information (PHI) can be accessed through IRB approval or other application approach.

4 DATA BROKERS

DATA MINERS

PATIENT NAME, SSN, AGE, SEX, ADDRESS, DOCTOR, MEDICAL HISTORY, DIAGNOSES, MEDICATIONS, LAB RESULTS, INSURER

Data miners use de-identified data including longitudinal records that track my longterm health and switch my name for a number. Data comes from my medical organization, pharmacy, insurance company, federal and state health department data, and more.[5]

Even de-identified, this data can provide valuable, population health insights and demographic profiling for individuals.

IQVIA, Symphony Health, LexisNexis, prognos

DATA BROKERS

NAME, AGE, SEX, HEALTH DATA FROM APPS AND WEARABLES, ZIP CODE, PURCHASES, BROWSER DATA, LIKES

Data brokers sell identified profiles. In 2014, the FTC reported that Acxiom had "over 3,000 data segments for nearly every U.S. consumer."[7]

Data brokers gather health data and health related digital footprint data, such as health related purchases, consumer genetic testing, and apps. EliteMate, a dating service, sells a list of individuals and their mailing addresses with AIDS/HIV.[10, 13]

ExamOne, acxiom, crossix, CentraForceHealth

5 DATA USERS

CLINICAL RESEARCH

Research Centers
Researchers use many data sources including Federal and State data sets, clinical study reports, and more. Some data brokers will give research centers a discount on population health data.[11]

MARKET ANALYSIS AND TARGETED ADVERTISING

Pharmaceutical companies
Population health data can help pharma companies determine which drugs to develop or invest in. Data inform Pharma where certain drugs are doing poorly and need more marketting. Profiles on doctors prescribing practices lead pharma companies to target certain providers to increase sales.[13] Pharma can also cross-reference de-identified and identified records from Miners and Brokers in order to learn more about individual customers.

Marketers
Marketers use health data to target consumers. For example, marketers have purchased "sick lists" of people presumed to have a certain ailment from Acxiom.[6]

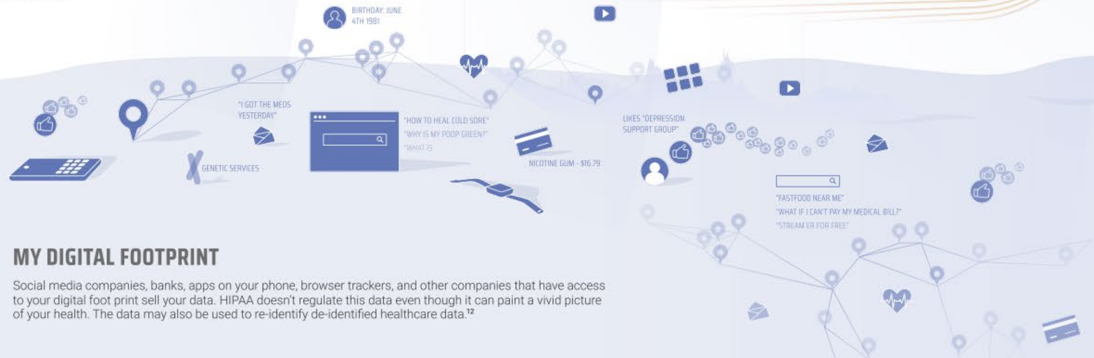
Digital Advertising (Facebook, Google, Amazon, etc)
Most have thier own sources of data but are interested in purchasing health data. In Feb of 2019, Facebook was caught matching ovulation health data from an app called Flo to their own users presumably for targeted advertising.[9]

RISK PROFILING

EHRs, Hospitals, and Physician Groups
It is often harder for doctors to get data about their patients from within the health system than from the outside. Re-identified data can flesh out a patient's record. Population data can predict patient risk. Some data brokers include "criminal records, online purchasing histories, retail loyalty programs and voter registration data" in their reports.[8]

Health Insurance
The ACA denies health insurers to exclude patients with pre-existing conditions. However, payers are interested in getting risk scores for their patient populations to manage populations, determine an individual's premium charges, and even deny coverage.[1, 13]

Car Insurance, House Insurance, Life Insurance, Job application, Cell phone or utility company
When assessing cutomers' financial risk, insurers and even employers may purchase health risk profiles.



Q Is it difficult to re-identify data?

Researchers have long demonstrated that it is not difficult to re-identify de-identified data.[10] One study found that "63% of the population can be uniquely identified by the combination of their gender, date of birth, and zip code alone."[3]

HIPAA AND MY MEDICAL RECORD

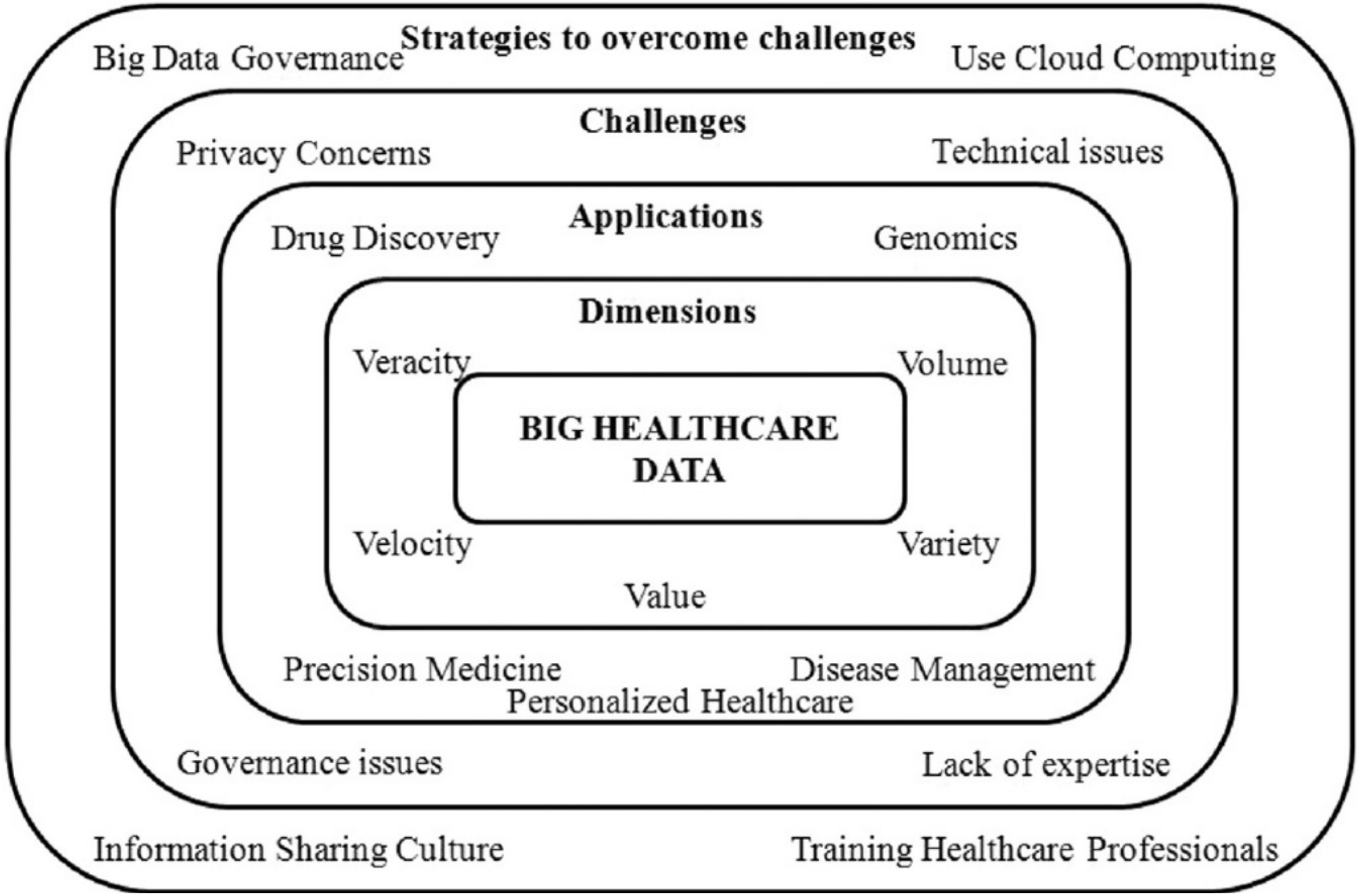
Medical records can contain history of my health events including hospitalizations, diagnoses, medication lists, family history. In 1996, HIPAA ruled that medical record data could be shared if it was de-identified by removing name and a few other personally identifying data.

MY DIGITAL FOOTPRINT

Social media companies, banks, apps on your phone, browser trackers, and other companies that have access to your digital foot print sell your data. HIPAA doesn't regulate this data even though it can paint a vivid picture of your health. The data may also be used to re-identify de-identified healthcare data.[12]

https://www.goinvo.com/vision/who-uses-my-health-data/

Impact of Dataset Scale on Performance



Mehta, N., & Pandit, A. (2018). Concurrence of big data analytics and healthcare: A systematic review. International journal of medical informatics, 114, 57-65.

Decision Support Algorithms

fuzzy boundaries : Data Science, Complex Systems

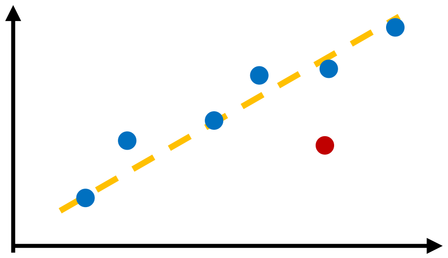
Rule-based Decision Making

if condition fulfilled *then*
    action 1
else
    action 2

➔ Expert System
- Binary data

Example
- Clinical diagnostic criteria
- Simple pattern matching
- Threshold based alarms
  eg. Drug use alarms
- Time based alarms

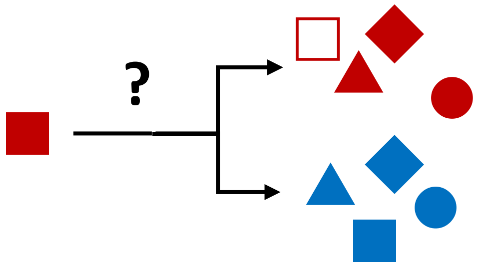
Statistical Reasoning



➔ Simple Regression
- Numerical data

Example
- Outlier detection
- Extra- and interpolation
- Predictive maintenance

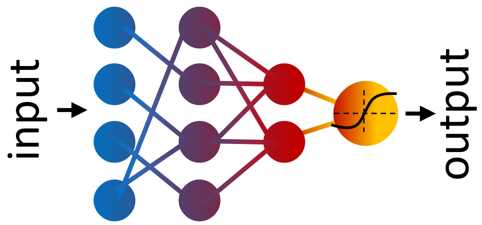
Machine Learning



➔ Classification Task
- Arbitrary data

Example
- Identification of relevant features from large input datasets
- QC using various metrics

Artificial Intelligence



➔ Dynamic Adaptation to Novelty

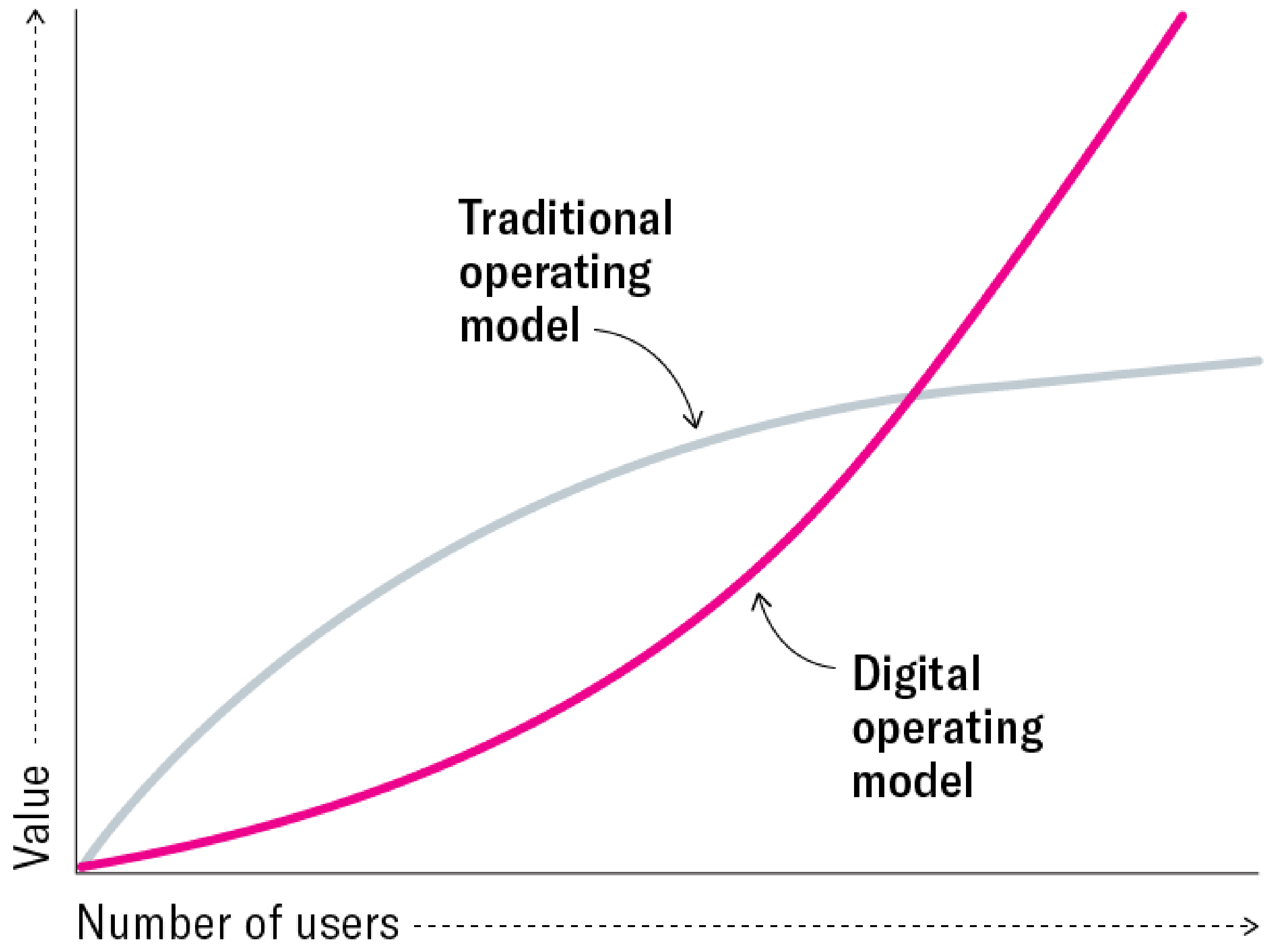
Example
- Recommendation System
- Treatment Effect Prediction
- Telemedicine
- Precision Healthcare with IoT

Differences: Generative AI vs Machine Learning vs Deep Learning

Point of Difference	Generative AI	Machine Learning	Deep Learning
Focus	Focuses on creating new content autonomously	Trains algorithms to learn patterns from data	Utilizes neural networks with multiple layers
Core Functions	Generates new content based on learned patterns	Analyzes data to make predictions or decisions	Learn Complex patterns in data for accurate predictions
Key Algorithms	Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Diffusion Models	Decision Trees, Support Vector Machines, Random Forests Naive Bayes	Convolutional Neural Networks (CNNs), Recurrent Networks (RNNs), Transformers
Application	Text generation, image synthesis, music creation, drug discovery	Spam detection, credit scoring, recommender systems, predictive maintenance	Computer vision, Natural language processing speech recognition, autonomous vehicles
Complexity Area	Incorporation of probailistic models and algorithms for content generation	Utilizing algorithms like decision trees, SVMs & Neural Networks	Involves intricate neural network architecture with multiple layers

https://k21academy.com/ai-ml/deep-learning-ml-generative-ai/

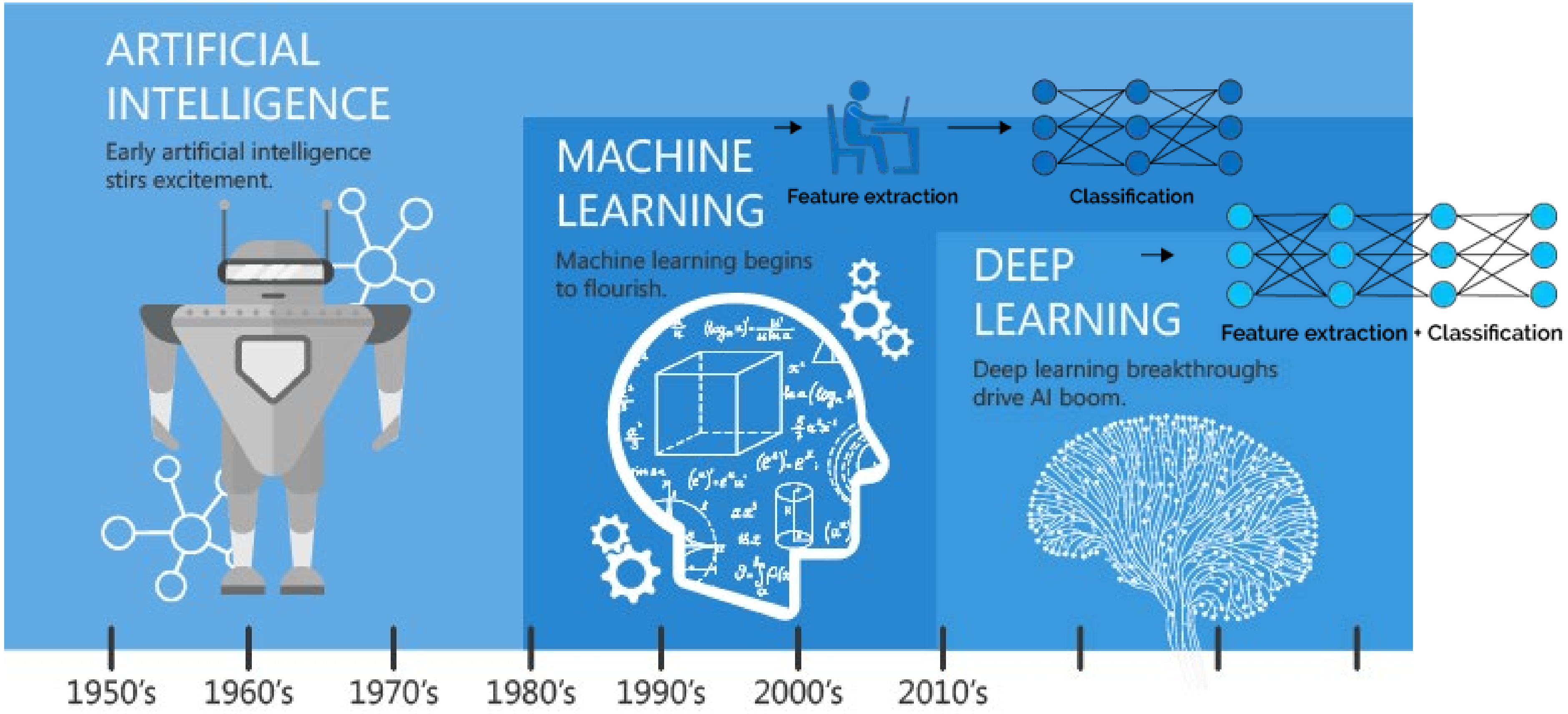
AI-Driven Device Can Outstrip Traditional Firms



From: "Competing in the Age of AI," by Marco Iansiti and Karim R. Lakhani, January–February 2020

HBR

Machine Learning & Deep Learning



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

Machine Learning & Deep Learning

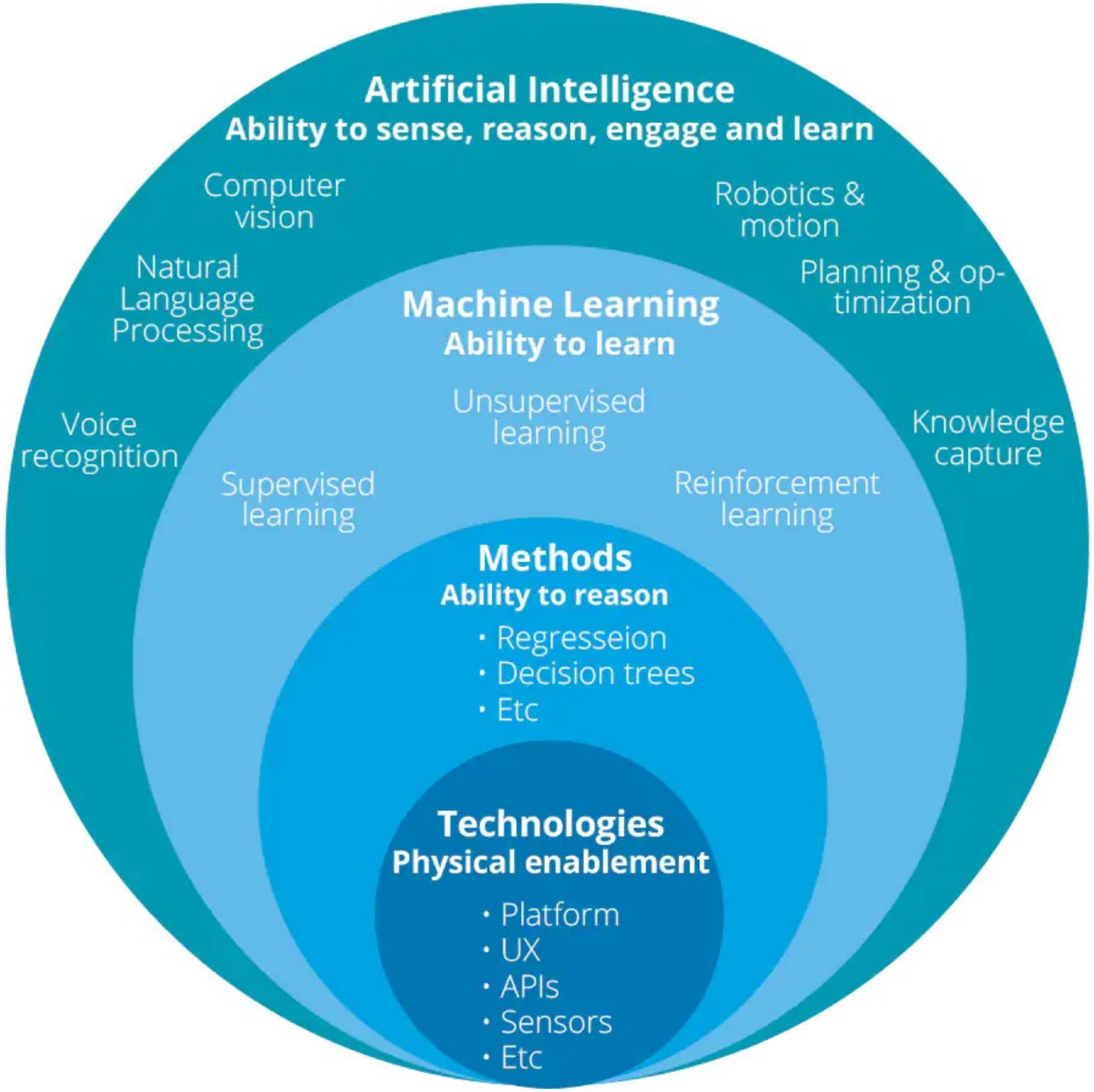
- **Machine learning** uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned.

- **Deep learning** structures algorithms in layers to create an “artificial neural network” that can learn and make intelligent decisions on its own.

The Ultimate Goal of Artificial Intelligence

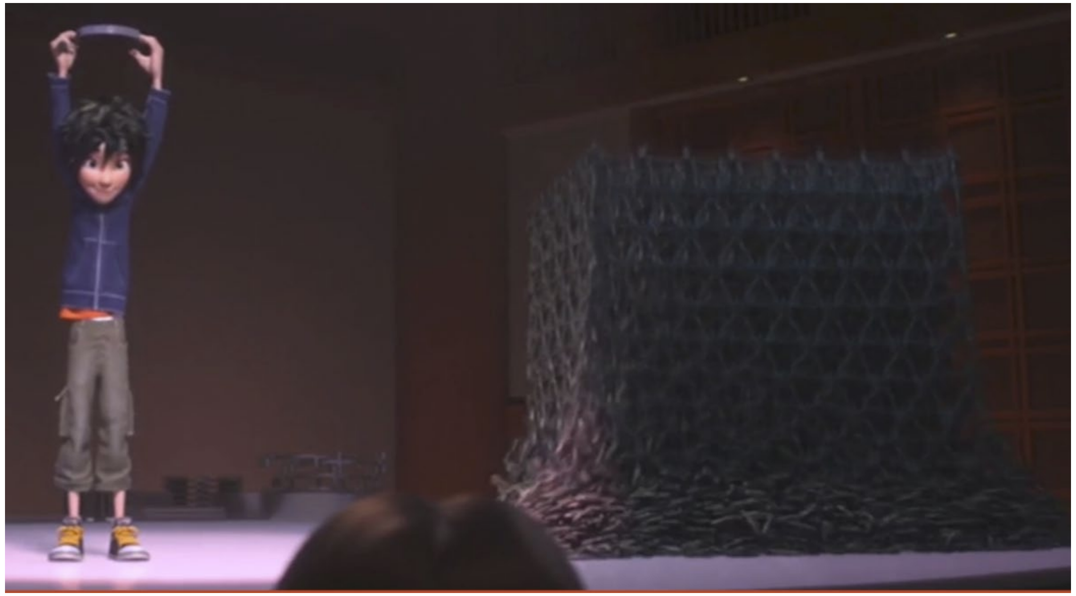
- Building a machine with human-like behaviors

→ Artificial General Intelligence (AGI)



The Ultimate Goal of Artificial Intelligence

- Brain-machine intelligence

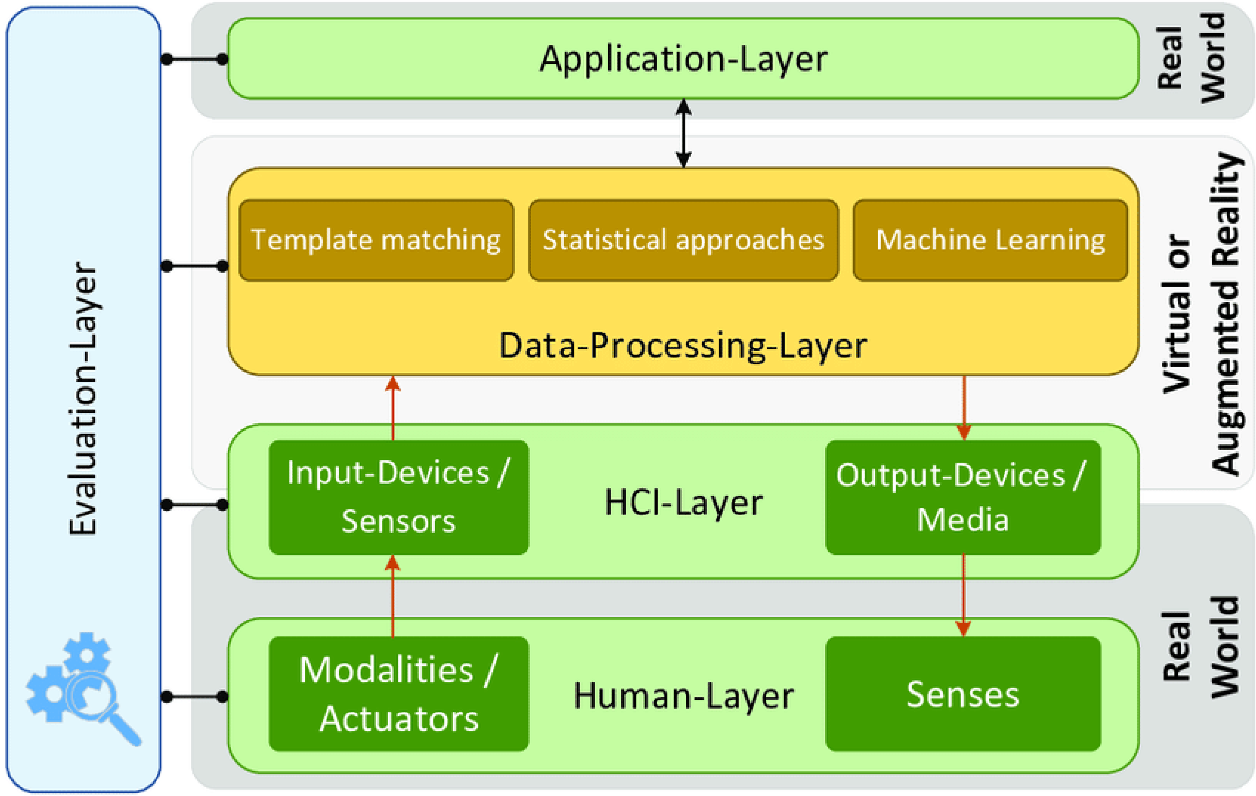


Big Hero 6, 2014

Neuralink, 2020

The Ultimate Goal of Artificial Intelligence

- Human-Computer Interaction (HCI)



Bachmann, D., Weichert, F., & Rinkenauer, G. (2018). Sensors, 18(7), 2194.

The Ultimate Goal of Artificial Intelligence

- Human-Computer Interaction (HCI)

Minority Report, 2002

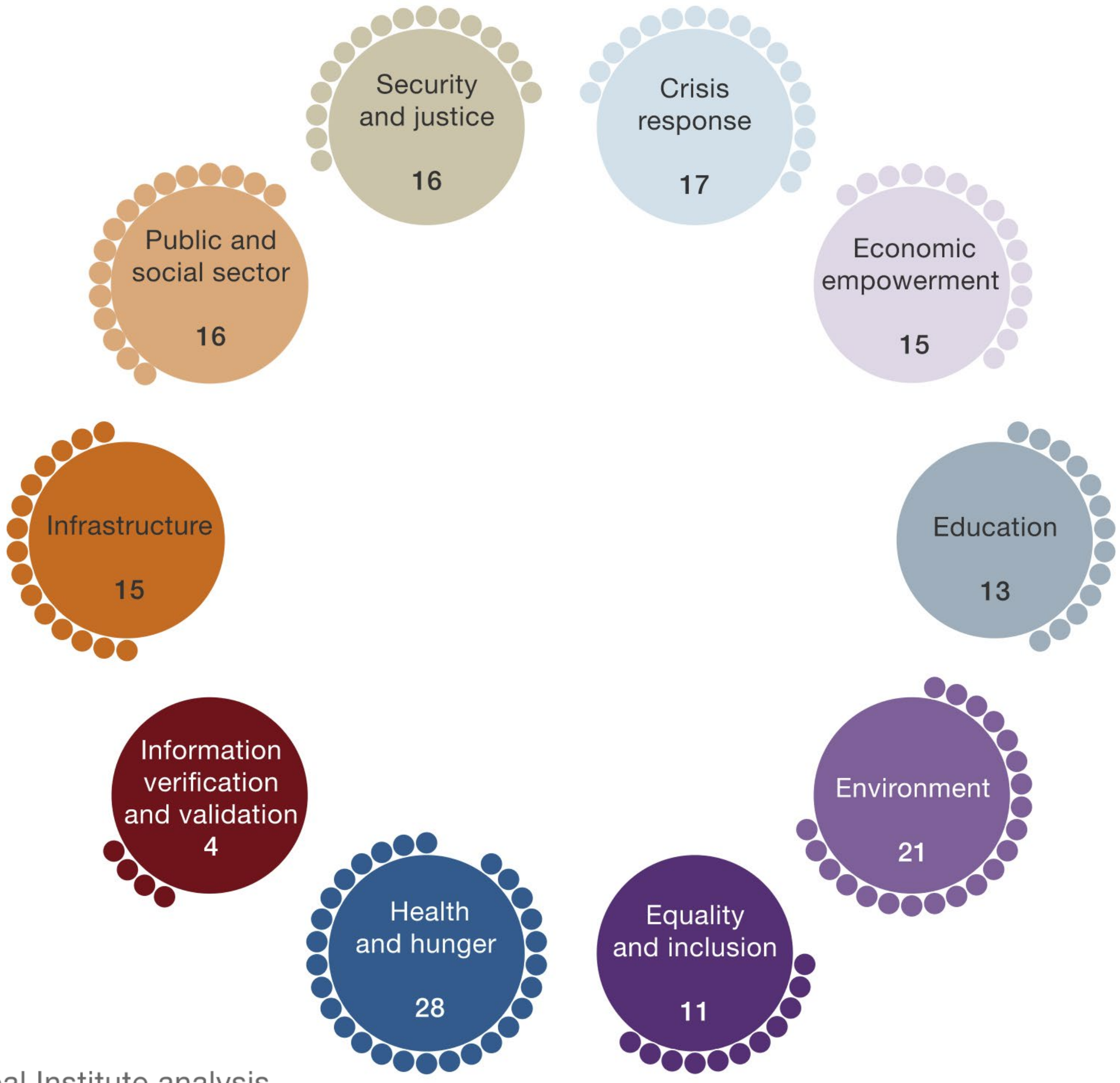
Machine learning is everywhere…

"Google's always used machine learning. In all the areas we applied it to, speech recognition, then image understanding, and eventually language understanding, we saw tremendous improvements.

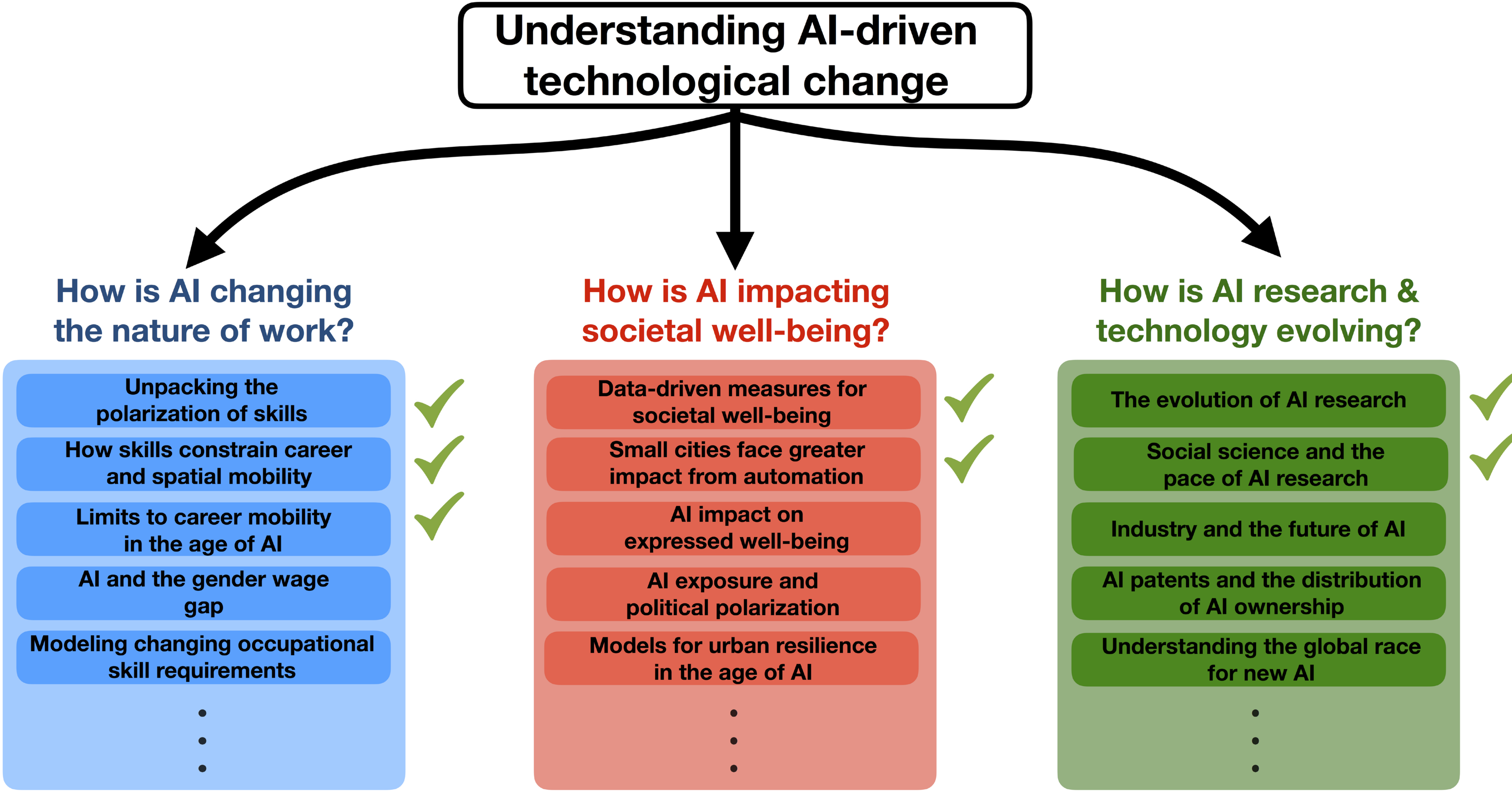
- John Gianadrea, then VP of Engineering, Google"

- "A craftsman who wishes to practice his craft well must first sharpen his tools."(工欲善其事 必先利其器)

ML/ DL/ AI in Domains of Social Good



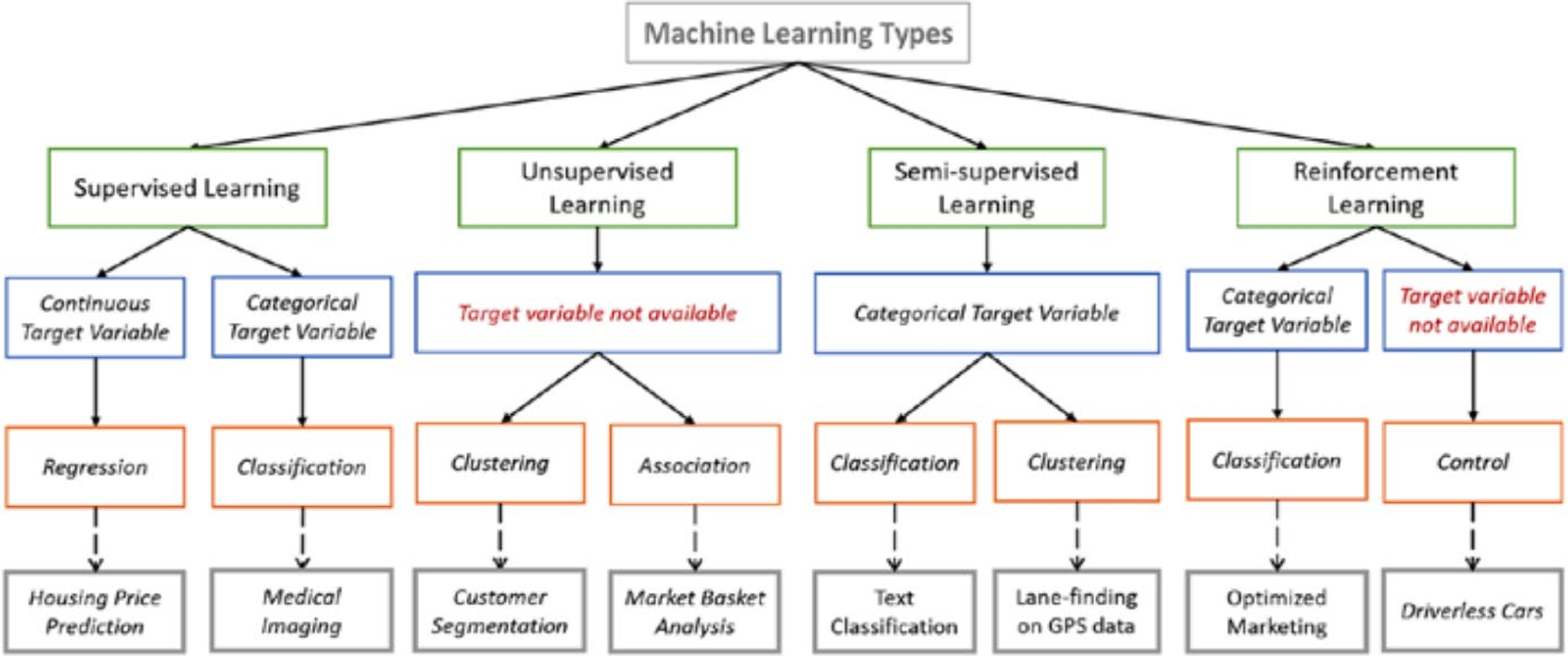
McKinsey&Company | **Source:** McKinsey Global Institute analysis


Understanding AI-driven technological change
How is AI changing the nature of work?
Unpacking the polarization of skills
How skills constrain career and spatial mobility
Limits to career mobility in the age of AI
AI and the gender wage gap
Modeling changing occupational skill requirements
How is AI impacting societal well-being?
Data-driven measures for societal well-being
Small cities face greater impact from automation
AI impact on expressed well-being
AI exposure and political polarization
Models for urban resilience in the age of AI
How is AI research & technology evolving?
The evolution of AI research
Social science and the pace of AI research
Industry and the future of AI
AI patents and the distribution of AI ownership
Understanding the global race for new AI

What is Machine Learning

by OxfordSparks

https://www.youtube.com/watch?v=f_uwKZIAeM0



- **NNs**
- GLM
- **Decision Trees**

- **SVM**
- Discriminant Analysis
- Naïve Bayes
- Nearest Neighbor

- **NNs**
- K-Means
- Gaussian Mixture
- Hidden Markov Model

The Fundamentals of Machine Learning

- What is Machine Learning? What problems does it try to solve? What are the main categories and fundamental concepts of Machine Learning systems?

  - The main steps in a typical Machine Learning project.
  - Learning by fitting a model to data.
  - Optimizing a cost function.
  - Handling, cleaning, and preparing data.
  - Selecting and engineering features.
  - Selecting a model and tuning hyperparameters using cross-validation.
  - The main challenges of Machine Learning, in particular underfitting and overfitting (the bias/variance tradeoff).
  - Reducing the dimensionality of the training data to fight the curse of dimensionality.
  - The most common learning algorithms: Linear and Polynomial Regression, Logistic Regression, k-Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forests, and Ensemble methods.

Neural Networks and Deep Learning

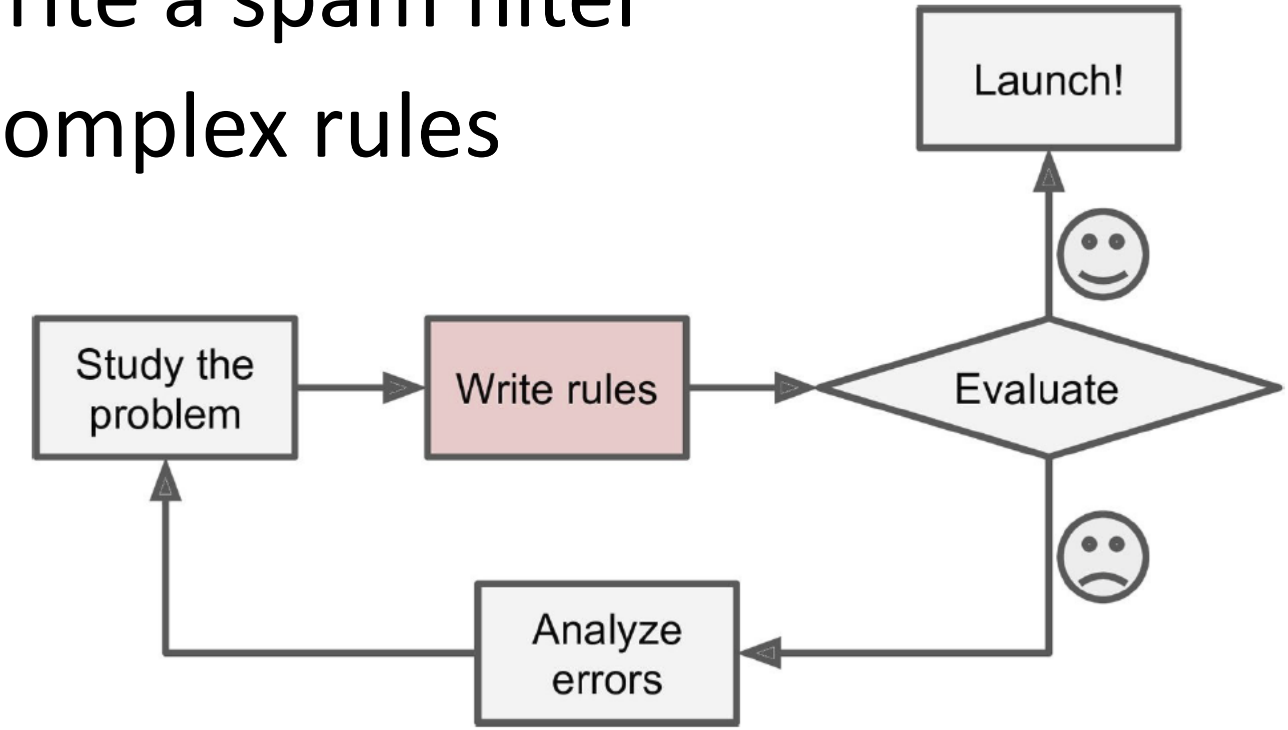
- What are neural nets? What are they good for?
- Building and training neural nets using TensorFlow.
- The most important neural net architectures: feedforward neural nets,
- convolutional nets, recurrent nets, long short-term memory (LSTM) nets, and autoencoders.
- Techniques for training deep neural nets.
- Scaling neural networks for huge datasets.
- Reinforcement learning.

What Is Machine Learning?

- The science (and art) of programming computers so they can learn from data.
- The field of study that gives computers the ability to learn without being explicitly programmed.
- To learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

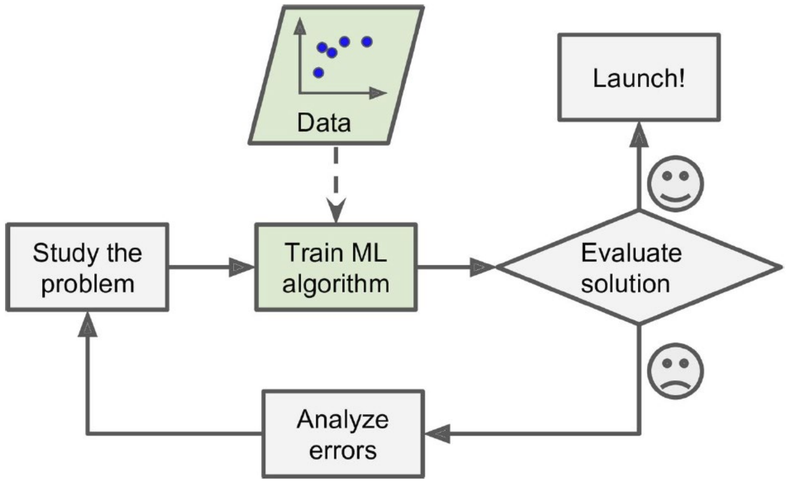
Why Use Machine Learning

- Traditional approach: detection algorithm for each of the patterns that you noticed in subjects or senders Example: to write a spam filter
- A long list of complex rules



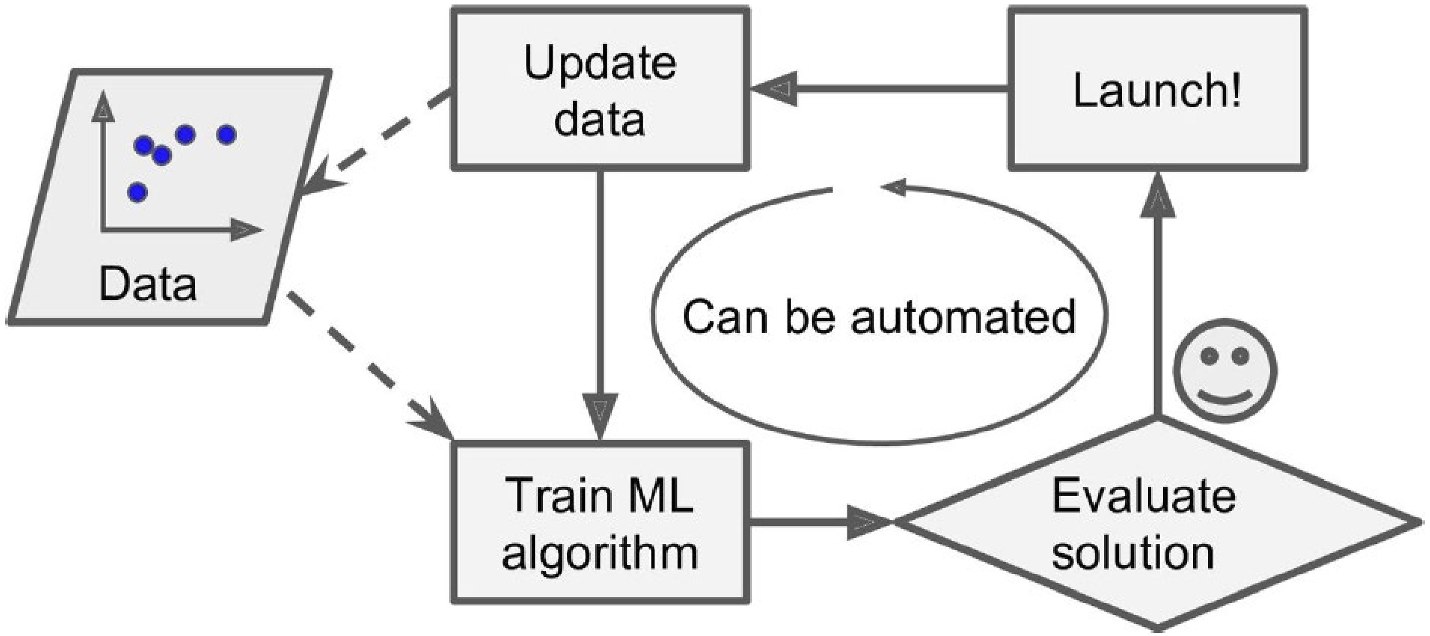
Why Use Machine Learning

- Machine learning approach: automatically leans which words and phrases are good predictors of spam by detecting unusually frequent patterns of words
- Much shorter, easier to maintain, and more accurate



Adaptive Machine Learning

- Automatically notices that some pattern has become unusually frequent in spam flagged by users



Machine Learning is great for…

- Problems for which existing solutions require a lot of hand-tuning or long lists of rules
  - One ML algorithms can often simplify code and perform better.
- Complex problems for which there is no good solution at all using a traditional approach
  - The best ML techniques can find a solution.
- Fluctuating environments
  - A ML system can adapt to new data
- Getting insights about complex problems and large amounts of data.

Types of Machine Learning Systems

- Whether or not they are trained with human supervision
(supervised, unsupervised, semi-supervised, and reinforcement learning)
- Whether or not they can learn incrementally on the fly
(online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do
(instance-based versus model-based learning)

Supervised Learning

- The training data you feed to the algorithm includes the desired solutions, called labels.
  - Classification, regression

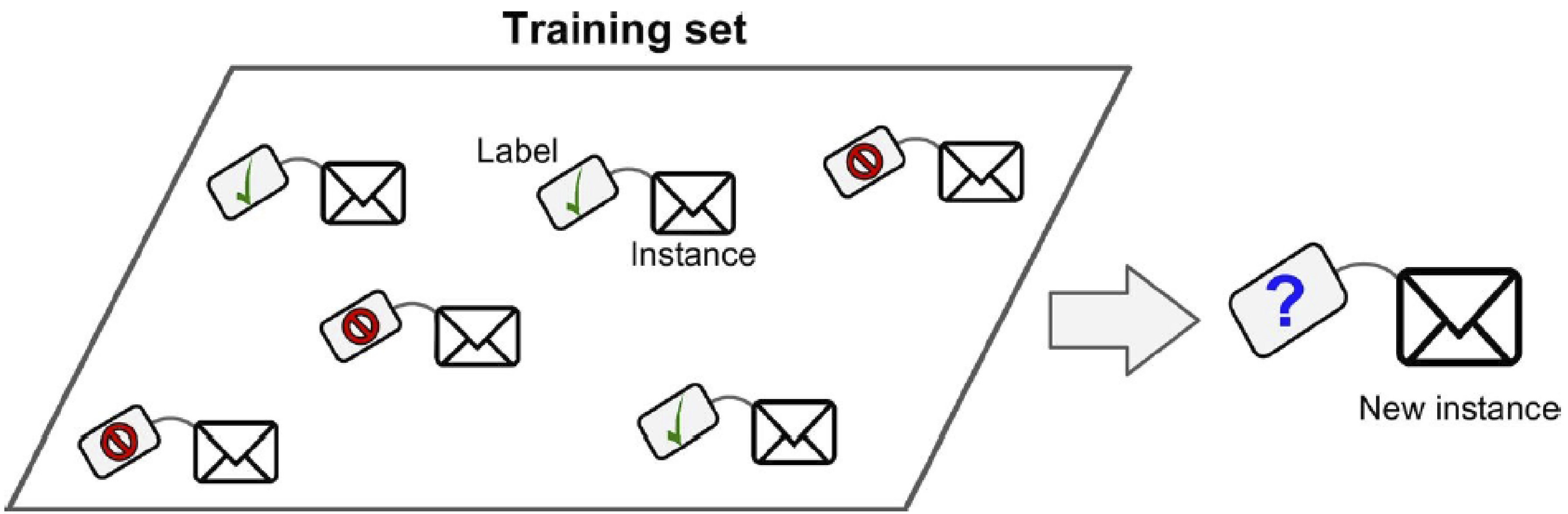


Figure 1-5. A labeled training set for supervised learning (e.g., spam classification)

Regression

- To predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.) called predictors

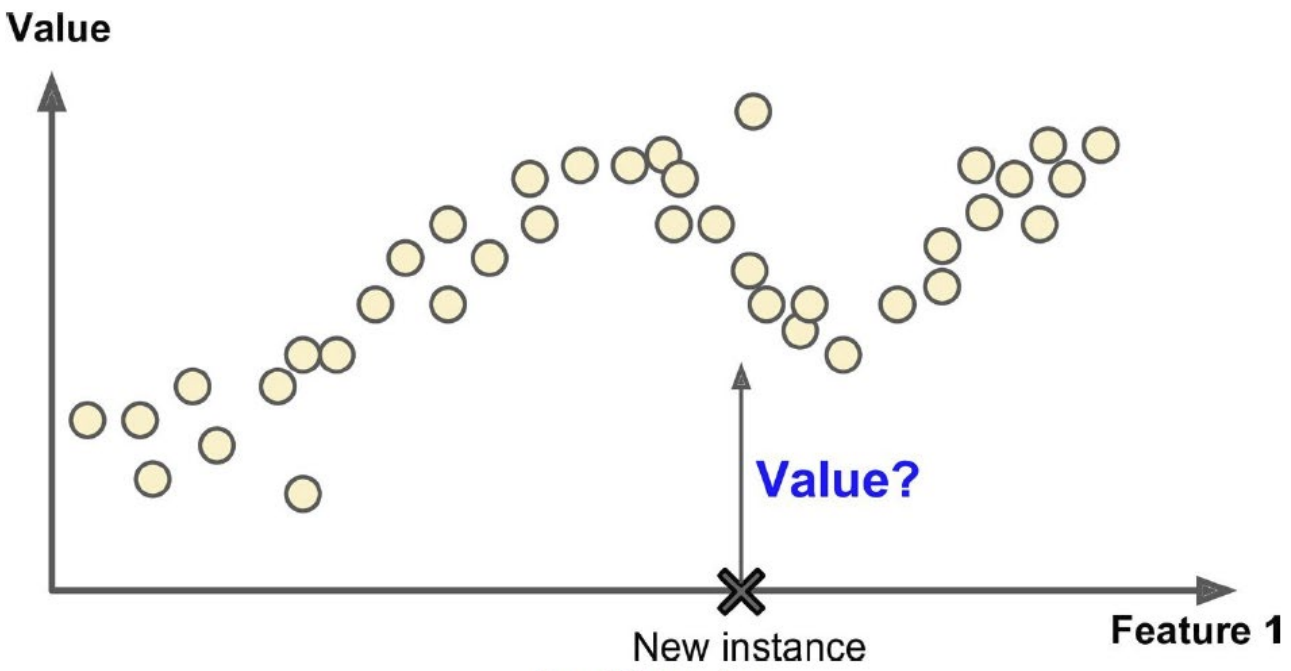
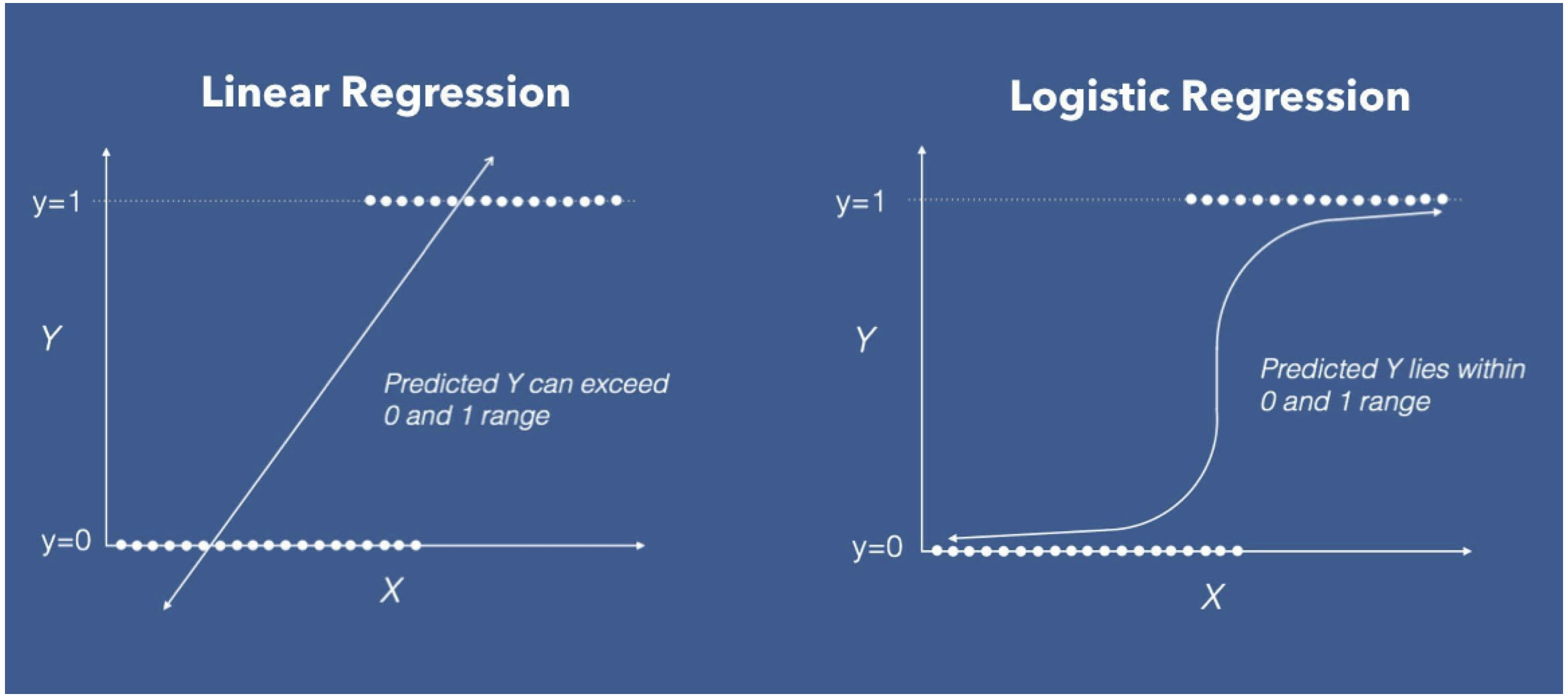


Figure 1-6. Regression

Logistic Regression

- Logistic Regression is commonly used for classification, as it can output a value that corresponds to the probability of belonging to a given class



Support Vector Machine

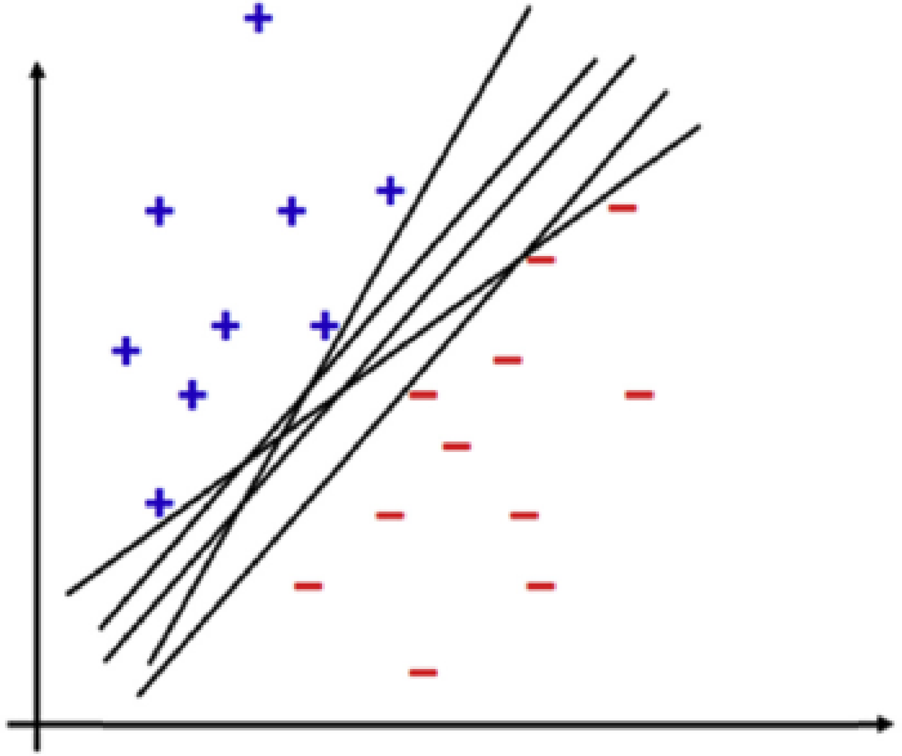


Fig. 1. Example of different linear functions for a given set of points.

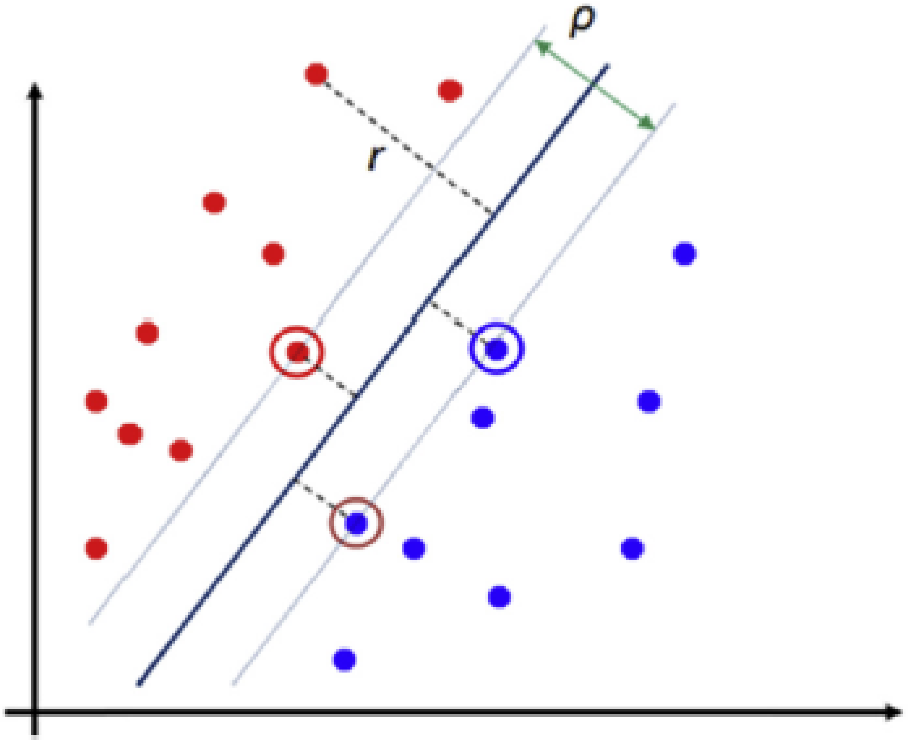


Fig. 2. Margins in an support vector machine model.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Support Vector Machine

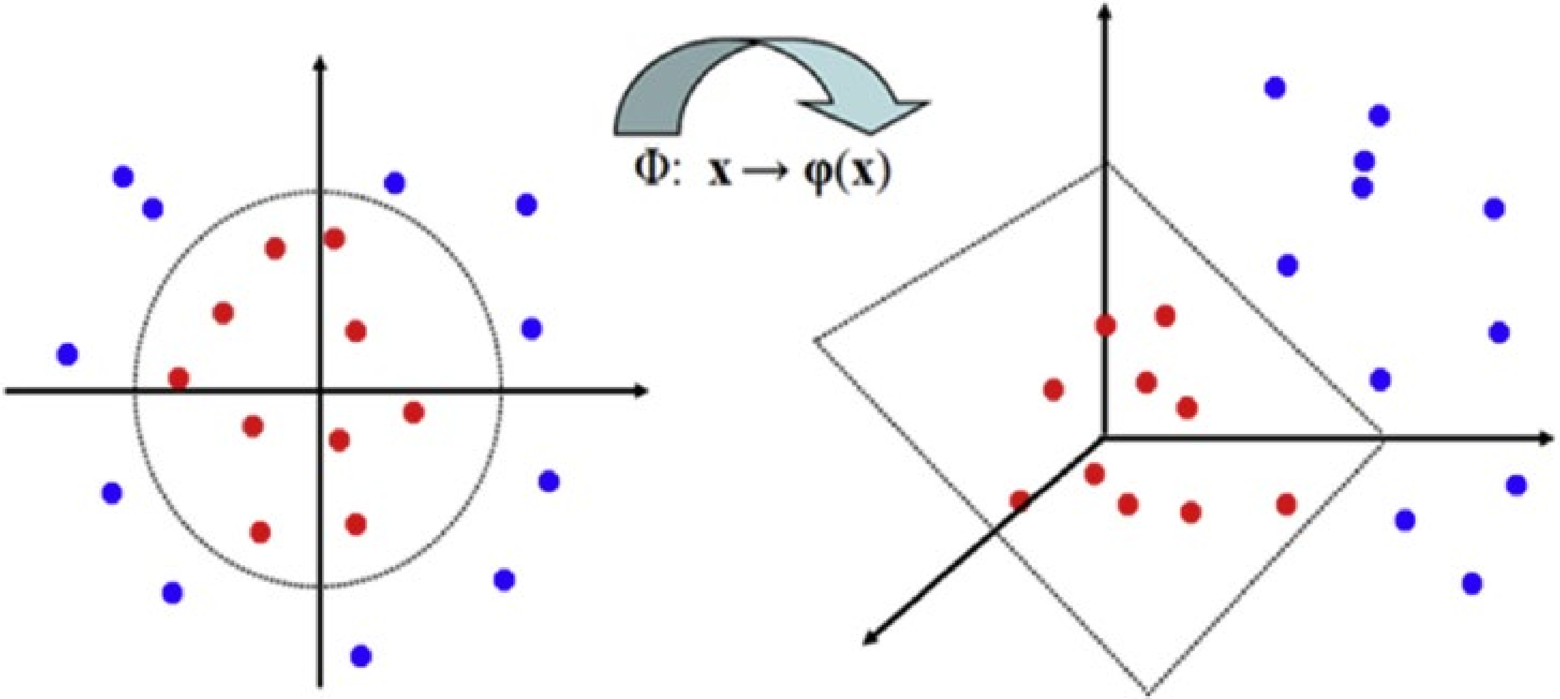


Fig. 3. Kernel function mapping of examples to a higher dimension.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Machine Learning Overview

- Support Vector Machine

Table 1
Advantages and disadvantages of SVMs

Advantages	Disadvantages
Nonlinear data types can be modeled with the use of kernel methods by SVMs. These have been very popular for many challenging tasks in the past decade.	Requires setting of many parameters and heavily dependent on choosing a good kernel for nonlinear data; typically requires a machine learning expert rather than a domain expert.
Kernels allow for flexible hypothesis.	The learned models can be difficult to interpret.

Abbreviation: SVM, support vector machine.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Random Forest

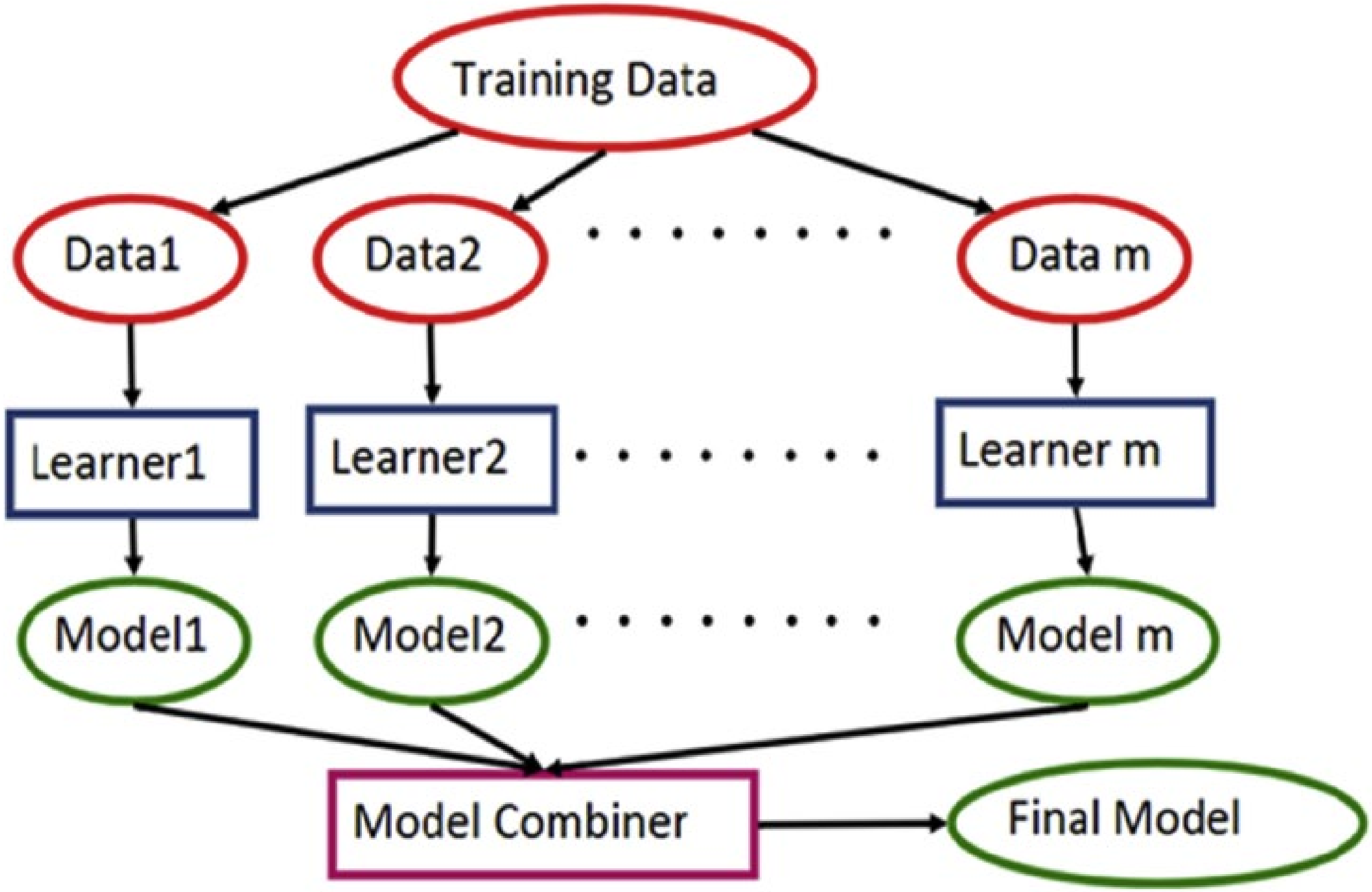


Fig. 4. An ensemble classifier. For random forest, the learners are decision trees.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Random Forest

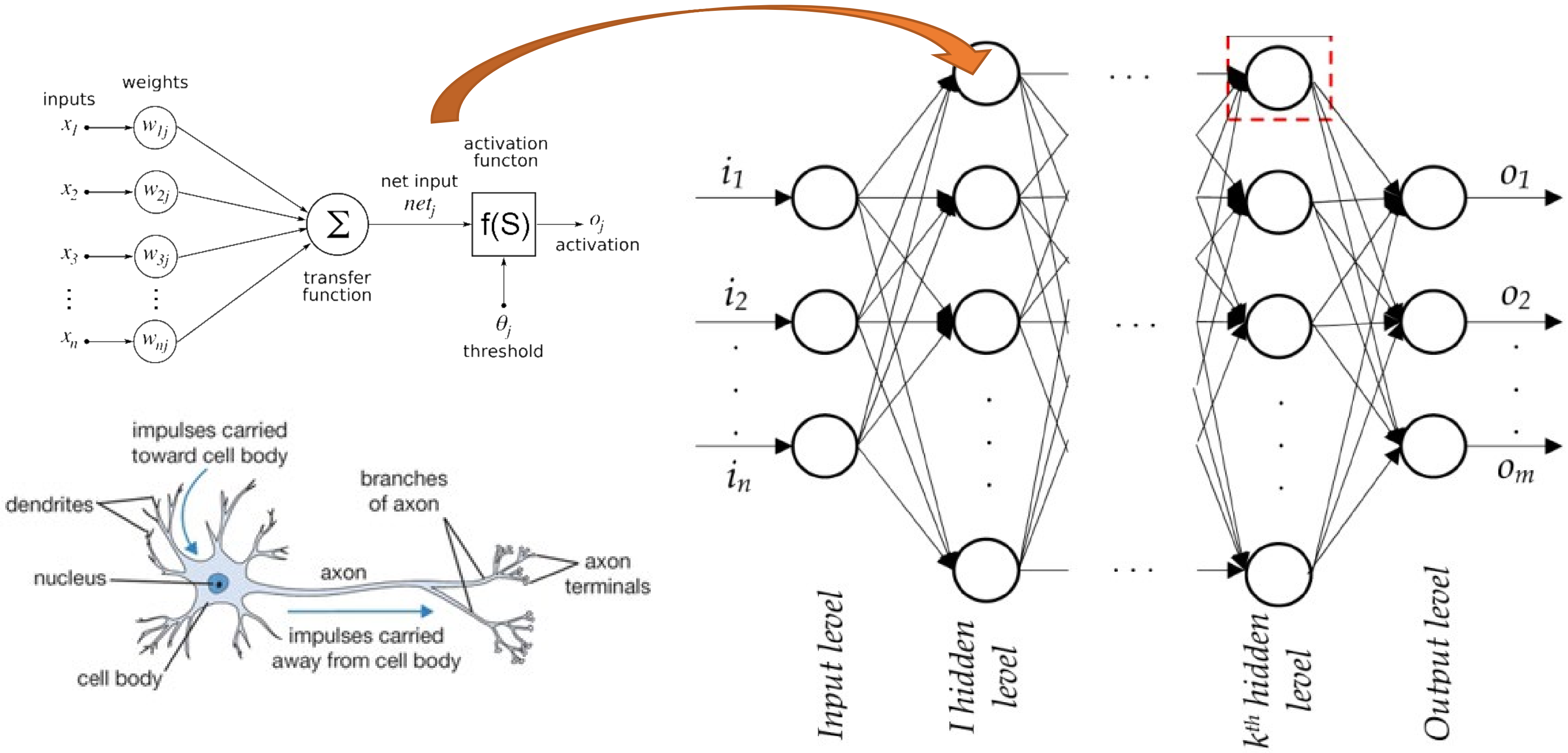
- Random Forests

Table 2
Advantages and disadvantages of random forests

Advantages	Disadvantages
The model is scalable and robust. Popular ensemble method that is inevitably the first approach taken for large datasets.	Performance can be lower in the presence of noise and outliers.
Can handle missing data. Can be extended to learning multiple types of models; there is no necessity for the base classifiers to be of the same type, that is, trees.	Interpretability is sometimes an issue because, although the individual models are interpretable, their combination is not necessarily interpretable.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Artificial Neural Networks



Artificial Neural Networks

- Artificial Neural Networks

Table 3
Advantages and disadvantages of artificial neural networks

Advantages	Disadvantages
The model can approximate any function, linear or nonlinear.	The models are not interpretable.
Scalable to very large problems and are recently very popular owing to their ability to handle millions of features during training.	A reasonably large amount of data is required for training.

Billings et al. Neuroimag Clin N Am 27 (2017) 609–620

Some of the most important supervised learning algorithms

- Linear Regression
- Logistic Regression
- k-Nearest Neighbors
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Unsupervised Learning

- The training data is unlabeled.
- The system tries to learn without a teacher

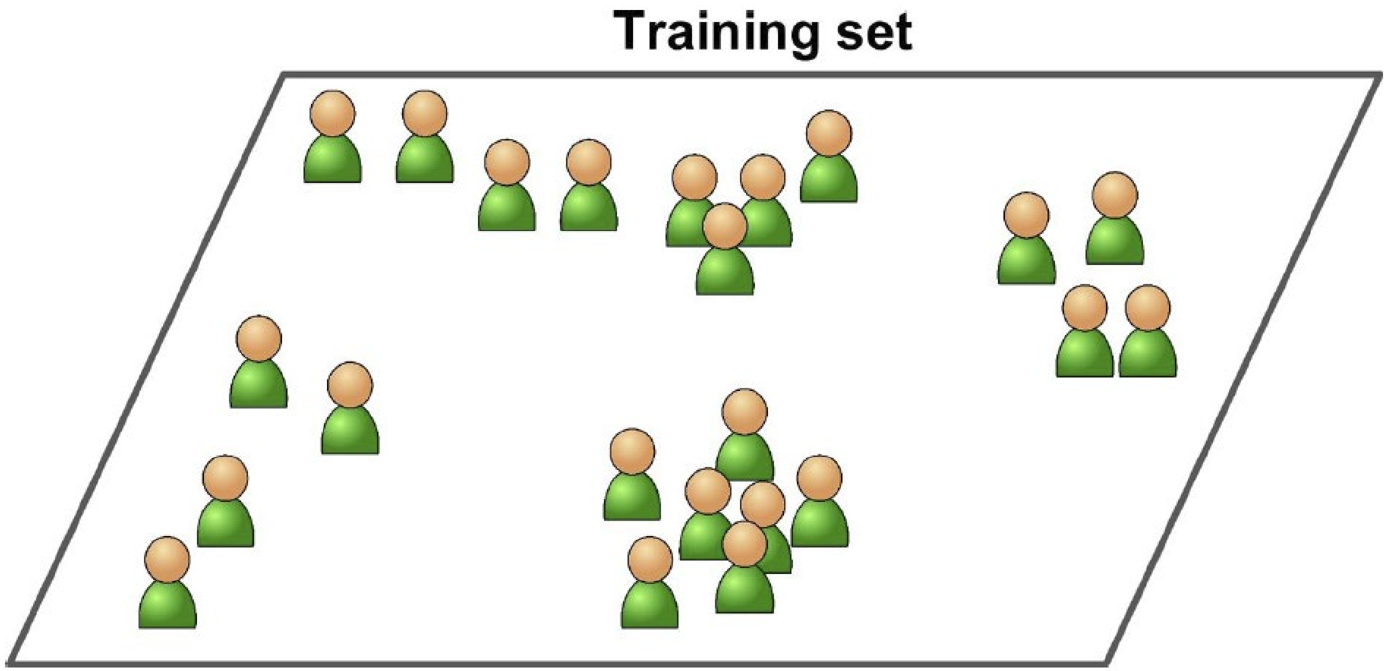


Figure 1-7. An unlabeled training set for unsupervised learning

Clustering

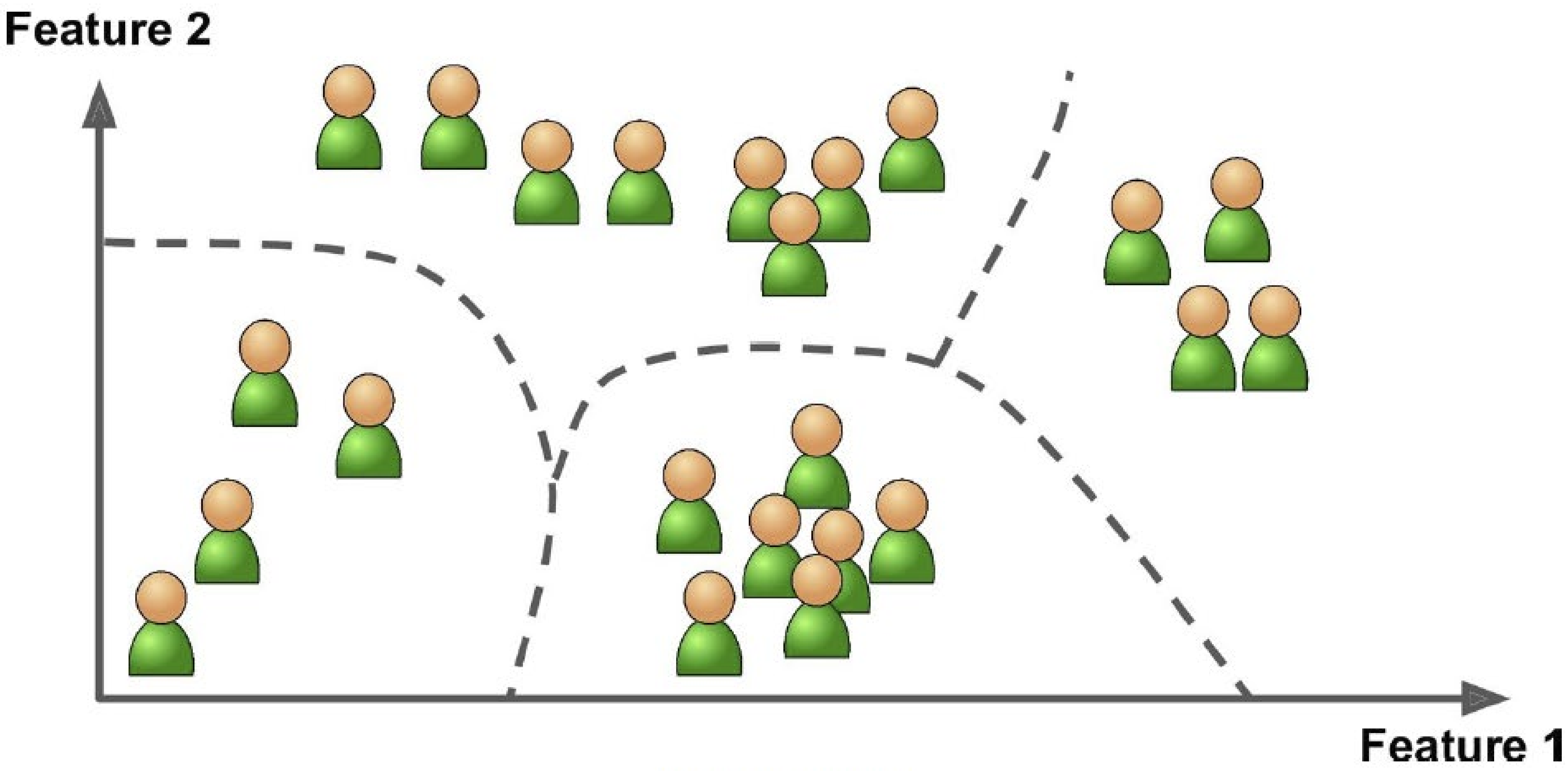


Figure 1-8. Clustering

Whether or not they can learn incrementally on the fly
Batch or **Online Learning**

- Human learns online/ incrementally
- Typical machine learning **is batch learning** :
  - the system is incapable of learning incrementally
  - it must be trained using all the available data
  - this is called *offline* learning
  - new data -> update the data (old + new) and train a new version of the system from scratch as often as needed
  - drawbacks: requires a lot of computing resources (CPU, memory space, disk space, disk I/O, network I/O, etc.

Online/ incremental Learning

- Train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches.
- Each learning step is fast and cheap
- This whole process is usually done offline (i.e., not on the live system)

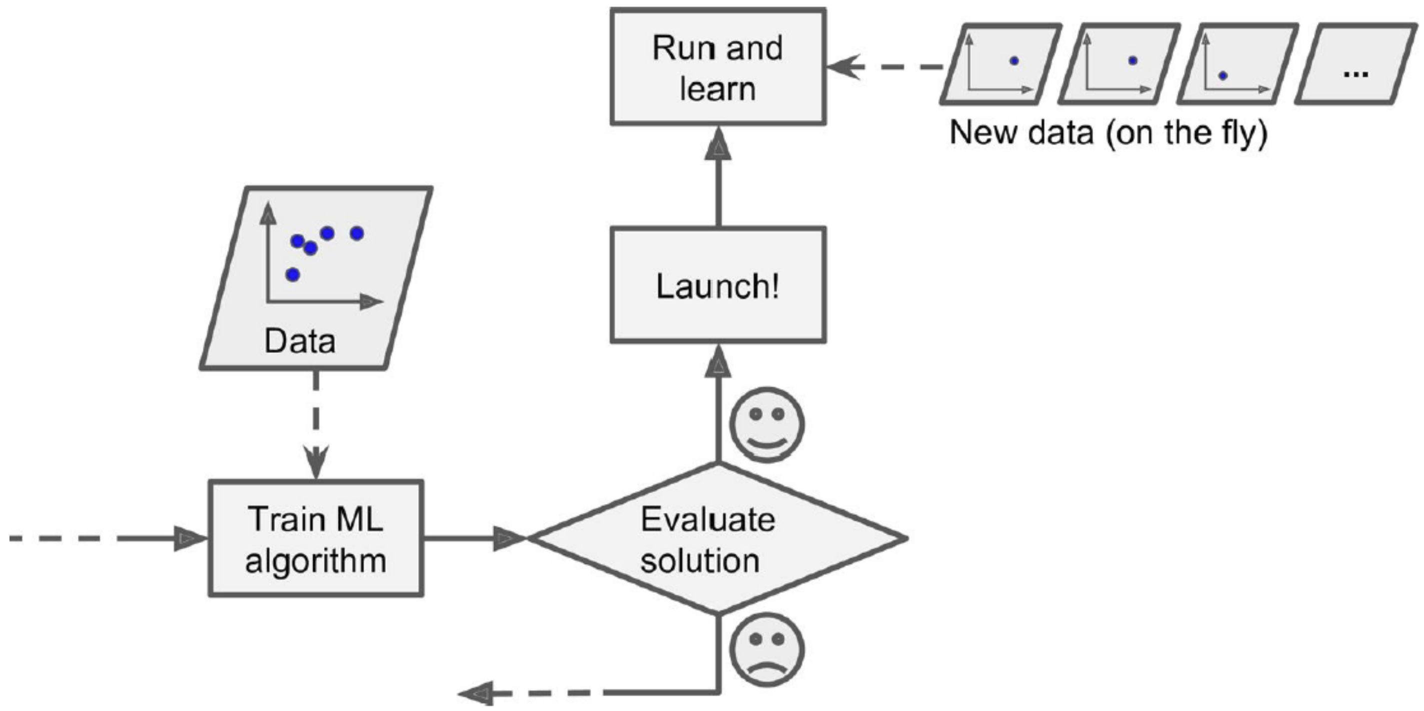


Figure 1-13. Online learning

Online/ incremental Learning

- Learning rate: how fast the system should adapt to changing data
  - high: rapidly adapt to new data and forgot old ones
  - low: learn slowly, sensitive to noise in the new data or nonrepresentative data points

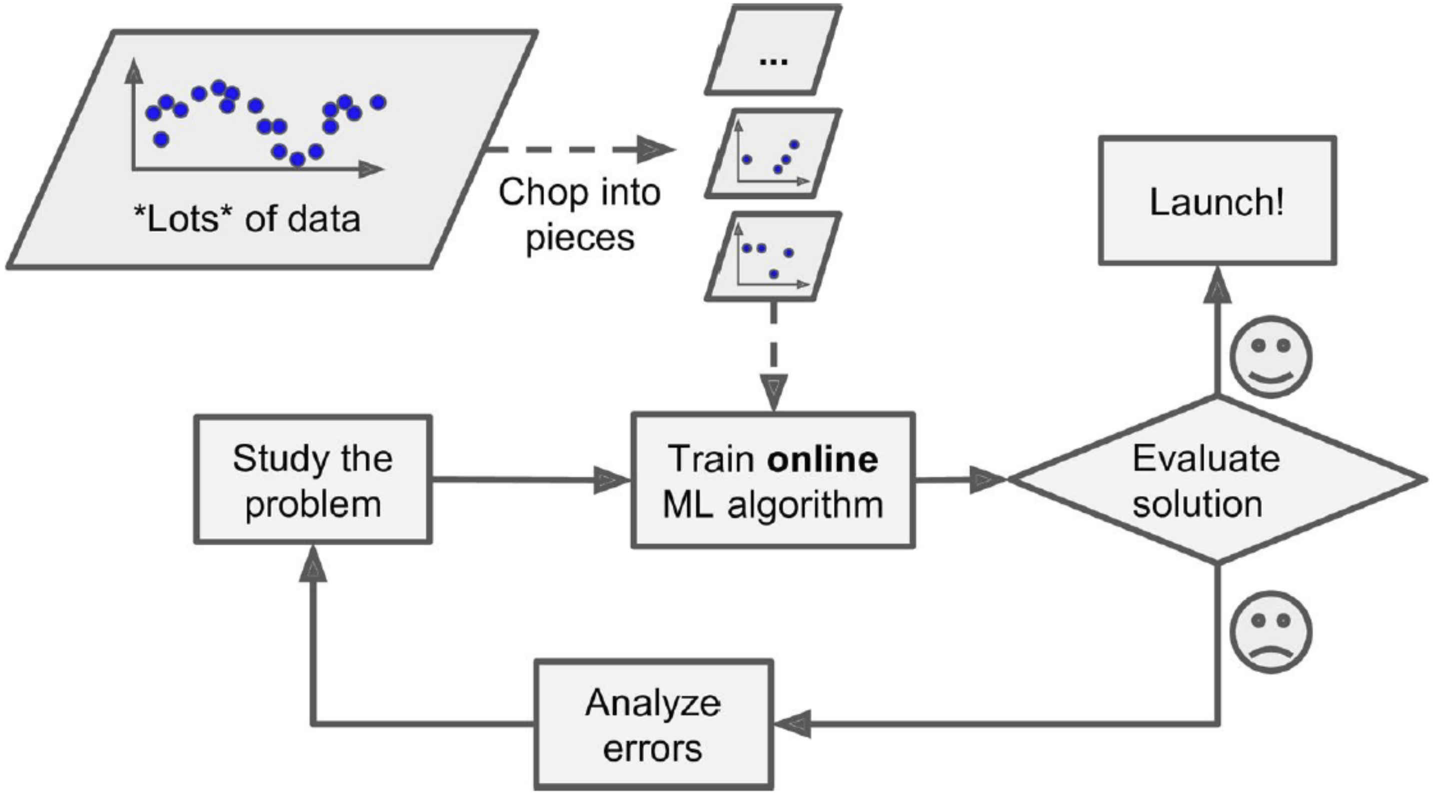


Figure 1-14. Using online learning to handle huge datasets

Online/ incremental Learning

- Bad data: how fast the system should adapt to changing data

  e.g. come from a malfunctioning sensor on a robot

  - the system's performance will gradually decline

  Solution:

  - to monitor your system closely and promptly switch learning off
  - to monitor the input data and react to abnormal data (e.g., using an anomaly detection algorithm)

Instance-based Learning or Model-based Learning

- Generalization
  Having a good performance measure on the training data is good, but insufficient; the true goal is to perform well on new instances.
- Instance-based learning
  the most trivial form of learning
  the system learns the examples by heart, then generalizes to new cases using a **similarity measure**

Instance-based Learning

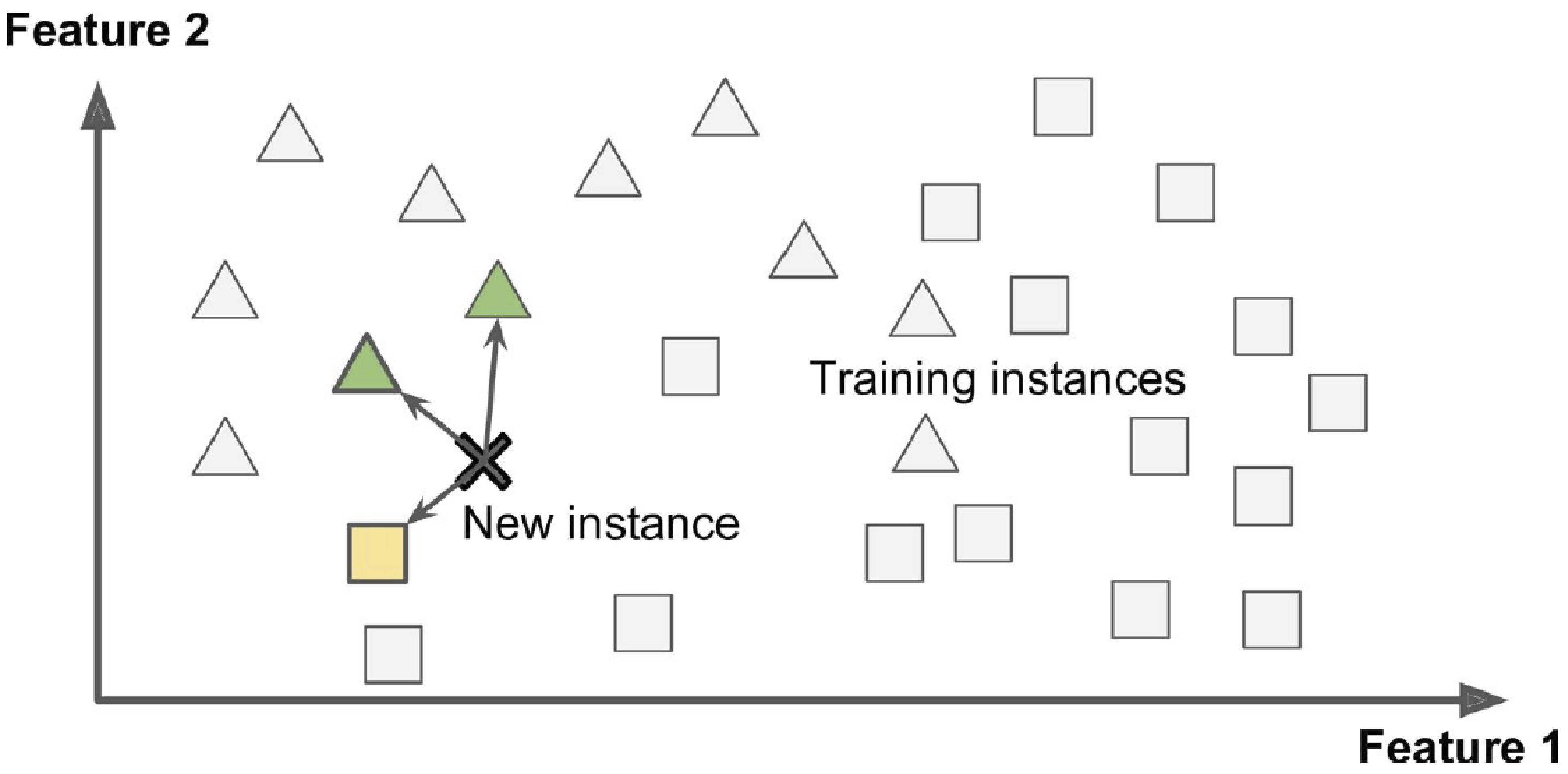
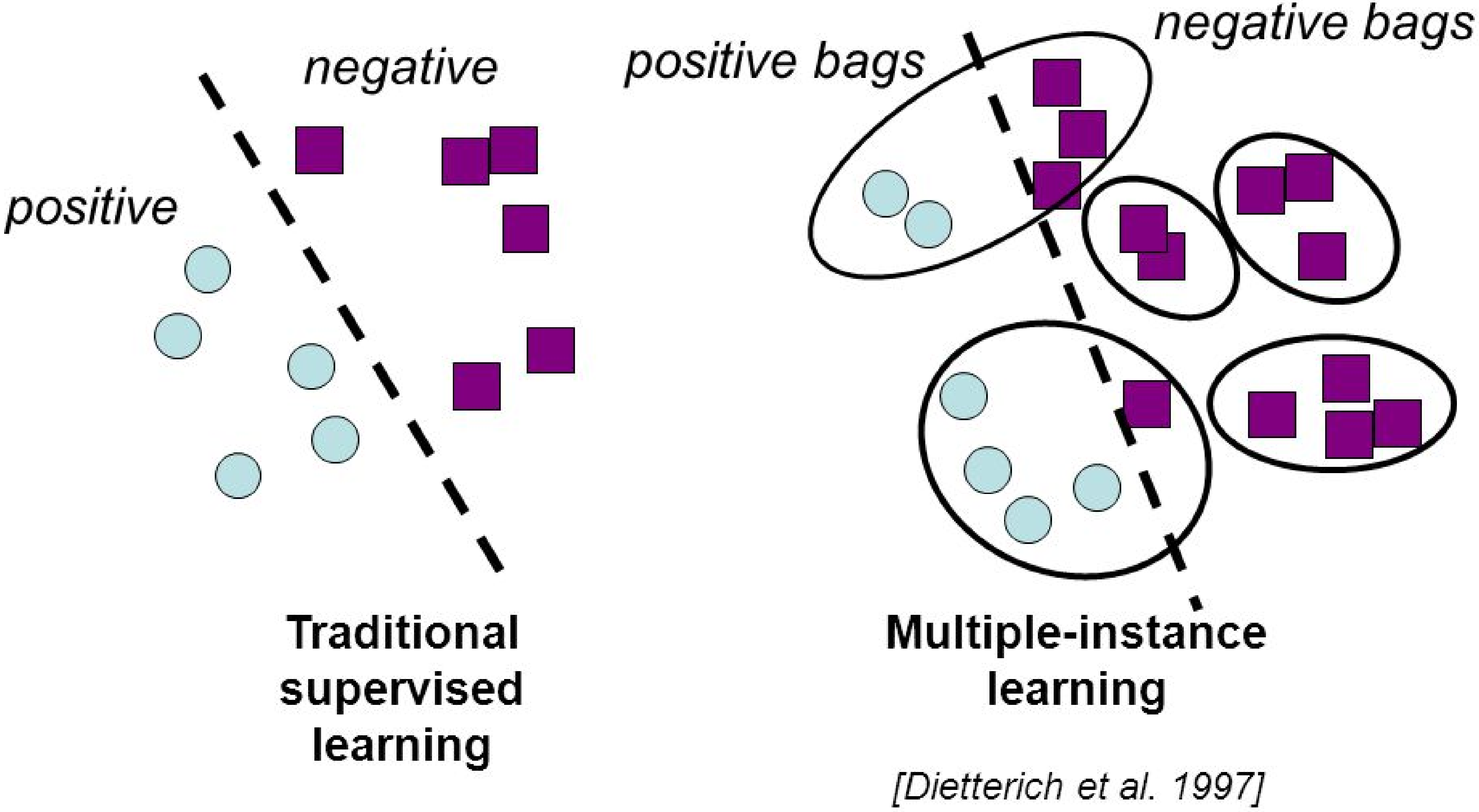


Figure 1-15. Instance-based learning

Multiple-Instance Learning



Model-based Learning

- To build a model of the training data, then use that model to make predictions for new data

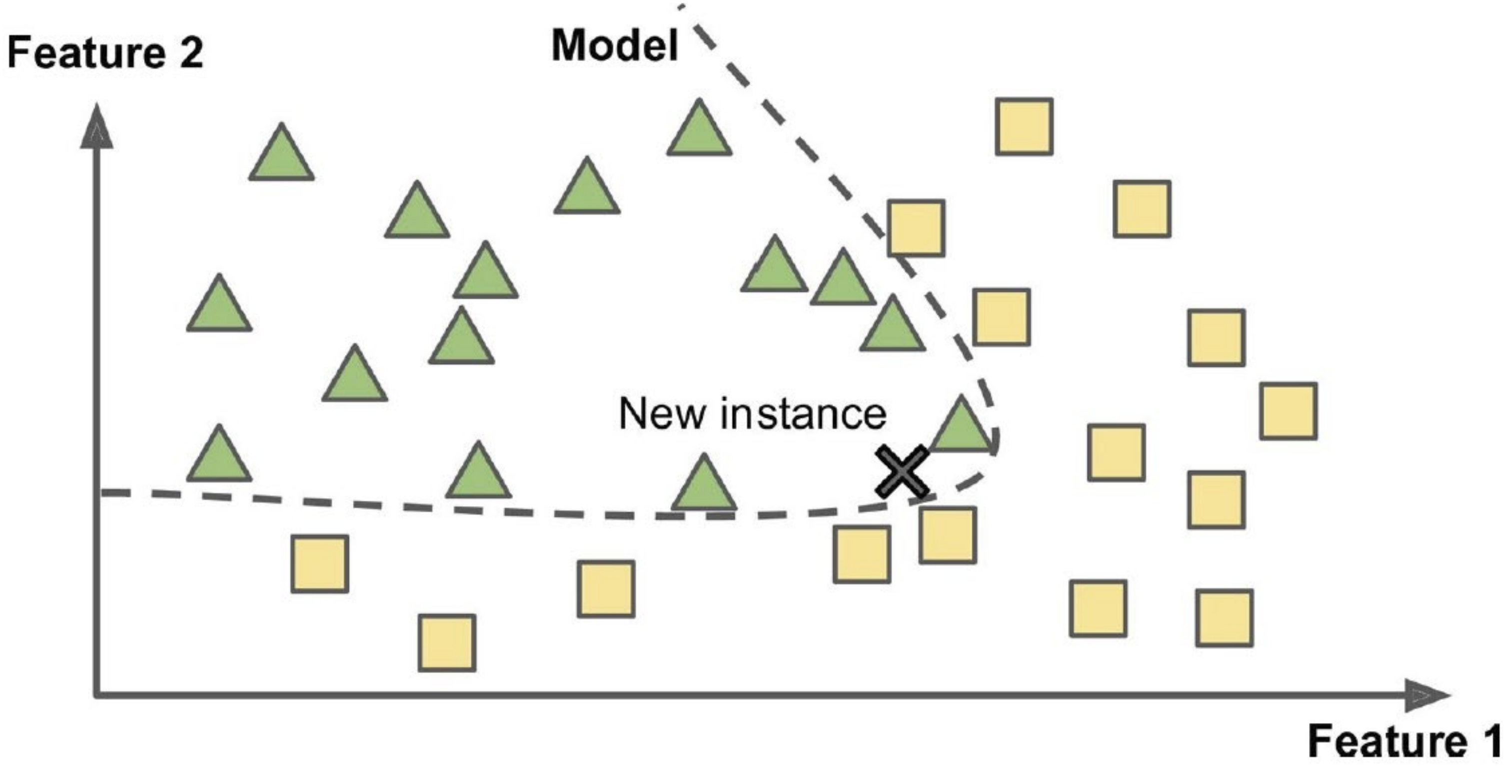


Figure 1-16. Model-based learning

Some of the most important unsupervised learning algorithms

- Clustering
  - k-Means
  - Hierarchical Cluster Analysis (HCA)
  - Expectation Maximization
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning:
  discover interesting relationship between attributes (e.g. supermarket)
  - Apriori
  - Eclat

Visualization

- To understand how the data is organized and perhaps identify unsuspected patterns.
- Dimension reduction, feature extraction

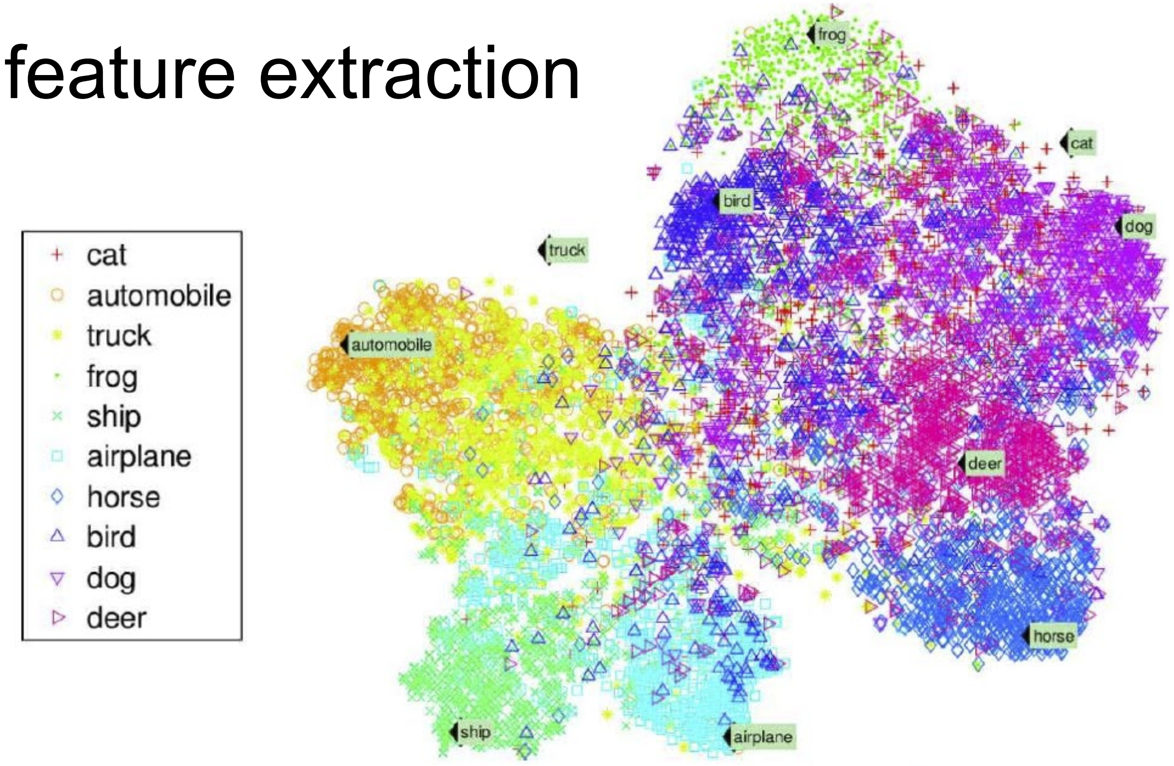


Figure 1-9. Example of a t-SNE visualization highlighting semantic clusters[3]

Anomaly Detection

- The system is trained with normal instances, and when it sees a new instance it can tell whether it looks like a normal one or whether it is likely an anomaly

Figure 1-10. Anomaly detection

Semi-supervised learning

- Combinations of unsupervised and supervised algorithms

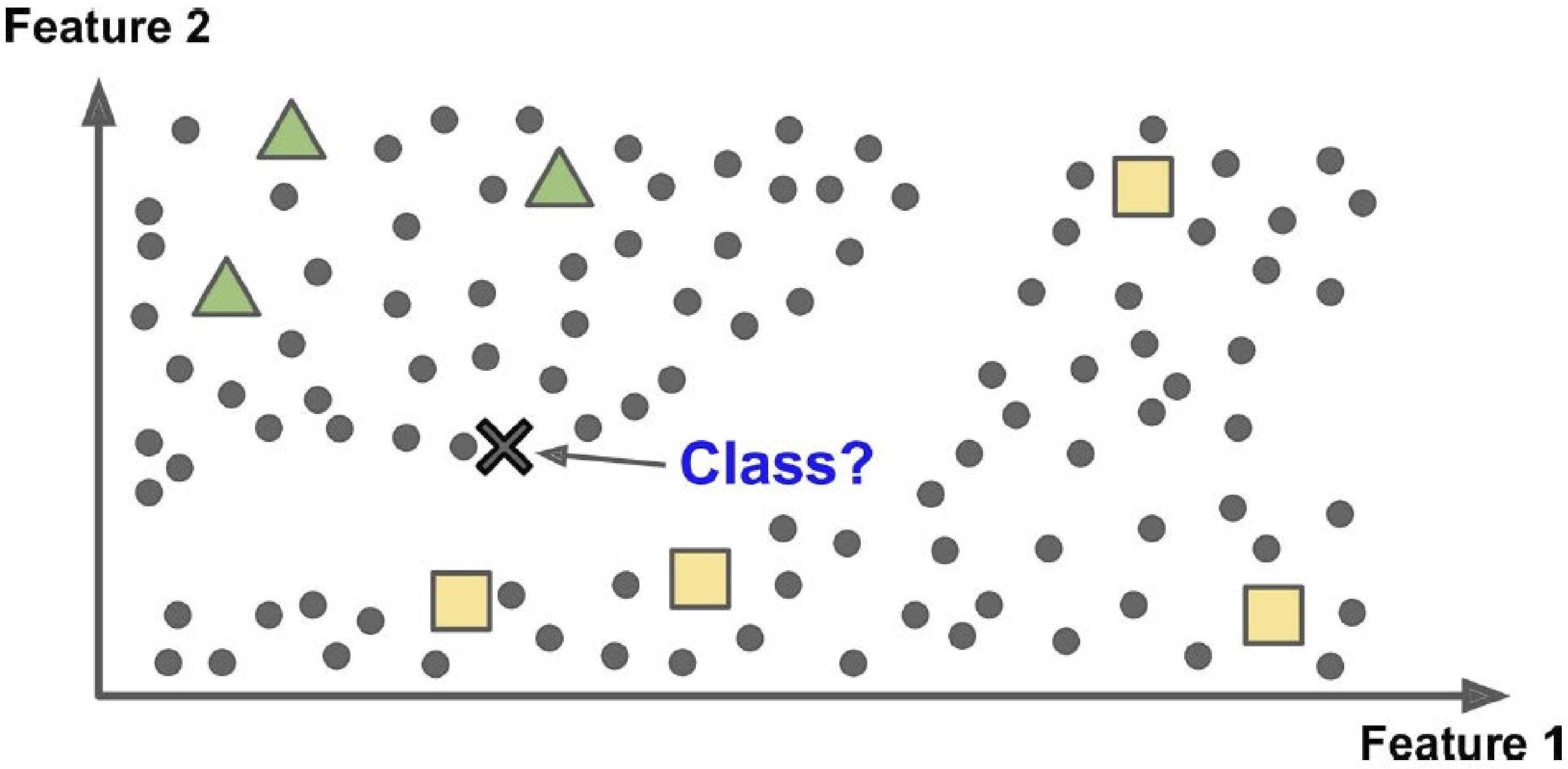


Figure 1-11. Semisupervised learning

Reinforcement Learning

- Agent
- Action
- Rewards

→ to learn the policy

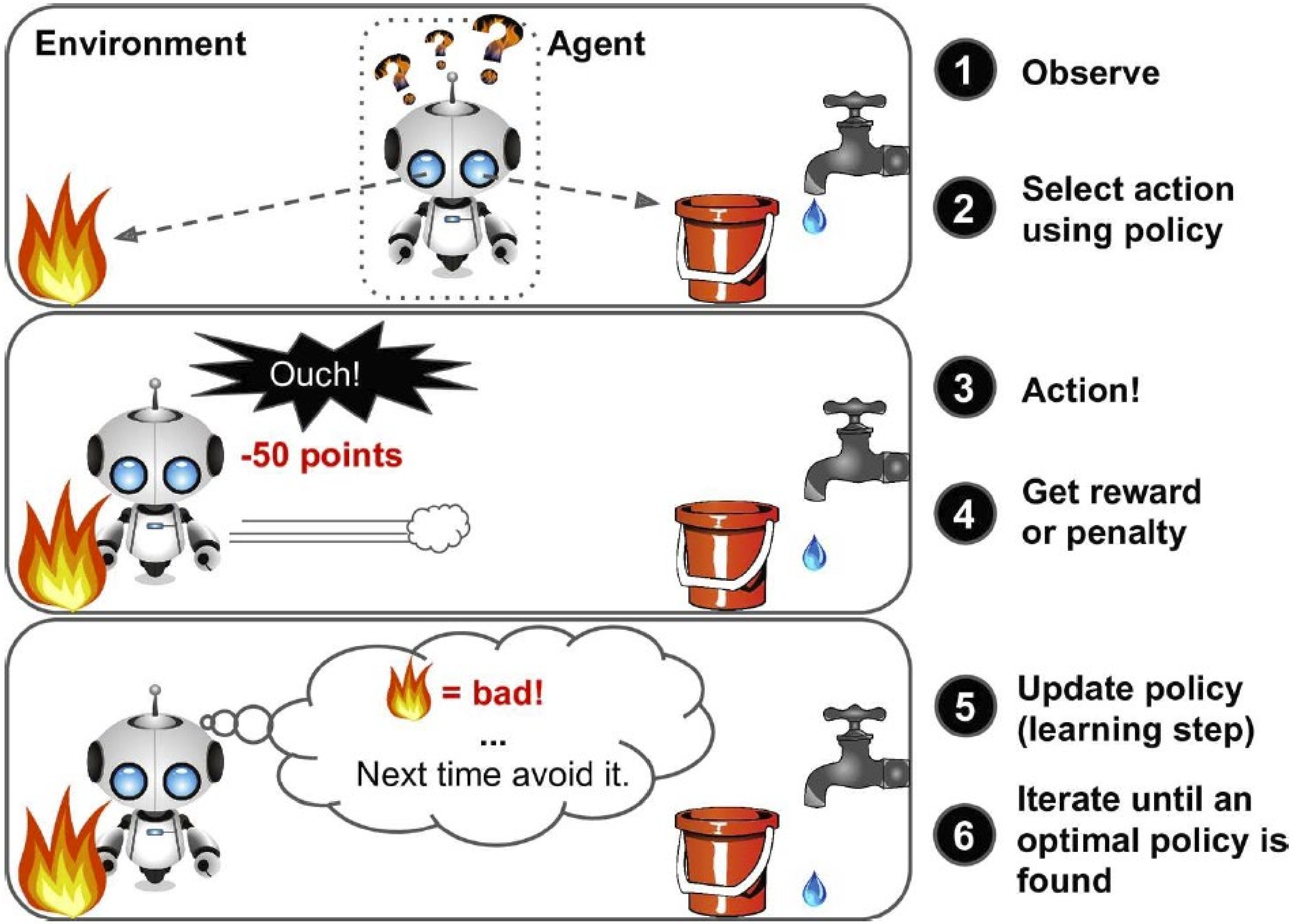
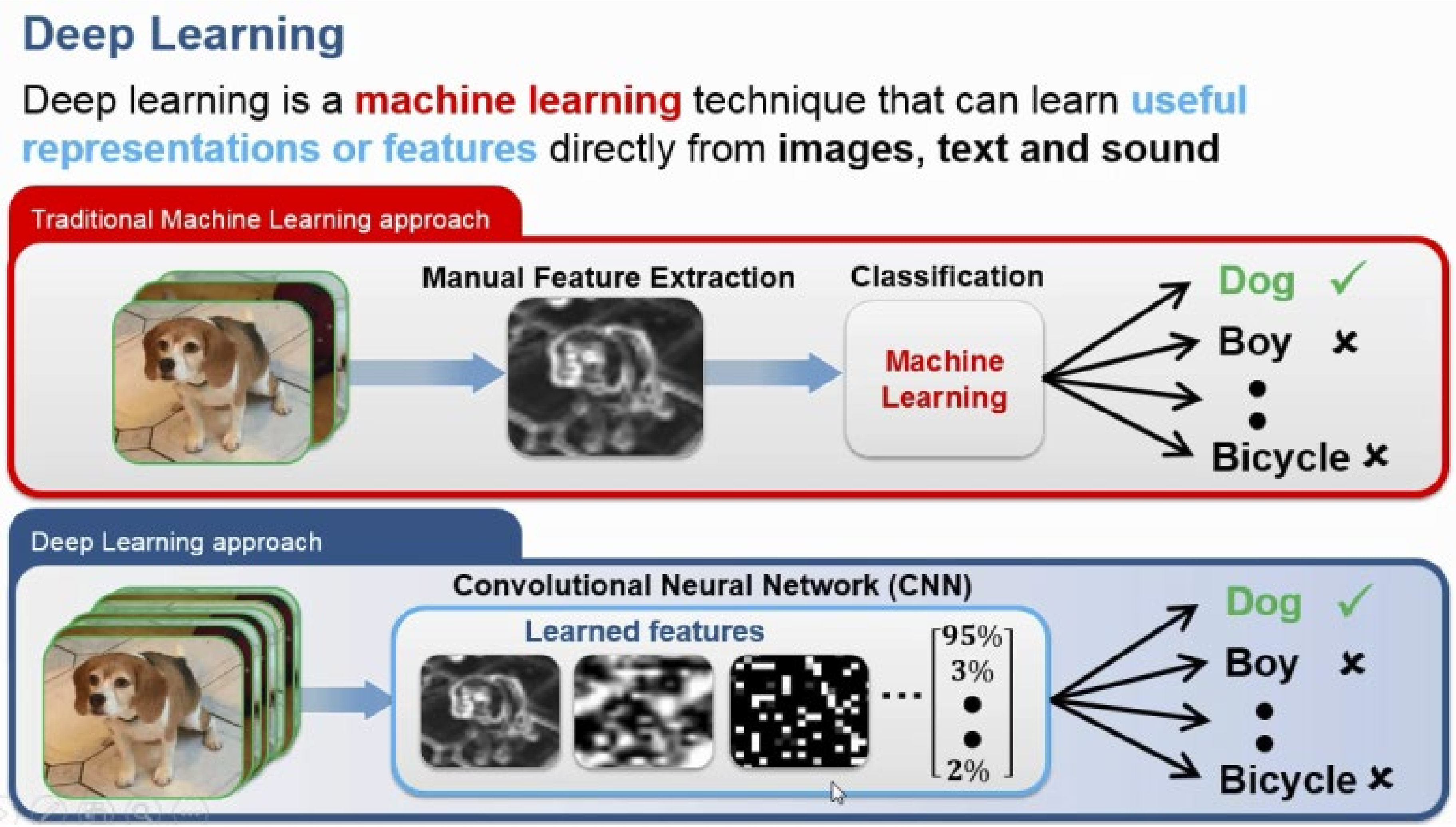


Figure 1-12. Reinforcement Learning

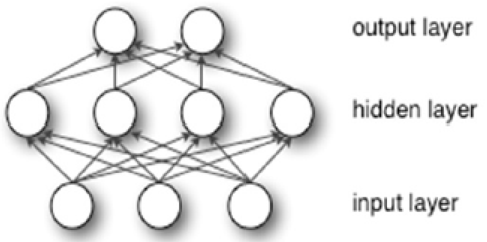
Step away

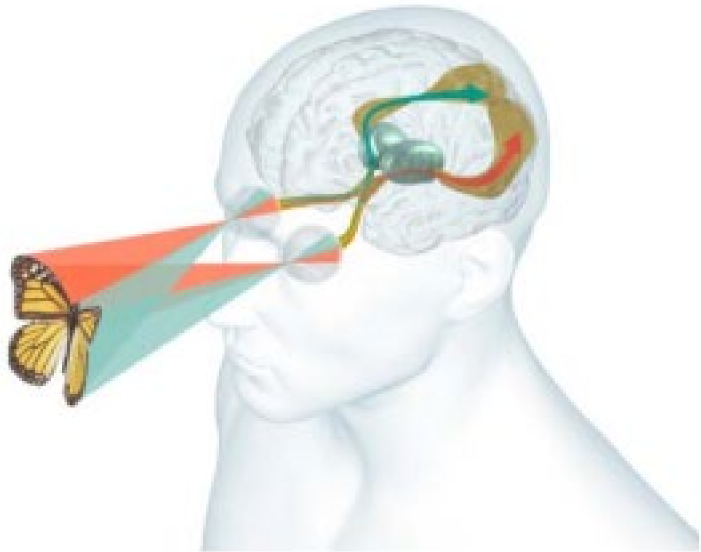
Do you ever notice how your brain can figure things out by itself? All it takes is to step away from the computer and take a break to think about something totally unrelated.

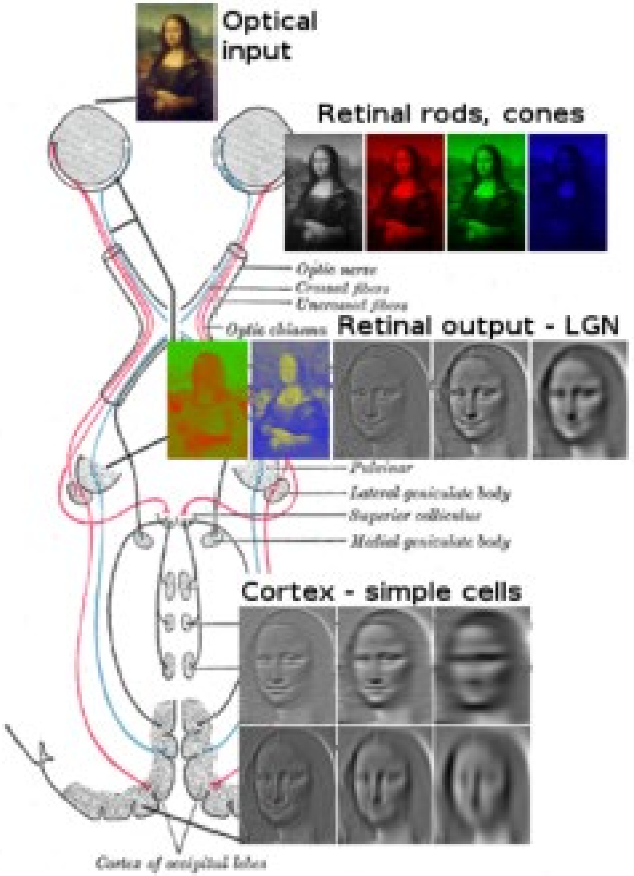
Traditional ML and Modern DL

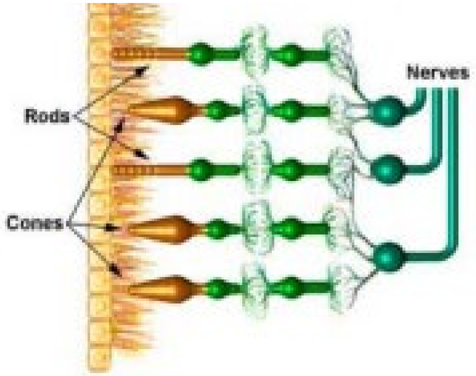


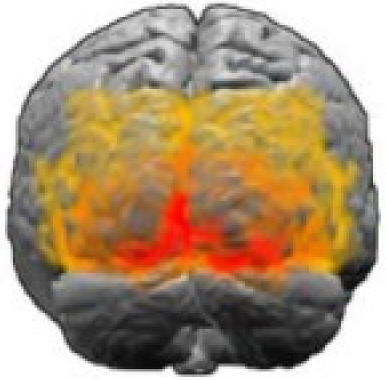
From Eyes to Brain

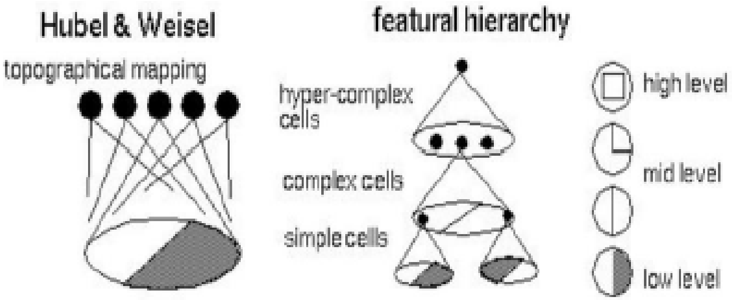




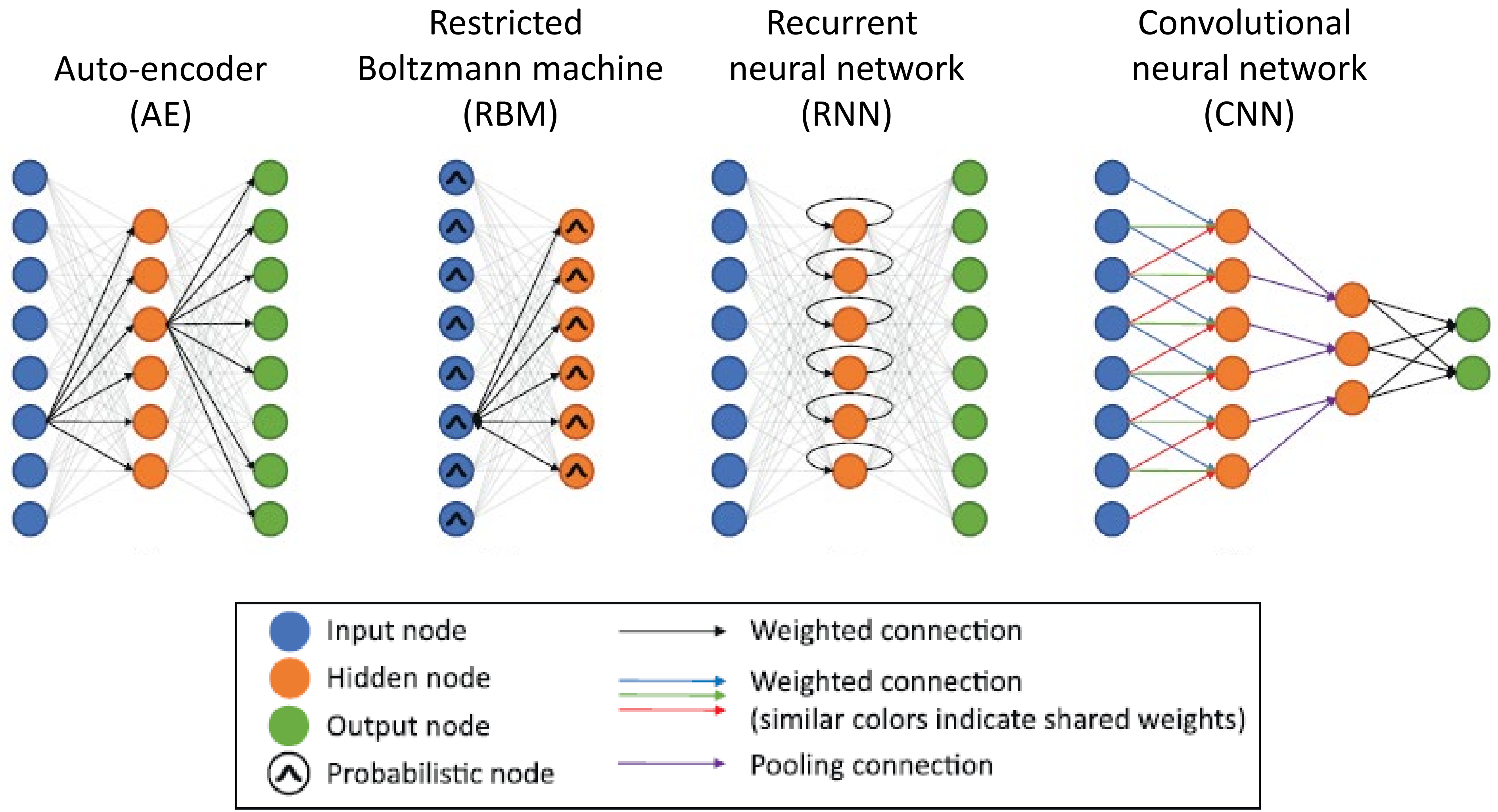






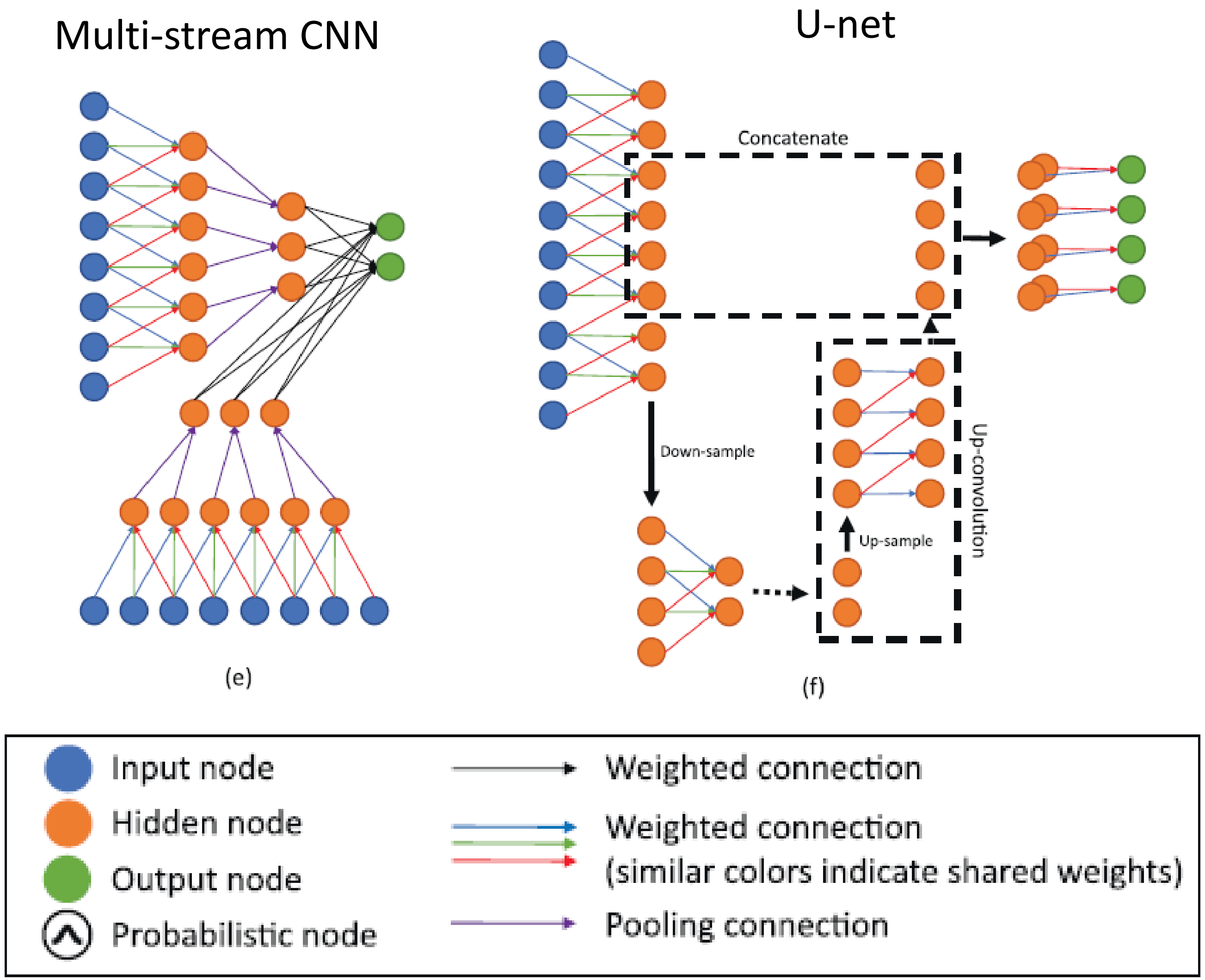


Frequent Used Model Structures



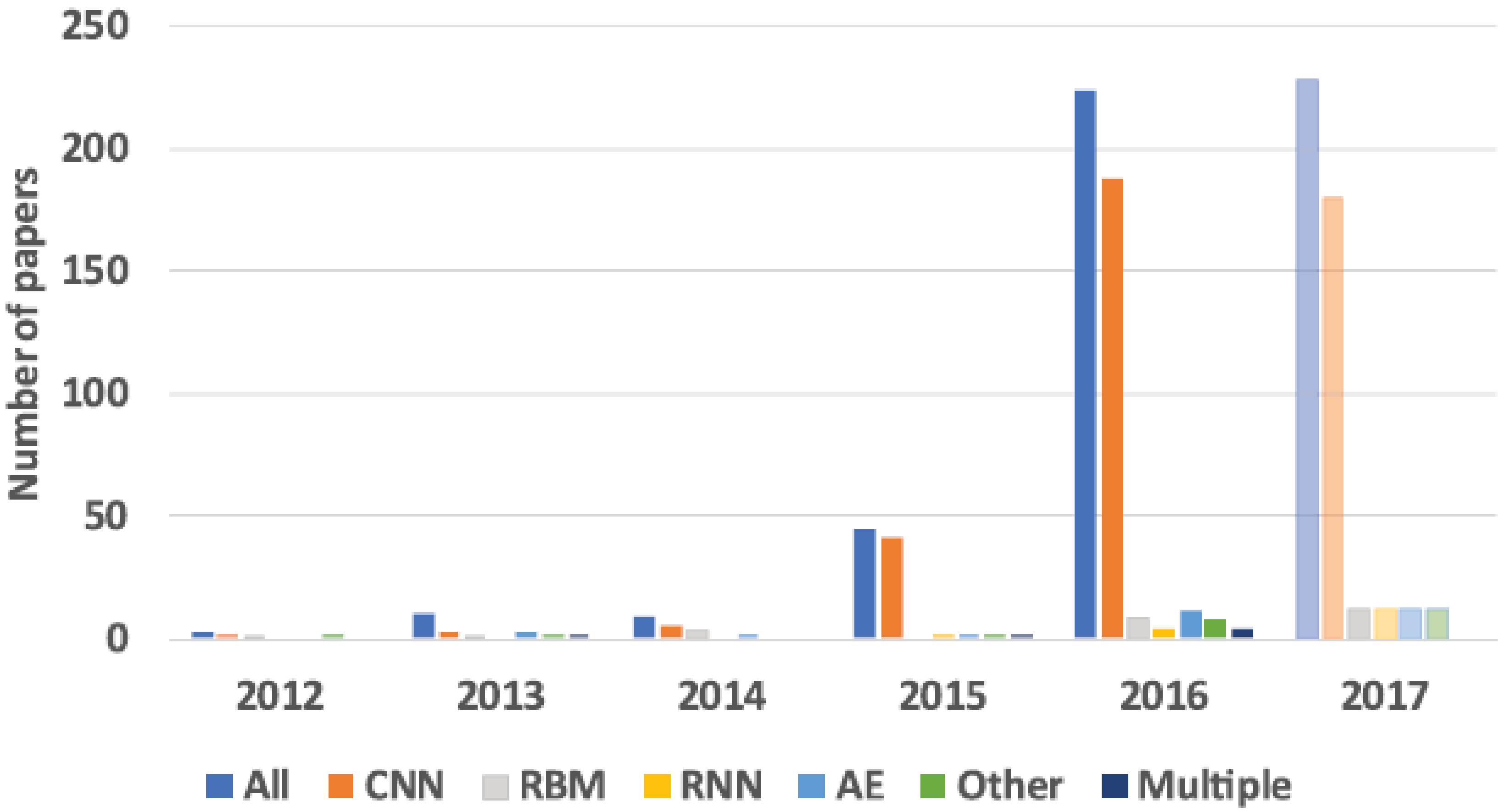
Litjens st al (2017) Med Image Anal 42. 60-88

Frequent Used Model Structures



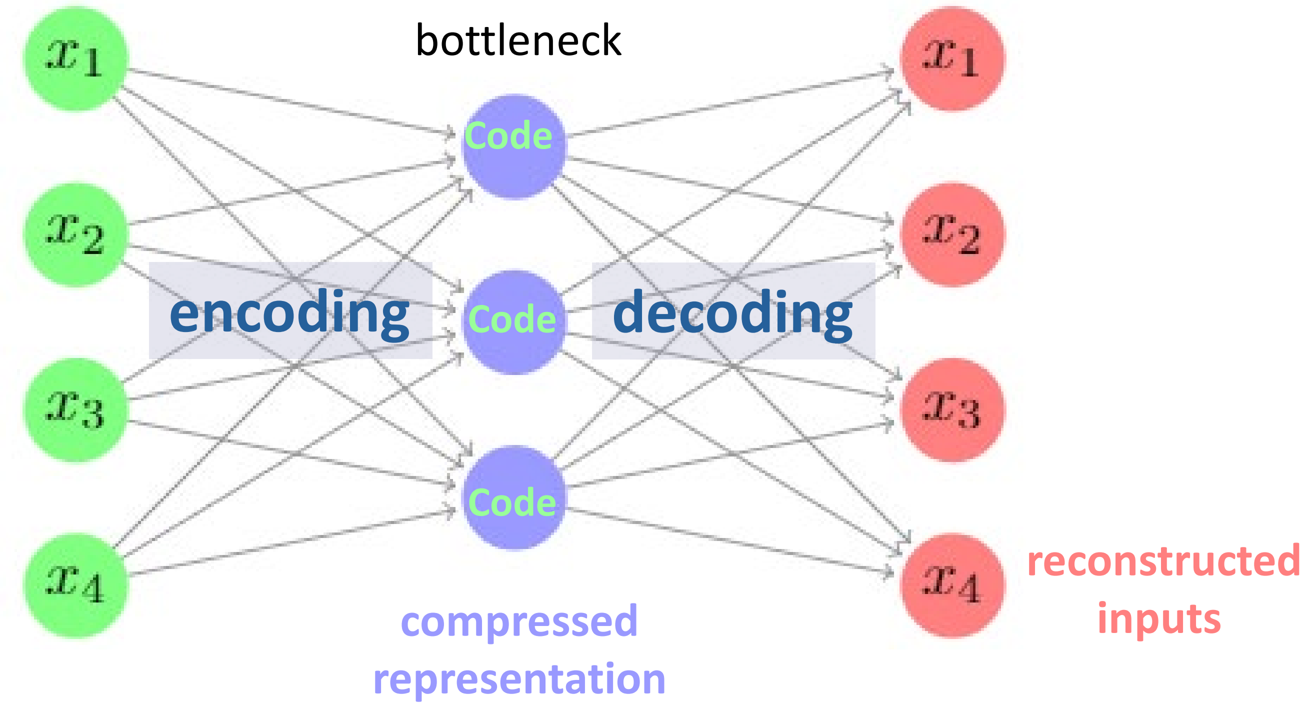
Litjens st al (2017) Med Image Anal 42. 60-88

Frequent Used Models



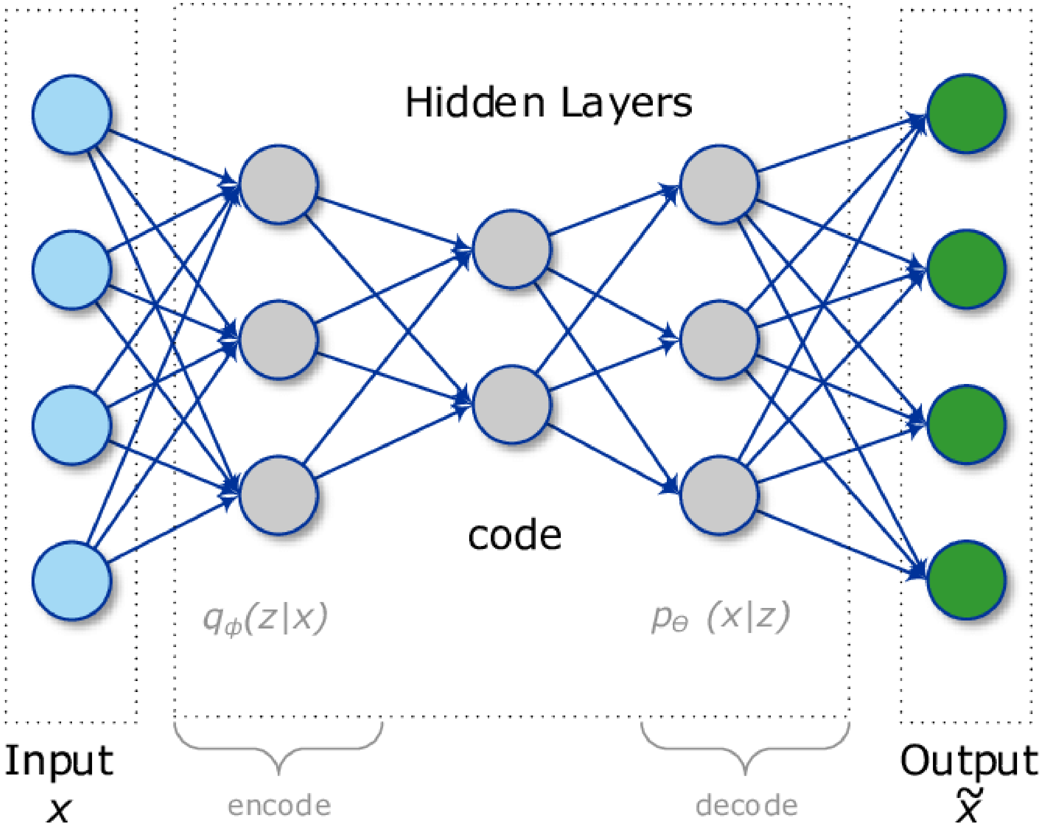
Litjens st al (2017) Med Image Anal 42. 60-88

Model Structure : Auto-encoder (AE)



- For feature extraction and dimension reduction
- # of input = # of output
- Pro: unsupervised learning   Con : needs pre-training set

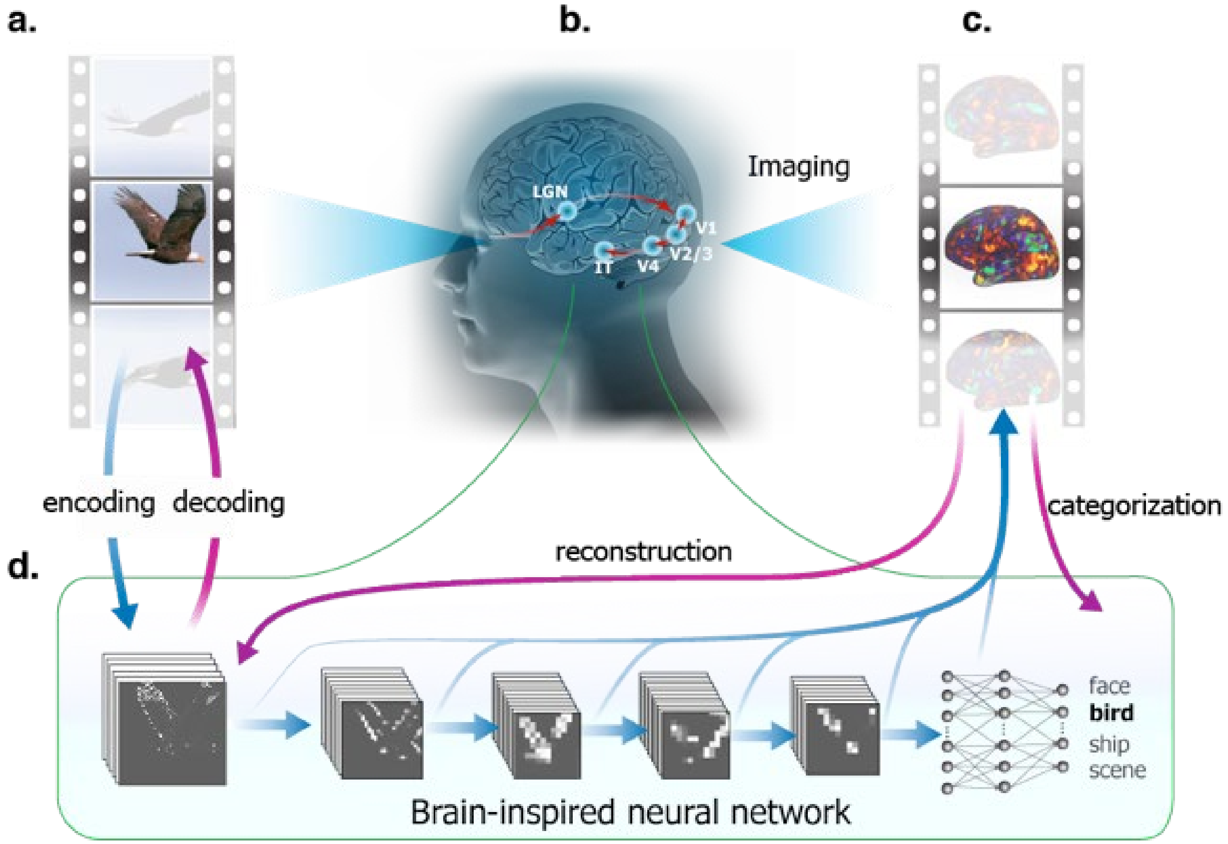
Model Structure : Auto-encoder (AE)



Sparse AE

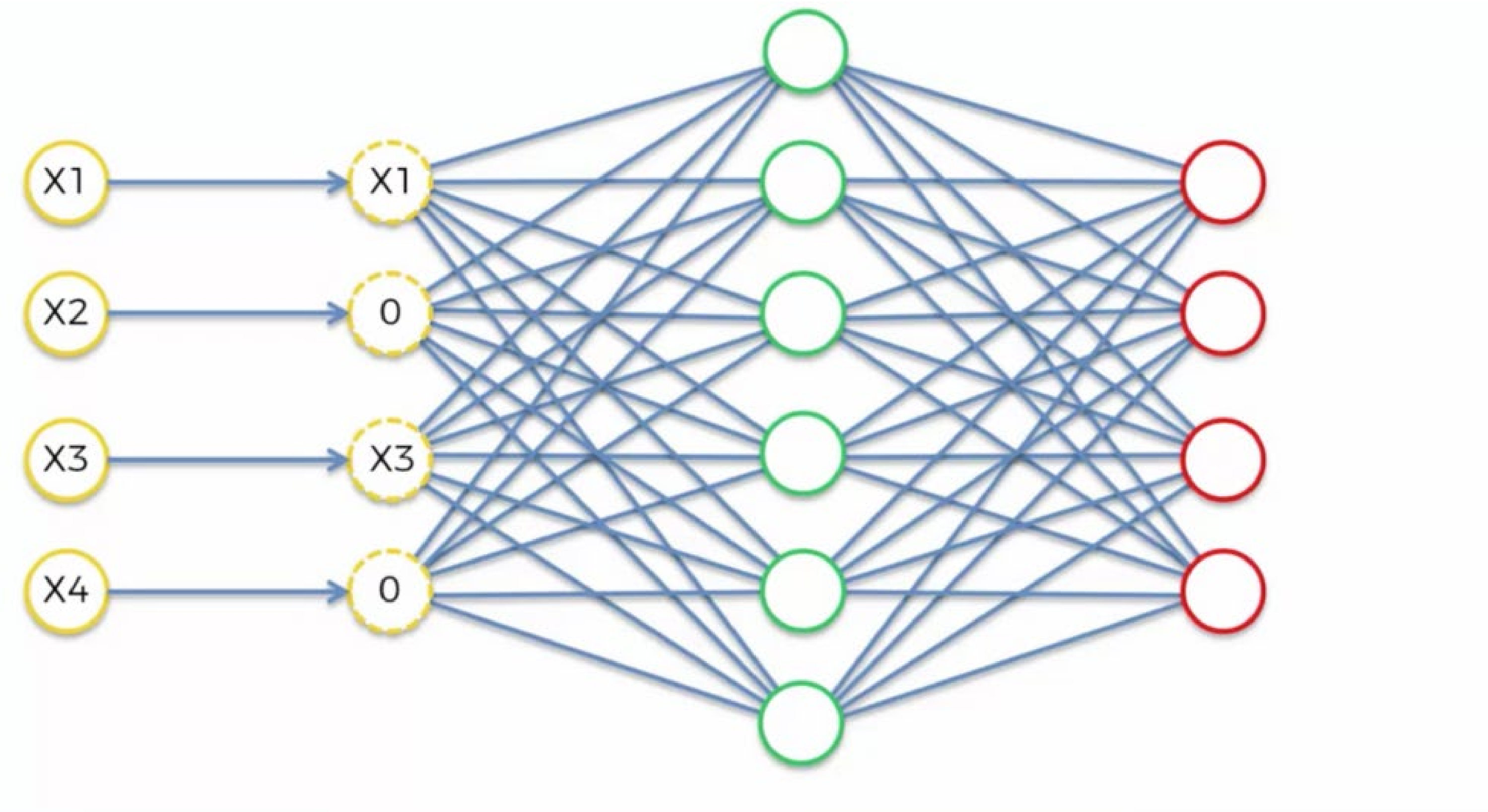
Model Structure : Auto-encoder (AE)

For example: Modeling and Decoding fMRI Activity in Visual Cortex



Han et al 2017 bioRxiv 10.1101/214247

Model Structure : Auto-encoder (AE)

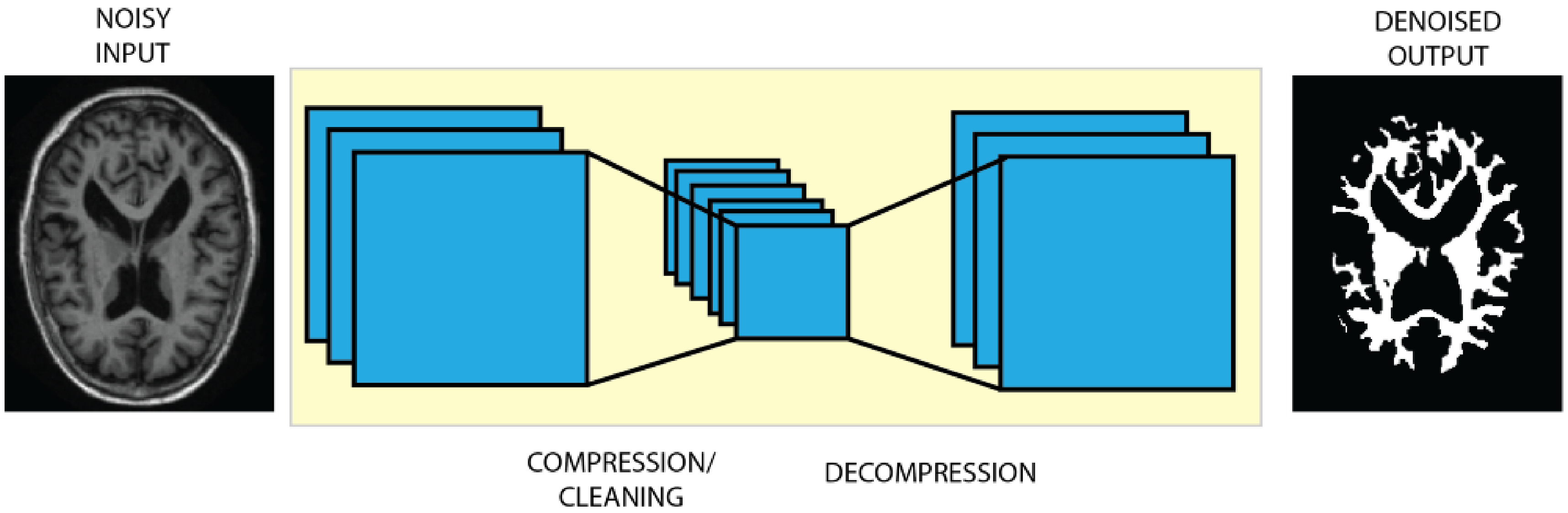


De-nosing AE

To avoid null function.

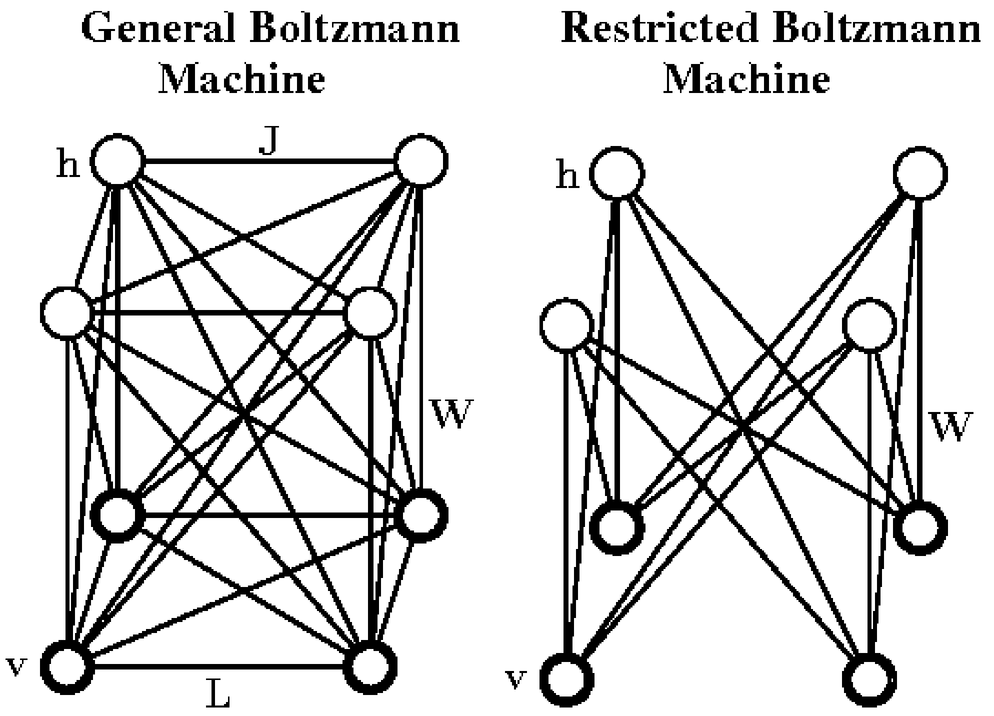
Model Structure : Auto-encoder (AE)

For example: Brain MRI image segmentation using Stacked Denoising Autoencoders



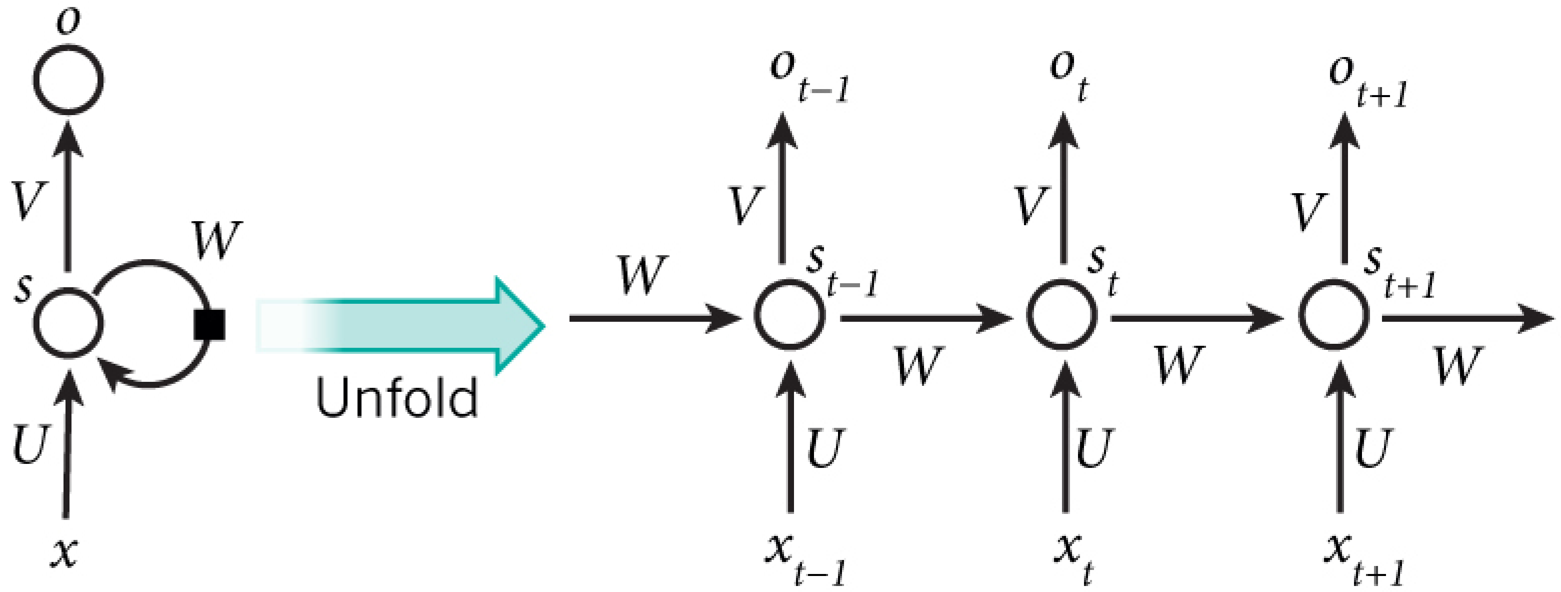
https://blog.insightdatascience.com/brain-mri-image-segmentation-using-stacked-denoising-autoencoders-4e91417688f6

Model Structure :Boltzmann machine (RBM)



- Undirectional connections between all hidden layers
- Pro: for robust inference, top-down feed-back incorporates
  with ambiguous data
  Con : unable for optimization for big dataset

Model Structure :Recurrent Neural Network (RNN)

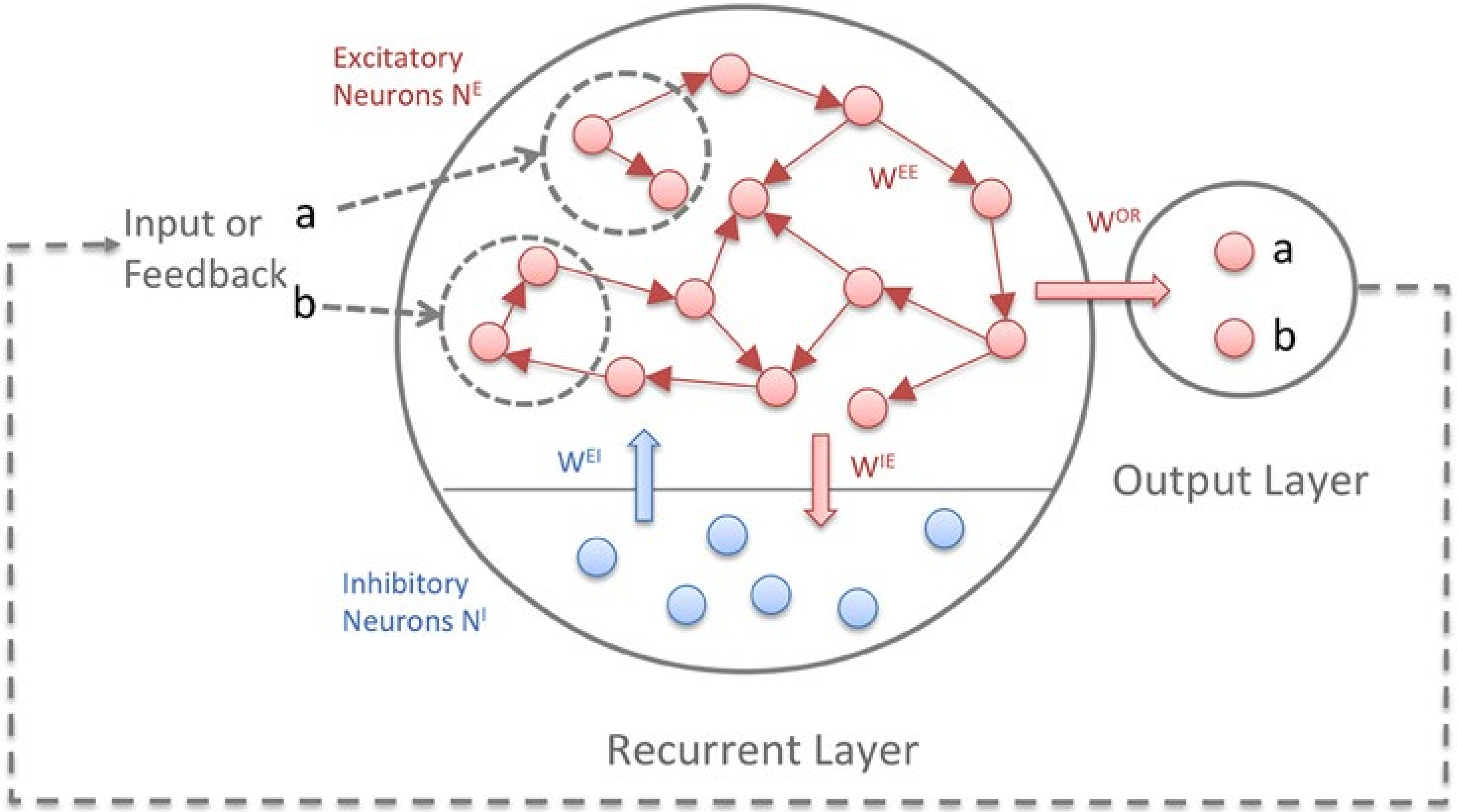


- Learning sequence. Weights are sharing across steps /neurons.
- Pro:allow time dependencies modeling
  Con : gradient vanishing / need big dataset

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

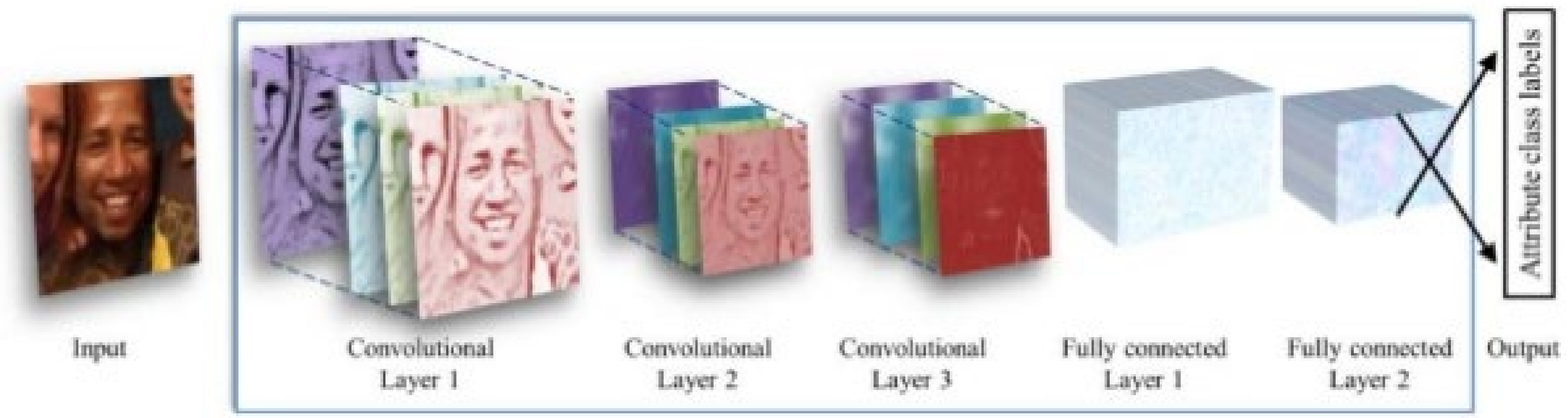
Model Structure : Recurrent Neural Network (RNN)

For example:A reward-modulated self-organizing recurrent neural network



Aswolinskiy & Pipa Frontiers in Computational Neuroscience 9(36):36 · March 2015

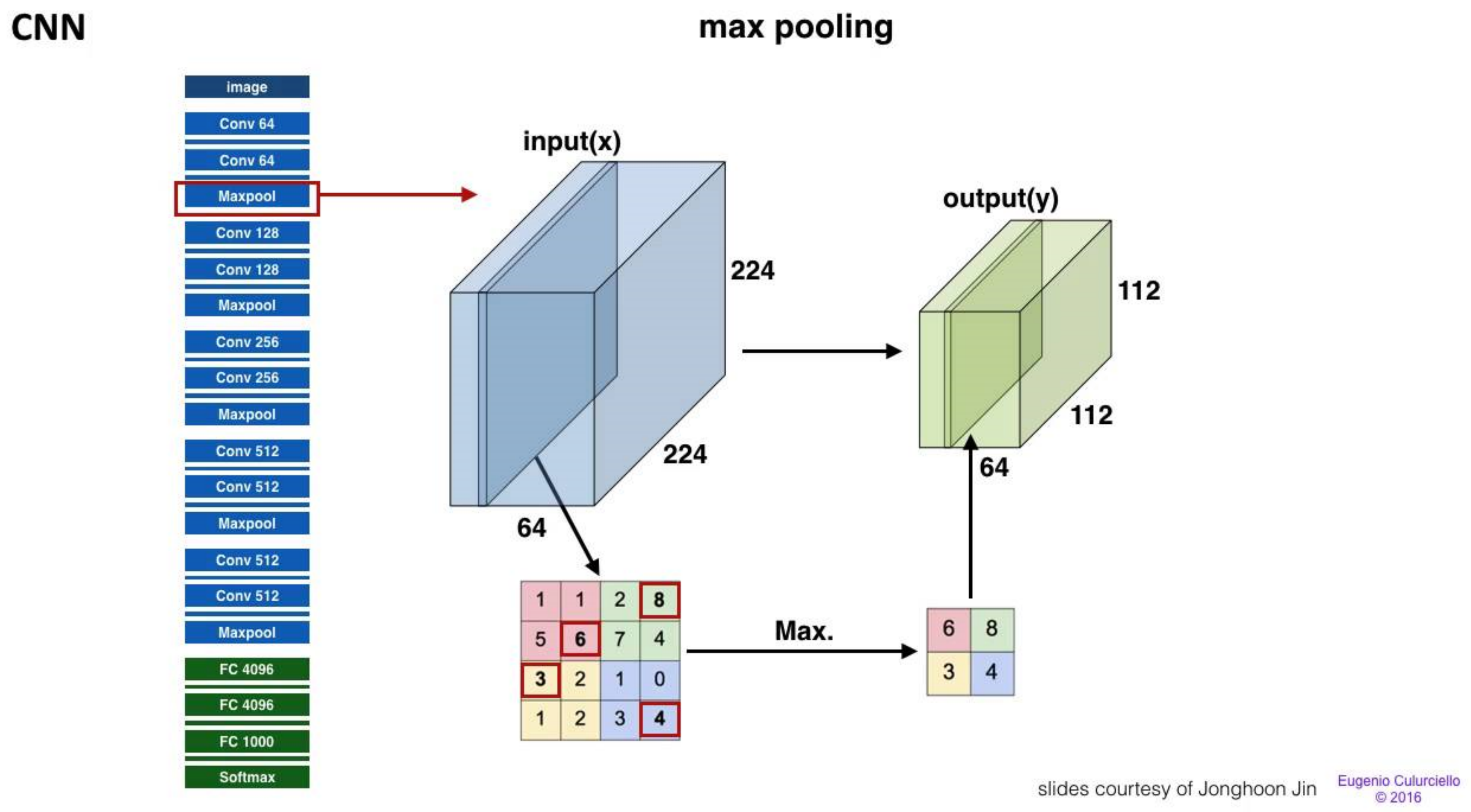
Model Structure : Convolutional Neural Network (CNN)



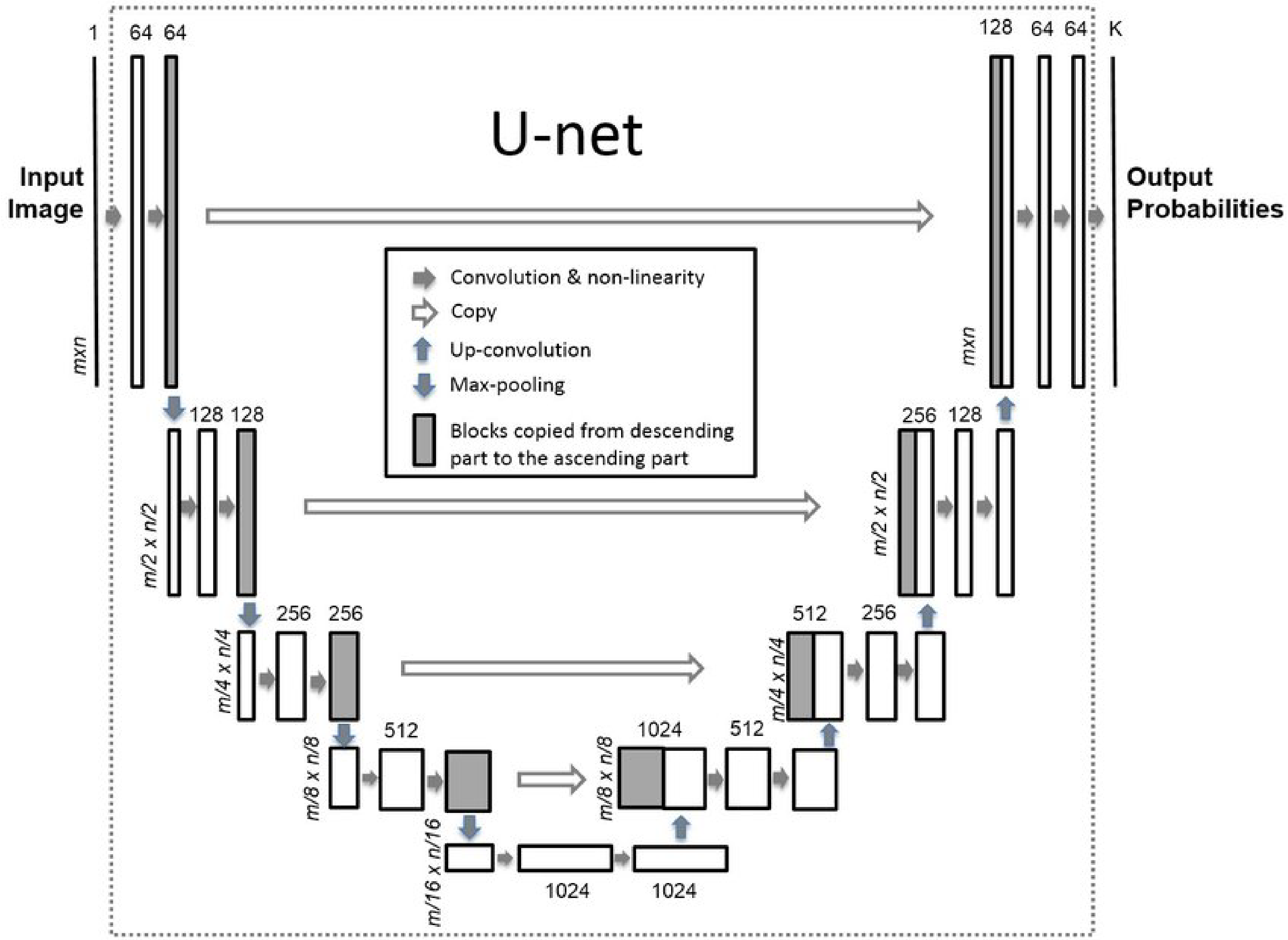
- Pro: fast, good with images analysis
  Con : big data required

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

Model Structure : Convolutional Neural Network (CNN)

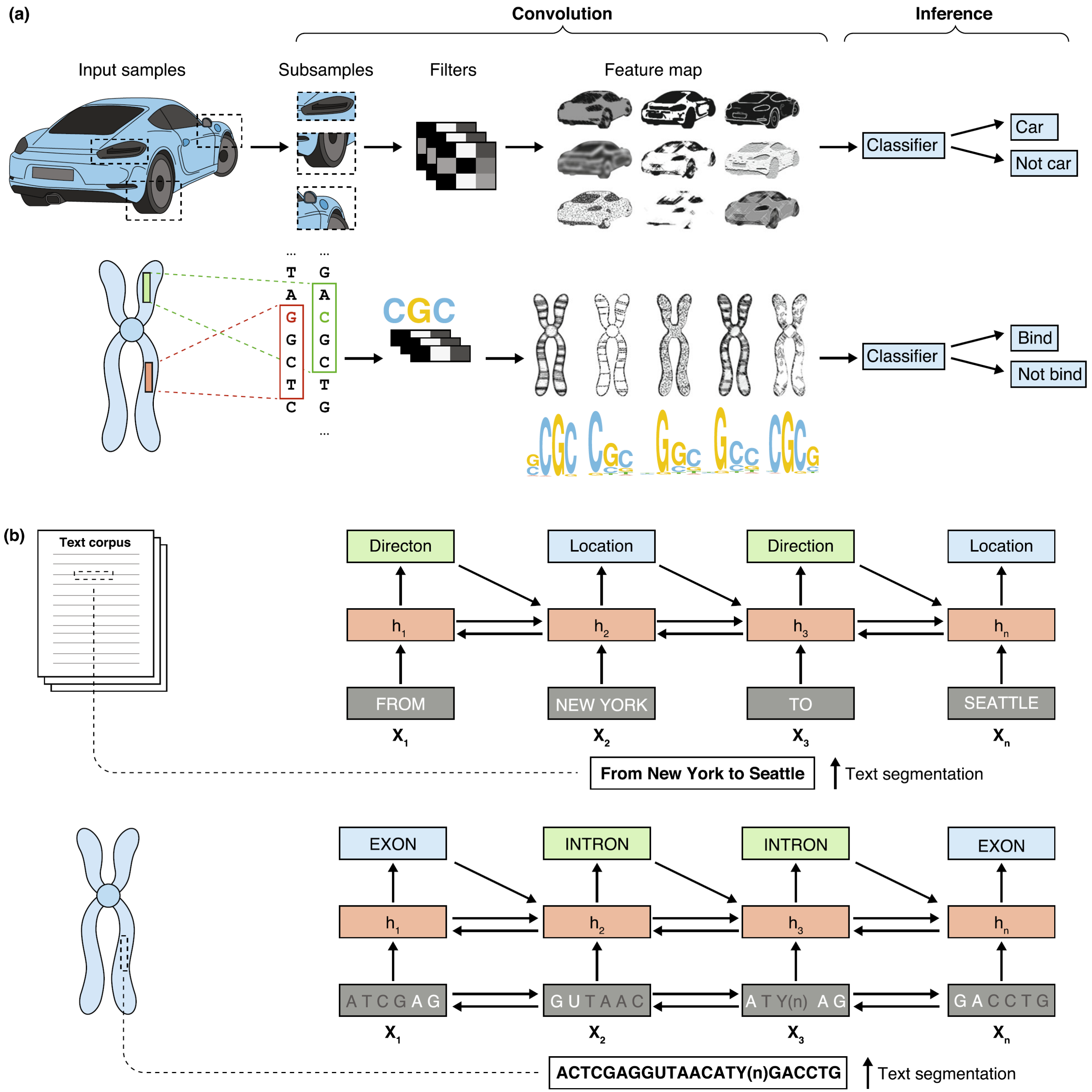


Model Structure : U-Net



Examples of different NN architectures, their typical workflow, and applications in genomics

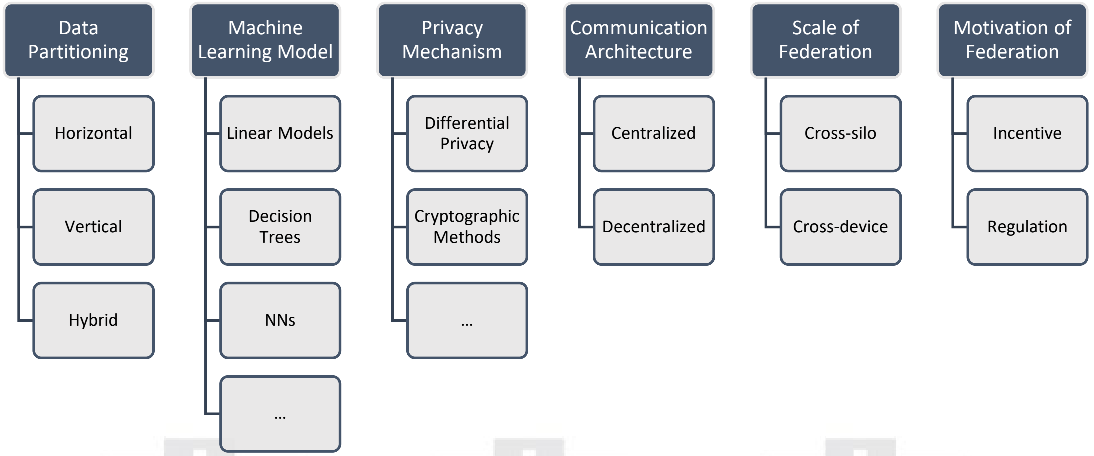
AI software and hardware, especially deep learning algorithms and the graphics processing units (GPUs) that power their training, have led to a recent and rapidly increasing interest in medical AI applications.

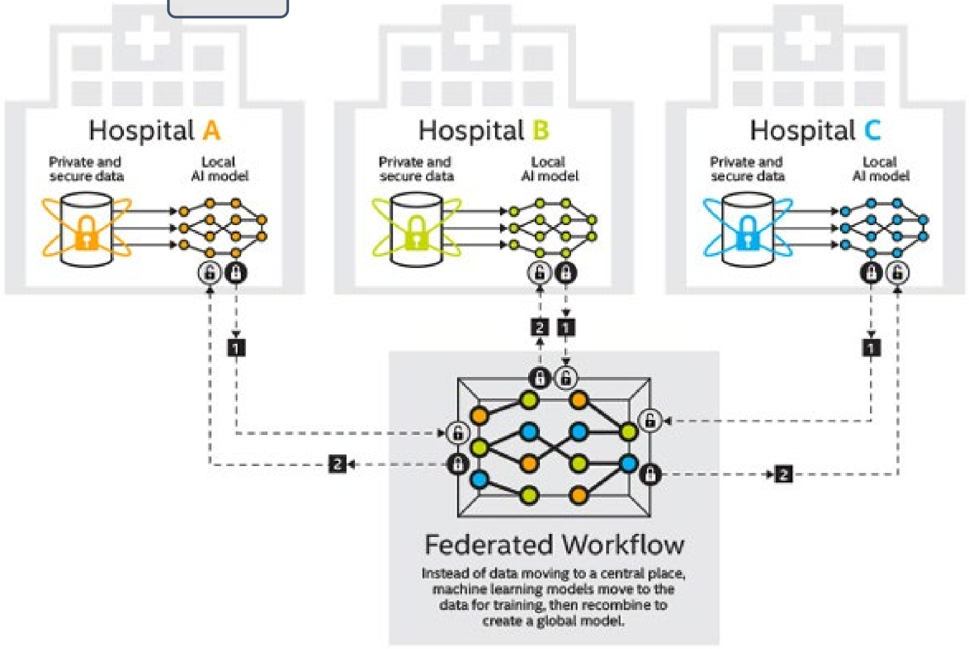


Dias, R., & Torkamani, A. (2019). Artificial intelligence in clinical and genomic diagnostics. Genome medicine, 11(1), 1-12.

Data/Skill United is Research/Nation Strength

Federated Learning System



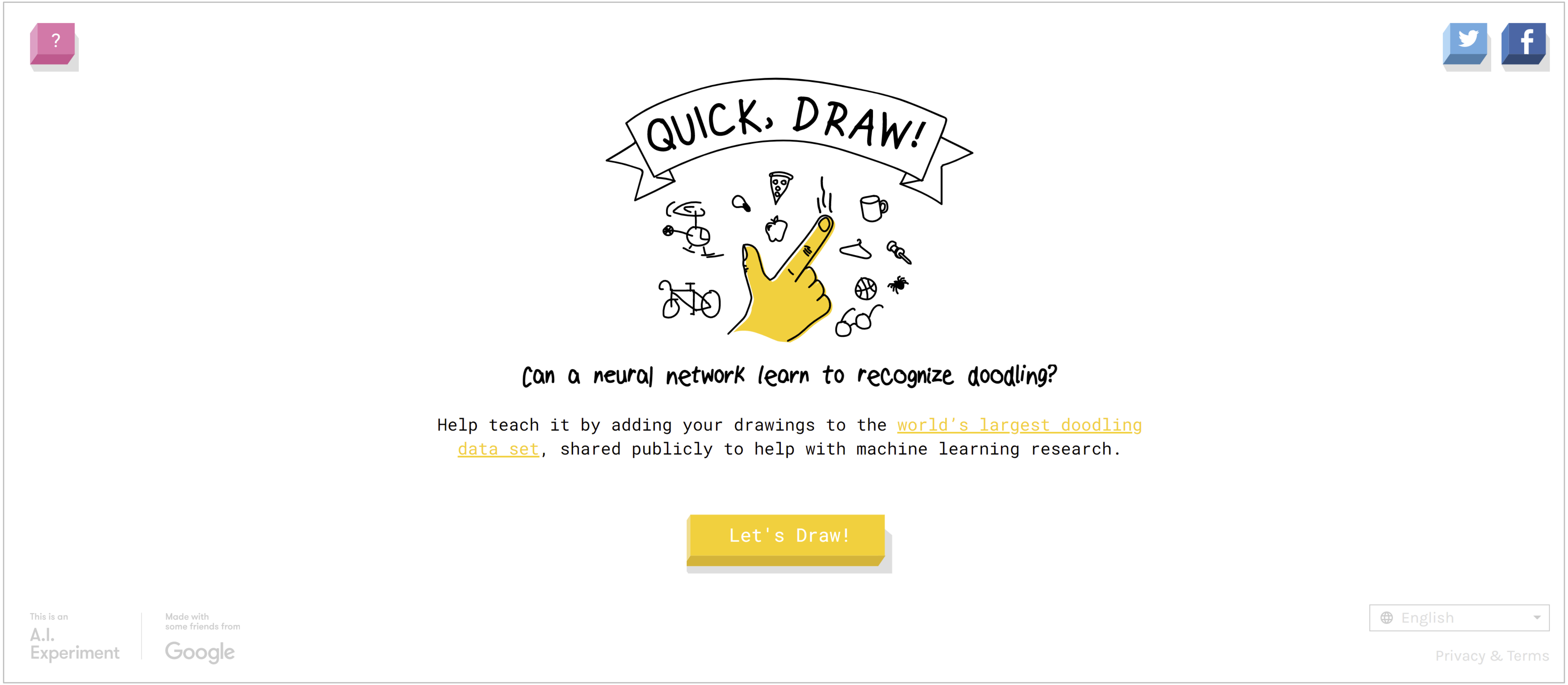


Quantum Computing

Source: Intel (https://www.edge-ai-vision.com/2020/06/intel-works-with-university-of-pennsylvania-in-using-privacy-preserving-ai-to-identify-brain-tumors/)

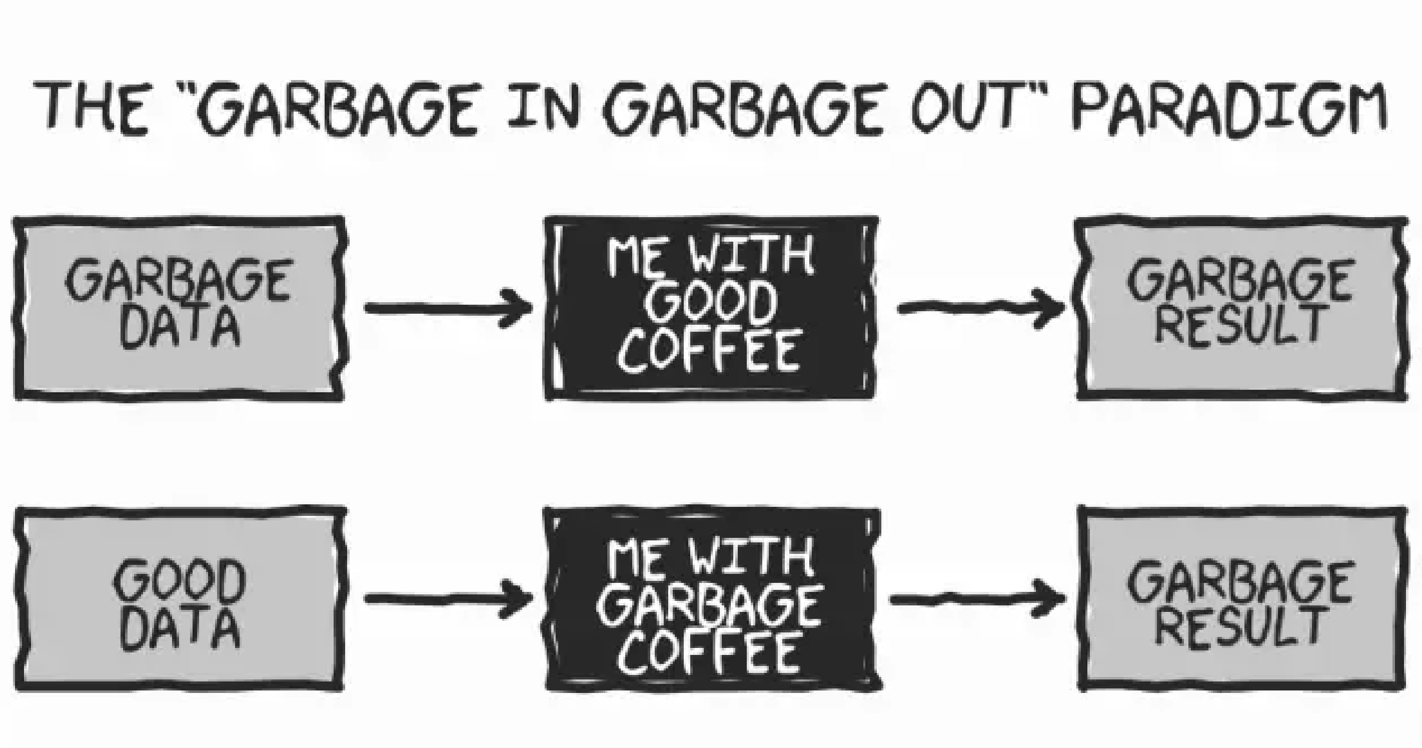
A fun example :

https://quickdraw.withgoogle.com/?locale=en_US



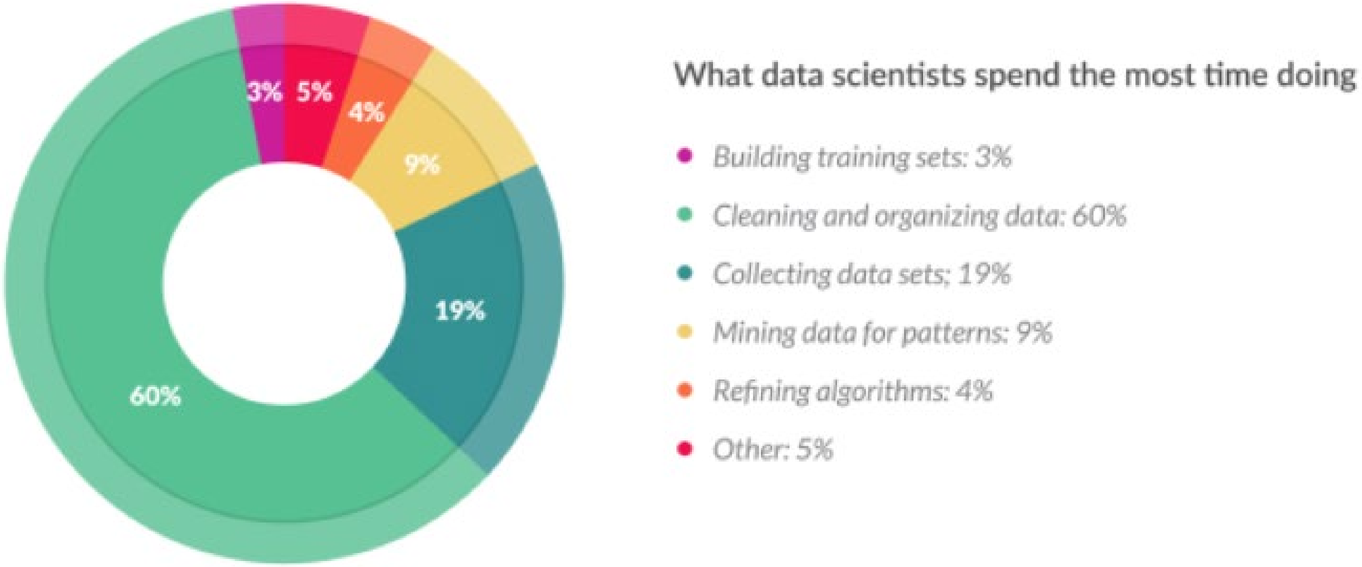
Main Challenges of Machine Learning

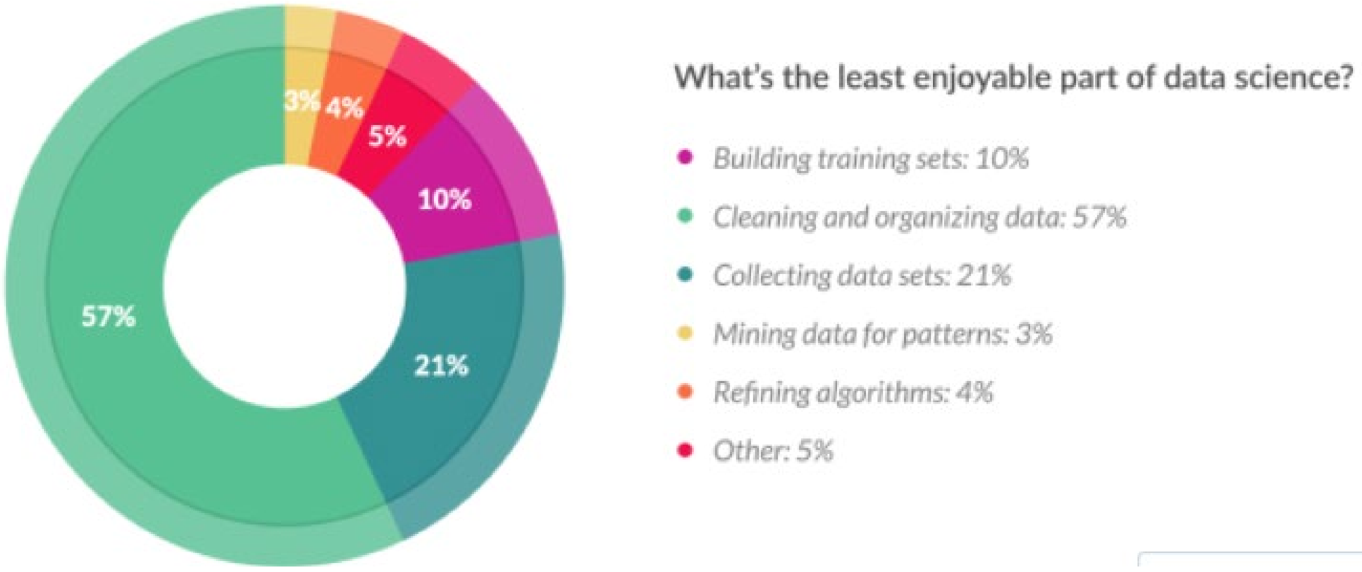
- Since our main task is to select a learning algorithm and train it on some data, the two things that can go wrong are "bad algorithm" and "bad data."



Data

- **Size**
- **Representativeness**
- **Quality**





https://whatsthebigdata.com/2016/05/01/data-scientists-spend-most-of-their-time-cleaning-data/

Bad Data: Insufficient Quantity of Training Data

- **Data** matters more than algorithms for complex problems
- Very different Machine Learning algorithms, including fairly simple ones, performed almost identically well on a complex problem of natural language disambiguation once they were given enough data.

Non-representative Training Data

- Sampling noise, sampling bias.

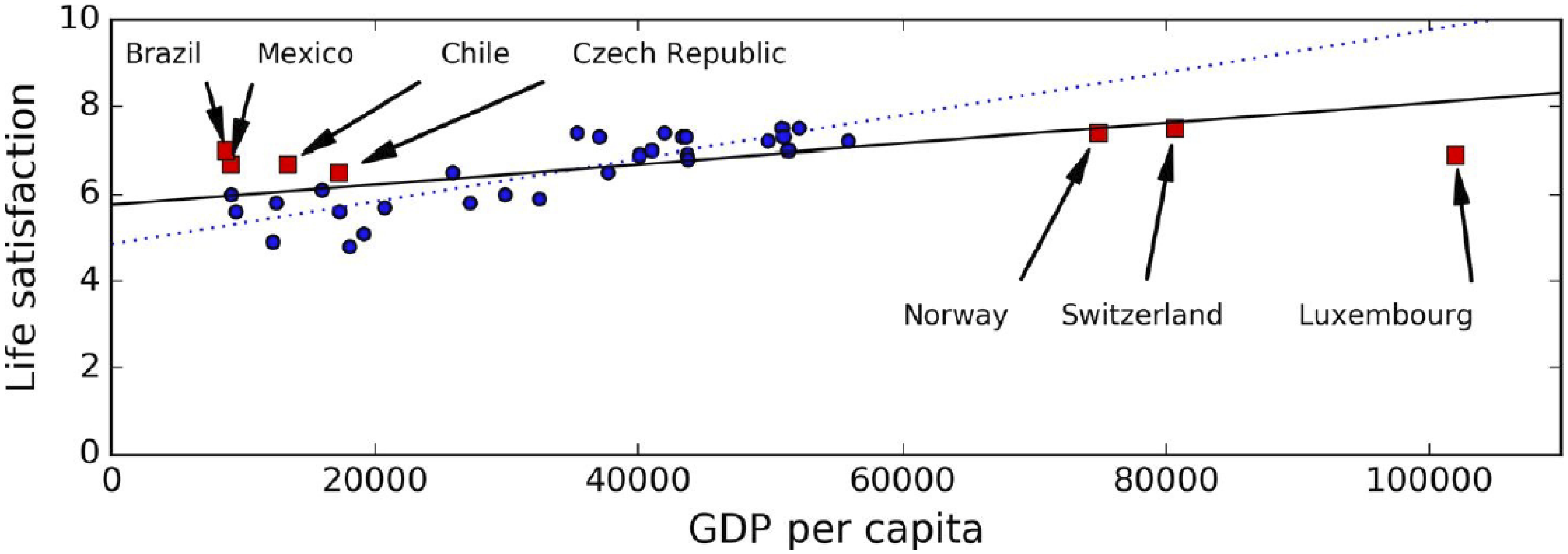


Figure 1-21. A more representative training sample

It seems that very rich countries are not happier than moderately rich countries (in fact they seem unhappier), and conversely some poor countries seem happier than many rich countries.

Poor Quality Data

- There is no substitute for good data.
- Cleaning data is very important!
- Outlier detection
- Detection of missing data

Irrelevant Features

- A critical part of the success of a Machine Learning project is coming up with a **good set of features** to train on.

Feature engineering

- Feature selection: selecting the most useful features to train on among existing features.
- Feature extraction: combining existing features to produce a more useful one (as we saw earlier, dimensionality reduction algorithms can help).
- **Creating new features** by gathering new data.

Bad Algorithms Overfitting the Training Data

- **Overfitting** happens when the model is too complex relative to the amount and noisiness of the training data
- Example: a high-degree polynomial model

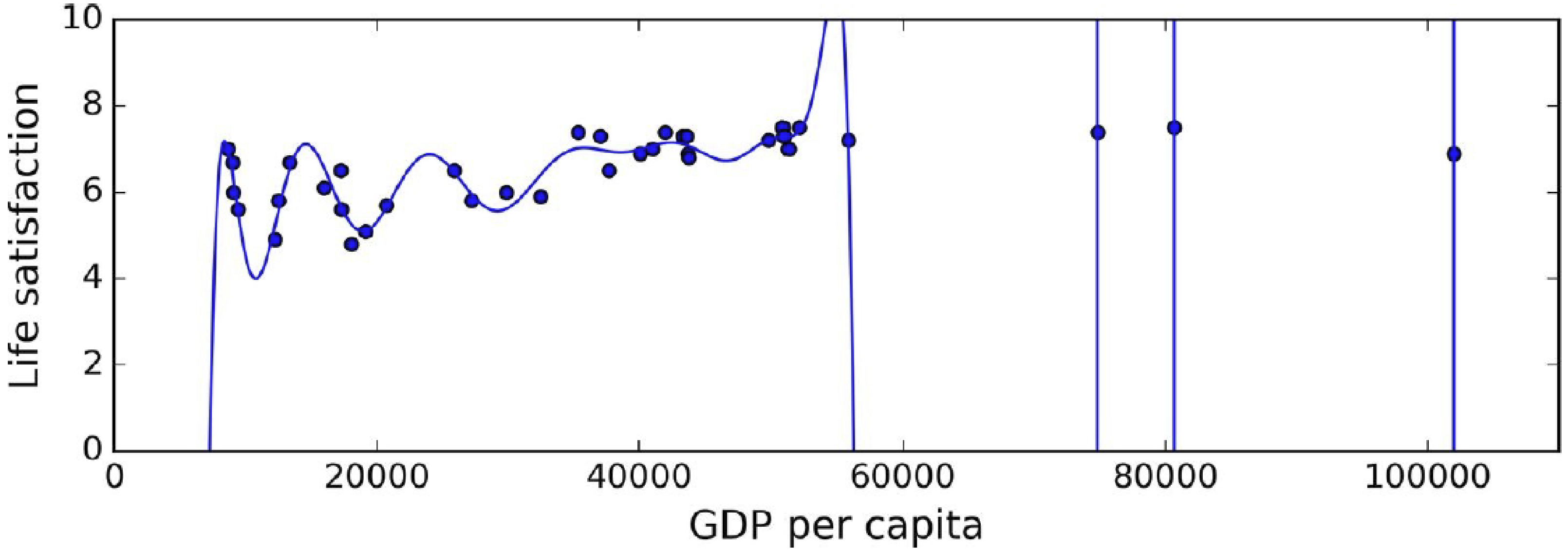
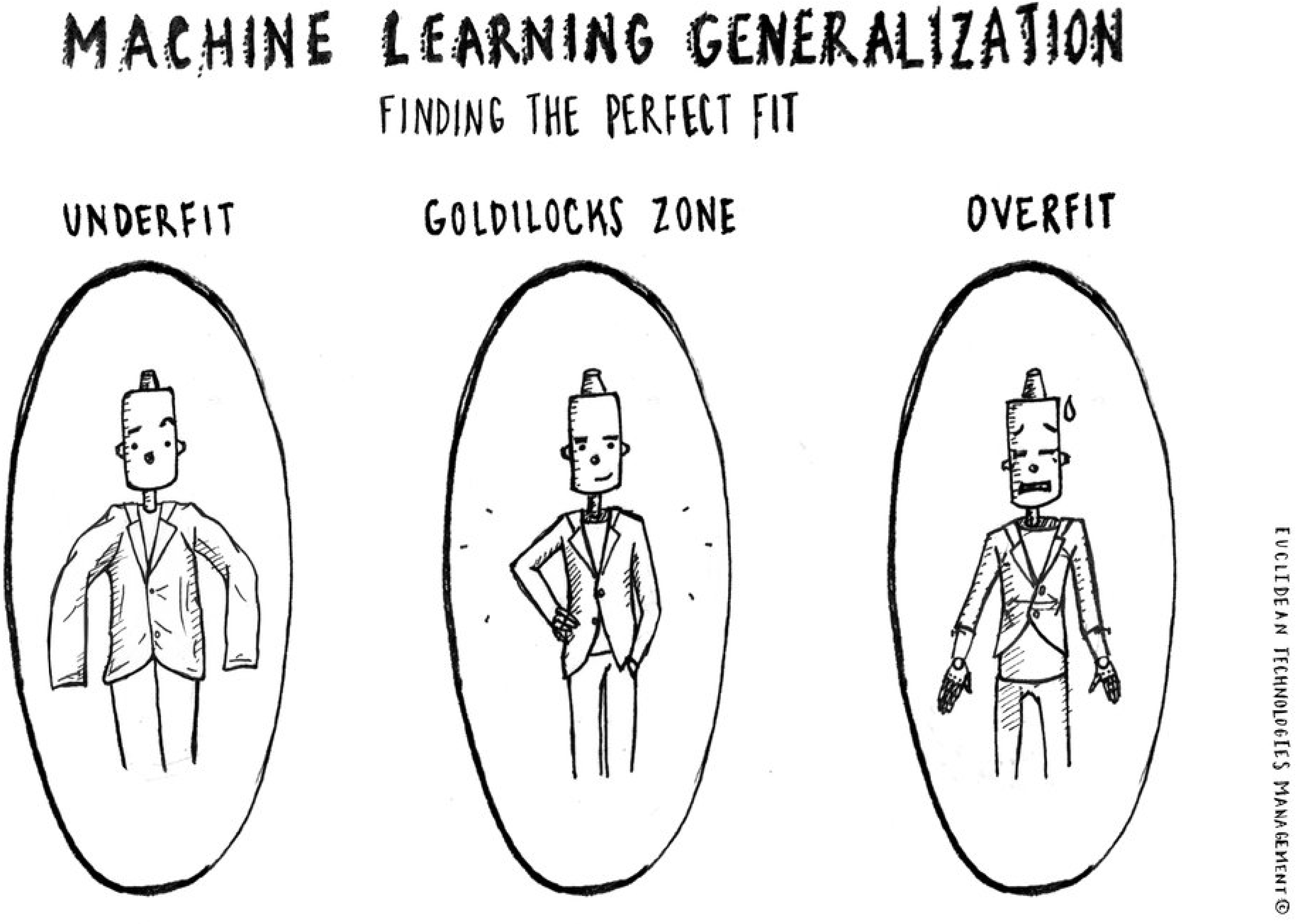


Figure 1-22. Overfitting the training data

- **C**
  a
- E


MACHINE LEARNING GENERALIZATION
FINDING THE PERFECT FIT
UNDERFIT
GOLDILOCKS ZONE
OVERFIT
EUCLIDEAN TECHNOLOGIES MANAGEMENT ©

Overfitting the Training Data

- Complex models such as deep neural networks can detect subtle patterns in the data, but if the training set is noisy, or if it is too small (which introduces sampling noise), then the model is likely to detect patterns in the noise itself.

Possible solutions:

- To simplify the model by selecting one with fewer parameters or by regularization (balancing between fitting the data perfectly and keeping the model simple)
- To gather more training data
- To reduce the noise in the training data

Underfitting the Training Data

- It occurs when your model is too simple to learn the underlying structure of the data.

Possible solutions:

- Selecting a more powerful model, with more parameters
- Feeding better features to the learning algorithm (feature engineering)
- Reducing the constraints on the model (e.g., reducing the regularization hyperparameter)

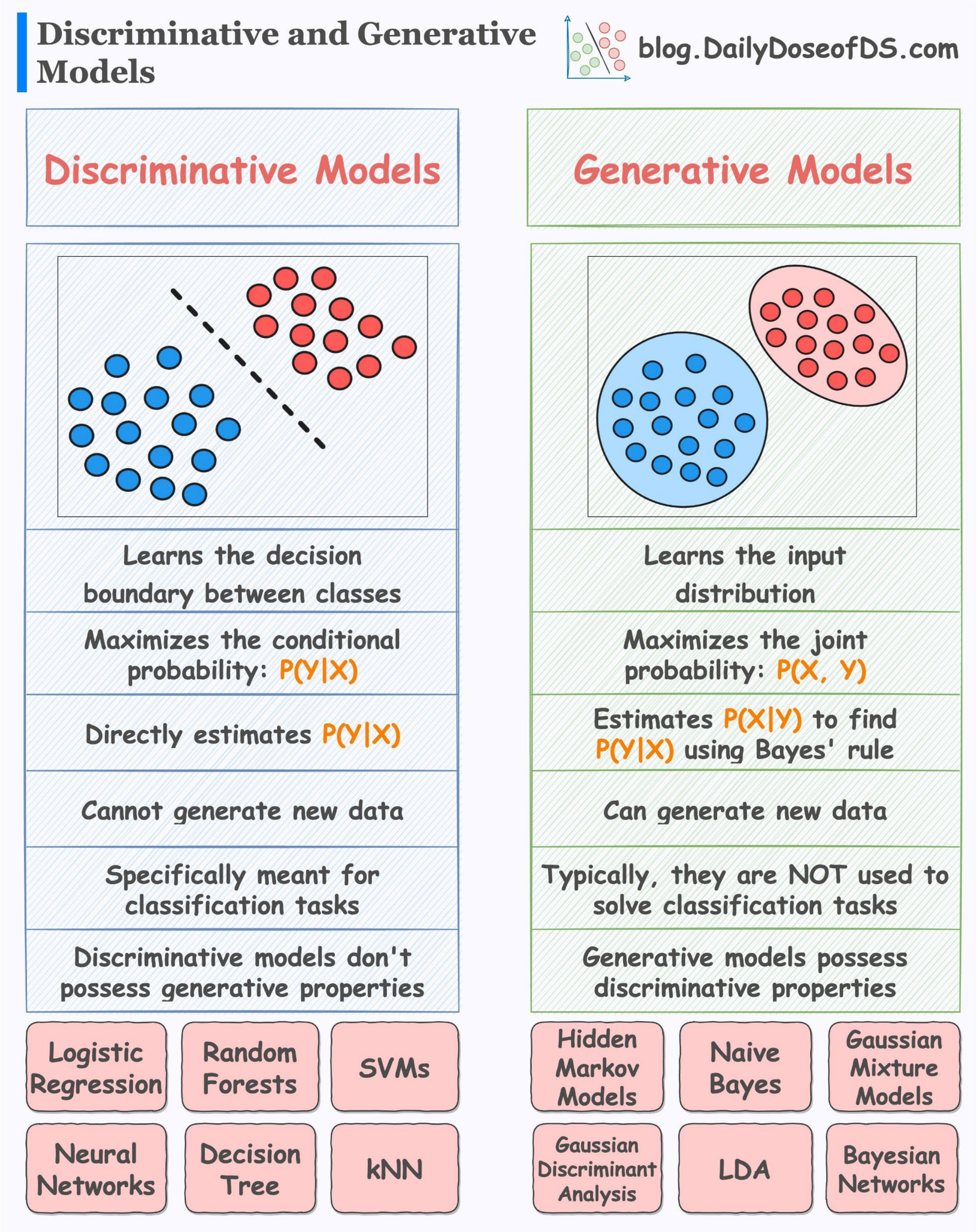
Testing and Validating

- The only way to know how well a model will generalize to new cases is to actually try it out on new cases.
- A better option is to split your data into two sets: the training set and the test set.
  - It is common to use 80% of the data for training and hold out 20% for testing.
- N-fold **cross-validation**: examining model parameters using training dataset (+ validation set)
  - For model selection

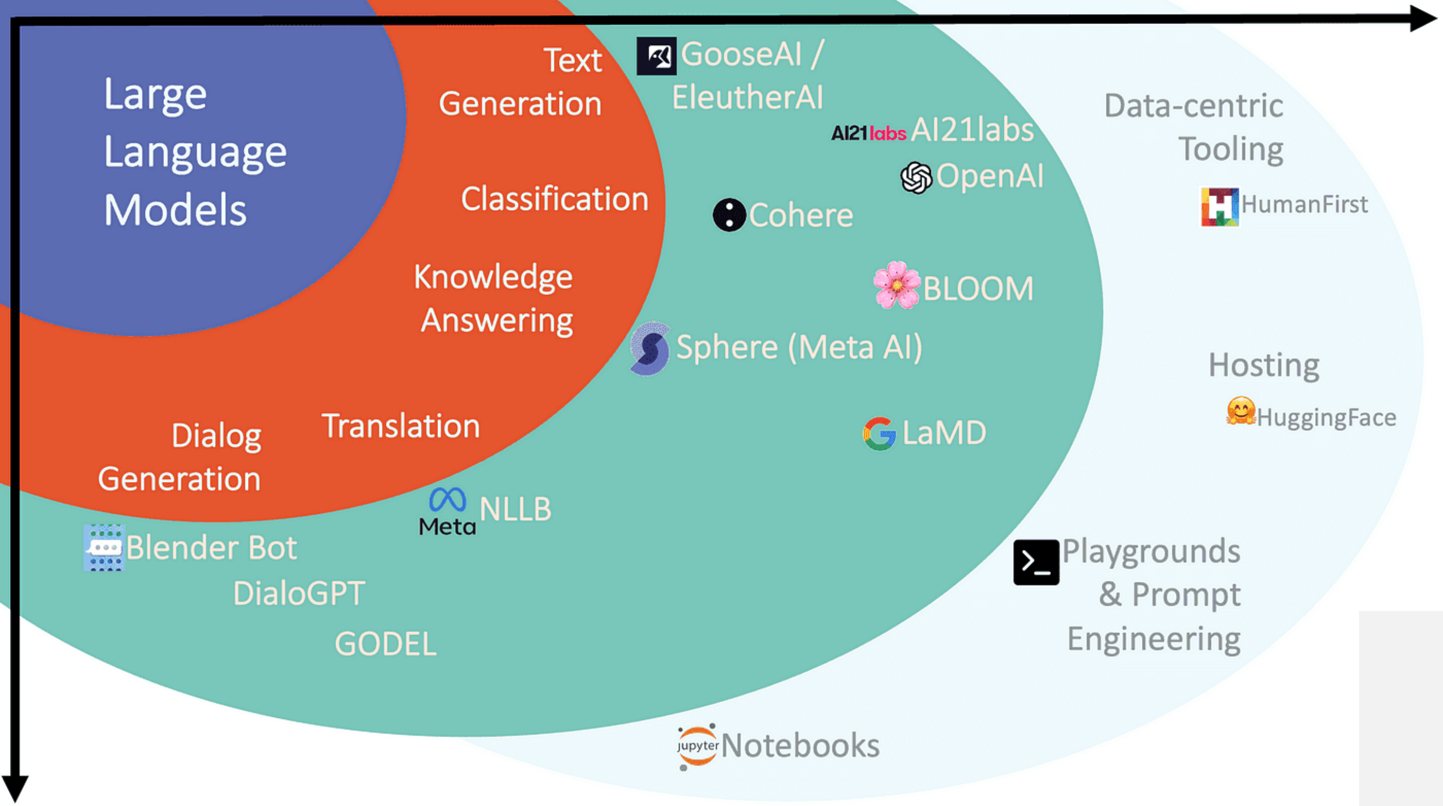
Recap

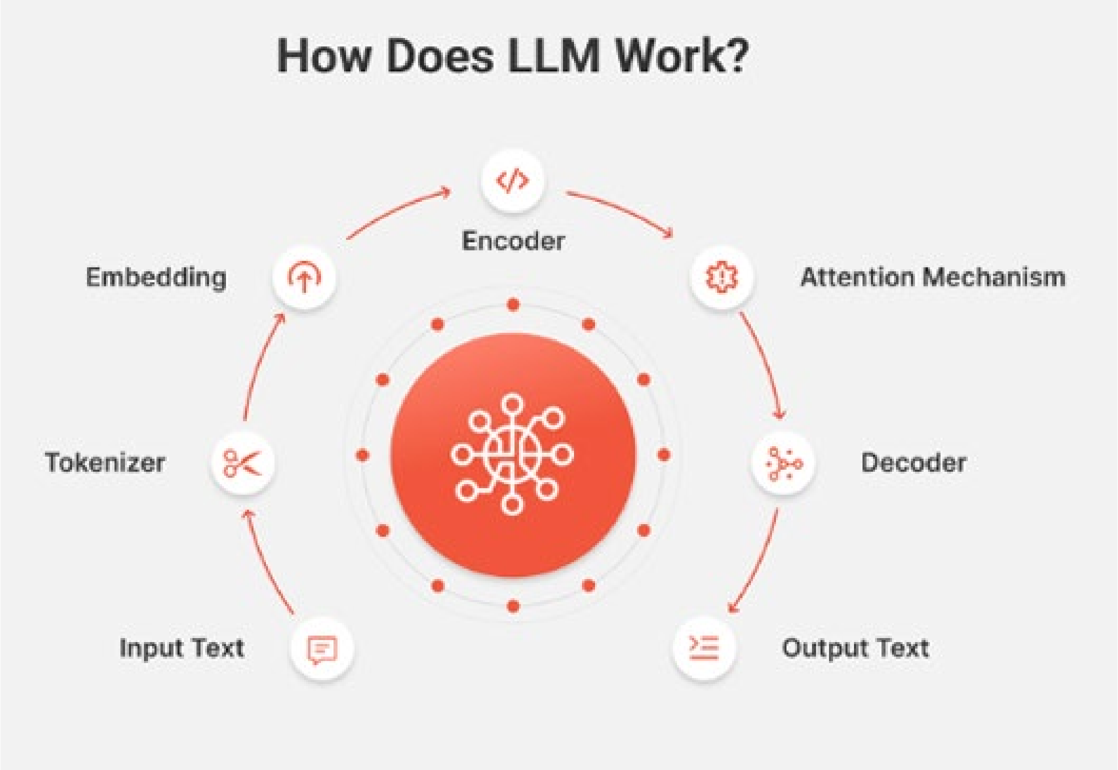
- Machine Learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules.
- There are many different types of ML systems: supervised or not, batch or online, instance-based or model-based, and so on.
- In a ML project you gather data in a training set, and you feed the training set to a learning algorithm.
- The system will not perform well
  - if your training set is too small, or
  - if the data is not representative, noisy, or polluted with irrelevant features (garbage in, garbage out).
  - Lastly, your model needs to be neither too simple (in which case it will underfit) nor too complex (in which case it will overfit).

Generative and Discriminative Models in Machine Learning

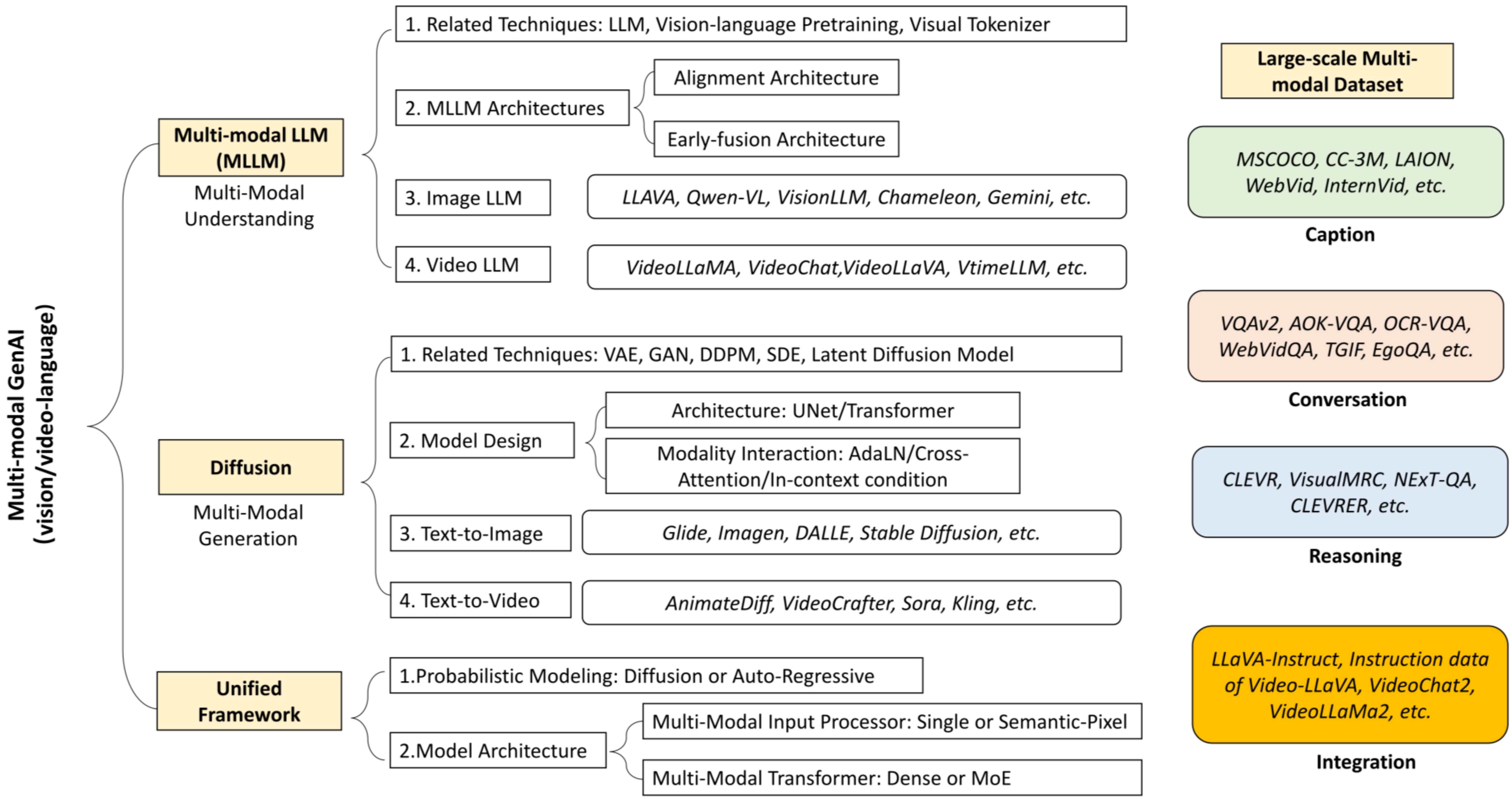


https://blog.dailydoseofds.com/p/an-intuitive-guide-to-generative





https://www.deepchecks.com/glossary/llm-as-a-service/
https://www.teneo.ai/blog/understanding-large-language-models-llms



Multi-modal GenAI (vision/video-language)
Multi-modal LLM (MLLM)
Multi-Modal Understanding
1. Related Techniques: LLM, Vision-language Pretraining, Visual Tokenizer
2. MLLM Architectures
Alignment Architecture
Early-fusion Architecture
3. Image LLM
LLAVA, Qwen-VL, VisionLLM, Chameleon, Gemini, etc.
4. Video LLM
VideoLLaMA, VideoChat,VideoLLaVA, VtimeLLM, etc.
Diffusion
Multi-Modal Generation
1. Related Techniques: VAE, GAN, DDPM, SDE, Latent Diffusion Model
2. Model Design
Architecture: UNet/Transformer
Modality Interaction: AdaLN/Cross-Attention/In-context condition
3. Text-to-Image
Glide, Imagen, DALLE, Stable Diffusion, etc.
4. Text-to-Video
AnimateDiff, VideoCrafter, Sora, Kling, etc.
Unified Framework
1.Probabilistic Modeling: Diffusion or Auto-Regressive
2.Model Architecture
Multi-Modal Input Processor: Single or Semantic-Pixel
Multi-Modal Transformer: Dense or MoE
Large-scale Multi-modal Dataset
MSCOCO, CC-3M, LAION, WebVid, InternVid, etc.
Caption
VQAv2, AOK-VQA, OCR-VQA, WebVidQA, TGIF, EgoQA, etc.
Conversation
CLEVR, VisualMRC, NExT-QA, CLEVRER, etc.
Reasoning
LLaVA-Instruct, Instruction data of Video-LLaVA, VideoChat2, VideoLLaMa2, etc.
Integration

Step away

Do you ever notice how your brain can figure things out by itself? All it takes is to step away from the computer and take a break to think about something totally unrelated.

AI Tools for Academic Research:

Programming and Data Analysis in Biomedical Sciences

Latest Trends and Revolutionary Applications (2024-2025)

- **Biomedical-Specific AI Tools**
  - Protein analysis and drug discovery tools
  - Medical image analysis platforms
  - Genomics and omics data analysis
- **General Academic Programming Tools**
  - AI coding assistants
  - Data analysis and visualization
  - Literature review and writing aids
- **Setup and Integration Guide**
  - Installation instructions
  - API access and pricing
  - Best practices

AI Tools for Academic Research:
AlphaFold 3 & ColabFold - Protein Structure Analysis

- **Background & Use Case:**
  Need to predict protein structures for your research? Traditional methods take weeks/months and require expensive equipment.
- **Tool Overview:**
  - **AlphaFold 3:** Google DeepMind's Nobel Prize-winning protein structure predictor
  - **ColabFold:** Free, faster implementation running on Google Colab
- **Key Features:**
  - **Free access** through Google Colab
  - **Results in minutes** instead of weeks
  - **High accuracy** (90%+ for most proteins)
  - **Direct PDB file output** for visualization
- **Links:**
  - **AlphaFold Database:** https://alphafold.ebi.ac.uk/
  - **ColabFold:** https://colab.research.google.com/github/deepmind/alphafold/
  - **Tutorial:** https://alphafold.ebi.ac.uk/help

AI Tools for Academic Research:

AlphaFold 3 & ColabFold - Protein Structure Analysis

- **Practical Example**

```
# ColabFold example for COVID-19 spike protein sequence =
"MFVFLVLLPLVSSQCVNLTTRTQLPPAYTNSFTRGVYYPDKVFRSSVLHSTQDLFLPFFSNVTWFHAIHVSGTNGTKRFDNPVLPFNDGVYFASTEKSNII
RGWIFGTTLDSKTQSLLIVNNATNVVIKVCEFQFCNDPFLGVYYHKNNKSWMESEFRVYSSANNCTFEYVSQPFLMDLEGKQGNFKNLREFVFKNIDGYF
KIYSKHTPINLVRDLPQGFSALEPLVDLPIGINITRFQTLLALHRSYLTPGDSSSGWTAGAAAYYVGYLQPRTFLLKYNENGTITDAVDCALDPLSETKCTLKS
FTVEKGIYQTSNFRVQPTESIVRFPNITNLCPFGEVFNATRFASVYAWNRKRISNCVADYSVLYNSASFSTFKCYGVSPTKLNDLCFTNVYADSFVIRGDEV
RQIAPGQTGKIADYNYKLPDDFTGCVIAWNSNNLDSKVGGNYNYLYRLFRKSNLKPFERDISTEIYQAGSTPCNGVEGFNCYFPLQSYGFQPTNGVGYQP
YRVVVLSFELLHAPATVCGPKKSTNLVKNKCVNFNFNGLTGTGVLTESNKKFLPFQQFGRDIADTTDAVRDPQTLEILDITPCSFGGVSVITPGTNTSNQVA
VLYQDVNCTEVPVAIHADQLTPTWRVYSTGSNVFQTRAGCLIGAEHVNNSYECDIPIGAGICASYQTQTNSPRRARSVASQSIIAYTMSLGAENSVAYSNN
SIAIPTNFTISVTTEILPVSMTKTSVDCTMYICGDSTECSNLLLQYGSFCTQLNRALTGIAVEQDKNTQEVFAQVKQIYKTPPIKDFGGFNFSQILPDPSKPSK
RSFIEDLLFNKVTLADAGFIKQYGDCLGDIAARDLICAQKFNGLTVLPPLLTDEMIAQYTSALLAGTITSGWTFGAGAALQIPFAMQMAYRFNGIGVTQNVLY
ENQKLIANQFNSAIGKIQDSLSSTASALGKLQDVVNQNAQALNTLVKQLSSNFGAISSVLNDILSRLDKVEAEVQIDRLITGRLQSLQTYVTQQLIRAAEIRAS
ANLAATKMSECVLGQSKRVDFCGKGYHLMSFPQSAPHGVVFLHVTYVPAQEKNFTTAPAICHDGKAHFPREGVFVSNGTHWFVTQRNFYEPQIITTDNT
FVSGNCDVVIGIVNNTVYDPLQPELDSFKEELDKYFKNHTSPDVDLGDISGINASVVNIQKEIDRLNEVAKNLNESLIDLQELGKYEQYIKWPWYIWLGFIAG
LIAIVMVTIMLCCMTSCCSCLKGCCSCGSCCKFDEDDSEPVLKGVKLHYT“

 # This sequence can be processed through ColabFold
# Link: https://colab.research.google.com/github/deepmind/alphafold/
```

AI Tools for Academic Research:
ChemBERTa & RDKit-AI - Drug Discovery Assistant

- **Background & Use Case:**
  Screening thousands of compounds for drug properties manually is impossible. Need AI to predict toxicity, solubility, and bioactivity.
- **Tool Overview:**
  - **AChemBERTa:** BERT-based model for molecular property prediction
  - **RDKit-AI:** Enhanced molecular informatics with AI capabilities
- **Key Features:**
  - **SMILES string input** (standard chemical notation)
  - **Multiple property predictions** (toxicity, solubility, permeability)
  - **Pre-trained on millions** of chemical compounds
  - **Integration with RDKit** for visualization
- **Use Cases:**
  - Virtual screening of compound libraries
  - ADMET (Absorption, Distribution, Metabolism, Excretion, Toxicity) prediction
  - Lead compound optimization
- **Links:**
  - **Hugging Face ChemBERTa:** https://huggingface.co/seyonec/ChemBERTa-zinc-base-v1
  - **RDKit:** https://www.rdkit.org/
  - **Tutorial:** https://github.com/seyonechithrananda/bert-loves-chemistry

```
# Install dependencies
!pip install transformers rdkit-pypi chembl_webresource_client

from transformers import AutoTokenizer,
AutoModelForSequenceClassification import pandas as pd

# Load pre-trained ChemBERTa model
tokenizer = AutoTokenizer.from_pretrained("seyonec/ChemBERTa-zinc-
base-v1")
model =
AutoModelForSequenceClassification.from_pretrained("seyonec/ChemB
ERTa-zinc-base-v1")

# Example: Predict properties of aspirin
aspirin_smiles = "CC(=O)OC1=CC=CC=C1C(=O)O"
inputs = tokenizer(aspirin_smiles, return_tensors="pt")
outputs = model(**inputs)

# Get toxicity prediction
toxicity_score = outputs.logits.softmax(dim=-1)
print(f"Toxicity prediction: {toxicity_score}")
```

AI Tools for Academic Research:
MONAI & MedSAM- Medical Image Analysis

- **Background & Use Case:**
  Analyzing thousands of medical images (CT, MRI, X-rays)
  for research requires automated segmentation and analysis.
- **Tool Overview:**
  - **MONAI**: Medical Open Network for AI - comprehensive medical imaging toolkit
  - **MedSAM:** Medical Segment Anything Model for universal medical image segmentation
- **Key Features:**
  - **Pre-trained medical models** (no training required)
  - **Multiple modalities** (CT, MRI, X-ray, ultrasound)
  - **Automatic segmentation** with minimal input
  - **Research-ready pipeline** with data loaders
- **Links:**
  - **MONAI:** https://monai.io/
  - **MedSAM:** https://github.com/bowang-lab/MedSAM
  - **Tutorials:** https://tutorials.monai.io/

```
# MONAI installation and basic usage
!pip install monai[all]
import monai
from monai.data import DataLoader, Dataset
from monai.transforms import Compose, LoadImaged, EnsureChannelFirstd,
Spacingd

# Example: Lung CT scan analysis
transforms = Compose([
LoadImaged(keys=["image", "label"]),
EnsureChannelFirstd(keys=["image", "label"]),
Spacingd(keys=["image", "label"],
pixdim=(1.5, 1.5, 2.0)), ])

# MedSAM for automatic segmentation
from segment_anything import sam_model_registry, SamPredictor
import torch
# Load MedSAM model
model_type = "vit_b“
checkpoint = "medsam_vit_b.pth" # Download from GitHub
sam = sam_model_registry[model_type](checkpoint=checkpoint) predictor =
SamPredictor(sam)

# Process medical image
predictor.set_image(medical_image)
masks, scores, logits = predictor.predict( point_coords=input_point,
point_labels=input_label, multimask_output=True, )
```

AI Tools for Academic Research:
scGPT & CellTypist - Single-Cell Analysis

- **Background & Use Case:**
  Single-cell RNA sequencing generates massive datasets (millions of cells). Manual analysis is impossible; need AI for cell type identification and trajectory analysis.
- **Tool Overview:**
  - **scGPT**: GPT-based foundation model for single-cell
  - **genomicsCellTypist:** Automated cell type annotation tool
- **Key Features:**
  - **Foundation model pre-training** on millions of cells
  - **Automatic cell type annotation** with confidence scores
  - **Trajectory inference and** developmental analysis
  - **Integration with Scanpy** ecosystem
- **Use Cases:**
- **Use Cases:**
  - Developmental biology studies
  - Disease progression analysis
  - Drug response prediction
  - Biomarker discovery
- **Links:**
  - **scGPT:** https://github.com/bowang-lab/scGPT
  - **CellTypist:** https://www.celltypist.org/
  - **Documentation:** https://scgpt.readthedocs.io/

```
# scGPT installation and usage

!pip install scgpt scanpy pandas
import scgpt
import scanpy as sc
import pandas as pd

# Load your single-cell data
adata = sc.read_h5ad("your_single_cell_data.h5ad")

 # Preprocess with scGPT
from scgpt.preprocess import Preprocessor
preprocessor = Preprocessor(
use_key="X", # the key in adata.layers to use as raw data
filter_gene_by_counts=3, # step 1
filter_cell_by_counts=False, # step 2
normalize_total=1e4, # 3. whether to normalize the raw data
result_normed_key="X_normed", # the key in adata.layers to store normalized
data log1p=True, # 4. whether to log1p the normalized data
result_log1p_key="X_log1p",
)

# Cell type prediction with CellTypist
import celltypist predictions = celltypist.annotate(adata,
model='Immune_All_Low.pkl')
```

Foundation Models - The Game Changer

- **What Are Foundation Models?**
- **Definition:**
  Large-scale AI models trained on massive datasets that can be adapted for multiple downstream tasks without task-specific training
- **Biomedical Examples:**
  - **Med-PaLM 2:** Achieved **67.6% passing score** on US Medical Licensing Examination
  - **BioGPT:** Specialized for biomedical text generation and mining
  - **AlphaFold 3: 2024 Nobel Prize** for protein structure prediction
- **Key Advantage:** One model, multiple applications - from diagnosis to drug discovery to patient care
- **Why This Matters for You:**
  These models can understand and process the same types of data you work with daily, but at unprecedented scale and accuracy
- **References:**

1. Singhal, K. et al. "Large language models encode clinical knowledge." *Nature*, 2023
2. Abramson, J. et al. "Accurate structure prediction with AlphaFold 3." *Nature*, 2024
3. Luo, R. et al. "BioGPT: generative pre-trained transformer for biomedical text." *Briefings in Bioinformatics*, 2022

Med-PaLM Multimodal - The Universal Biomedical AI

Revolutionary Capabilities:

- **What It Does:**
    - **Single model** processes text, images, and genomic data simultaneously
    - **14 diverse tasks** from medical Q&A to radiology report generation
    - **Zero-shot learning** for novel medical concepts
- **Performance Highlights:**
    - **86.1% accuracy** in medical visual question answering
    - **Competitive with specialists** across multiple medical domains
    - **40.5% preference rate** over human radiologist reports
- **Real-World Impact:**
    - Radiology workflow acceleration
    - Consistent diagnostic quality across institutions
    - 24/7 availability for medical consultation
- **For Your Practice:**
  Imagine having an AI assistant that understands your field as well as a colleague, available instantly
- **References:**

1. Tu, T. et al. "Towards Generalist Biomedical AI." *arXiv*, 2023
2. Singhal, K. et al. "Expert-level medical question answering." *Nature Medicine*, 2025

BioMedLM & Specialized Language Models

Cost-Effective Specialized AI:

- **BioMedLM Specifications:**
  - **2.7 billion parameters** (smaller but smarter)
  - Trained exclusively on **PubMed abstracts and full articles**
  - **Domain-specific tokenizer** for biomedical terminology
- **Impressive Performance:**
  - **57.3% on MedMCQA** (competitive with much larger models)
  - **69.0% on MMLU Medical Genetics**
  - **74.4% on PubMedQA** tasks
- **Economic Advantage:**
  - **90% lower computational costs** than GPT-4 scale models
  - Suitable for individual institutions and research groups
  - **Privacy-preserving** (can run locally)
- **Practical Applications:**
  - Literature review automation
  - Medical report summarization
  - Clinical decision support
- **References:**

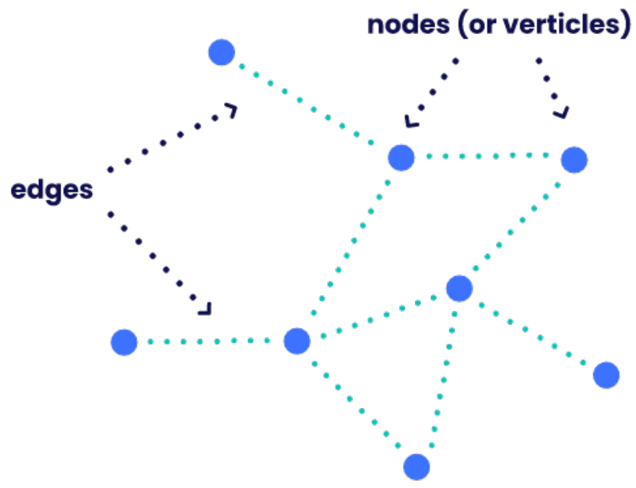
  1. Bolton, E. et al. "BioMedLM: A 2.7B Parameter Language Model." 2024
  2. Labrak, Y. et al. "BioMistral: Open-source biomedical language model." 2024

Graph Neural Networks - Molecular Intelligence

Understanding Molecular Relationships:

- **Why Graphs for Molecules?**
  - **Atoms as nodes**, **bonds as edges**
  - Captures **3D spatial relationships**
  - Preserves **chemical structure information**
- **Drug Discovery Applications:**
  - **Drug-target interaction prediction:** 95%+ accuracy
  - **Molecular property prediction:** Enhanced vs traditional methods
  - **Drug synergy analysis:** Optimizing combination therapies
- **Recent Breakthroughs (2024):**
  - **GCN-DTI models:** Identifying novel drug-target interactions
  - **CCL-DTI algorithm:** Contrastive learning for better predictions
  - **Multi-modal fusion:** Combining molecular graphs with other data
- **Impact on Pharmaceutical Research:**
  - **2-3 year reduction** in drug discovery timelines
  - **40%+ success rate** in drug repurposing identification
  - **Cost reduction** of millions per successful drug
- **References:**

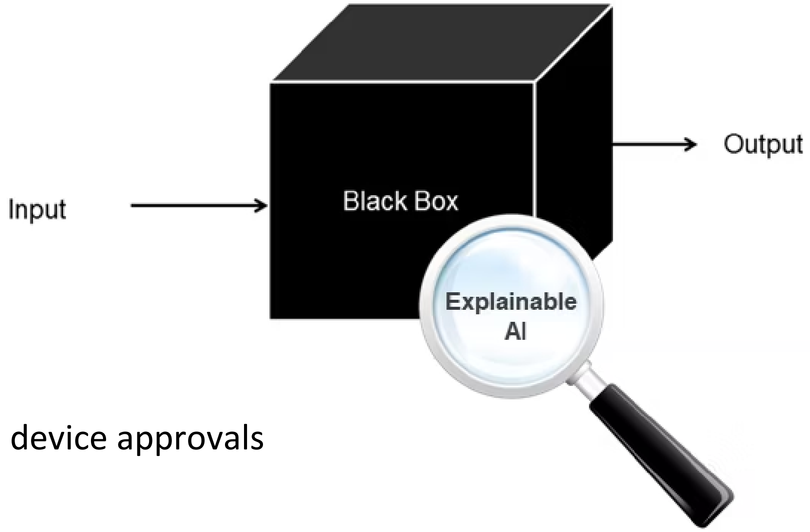
1. Yao, R. et al. "Graph neural networks for drug discovery: bibliometric analysis." *Frontiers in Pharmacology*, 2024
2. Dehghan, A. et al. "CCL-DTI: contrastive loss in drug-target interaction prediction." *BMC Bioinformatics*, 2024



Explainable AI - Building Clinical Trust

Making AI Transparent:

- **The Black Box Problem:** Traditional AI models make decisions **without explaining reasoning** - problematic for clinical use
- **Explainable AI Solutions:**
  - **SHAP/LIME explanations:** Feature importance visualization
  - **Attention mechanisms:** Highlighting relevant image regions
  - **Uncertainty quantification:** Confidence levels for predictions
- **Clinical Applications:**
  - **Cancer diagnosis:** Showing which image features indicate malignancy
  - **Drug interactions:** Explaining molecular mechanisms
  - **Treatment recommendations:** Justifying therapeutic choices
- **Clinical Acceptance:**
  - **85%+ clinician acceptance** with explainable models
  - **40% reduction** in false positive rates
  - **Enhanced patient trust** through transparency
- **Regulatory Requirements:** FDA increasingly requires **explainability** for AI medical device approvals
- **References:**



1. IEEE Journal on Biomedical Health Informatics. "Explainable AI-Driven Medical Imaging." 2025
2. Nature Machine Intelligence. "Expert-level pathology detection with explanations." 2024

General Academic Programming Tools
Github Copilot & Cursor - AI Programming Assistants

- **Background & Use Case:**
  Writing research code (data analysis, algorithms, visualizations) is time-consuming and error-prone. Need AI assistance for faster, better coding.
- **Tool Overview:**
  - **GitHub Copilot:** AI pair programmer by Microsoft/OpenAI
  - **Cursor:** AI-first code editor with advanced contextual understanding
- **How It Works:**
  - **Type comments** describing what you want to do
  - **AI suggests complete functions** based on your description
  - **Real-time code completion** as you type
  - **Debugging assistance** and optimization suggestions
- **Key Features:**
  - **Context-aware suggestions** based on your existing code
  - **Multiple language support** (Python, R, Julia, MATLAB, etc.)
  - **Research-specific patterns** (data analysis, ML, statistics)
  - **Natural language commands** ("add error handling to this function")
  - **Codebase-wide understanding** (knows your entire project)
- **Practical Applications:**
  - **Gene expression analysis** correlation matrices and heatmaps
  - **Statistical testing** automated hypothesis testing
  - **Data visualization** publication-ready plots
  - **Machine learning pipelines** model training and evaluation
- **Pricing:**
  - **GitHub Copilot:** $10/month (free for students)
  - **Cursor:** $20/month with free tier
- **Links:**
  - **GitHub Copilot:** https://github.com/features/copilot
  - **Cursor:** https://cursor.sh/
  - **Student discount:** https://education.github.com/

General Academic Programming Tools
Claude & ChatGPT Code Interpreter - Data Analysis

- **Background & Use Case:**
  Need quick data analysis, statistical tests, or visualization without writing complex code. Want AI to understand your research context.
- **Tool Overview:**
  - **Claude (Anthropic):** Advanced reasoning with code execution capabilities
  - **ChatGPT Code Interpreter:** OpenAI's data analysis tool with Python environment
- **How It Works:**
  1. **Upload your data files** directly to the platform
  2. **Describe your analysis needs** in natural language
  3. **AI generates and executes code** automatically
  4. **Get results with interpretation** and explanations
- **Example Use Cases:**
  - **RNA-seq differential expression analysis** with publication-ready plots
  - **Clinical trial statistical analysis** with appropriate tests
  - **Volcano plots and heatmaps** for genomics data
  - **Patient outcome correlations** with demographic factors

- **Key Features:**
  - **Upload data files** directly (CSV, Excel, etc.)
  - **Automatic statistical analysis** with interpretation
  - **Publication-ready visualizations**
  - **Research methodology suggestions**
  - **Natural language explanations** of results
- **Practical Applications:**
  - Quick exploratory data analysis
  - Statistical test selection and execution
  - Data visualization and interpretation
  - Manuscript figure generation
- **Links:**
  - **Claude:** https://claude.ai/
  - **ChatGPT Plus:** https://chat.openai.com/
  - **Pricing:** $20/month each

General Academic Programming Tools
Semantic Scholar API & Research Rabbit - Literature Review

- **Background & Use Case:**
  Manually searching through thousands of papers for literature review is inefficient. Need AI to find relevant papers and extract key insights.

- **Tool Overview:**
  - **Semantic Scholar API:** AI-powered academic search with paper insights
    Research Rabbit: Visual literature exploration and recommendation system
- **How They Work:**
  - **Semantic Scholar Features:**
    - **Semantic search understanding** (not just keyword matching)
    - **Citation analysis** and impact metrics
    - **Author and venue insights**
    - **Trend analysis** over time
    - **Free API access** for researchers
  - **Research Rabbit Features:**
    - **Visual paper network** showing connections between papers
    - **Automatic recommendations** based on your interests
    - **Collaboration features** for team research
    - **Export to reference managers** (Zotero, Mendeley)

- **Practical Applications:**
  - **Comprehensive literature searches** with better relevance
  - **Citation analysis** to find high-impact papers
  - **Research trend identification** over time periods
  - **Author network analysis** for collaboration opportunities
  - **Gap identification** in current research
- **Key Benefits:**
  - **AI-powered search** with semantic understanding
  - **Visual exploration** of research landscapes
  - **Time savings** of 50-70% in literature review
  - **Discovery of relevant papers** you might miss
- **Use Cases:**
  - Systematic literature reviews
  - Research proposal background
  - Grant application literature support
  - Staying updated with latest developments
- **Links:**
  - **Semantic Scholar:** https://www.semanticscholar.org/
  - **API Documentation:** https://api.semanticscholar.org/
  - **Research Rabbit:** https://www.researchrabbit.ai/

General Academic Programming Tools
Grammarly & Writefull - Academic Writing AI

- **Background & Use Case:**
  Academic writing requires precision, clarity, and proper style. Non-native speakers especially need assistance with grammar and academic tone.
- **Tool Overview:**
  - **Grammarly:** Comprehensive writing assistant with academic features
    Writefull: AI writing tool specifically designed for academic writing
- **Key Features:**
  - **Grammarly:**
    - **Grammar and spelling** correction
    - **Tone adjustment** (formal, academic)
    - **Plagiarism detection**
    - **Citation format** checking
  - **Writefull:**
    - **Academic phrase suggestions**
    - **Journal-specific writing patterns**
    - **Sentence variety** recommendations
    - **Abstract and title** optimization
- **Integration:**
  - **Microsoft Word** add-ins
  - **Overleaf** (LaTeX) integration
  - **Browser extensions** for web writing
  - **Desktop applications**
- **Pricing:**
  - **Grammarly:** $12/month for Premium
  - **Writefull:** $4.99/month for academics
- **Links:**
  - **Grammarly:** https://www.grammarly.com/
  - **Writefull:** https://www.writefull.com/
  - **Academic discounts:** Available for both platforms

General Academic Programming Tools
Perplexity & Elicit - Research Question Answering

- **Background & Use Case:**
  Quick answers to research questions with citations. Understanding complex topics across disciplines without extensive literature review.

- **Tool Overview:**
  - **Perplexity:** AI search engine with real-time web access and citations
  - **Elicit:** AI research assistant for scientific questions

- **Key Features:**
  - **Perplexity:**
    - **Real-time information** from latest publications
    - **Automatic citations** with links to sources
    - **Follow-up question** suggestions
    - **Multi-source synthesis**
  - **Elicit:**
    - **Research workflow** optimization
    - **Paper summarization** with key findings
    - **Claim verification** against literature
    - **Methodology extraction** and comparison

- **Use Cases:**
  - Quick background research for grant proposals
  - Staying updated with latest developments in your field
  - Verifying claims and finding supporting sources
  - Generating research hypotheses and directions
- **Links:**
  - **Perplexity:** https://perplexity.ai/
  - **Elicit:** https://elicit.org/
  - **Pricing:** Free tiers available, Pro versions ~$20/month

Emergency Contacts:
- **GitHub Copilot:** https://support.github.com/
- **OpenAI:** https://help.openai.com/
- **Institutional IT:** Your university's IT helpdesk
- **Communities:** Join Discord/Slack channels for each tool
- **Support:** Most paid tools offer email support
- **Forums:** Stack Overflow, Reddit communities

Often Asked Questions

- **Memory Errors with Large Datasets:**
  - **Solution:** Process data in smaller batches
  - **Use cloud computing** for resource-intensive tasks
  - **Optimize data formats** (use compressed files)
  - **Monitor resource usage** during processing

- **API Rate Limiting:**
  - **Implement delays** between requests
  - **Use multiple API keys** if allowed
  - **Batch requests** when possible
  - **Monitor usage limits** proactively

- **Tool Integration Issues:**
  - **Check version compatibility** between tools
  - **Use virtual environments** to avoid conflicts
  - **Test integrations** with small datasets first
  - **Document working configurations**

- **Data Privacy Concerns:**
  - **Use institutional licenses** when available
  - **Check data policies** before uploading sensitive data
  - **Consider local installations** for confidential research
  - **Anonymize data** when possible before AI processing

- **Quality Control:**
  - **Always validate** AI outputs independently
  - **Use multiple tools** for cross-validation
  - **Maintain human oversight** for critical decisions
  - **Document AI tool versions** for reproducibility

- **Getting Help:**
  - **Documentation:** Each tool has comprehensive docs
  - **Communities:** Join Discord/Slack channels for each tool
  - **Support:** Most paid tools offer email support
  - **Forums:** Stack Overflow, Reddit communities

Future Tools & Emerging Technologies

Coming Soon (2025-2026)

- **Next-Generation Biomedical AI:**
  - **AlphaFold 4:** Multi-protein complex prediction
  - **Med-PaLM 3:** Enhanced multimodal capabilities
  - **scGPT 2.0:** Real-time single-cell analysis
  - **BioGPT-3:** Advanced biomedical reasoning

- **Programming Assistants:**
  - **GitHub Copilot X:** Full IDE integration
  - **Amazon CodeWhisperer:** AWS-integrated development
  - **DeepMind AlphaCode:** Competitive programming level
  - **Cursor AI:** Advanced contextual understanding

- **Research Automation:**
  - **Auto-GPT for Research:** Autonomous research pipelines
  - **LangChain for Science:** Complex research workflows
  - **Research Agent Networks:** Multi-agent research systems

- **Investment Strategy:**
  - **Start with free tools** to learn workflows
  - **Upgrade selectively** based on usage patterns
  - **Monitor new releases** for breakthrough capabilities
  - **Budget 10-15%** of research budget for AI tools

Key Features of LangChain

01 Modular Component

LangChain's modular design simplifies development, enabling effortless application building and experimentation.

02 Integration with External Data Sources

Integrates with services, enhancing response quality with contextually relevant data.

03 Prompt Engineering

Create and refine prompts using templates for consistent, accurate LLM responses.

04 Memory Capabilities

Remembers conversations, ensuring coherent responses and enhancing customer support satisfaction.

05 Retrieval Augmented Generation (RAG)

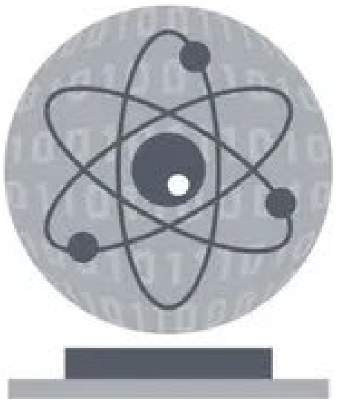
Combines data retrieval with LLMs, improving accuracy and reducing hallucinations.

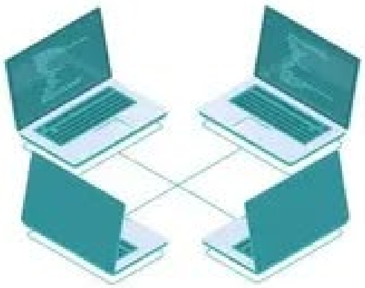
06 Deployment and Monitoring

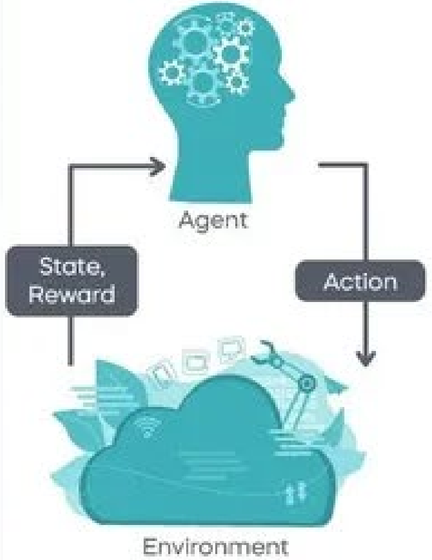
LangSmith and LangServe simplify debugging, testing, monitoring, and deployment of applications.

TOP 5 MACHINE LEARNING TRENDS TO WATCH IN THE FUTURE

The Quantum Computing Effect	The Big Model Creation	Distributed ML Portability	No-Code Environment	The Quantum Computing Effect
Quantum computing will optimize ML speed	Creation of an all-purpose model to perform tasks in various domains simultaneously	Businesses will run existing algorithms and datasets natively on various platforms and computer engines	Machine learning will become a branch of software engineering	Raise of new RL mechanisms for leveraging data to optimize resources in a dynamic setting
Reduced execution times in high-dimensional vector processing	Users can tailor such an uber ML model	Portability will eliminate the need for shifting to new toolkits constantly	Minimized coding effort and maximized access to machine learning programs	RL will shift economics, biology, and astronomy







https://365datascience.com/trending/future-of-machine-learning/

Ethical Considerations & Best Practices
Research Integrity

- Proper Attribution

```latex
% Example acknowledgment in papers
\section{Acknowledgments}
This research was assisted by AI tools including GitHub Copilot
for code development, Grammarly for manuscript preparation, and
Semantic Scholar API for literature analysis. All AI-generated
content was reviewed and validated by the authors.
```

- Transparency Guidelines
  - Disclose AI usage in methodology sections
  - Validate AI results with independent verification
  - Maintain human oversight for critical decisions
  - Document AI tool versions for reproducibility

Ethical Considerations & Best Practices
Data Privacy & Security

- Institutional Policies (Anonymization)

```python
# Example data anonymization before AI processing
def anonymize_patient_data(data):
    # Remove direct identifiers
    data = data.drop(['patient_id', 'name', 'ssn'], axis=1)

    # Add noise to sensitive measurements
    data['age'] = data['age'] + np.random.normal(0, 0.5, len(data))

    # Categorical masking
    data['location'] = data['location'].apply(generalize_location)

    return data

# Only process anonymized data with AI tools
anonymized_data = anonymize_patient_data(raw_patient_data)
ai_results = ai_tool.analyze(anonymized_data)
```

- Security Checklist:
    - Review institutional policies before using AI tools
    - Use secure connections (HTTPS/VPN)
    - Avoid uploading sensitive raw data
    - Check data residency requirements
    - Monitor access logs for compliance

Ethical Considerations & Best Practices
Quality Assurance

- Validation Framework

```python
class AIResultValidator:
    def __init__(self):
        self.validation_tests = [
            self.check_statistical_significance,
            self.verify_biological_plausibility,
            self.cross_reference_literature,
            self.peer_review_validation
        ]

    def validate_results(self, ai_output):
        validation_scores = []
        for test in self.validation_tests:
            score = test(ai_output)
            validation_scores.append(score)

        overall_confidence = np.mean(validation_scores)
        return overall_confidence > 0.8  # 80% confidence threshold
```

- Bias & Fairness:
  - Diverse training data awareness
  - Population representation in medical AI
  - Regular bias testing of AI outputs
  - Inclusive research practices

Potential Challenges of Generative AI on Healthcare

1. **Privacy and security**
   Patient privacy is strictly regulated. The use of generative AI in healthcare also raises concerns about protecting patient privacy, sensitive medical data and the potential for unauthorized access to the healthcare data.

2. **Bias and discrimination**
   Generative AI algorithms can be **prone to bias and discrimination**, especially if they are trained on healthcare data that is not representative of the population they are intended to serve. This can result in unfair or inaccurate medical diagnoses or treatment plans for underprivileged groups such as women or non-white races.

3. **Misuse and over-reliance**
   If generative AI algorithms are not used properly, they can lead to incorrect or harmful medical decisions. There is a risk that healthcare providers may become **overly reliant on these algorithms** and lose the ability to make independent judgments.

4. **Ethical considerations**
   Impact on **employment** in the healthcare sector.

How can AI Application Infrastructure be **Sustainable**?

Data

Model + **Sustainability**

Application

→ Value-added, Open Science

→ Robust Learning
Dynamic Adaptive System
Efficient information flow

→ Collaboration and Competition in the Academic and Research Community, Ecosystem Establishment

---

Interpretation / Generosity / Faith (Trust) / Ethics

Funding/ computation and storage sources

→ Human-(Relationship?-)centered
Home-based ?
Empirical-based knowledge
Problem-based approach
Reinforcement/federated …

Before We Start…

Step 1: **Adjust Mindset**. Believe you can practice and apply machine learning.

- What is Holding you Back From Your Machine Learning Goals?
- Why Machine Learning Does Not Have to Be So Hard
- How to Think About Machine Learning
- Find Your Machine Learning Tribe

Step 2: **Pick a Process**. Use a systemic process to work through problems.

- Applied Machine Learning Process

Step 3: **Pick a Tool**. Select a tool for your level and map it onto your process.

- Beginners: Weka Workbench
- Intermediate: Python Ecosystem
- Advanced: R Platform
- Programming Language for Machine Learning

Step 4: **Practice on Datasets**. Select datasets to work on and practice the process.

- Practice Machine Learning with Small In-Memory Datasets
- Tour of Real-World Machine Learning Problems
- Work on Machine Learning Problems That Matter To You

Step 5: **Build a Portfolio**. Gather results and demonstrate your skills.

- Build a Machine Learning Portfolio
- Get Paid To Apply Machine Learning
- Machine Learning For Money

Need Help Getting Started with Applied Machine Learning?

These are the Step-by-Step Guides that You've Been Looking For!

What do you want help with?

Foundations

- How Do I Get Started?
- Step-by-Step Process
- Probability
- Statistical Methods
- Linear Algebra
- Optimization
- Calculus

Beginner

- Python Skills
- Understand ML Algorithms
- ML + Weka (no code)
- ML + Python (scikit-learn)
- ML + R (caret)
- Time Series Forecasting
- Data Preparation
- Data Science

Intermediate

- Code ML Algorithms
- XGBoost Algorithm
- Imbalanced Classification
- Deep Learning (Keras)
- Deep Learning (PyTorch)
- ML in OpenCV
- Better Deep Learning
- Ensemble Learning

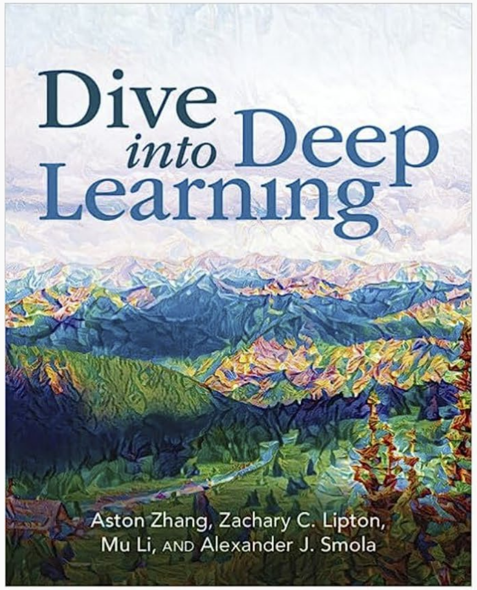
Advanced

- Long Short-Term Memory
- Natural Language (Text)
- Computer Vision
- CNN/LSTM + Time Series
- GANs
- Attention and Transformers

How Do I Get Started?

d2l.ai

DIVE INTO
DEEP LEARNING



Dive into Deep Learning

Interactive deep learning book with code, math, and discussions

Implemented with **PyTorch**, **NumPy/MXNet**, **JAX**, and **TensorFlow**

Adopted at 500 universities from 70 countries

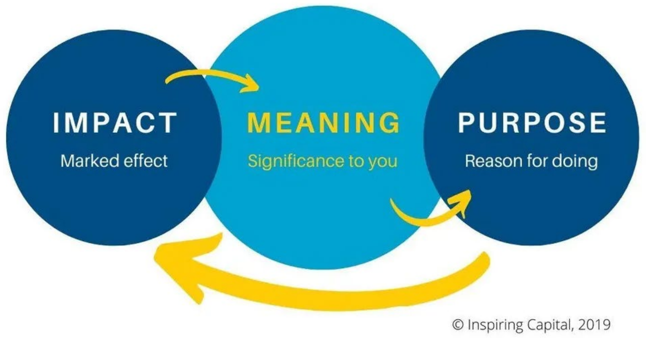
Star 26,496

Check List for Developing a Project

General guidelines, subject to change based on your aim.

1. **Frame the problem** and look at the big picture.
2. **Get the data**.
3. **Explore the data** to gain insights.
4. **Prepare the data** to better expose the underlying data patterns to ML/DL **algorithms**.
5. **Explore** many different models and short-list the best ones.
6. **Fine-tune your models** and combine them into a great solution.
7. Present your solution.
8. Launch, monitor, and maintain your system.

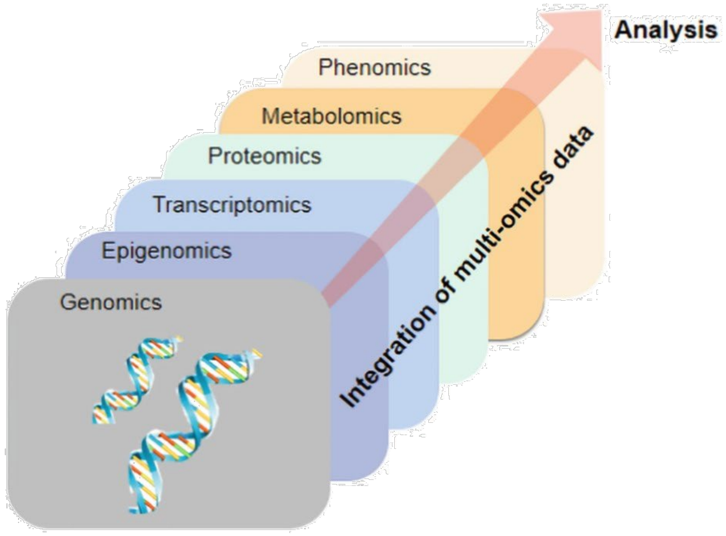
1. Looking at the Big Picture



Frame the Problem

1. Define the objective in terms.
2. How will your solution be used?
3. What are the current solutions/workarounds (if any)?
4. How should you frame this problem (supervised/unsupervised, online/offline, etc.)?
5. How should performance be measured?
6. Is the performance measure aligned with the business objective?
7. What would be the minimum performance needed to reach the business objective?
8. What are comparable problems? Can you reuse experience or tools?
9. Is human expertise available?
10. How would you solve the problem manually?
11. List the assumptions you (or others) have made so far.
12. Verify assumptions if possible.

2. Get the Data

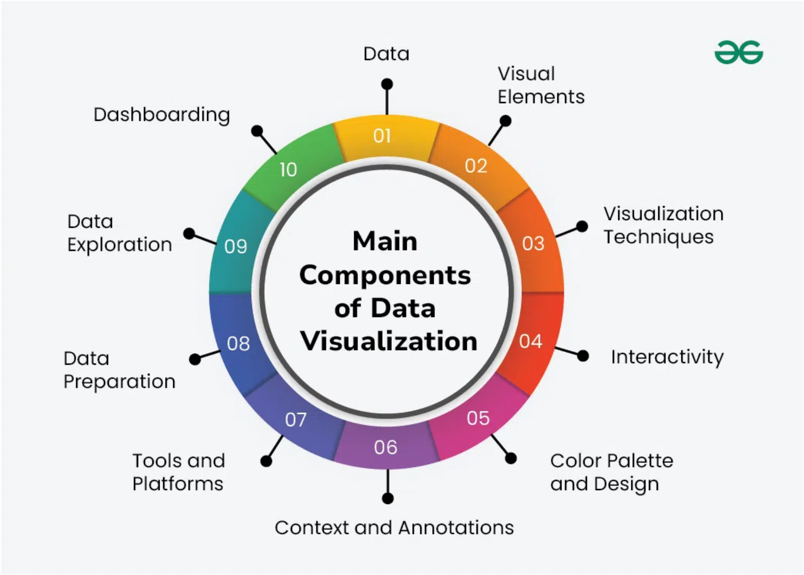


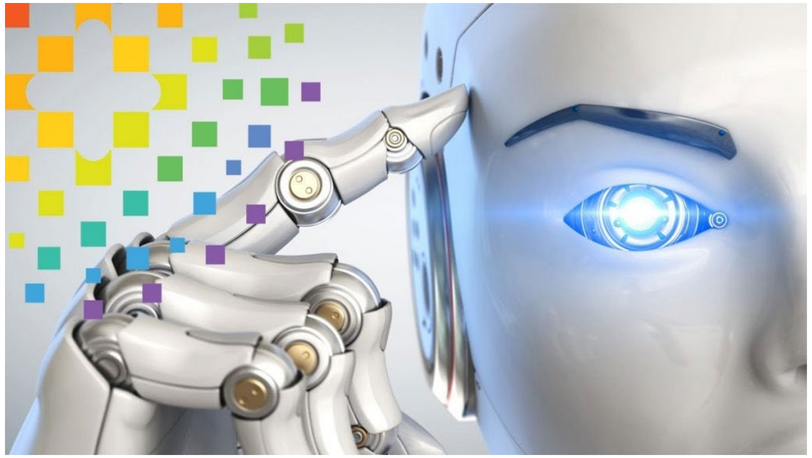
1. List the data you need and how much you need.
2. Find and document where you can get that data.
3. Check how much space it will take.
4. Check legal obligations, and get authorization if necessary.
5. Get access authorizations.
6. Create a workspace (with enough storage space).
7. Get the data.
8. Convert the data to a format you can easily manipulate (without changing the data itself).
9. Ensure sensitive information is deleted or protected (e.g., anonymized).
10. Check the size and type of data (time series, sample, geographical, etc.).
11. Sample a test set, put it aside, and never look at it (no data snooping!).

https://www.physalia-courses.org/courses-workshops/multiomics/

3. Explore the Data

1. Create a copy of the data for exploration
   (sampling it down to a manageable size if necessary).
2. Create a Jupyter notebook to keep a record of your data exploration.
3. Study each attribute and its characteristics:
   (1) Name
   (2) Type (categorical, int/float, bounded/unbounded, text, structured, etc.)
   (3) % of missing values
   (4) Noisiness and type of noise (stochastic, outliers, rounding errors, etc.)
   (5) Possibly useful for the task?
   (6) Type of distribution (Gaussian, uniform, logarithmic, etc.)
4. For supervised learning tasks, identify the target attribute(s).
5. **Visualize** the data.
6. Study the correlations between attributes.
7. Study how you would solve the problem manually.
8. Identify the promising transformations you may want to apply.
9. Identify extra data that would be useful (go back to "Get the Data").
10. Document what you have learned.





https://www.geeksforgeeks.org/data-visualization/what-are-the-key-components-of-data-visualization/
https://www.edge-ai-vision.com/2020/04/robot-senses-robots-that-can-see-hear-feel-and-more/

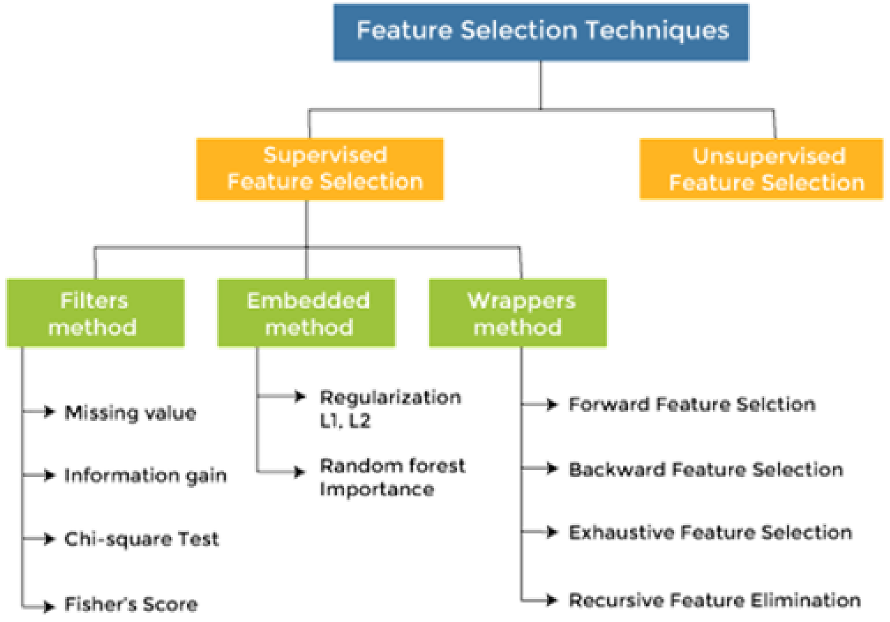
QC trade off

4. Prepare the Data

- Work on copies of the data
  (keep the original dataset intact).
- Write **functions** for all data transformations you apply, for five reasons:
  - So you can easily prepare the data the next time you get a fresh dataset
  - So you can apply these transformations in future projects
  - To clean and prepare the test set
  - To clean and prepare new data instances once your solution is live
  - To make it easy to treat your preparation choices as hyperparameters

https://www.mbaskool.com/business-concepts/operations-logistics-supply-chain-terms/2199-quality-tradeoff-curve.html

4. Prepare the Data



1. Data cleaning:
   - Fix or remove outliers (optional).
   - Fill in missing values (e.g., with zero, mean, median…) or drop their rows (or columns).
2. Feature selection (optional):
   - Drop the attributes that provide no useful information for the task.
3. Feature engineering, where appropriate:
   - Discretize continuous features.
   - Decompose features (e.g., categorical, date/time, etc.).
   - Add promising transformations of features (e.g., log(x), x^2, etc.).
   - Aggregate features into promising new features.
4. Feature scaling: standardize or normalize features.

https://medium.com/@favourphilic/an-easy-guide-to-feature-selection-77f7ac5ef5b0

5. Explore and training models

- If the data is huge, you may want to **sample smaller training sets** so you can train many different models in a reasonable time
(be aware that this penalizes complex models such as large neural nets or Random Forests).
- Once again, try to automate these steps as much as possible.

5. Explore and training models

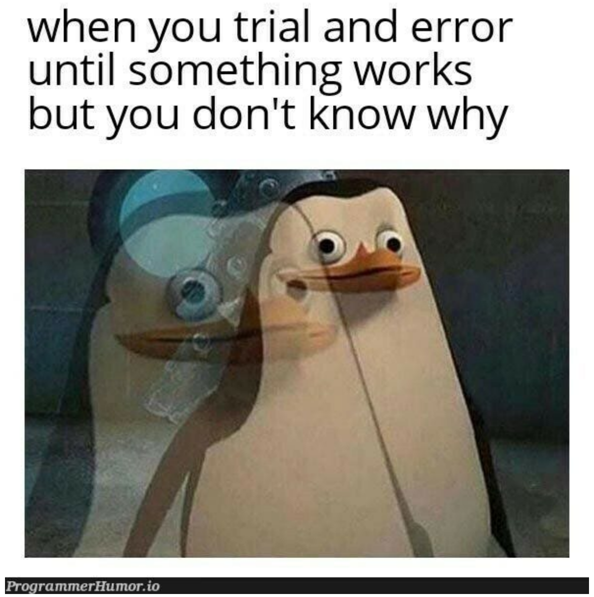
1. Train many quick and dirty models from different categories (e.g., linear, naive Bayes, SVM, Random Forests, neural net, etc.) using standard parameters.
2. Measure and compare their performance.

    For each model, use N-fold cross-validation and compute the mean and standard deviation of the performance measure on the N folds.

3. Analyze the most significant variables for each algorithm.
4. Analyze the types of errors the models make.

    What data would a human have used to avoid these errors?

5. Have a quick round of feature selection and engineering.
6. Have one or two more quick iterations of the five previous steps.
7. Short-list the top three to five most promising models, preferring models that make different types of errors.



6. Fine-tune the system

- You will want to use as much data as possible for this step, especially as you move toward the end of fine-tuning.
- As always **auto**mate what you can.

6. Fine-tune the system

1. Fine-tune the hyperparameters using cross-validation.

   Treat your data transformation choices as hyperparameters, especially when you are not sure about them. (e.g., should I replace missing values with zero or with the median value? Or just drop the rows?)

   Unless there are very few hyperparameter values to explore, prefer random search over grid search. If training is very long, you may prefer a Bayesian optimization approach.
   (e.g., using Gaussian process priors)

2. Try Ensemble methods. Combining your best models will often perform better than running them individually.

3. Once you are confident about your final model, measure its performance on the test set to estimate the generalization error.

7. Present your solution

1. Document what you have done.
2. Create a nice presentation.
   Make sure you highlight the big picture first.
3. **Explain why** your solution achieves the business objective.
4. Don’t forget to present interesting points you noticed along the way.
   Describe what worked and what did not.
   List your assumptions and your system’s limitations.
5. Ensure your key findings are communicated through beautiful visualizations or easy-to-remember statements (e.g., “the median income is the number-one predictor of housing prices”).

8. Launch

1. Get your solution ready for production (plug into production data inputs, write unit tests, etc.).

2. Write **monitoring code** to check your system's live performance at regular intervals and trigger alerts when it drops.
   - Beware of slow degradation too: models tend to “rot” as data evolves.
   - Measuring performance may require a human pipeline (e.g., via a crowdsourcing service).
   - Also monitor your **inputs' quality** (e.g., a malfunctioning sensor sending random values, or another team's output becoming stale). This is particularly important for online learning systems.

3. Retrain your models on a regular basis on fresh data (automate as much as possible).

https://openvalue.blog/posts/2024/08/12/codegreen-part1/

Check List for Developing a Project

1. **Frame the problem** and look at the big picture.
2. **Get the data**.
3. **Explore the data** to gain insights.
4. **Prepare the data** to better expose the underlying data patterns to ML/DL **algorithms**.
5. **Explore** many different models and short-list the best ones.
6. **Fine-tune your models** and combine them into a great solution.
7. Present your solution.
8. Launch, monitor, and maintain your system.

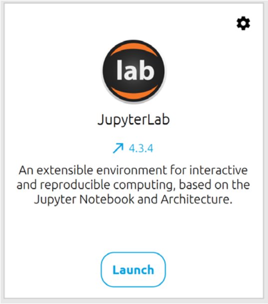
Practical Coding Session

- **Objectives of the Short Coding Course**
  At the end of this short course, attendees will have an understanding of what Python can do in addressing questions of relevance to precision medicine. With simple worked examples, attendees will see for themselves how items data from individual patients can generate, both in numerical terms and in graphical outputs , models of outcomes that have real world applicability.

Practical Coding Session

- **Materials Download** **(OXCEP- Training Program July 2025)**
  - https://shorturl.at/egcr1 (Google Drive)
  - https://github.com/griffinfarrow/OXCEP-Precision-Medicine?tab=readme-ov-file
- Installation Instruction- Anaconda

  We are using the Anaconda distribution, but if you'd rather just follow the course with your own Python version, there is a requirements.txt and environment.yml file in installation/.
  We use python 3.9. You can check whether all modules are installed using installation/verify_installation.py



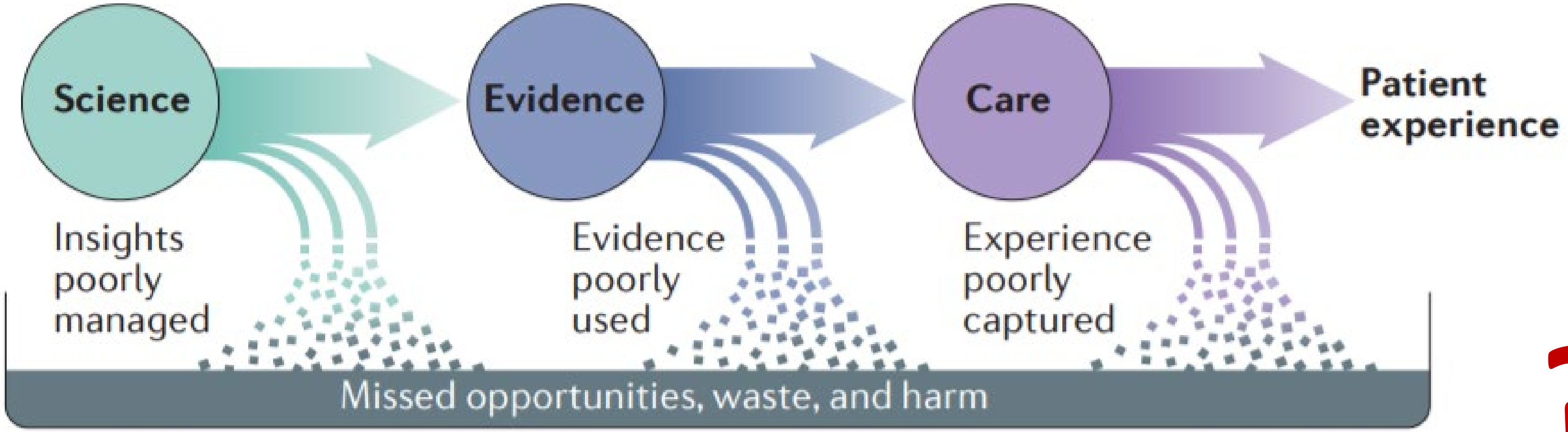
- Why Python?

  Python is an 'easy' to learn programming language that has libraries that can be used to explore a complex data set and generate answers in a user friendly format.

Practical Coding Session

- **Background skills prior to course**
  It is recommended that attendees have undertaken a 'Python 101.' Having done a preliminary online (or refresher) course will help attendees get more out of the course.
- The Practical hands-on will be divided in to 5 sessions,
  (1) Python basics
  (2) Data science best practices
  (3) Model Evaluation with Python
  (4) Identifying the most important inputs
  (5) Survival Analysis
  each session will have an Introductory tutorial, a demonstration, and tasks/support.

What can be wrong in medicine?



?

Figure 1 | **Health-care system today.** The current health-care system has important shortcomings and inefficiencies. Insights from research are poorly managed, the available evidence is poorly used, and the care experience is poorly captured, resulting in missed opportunities, wasted resources, and potential harm to patients. Reprinted with permission from *Best Care at Lower Cost: The Path to Continuously Learning Health Care in America* (2013) by the National Academy of Sciences, courtesy of the National Academies Press, Washington, D.C.

Rumsfeld, J., Joynt, K., & Maddox, T. (2016). Big data analytics to improve cardiovascular care: promise and challenges. Nature Reviews Cardiology, 13, 350-359.

Meaningful and Trust Worthy **AI**
&
Meaningful and Trust Worthy **Medicine**

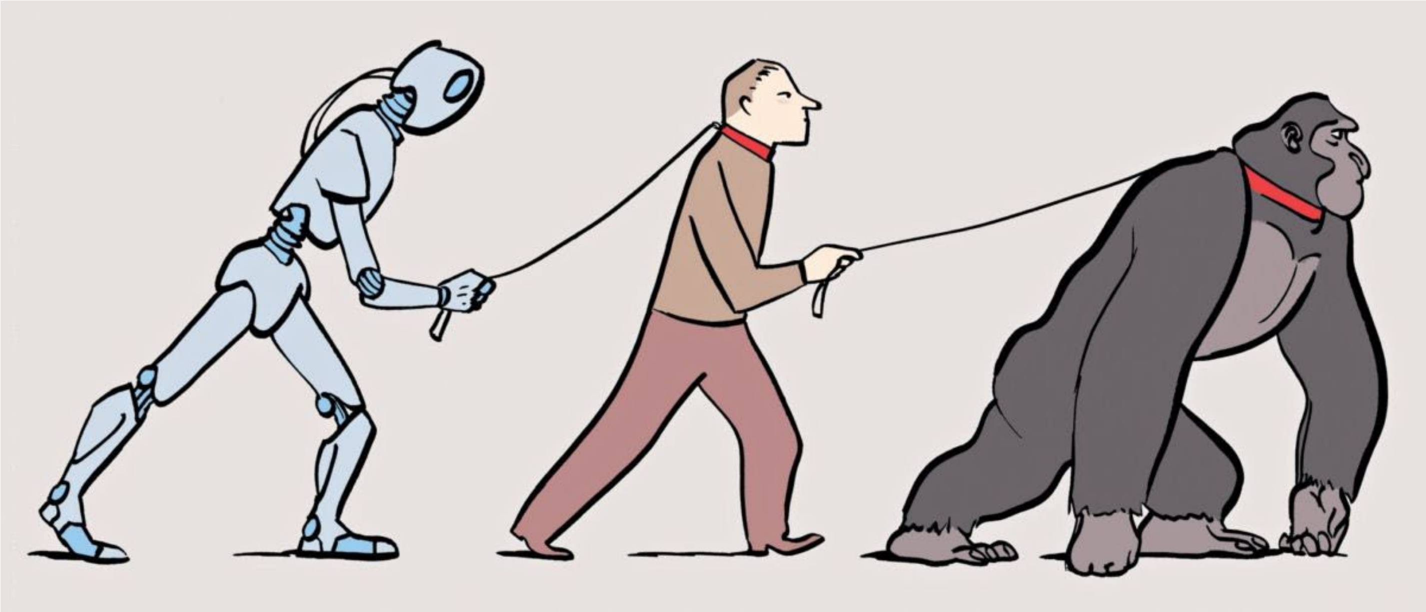
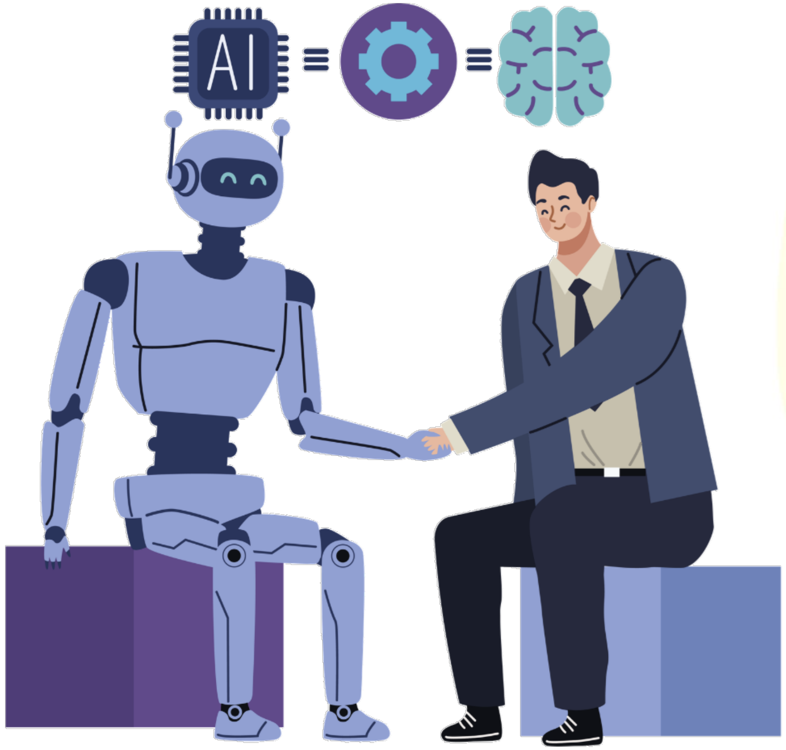


Figure source: the New Yorker November 28, 2016 Issue “If Animals Have Rights, Should Robots?” Illustration by Nishant Choksi



感謝您的聆聽，敬請建議指導

Thank you for your attention!

yijulee@stat.sinica.edu.tw