THE UNIVERSITY of TENNESSEE KNOXVILLE

Day 1: Introduction to Machine Learning

Sergei V. Kalinin

University of Tennessee, Knoxville, and
Pacific Northwest National Laboratory

Course outline

Day 1 – Intro and Decision Trees

Day 2 – Classification

Day 3 – Clustering and Dimensionality Reduction

Day 4 – Decisions and Bayesian worldview

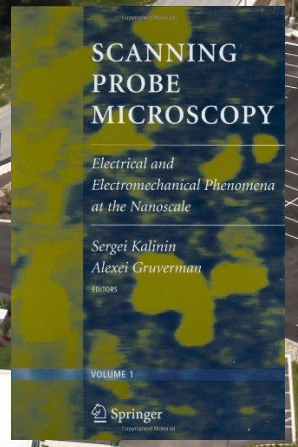
Day 5 – Deep Learning

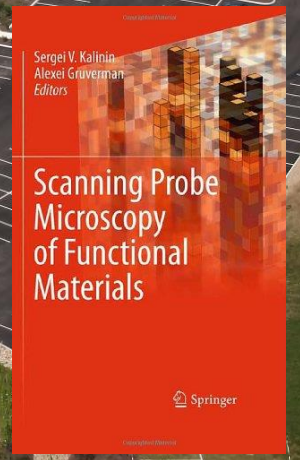
2002 - 2022

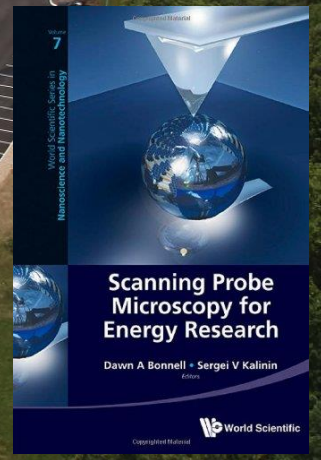
Since 2022

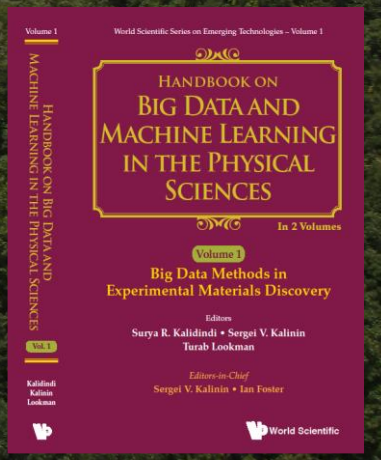
2022 -2023

amazon



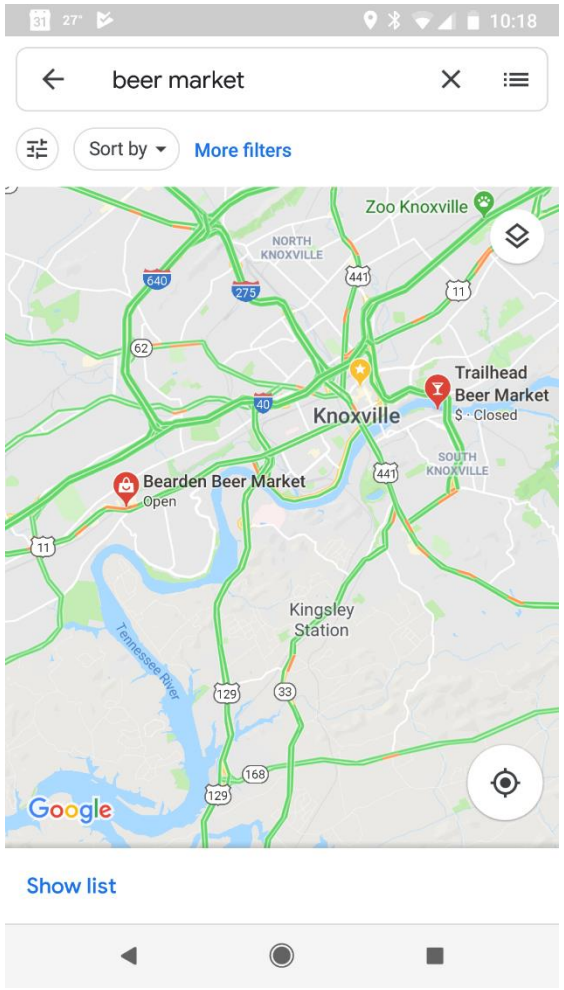


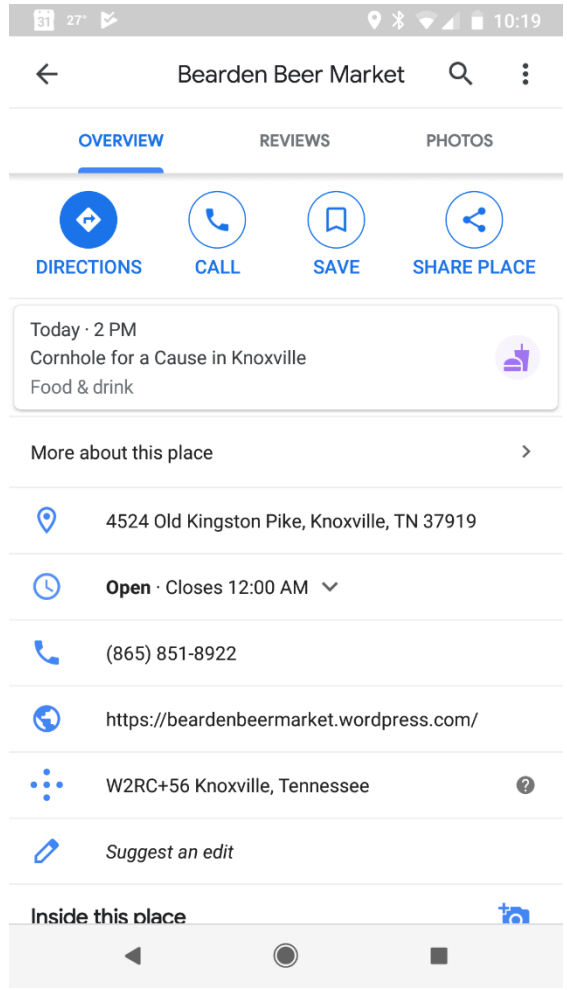




Modern day world

- Google
- Facebook
- Yelp
- Netflix
- Uber
- Lyft
- …



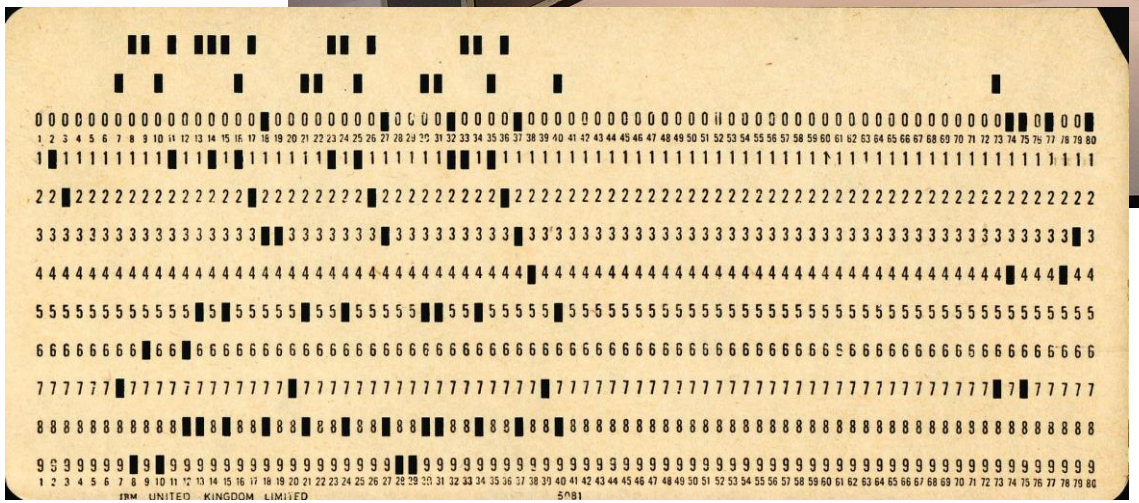


What was science like before 80ies?

- Computers existed only for specialized applications
- Many years of training before you can use one
- … and even if you can, required much patience

BESM-6 Computer

Punch card

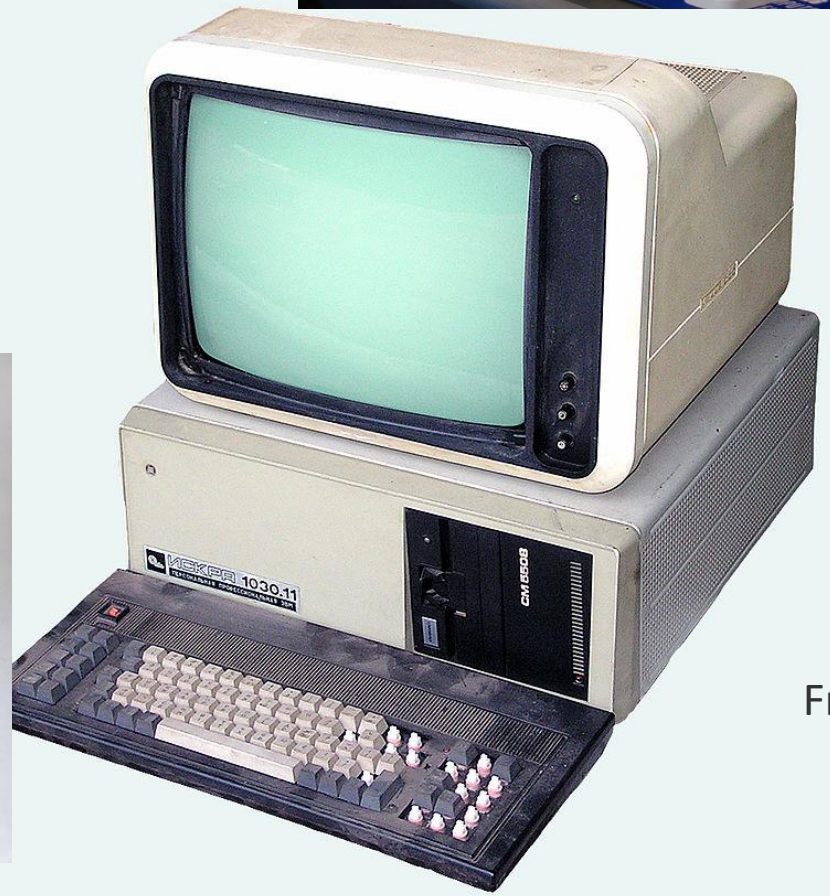


From Wikipedia

Altair(duino)

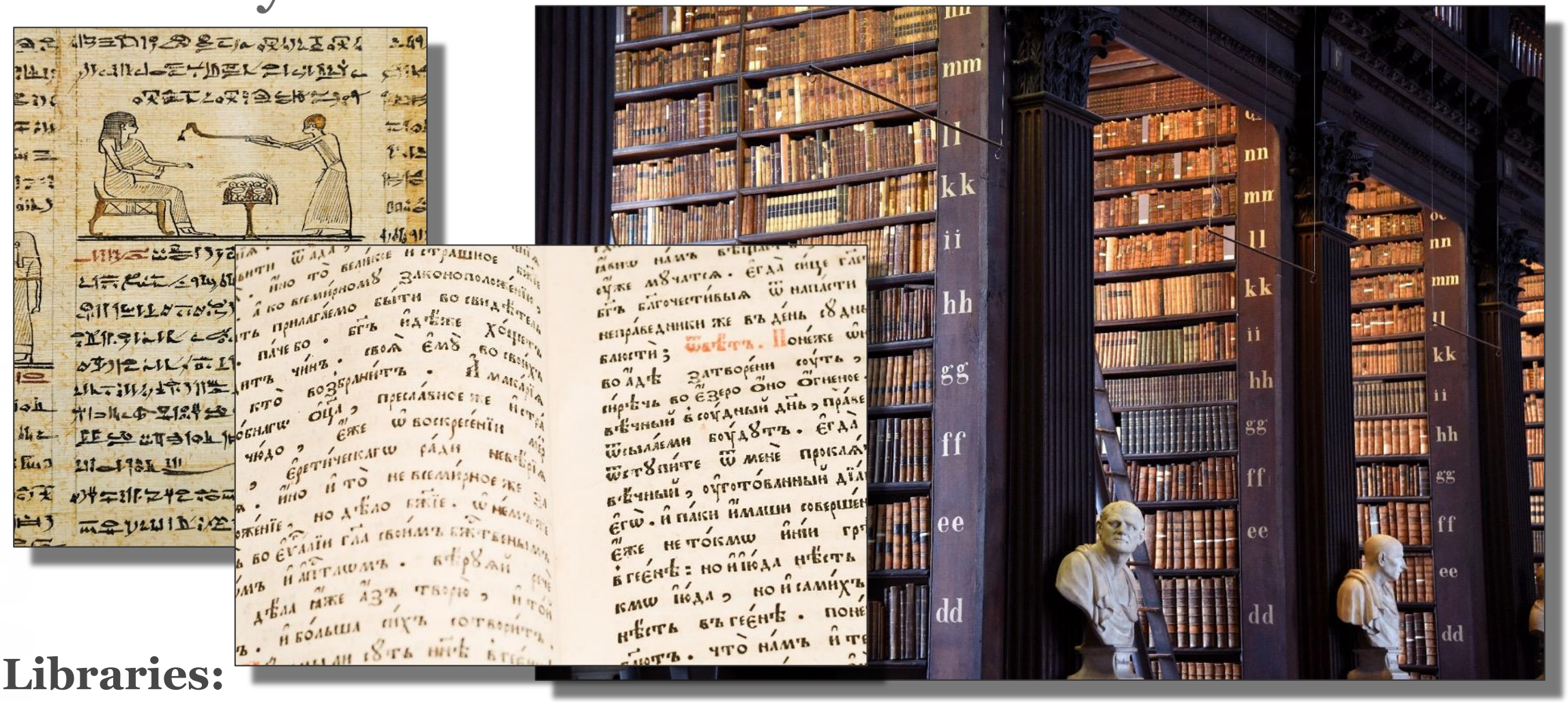
80ies

- First broadly available personal computers
- Can start programming (and get results) overnight
- First specialized scientific software
- What you have is what you bring



From Wikipedia

What if you want to know more?



Libraries:

- Annual abstract books to find papers
- Dewey system to find books
- Collections: Landolt-Bornstein, etc

For the last ~30 years:

Scientific information access:

- ISI and other data bases
- Electronic journals

General information

- Google and other search engines

Scientific software:

- Word
- Origin
- Digital micrograph

Programming languages

- MatLab
- Igor Pro

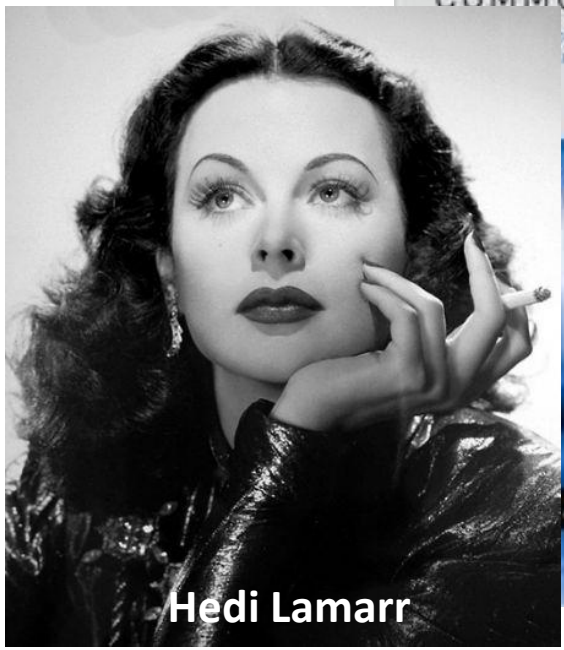
Instrument control (usually limited by manufacturer)

But relatively few changes since then….

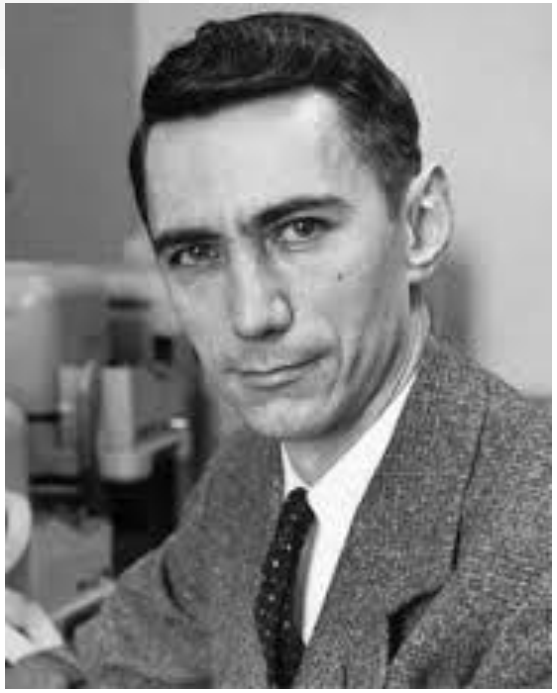
Could anyone have predicted it?

J. C. R. Licklider in 1965. A psycho-acoustician who saw computers as more than calculating machines, he was the first director of ARPA's Information Processing Techniques Office (IPTO). *(Photo courtesy of the MIT Museum)*

Ada Lovelace



Hedi Lamarr



Claude Shannon

Noel Bonnet

What is happening now?

THE UNIVERSITY of TENNESSEE KNOXVILLE

New data technologies

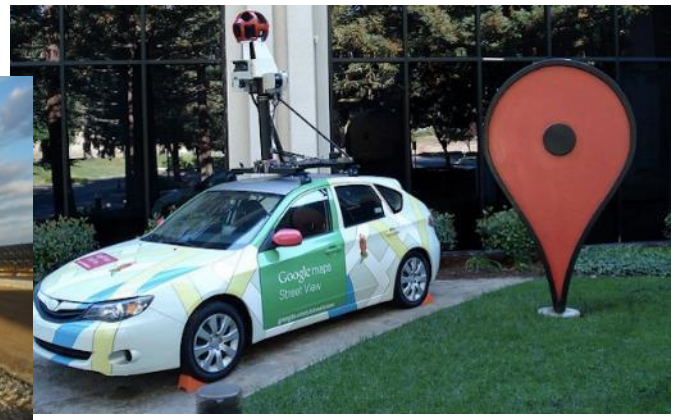
- Searches
- Social networks
- Recommender engines
- Connection to real world
- Large Language models

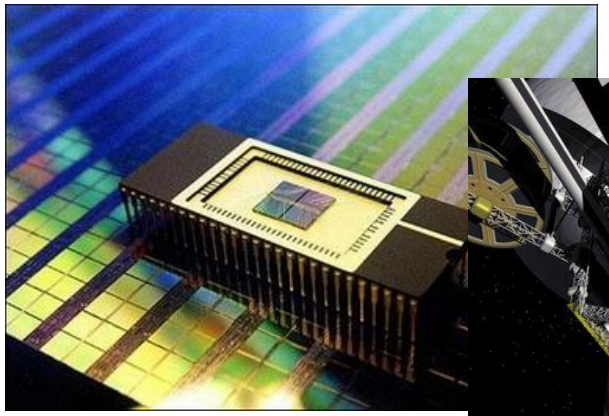
New opportunities:

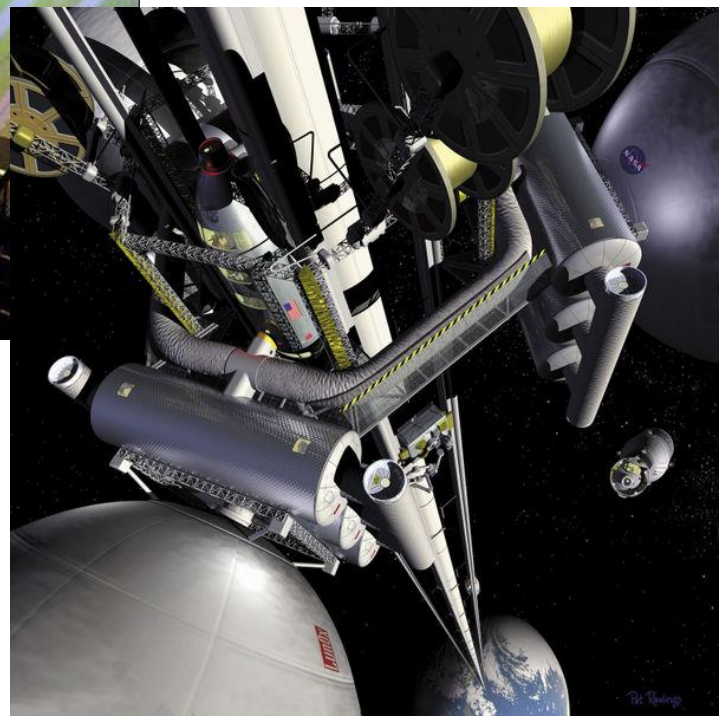
- 3D Printing
- IoT devices
- Laboratory robotics
- Open code
- Text analytics

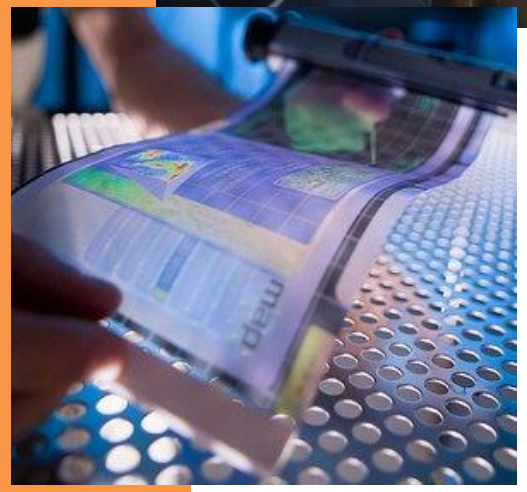
Colab – LLM for MD discovery

The World is Material Opportunity









Predicting crystal structure by merging data mining with quantum mechanics

CHRISTOPHER C. FISCHER[1], KEVIN J. TIBBETTS[1], DANE MORGAN[2] AND GERBRAND CEDER[1]*

[1]Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

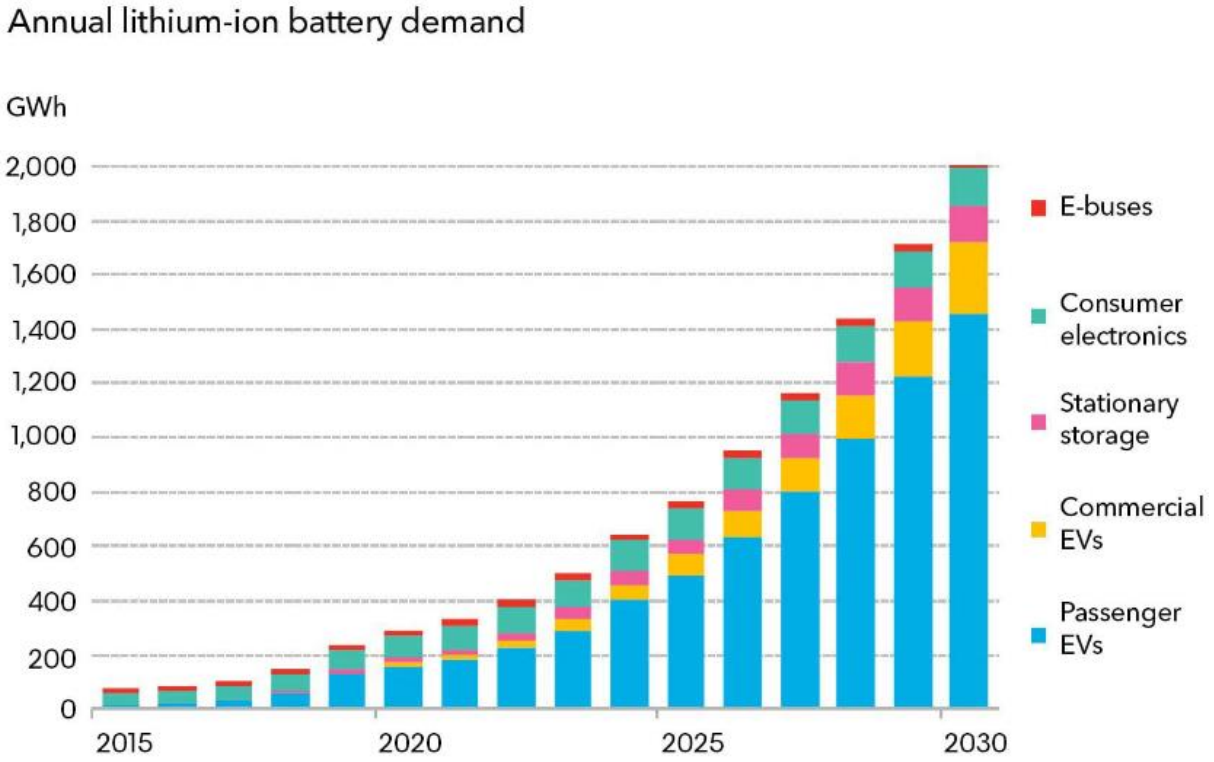
[2]Department of Materials Science and Engineering, University of Wisconsin, Madison, Wisconsin 53706, USA

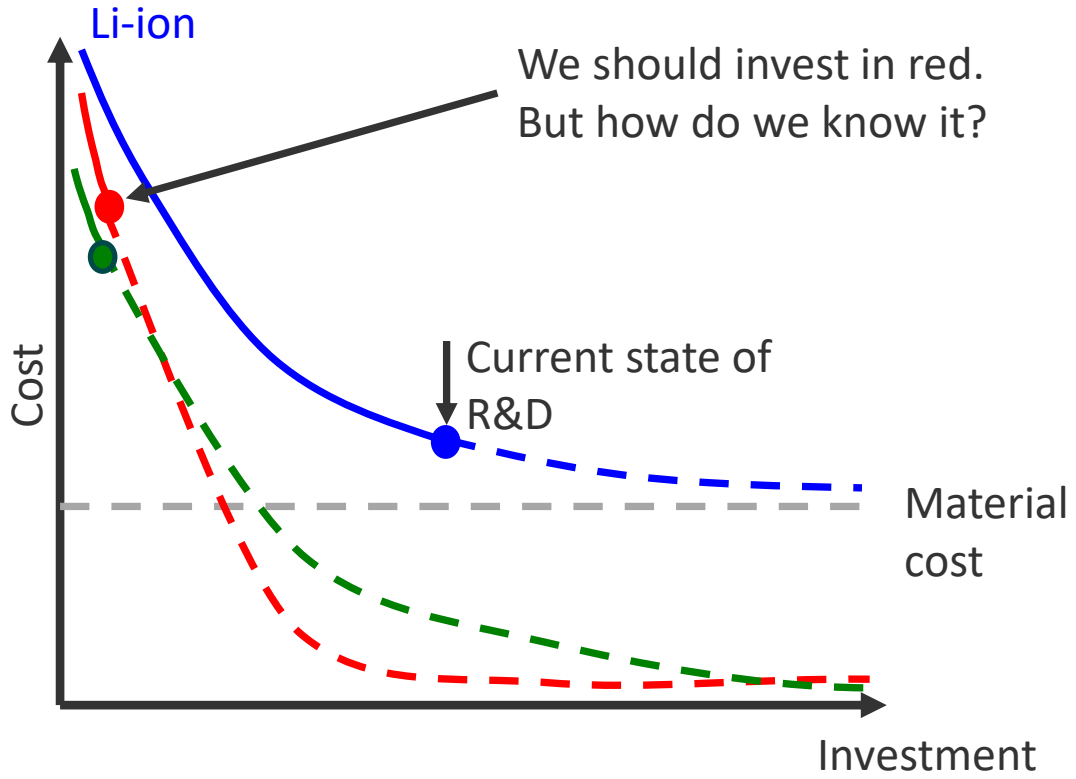
*e-mail: gceder@mit.edu

- **"Improve":** Renewable energy, self-driving cars, transparent displays, memory technologies
- **"Discover":** Room temperature superconductivity, high mechanical stress materials
- **"Engineer":** Quantum computing, single-atom catalysts, biomolecules

Functionality, manufacturability, cost

Batteries: Li-ion and Beyond

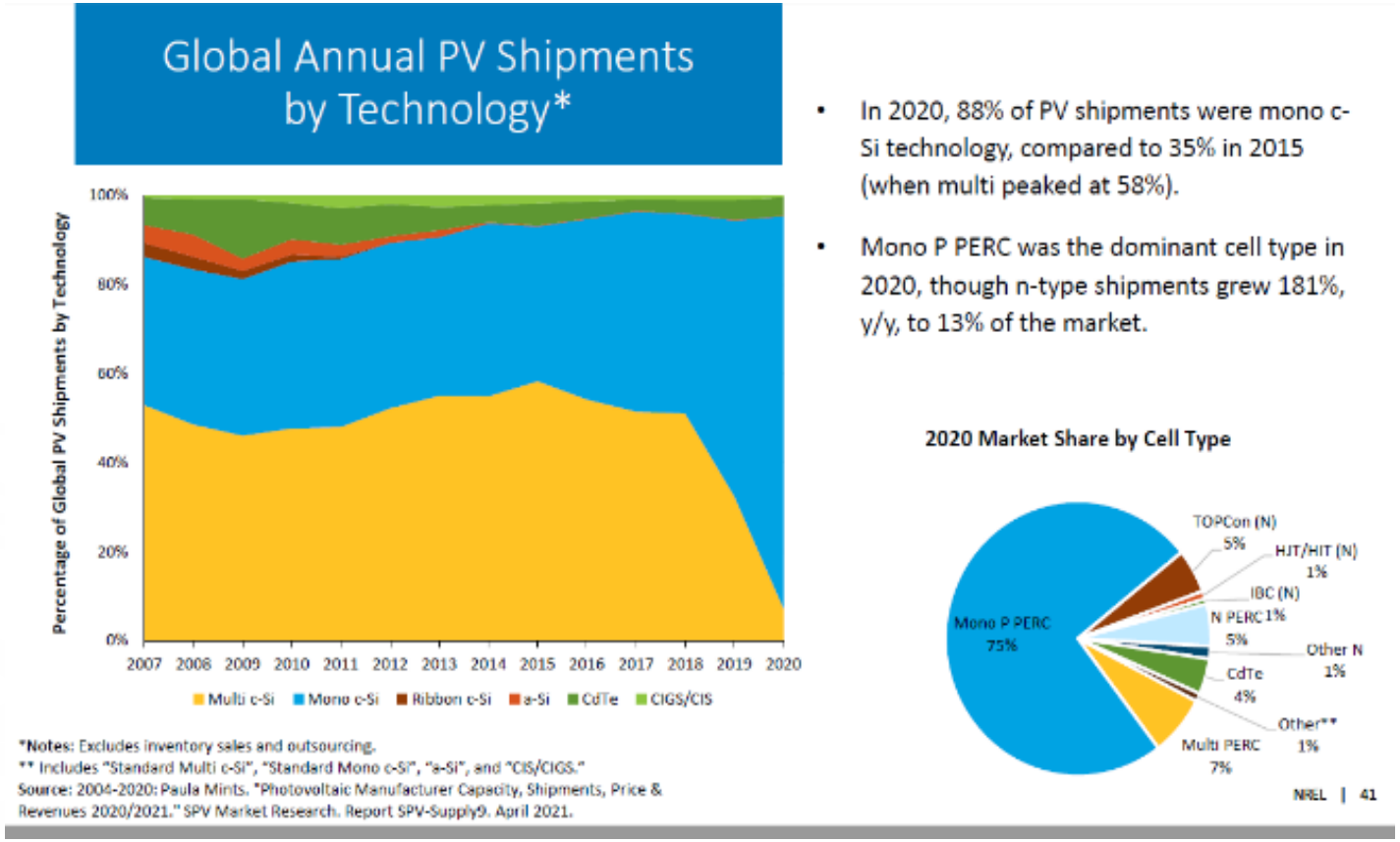




- Batteries are required element of energy transition (EVs, ESS, mobile devices)
- Currently Li-ion is the primary technology
- Optimization of Li-ion batteries takes years (even with same process on new Gigafactory)
- However, it is far from Goldilock zone for ESS or energy transport

- How can we optimize usage and safety for Li-ion batteries in EVs?
- How do we select beyond Li technologies for ESS?

Solar Energy: Will Silicon Ever Reign?



- In 2020, 88% of PV shipments were mono c-Si technology, compared to 35% in 2015 (when multi peaked at 58%).
- Mono P PERC was the dominant cell type in 2020, though n-type shipments grew 181%, y/y, to 13% of the market.

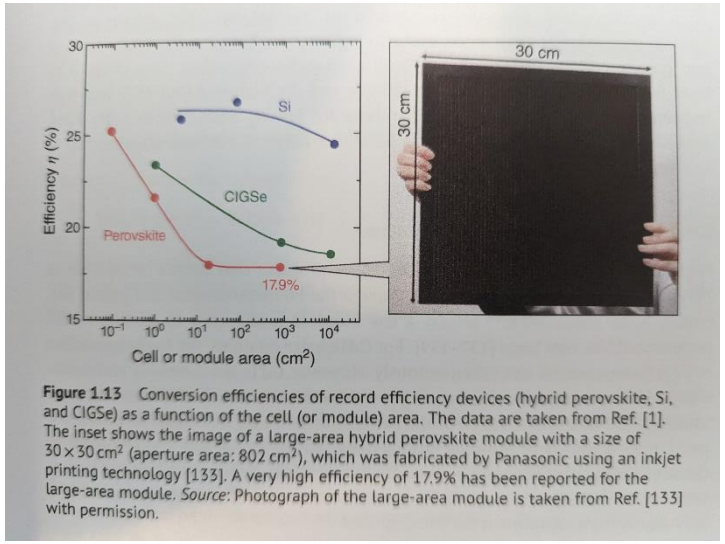
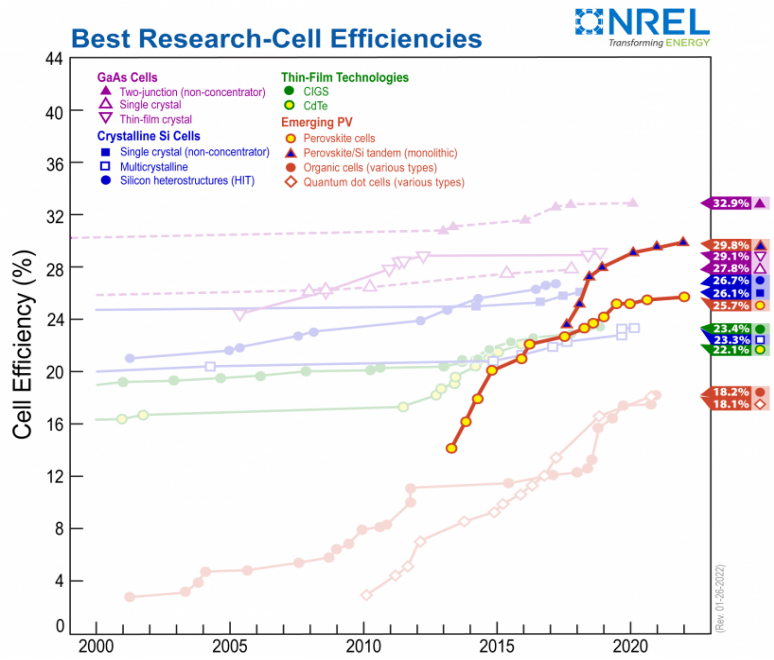
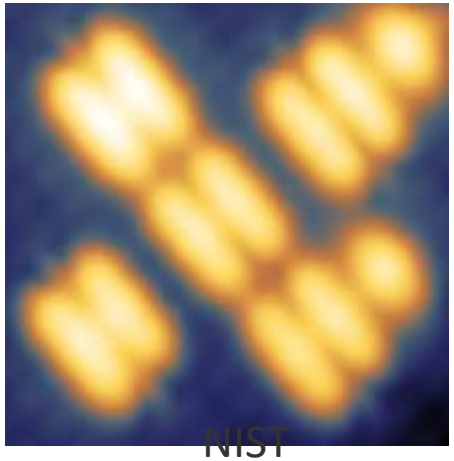


Figure 1.13 Conversion efficiencies of record efficiency devices (hybrid perovskite, Si, and CIGSe) as a function of the cell (or module) area. The data are taken from Ref. [1]. The inset shows the image of a large-area hybrid perovskite module with a size of $30 \times 30\ cm^2$ (aperture area: $802\ cm^2$), which was fabricated by Panasonic using an inkjet printing technology [133]. A very high efficiency of 17.9% has been reported for the large-area module. *Source*: Photograph of the large-area module is taken from Ref. [133] with permission.



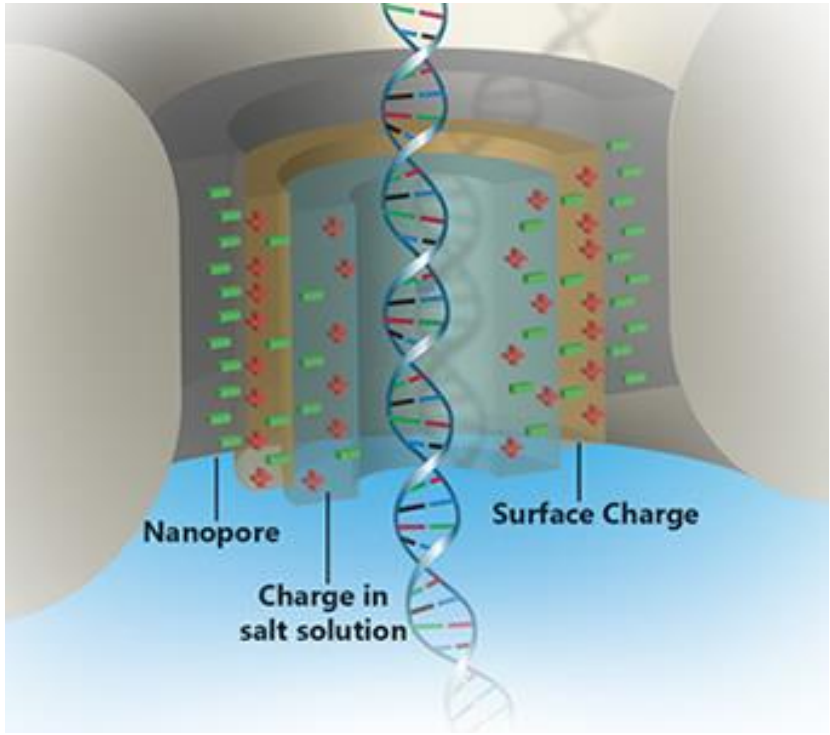
- Solar energy is the fastest growing energy sector
- Si is now reining material – however, it is really not the optimal material for PV (heavy, expensive)!
- Hybrid perovskites can be used as ideal PV materials – if we can make them stable and scale manufacturing!

Quantum Computing and Single Molecule Bio



NIST

Zyvex Labs



Oxford Nanopore

- Direct atomic fabrication: quantum communications and quantum computing, environmental sensing
- Single-molecule biological devices

- Success story 1: cryo-electron microscopy
- Success story 2: nanoelectron diffraction

Instrumentation

General synthesis/characterization:

- Multiple data generation tools
- Complex workflows
- 100s at each university in US

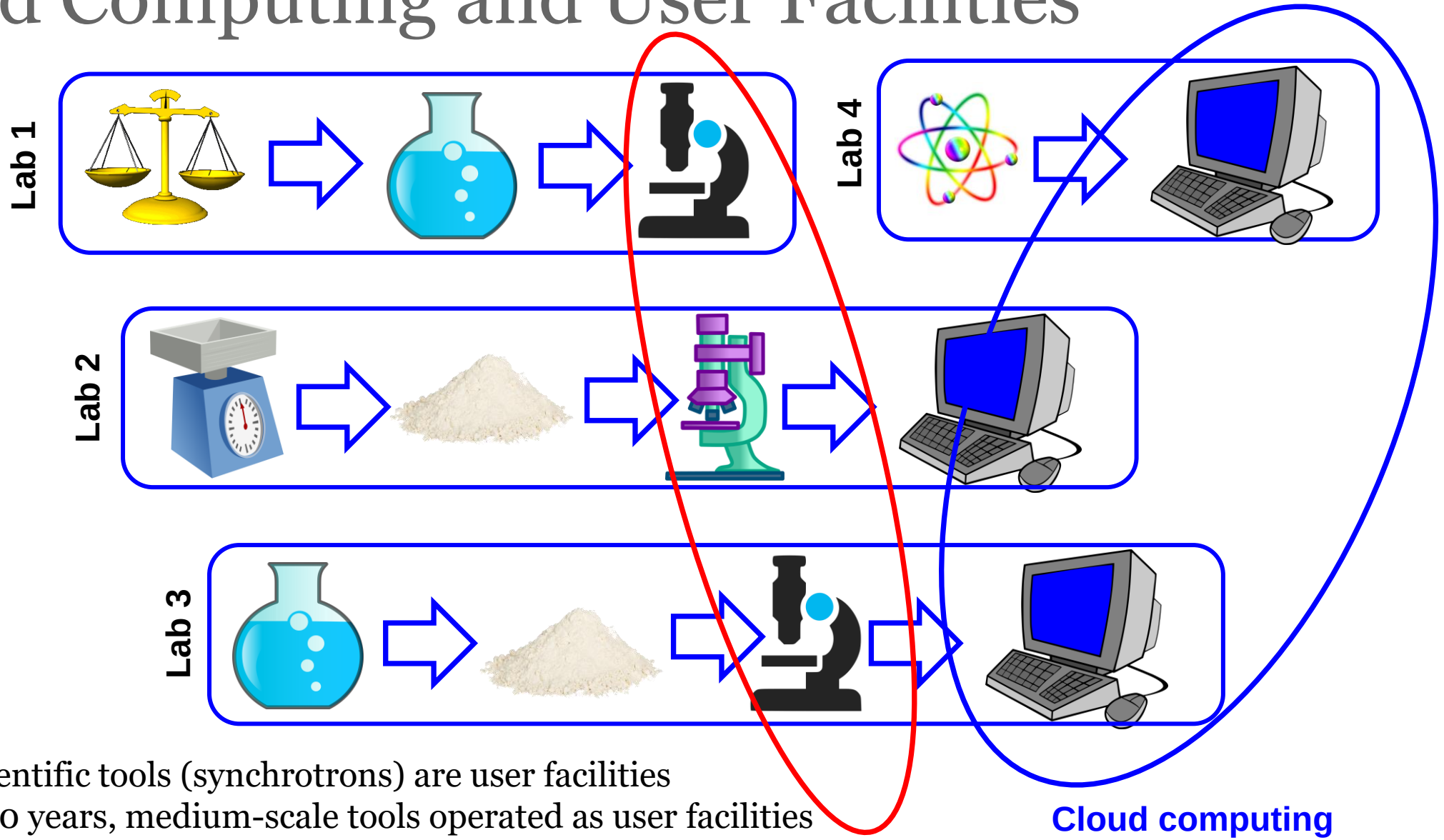
Surface science lab:

- Multiple data generation tools
- Complex controls and workflows
- 10s in each university

Electron microscope:

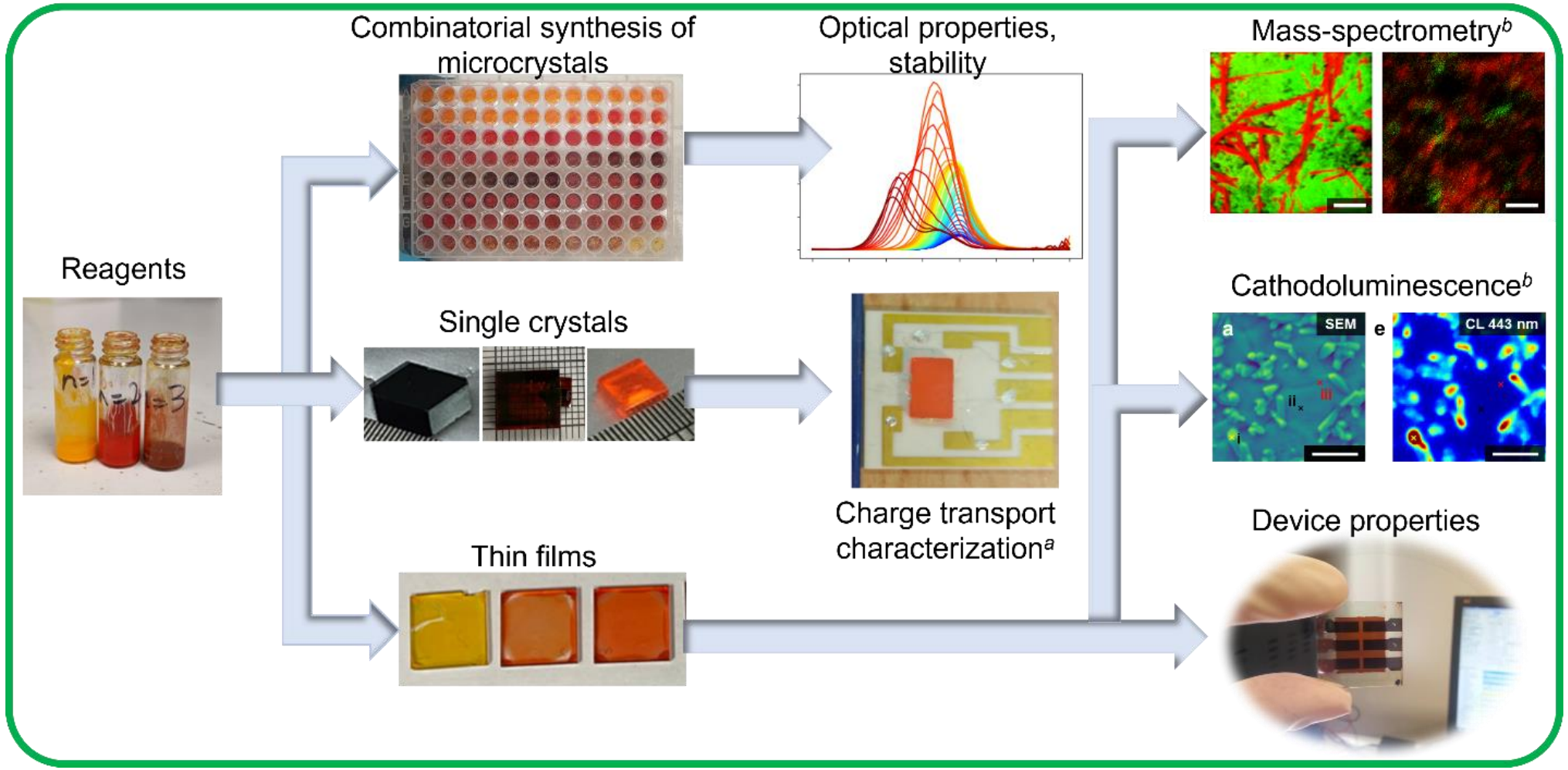
- >100k worldwide
- Can cost up to ~4-5$M
- Can generate data at the ~10GB/s

Cloud Computing and User Facilities



- Big scientific tools (synchrotrons) are user facilities
- For ~20 years, medium-scale tools operated as user facilities
- Enterprise computing -> cloud computing
- Over last 5 years, cloud labs are emerging
- What about the workflows?

What is A Workflow?



- **Workflow:**
- Ideation, orchestration, implementation
- Domain specific language
- Dynamic planning: latencies and costs
- Reward and value functions

Designed in academia and adopted by industry

- Are they optimal?
- Can we design them better?
- Can they be changed dynamically?

Value Proposition

1. You are interested in ML and AI and would like to try it hands-on on real world problems from materials science and microscopy

2. Learn the basics of the ML methods and build upon this knowledge - from simple principal component analysis to large language models.

3. Explore how ML is being adopted by industry - from IT leaders such as Amazon, Google, and Meta to instrumental, chemical, and materials companies.

4. Learn why next decade of ML will be transition from purely in-silico to real-world materials and device applications, and be a part of this transition.

5. And learn to work backwards from real-world problem to solution.

Prerequisites

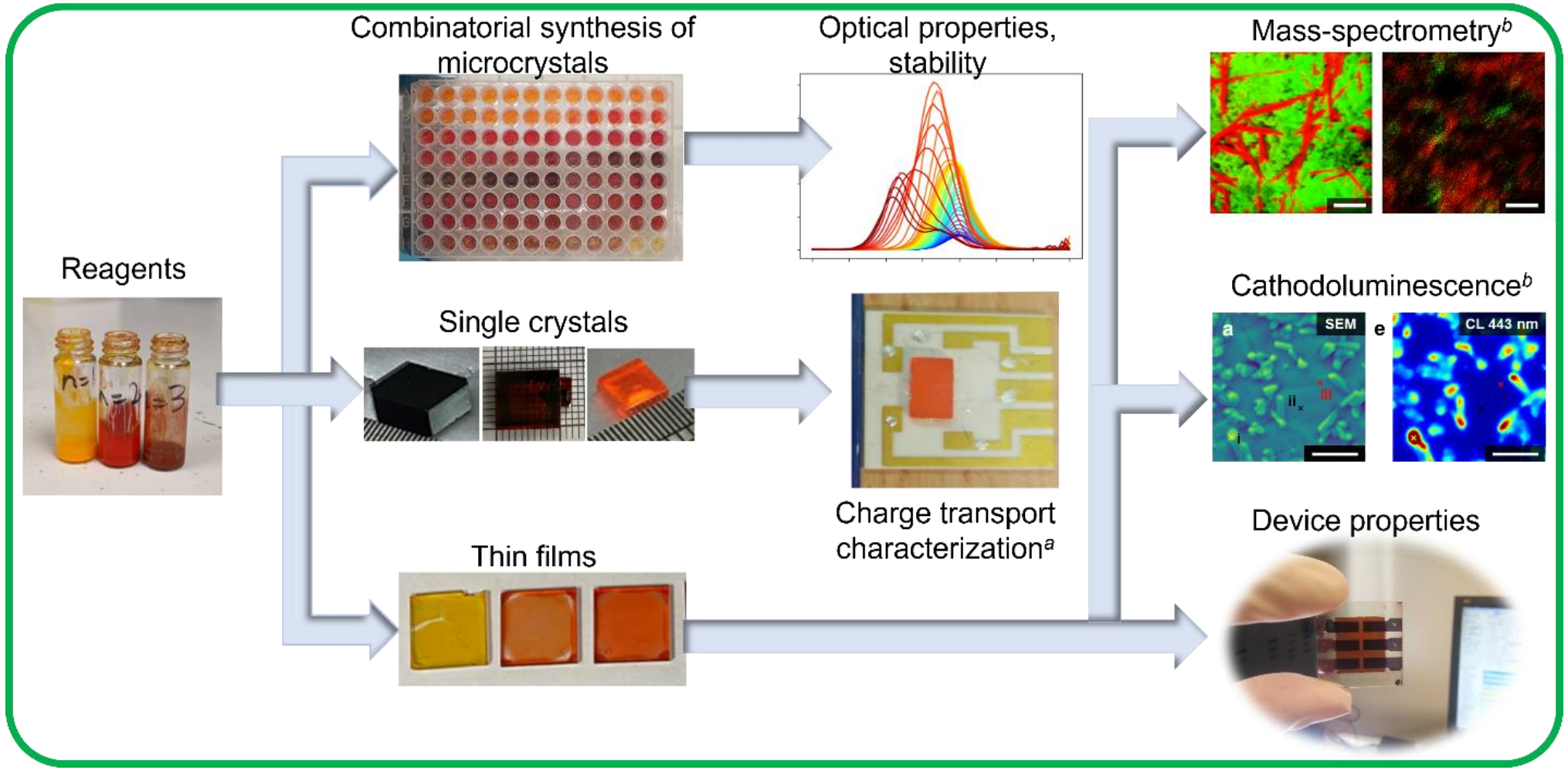
To be successful in this course you will need a general background in materials science. Python or similar programming experience, while not essential, will be extremely useful. Students without any prior programming experience should expect to spend extra time outside of class learning basic skills.

Reference Materials

I will provide copies of lecture notes, presentations, and Colabs on GitHub. There is no specific textbook and we will take material from a variety of sources including:

- Andrew Bird et al, *Python Workshop – Second Edition*, https://subscription.packtpub.com/book/programming/9781804610619/1
- Sebastian Raschka, *Machine Learning with PyTorch and Scikit-Learn*, https://subscription.packtpub.com/book/data/9781801819312/1
- Rowel Atienza, *Advanced Deep Learning with TensorFlow 2 and Keras - Second Edition*, https://www.packtpub.com/product/advanced-deep-learning-with-tensorflow-2-and-keras-second-edition/9781838821654

What is A Workflow?



- **Workflow:**
- Ideation, orchestration, implementation
- Domain specific language
- Dynamic planning: latencies and costs
- Reward and value functions

Designed in academia and adopted by industry

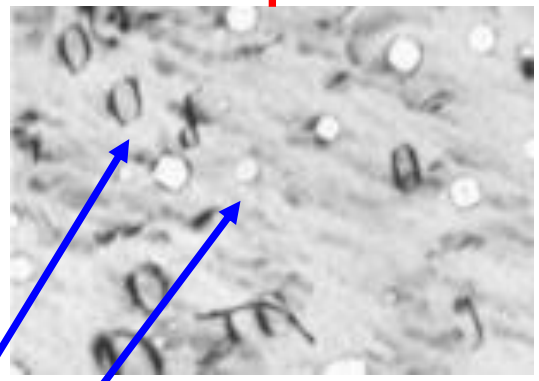
- Are they optimal?
- Can we design them better?
- Can they be changed dynamically?

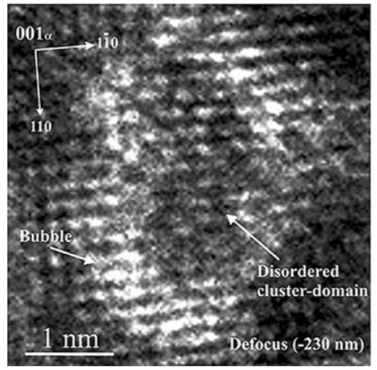
Workflows for Nuclear Materials Design

Traditional experiment:

1. Always based on workflows
2. Ideated, orchestrated, and implemented by humans
3. The “gain of value” during the workflow implementation is uncertain

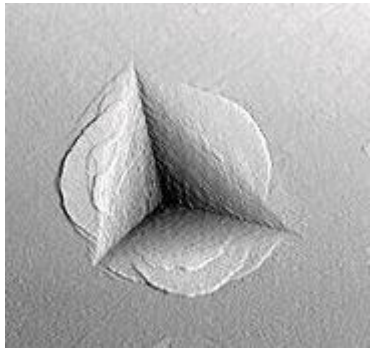
Materials selection





Prior knowledge

Operator curiosity



Value

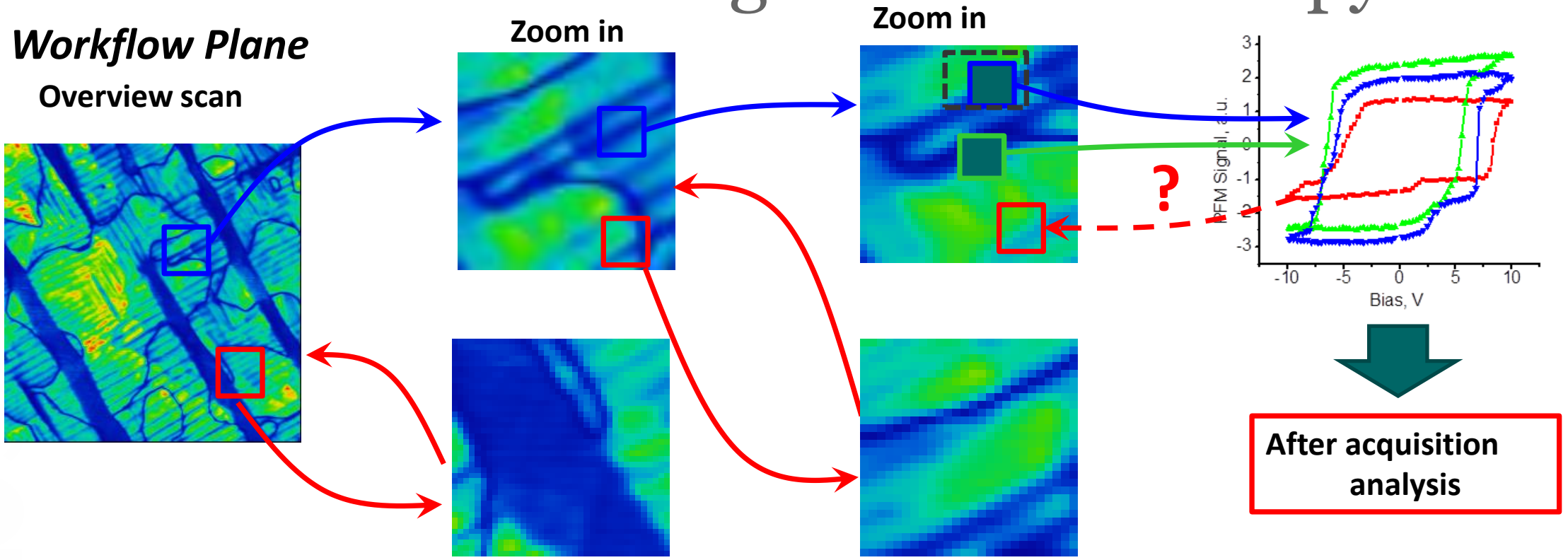
Value of the step is key element:

- Either based on prior knowledge
- Or defined in a sense of the reinforcement learning Q-function

Workflows in Scanning Probe Microscopy

Workflow plane

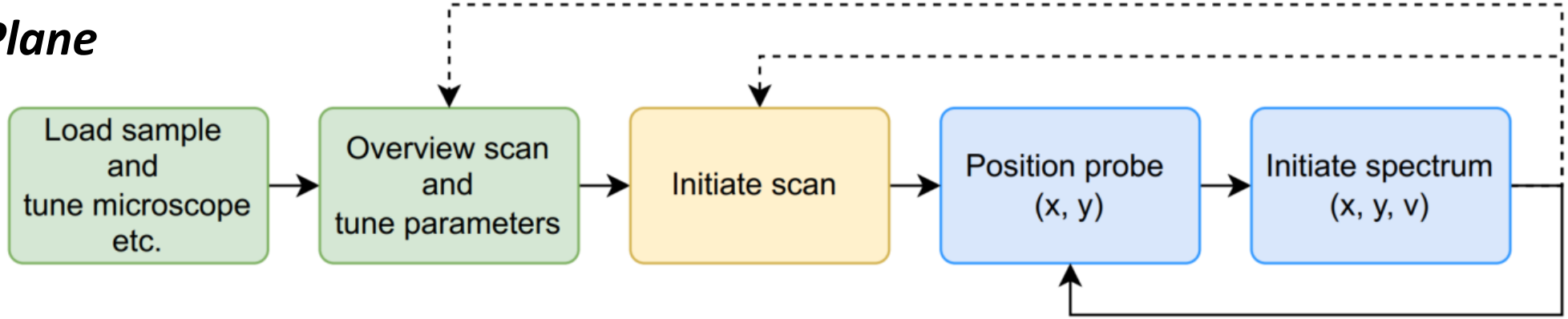
Workflow Plane



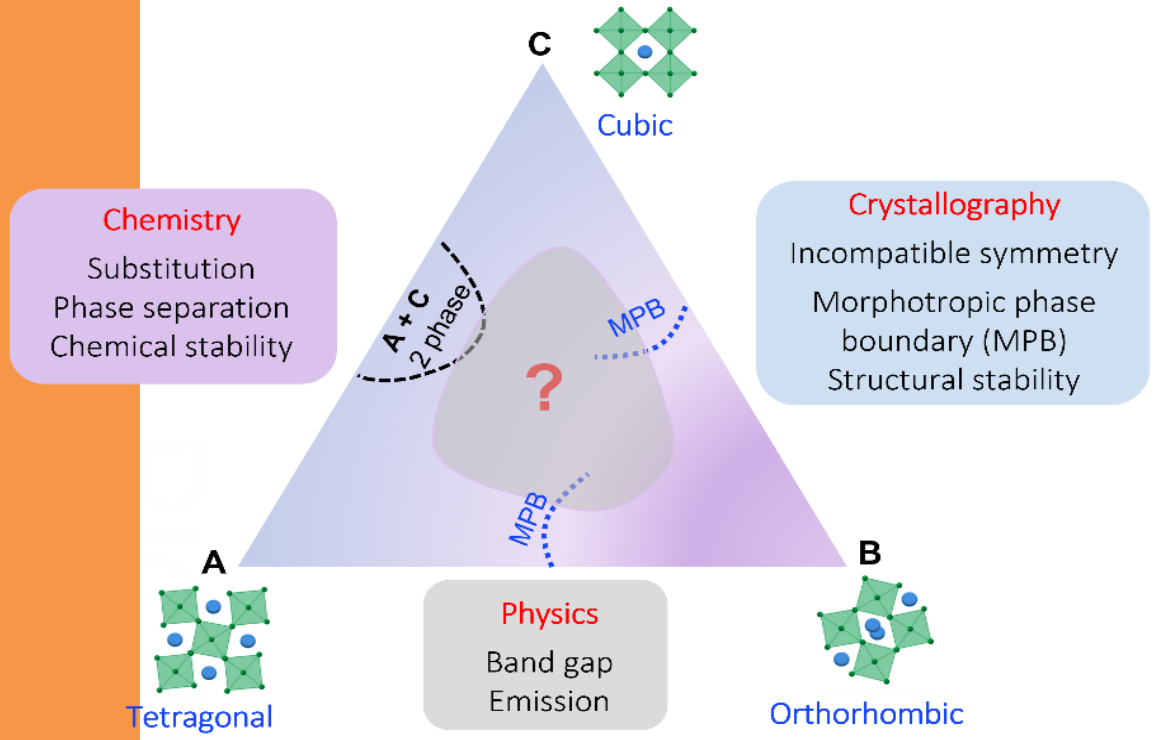
Instrument plane

Instrument Plane

Minimal instruction set control language



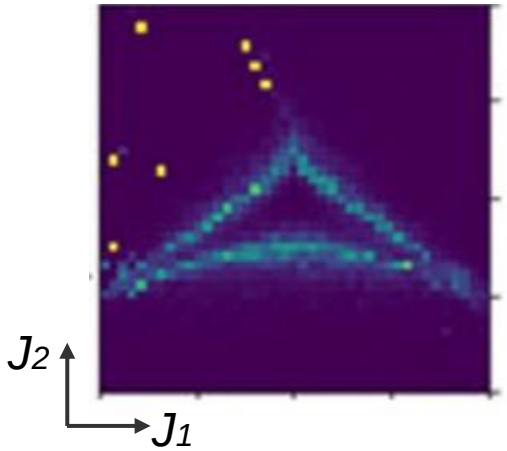
Why synthesis (or theory)?

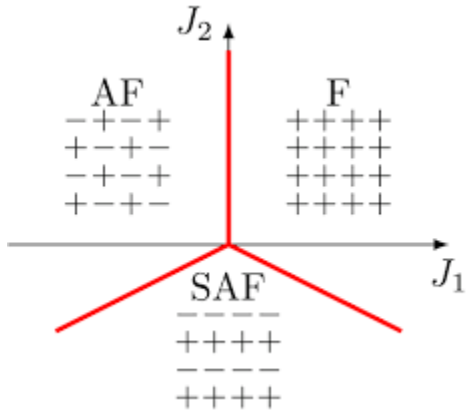


- Automated synthesis in its simplest form requires some way to navigate phase diagrams
- In more complex form, processing space.
- Ideally, incorporate physical knowledge

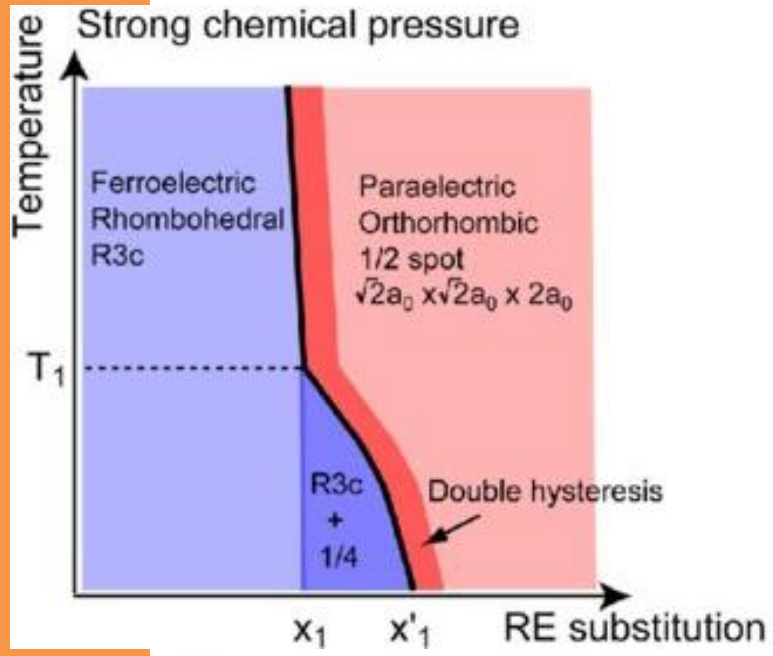
- Similar problem - theory

Ising model

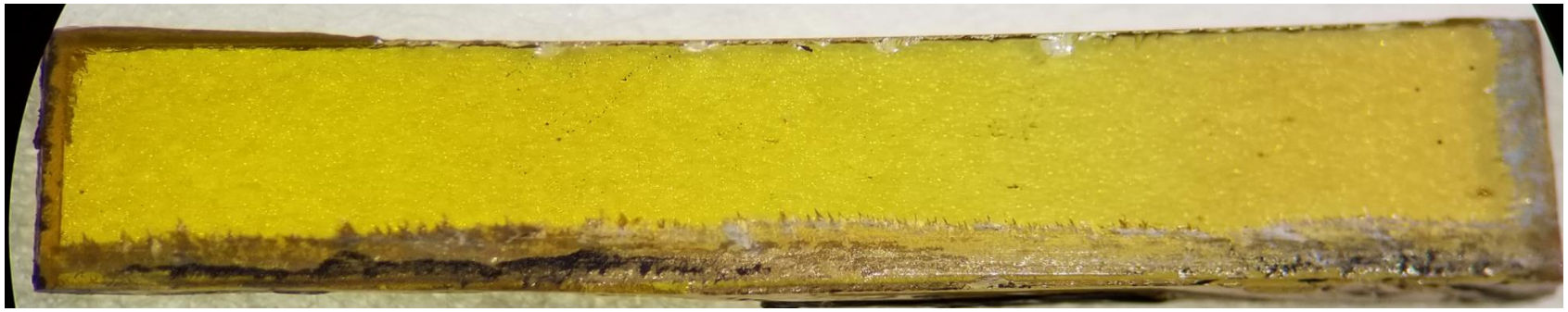




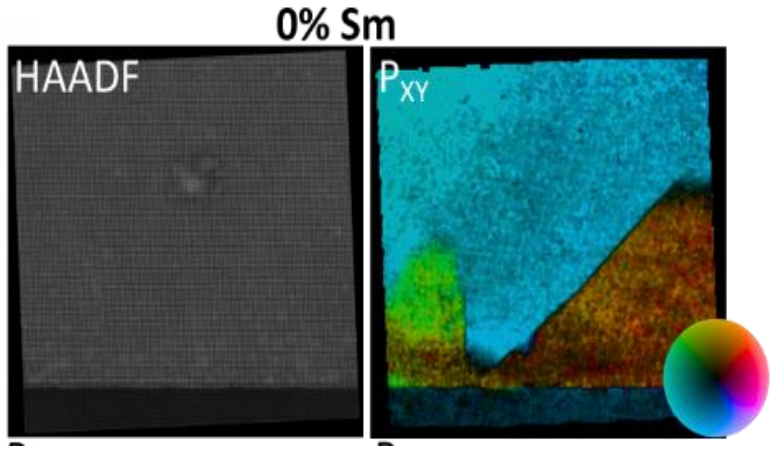
Combinatorial Synthesis

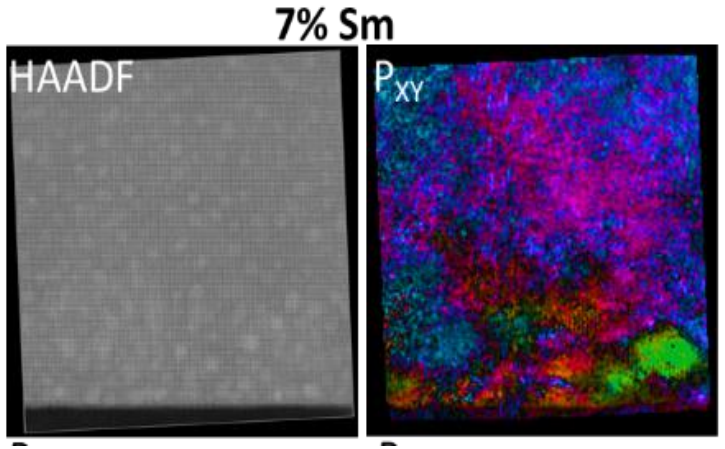


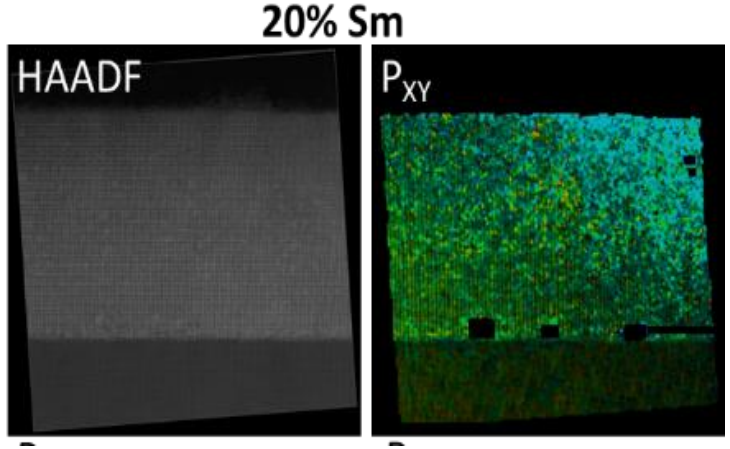
Sample by I. Takeuchi, UMD
Phase diagram by N. Valanoor et al.



$BiFeO_3$ Linear est. 7%Sm $BiFeO_3$ 20%Sm $BiFeO_3$







arXiv:2004.11817

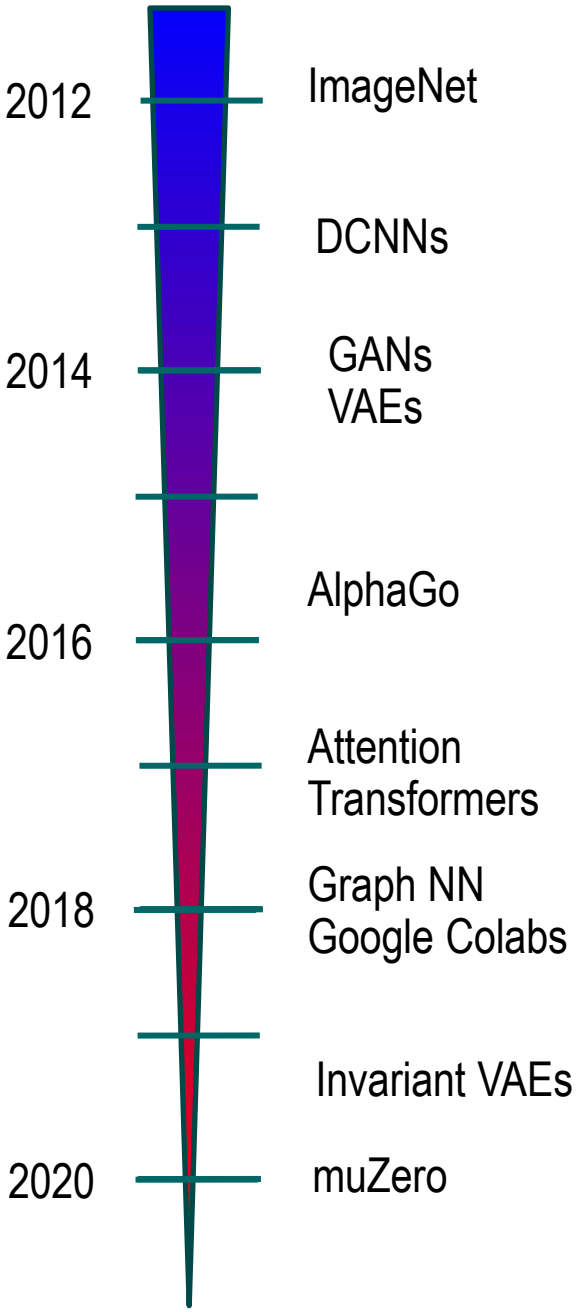
Why Machine Learning?

- Last decade has experienced an explosive growth of machine learning and artificial intelligence applications
- These developments have spanned areas from computer vision to medicine to autonomous systems and games
- However, the progress and impact as applied to experimental physical sciences has been minimal….

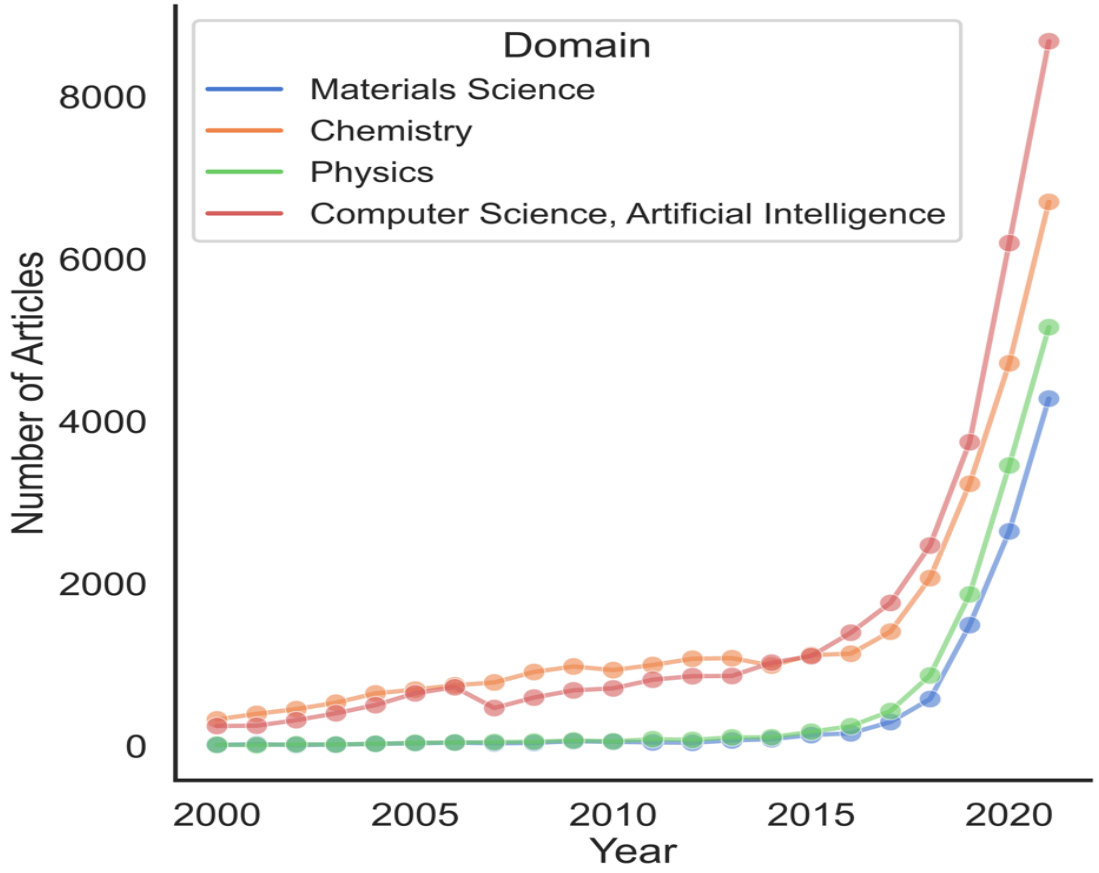
Why is it difficult?

- Requires domain expertise and domain-specific goals
- Deeply causal and hypothesis drive nature of domain sciences
- No single answer: culture, not a method
- Infrastructure, open code, open data
- **Most important:** active nature of scientific process

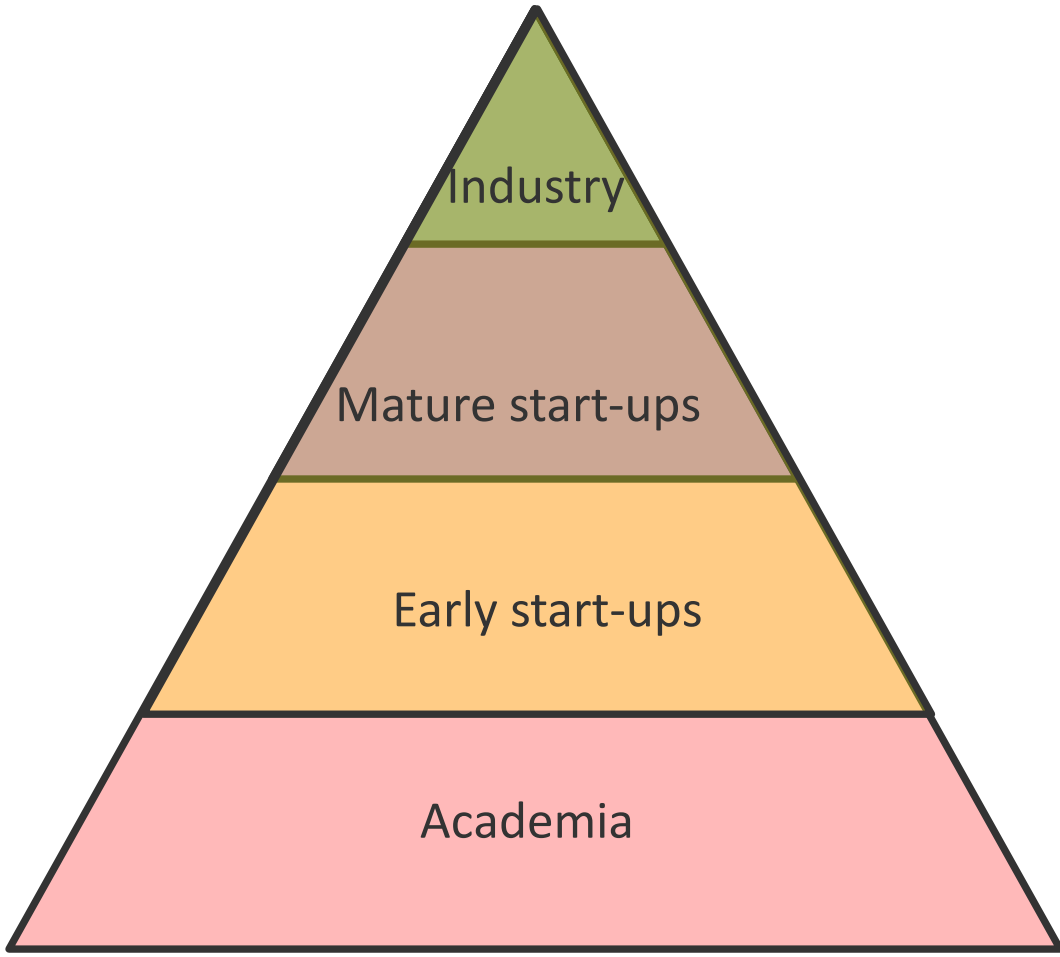
Microsoft: GitHub
Meta: Open Catalyst,
Meta: Papers with Code
Toyota: TRI
Google: AlphaFold
NVIDIA: protein folding



ML in Domain Sciences



Analysis by B. Blaiszik, Argonne



- The rapid adoption of ML in domain sciences and industrial R&D is a very recent trend
- Technologies and workforce emerge from academia into industry
- We can estimate potential growth rates comparing to cloud computing 15 - 20 years ago

“Eras” of ML in Industry

- **Before 2000:** It’s all about IT (dotcoms, Amazon, etc)
- **2000 - 2010:** It’s all about collecting and searching data (Facebook, Google, Uber)
- **2010 – 2020:** What do we learn from data (correlative era)
- **2020 – now:** Physics is the new data

- Classical machine learning is underpinned by the existence of the large static data sets – from MNIST to emerging medical, bio, faces, etc.
- Real world problems are associated with the large distribution shifts, often small data sets, and presence of uncontrollable exogenous factors
- Also, real world problems are often active learning: we interrogate the data generation process and provide feedback, not deal with static data sets
- However, we often have extensive prior knowledge of past data, physical laws generalizing them, and strong set of inferential biases

ML for real-world applications is different!

Approach

0. Getting big data: making imaging tools a part of data infrastructure

Physics: Why something happens

1. Big data:
How does
it happen?

- Unsupervised learning, clustering, and visualization
- **Biggest hurdle:** Language/ elementary tools

2. Deep data:
How can
we understand?

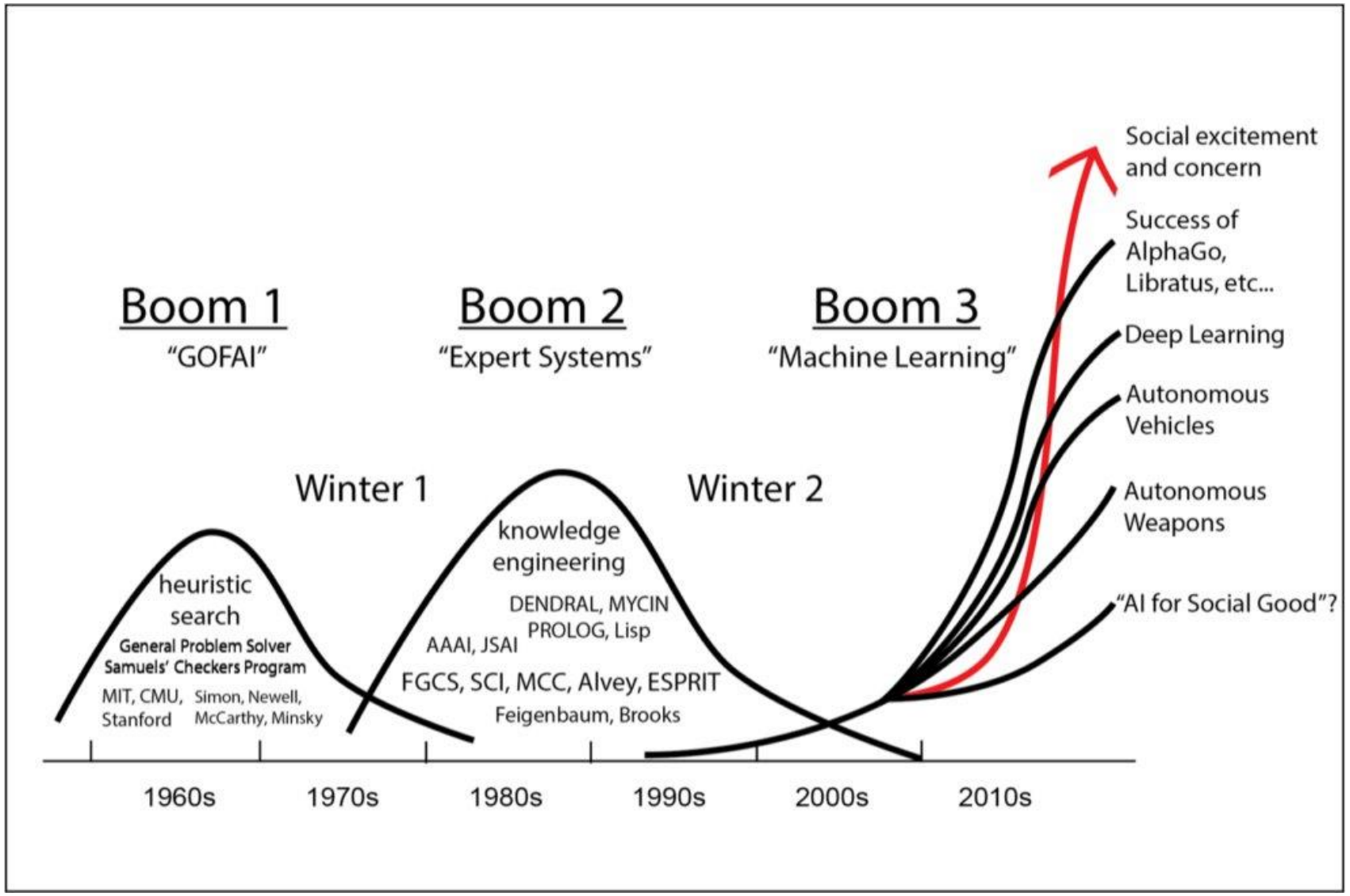
- Physics informed data analytics/ supervised methods
- **Biggest hurdles:** Mathematical framework, scalability of computational tools

3. Smart data: How can we do better?

- Feedback and expert/AI systems
- **Biggest hurdles:** With LLMs, it is possible

How it feels most of the time:

Zooming out on history



https://jaylatta.net/history-of-ai-from-winter-to-winter/

Types of Machine Learning

Supervised (inductive) learning
– Given: training data + desired outputs (labels)

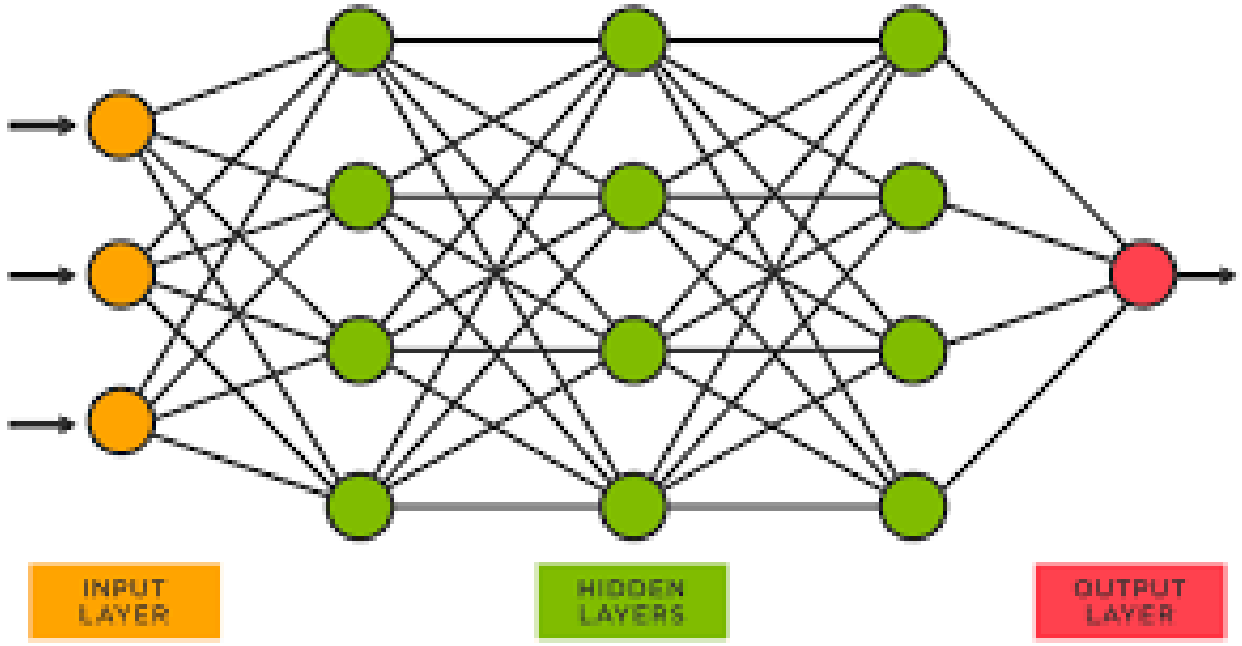
- **Unsupervised learning**
– Given: training data (without desired outputs)

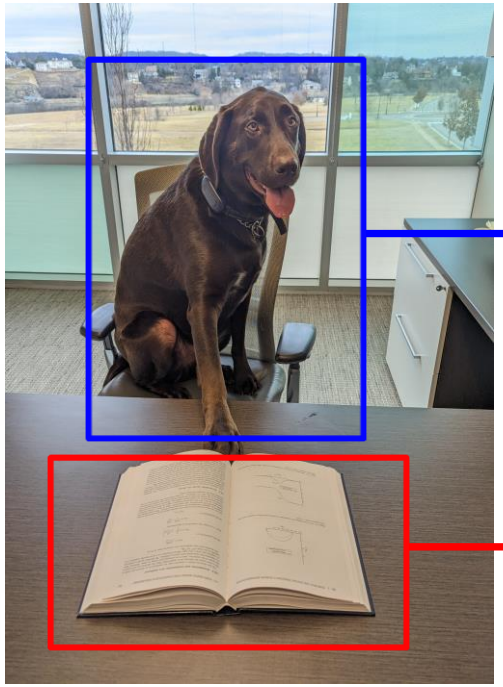
- **Semi-supervised learning**
– Given: training data + a few desired outputs

- **Reinforcement learning**
– Rewards from sequence of actions

Supervised Machine Learning

- Regression
- Classification
- Semantic segmentation
- Instance segmentation
- …





Dog

Book

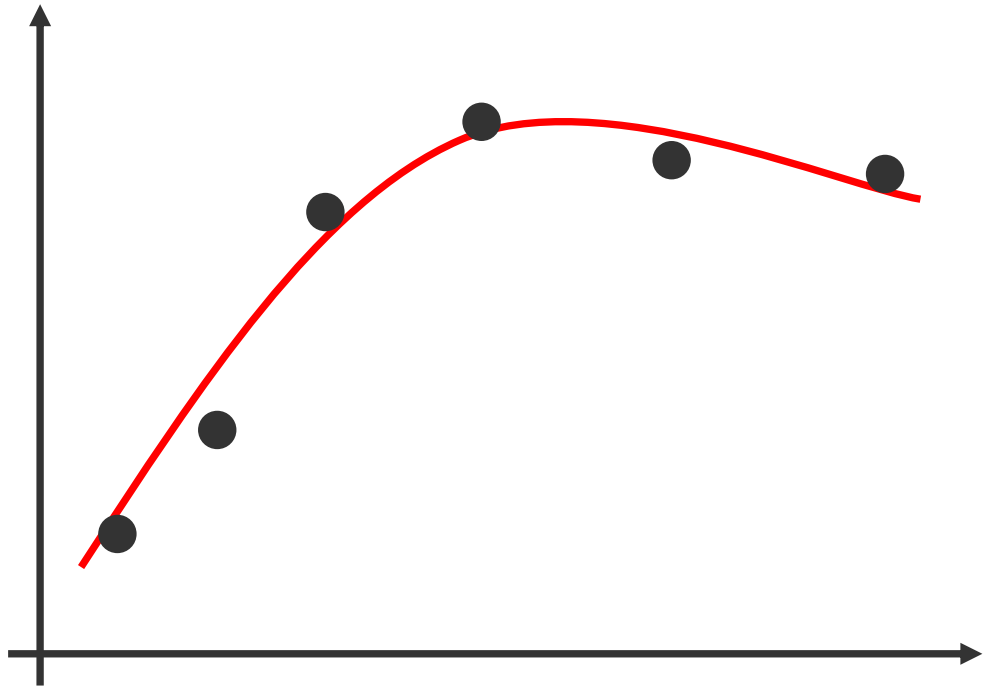
Classification

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function f(x) to predict y given x
- If y is categorical == classification

Application	Input Data	Classification
Medical Diagnosis	Noninvasive tests	Results from invasive measurements
Optical Character Recognition	Scanned bitmaps	Letter A-Z and digits 0-9
Protein Folding	Amino acid sequence	Protein shape (helices, loops, sheets)
Materials Discovery	Composition	Metal/Semiconducotr
Research Paper Acceptance	Words in paper title	Paper accepted or rejected

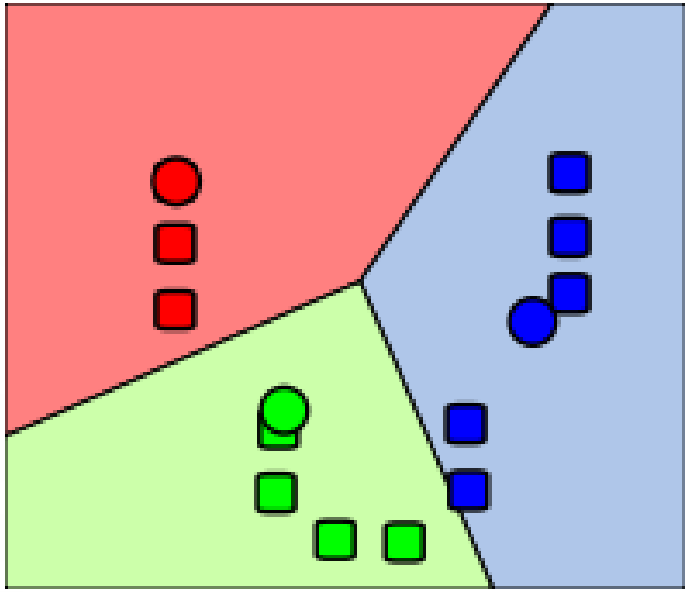
Regression

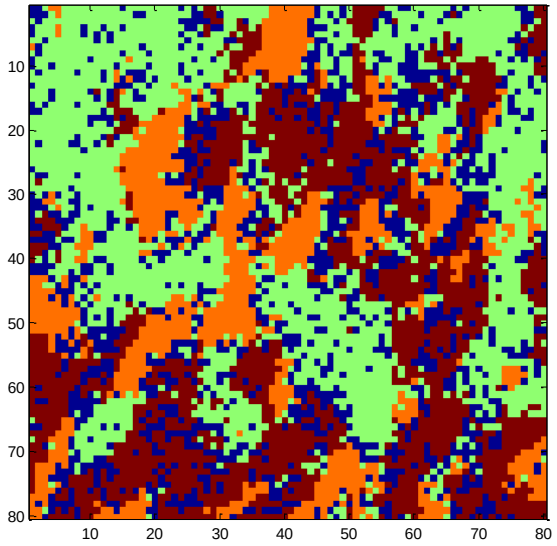
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function f(x) to predict y given x
- y is real-valued == regression

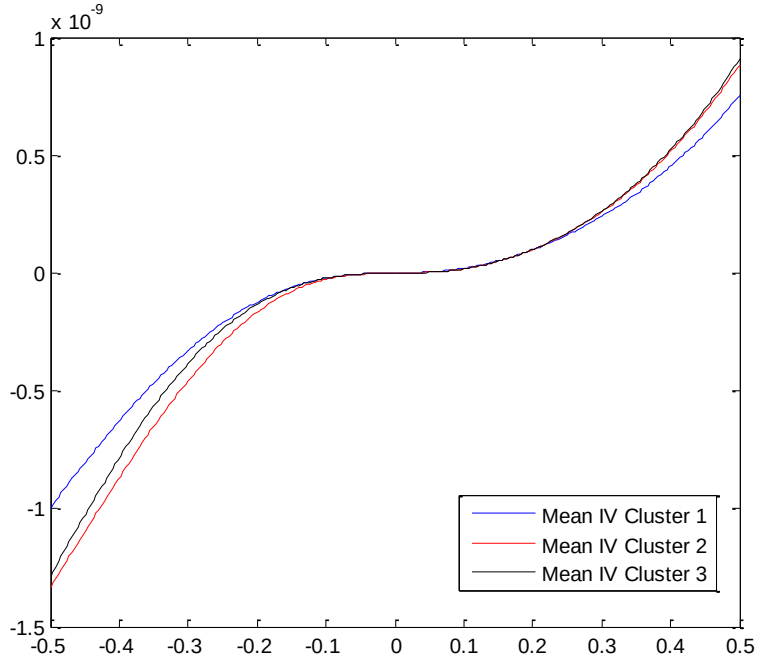


Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
- E.g., clustering



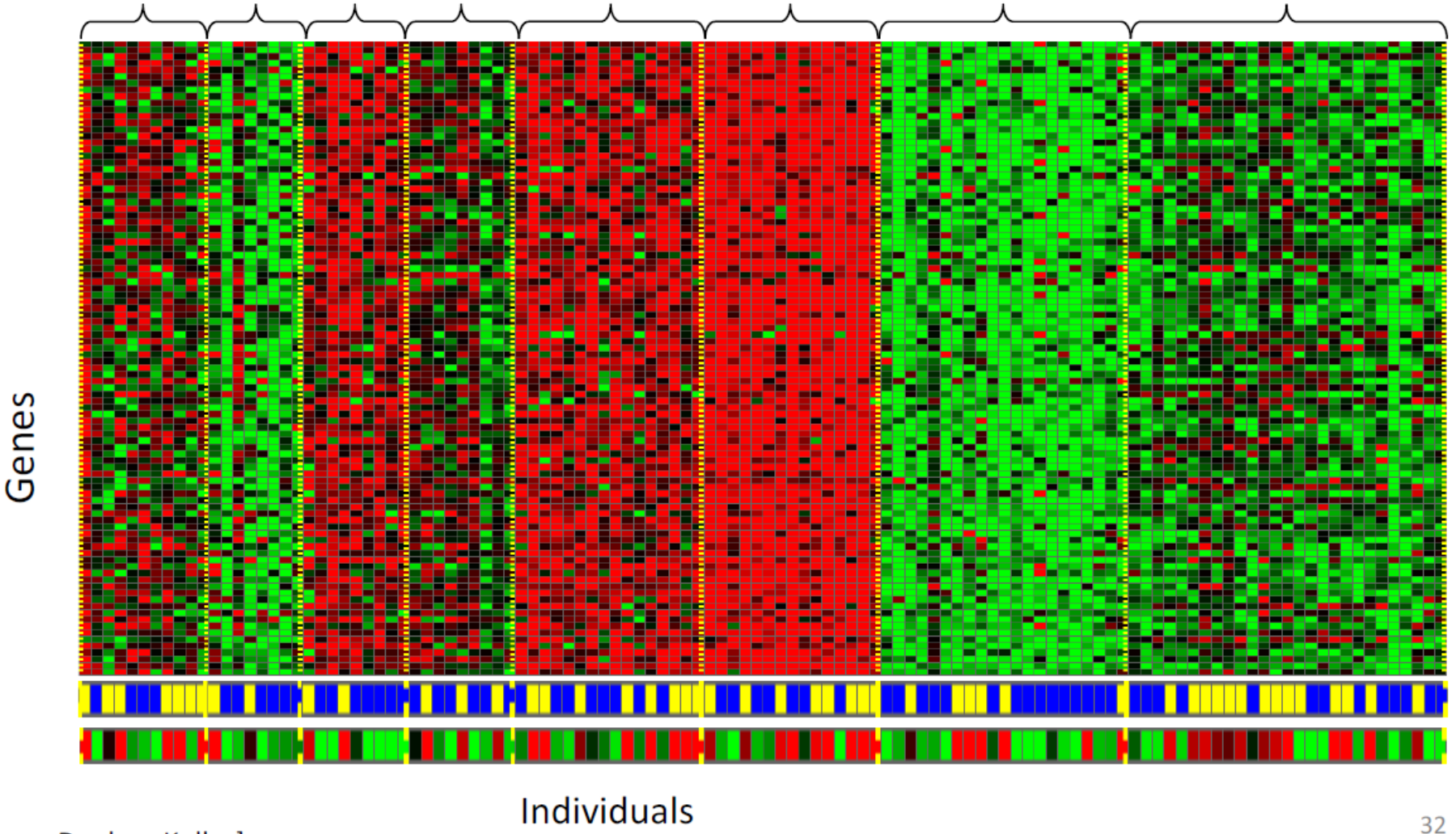




M. ZIATDINOV, A. MAKSOV, L. LI, A. SEFAT, P. MAKSYMOVYCH, and S.V. KALININ, *Deep data mining in a real space: Separation of intertwined electronic responses in a lightly-doped* $BaFe_2As_2$, Nanotechnology **27**, 475706 (2016).

Unsupervised Learning

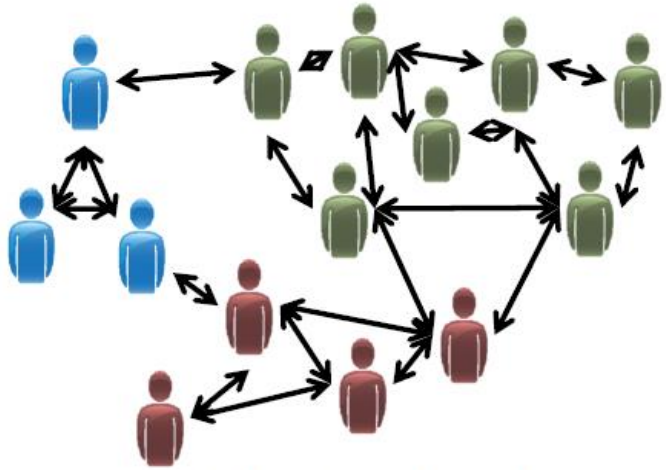
Genomics application: group individuals by genetic similarity



[Source: Daphne Koller]

Unsupervised Learning

Organize computing clusters



Social network analysis

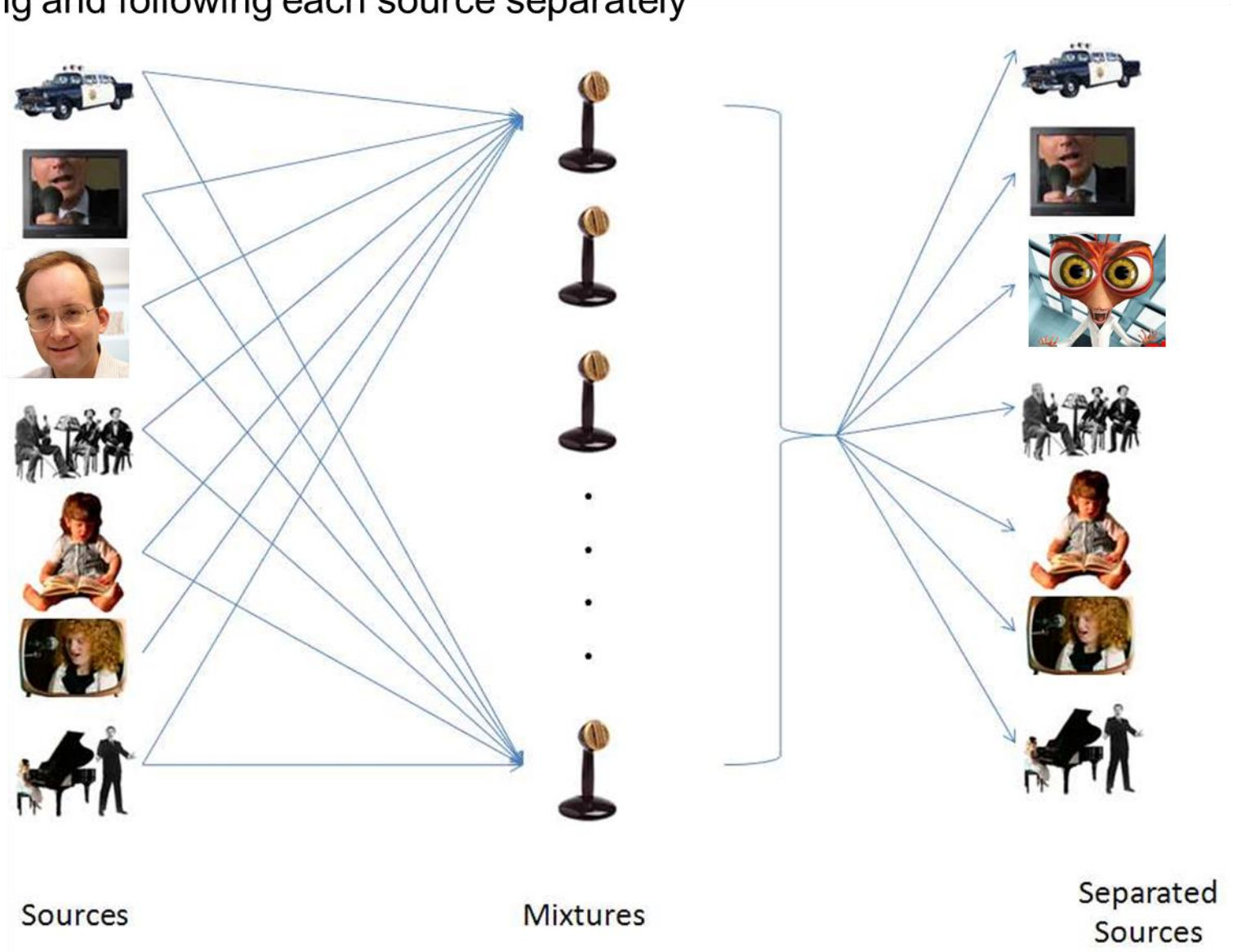
Market segmentation

Astronomical data analysis

Slide credit: Andrew Ng

Unsupervised Learning

Number of signals are being produced simultaneously; with the objective of separating and following each source separately

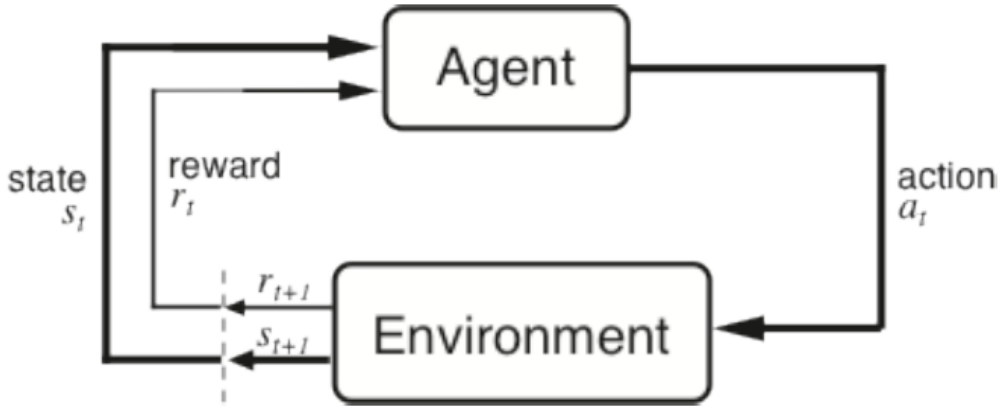


Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
- Policy is a mapping from states to actions that tells you what to do in a given state

• Examples:
– Credit assignment problem
– Game playing
– Robot in a maze
– Balance a pole on your hand

RL: Agent and Environment



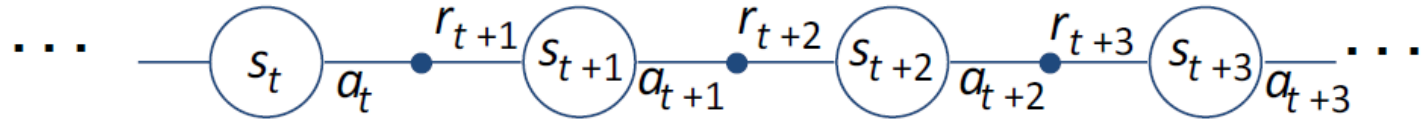
Agent and environment interact at discrete time steps : $t = 0, 1, 2, \mathrm{K}$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



Slide credit: Sutton & Barto

Reinforcement Learning in Action

https://www.youtube.com/watch?v=GtYlVxv0py8

Reinforcement Learning Applications

Chemical Synthesis and Drug Discovery

M. Ahmadi lab
UTK MSE

Cloud Laboratories

Emerald Cloud Lab,
SF and CMU

Build the case for machine learning

- Explore the business/scientific problem
- Build workflow (operations, costs, latencies)
- Identify bottlenecks
- Chart the solution
- Prototype the solution
- Test and iterate
- Deploy
- Support
- Upgrade
- Sunset

Identify the type of problem

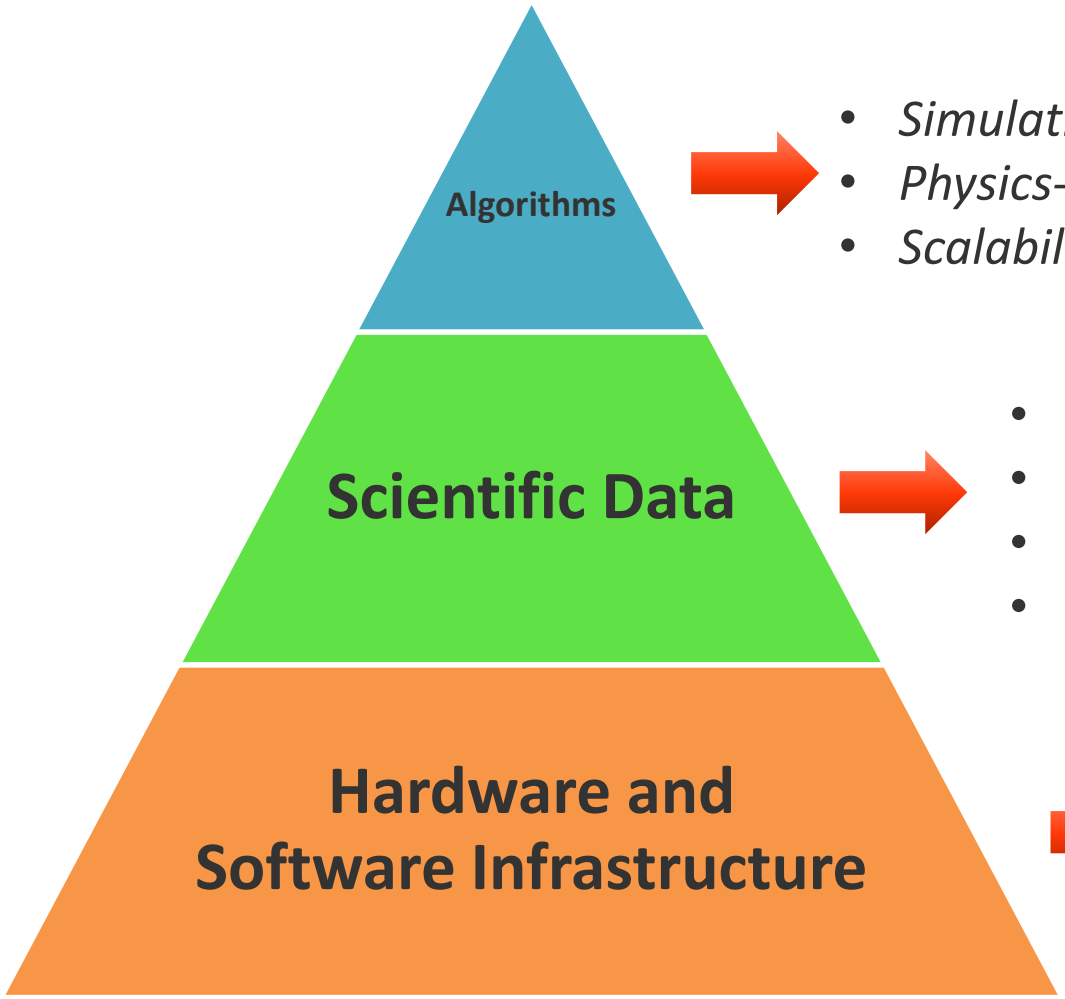
Supervised (inductive) learning
– Given: training data + desired outputs (labels)

• **Unsupervised learning**
– Given: training data (without desired outputs)

• **Semi-supervised learning**
– Given: training data + a few desired outputs

• **Reinforcement/active learning**
– Rewards from sequence of actions

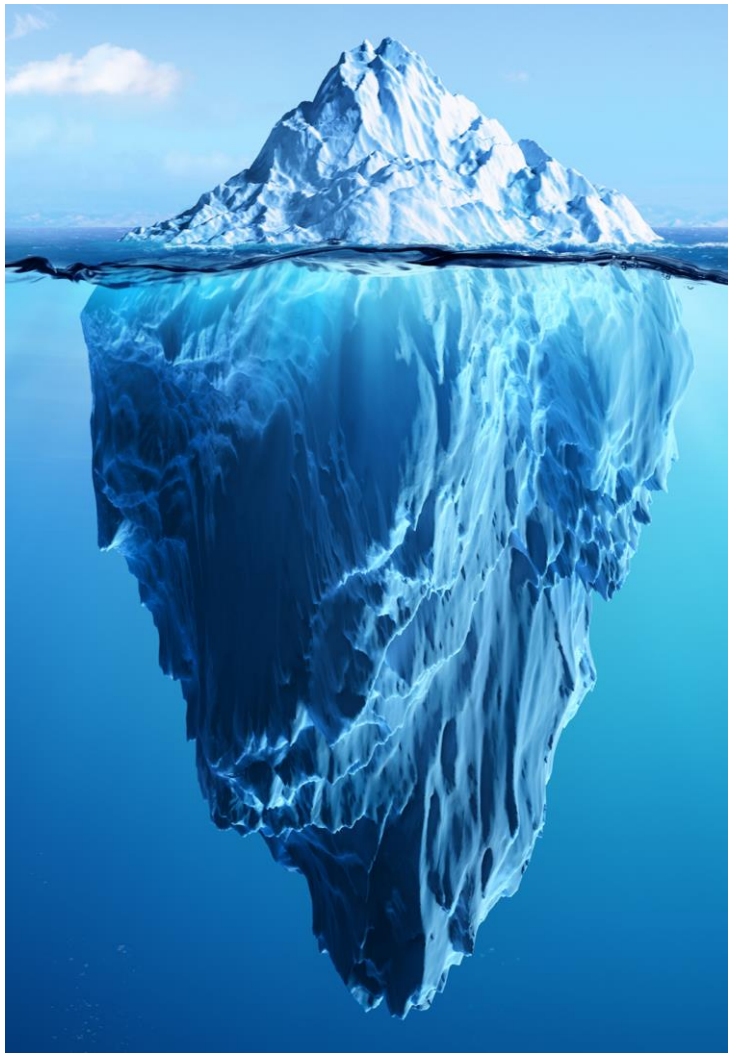
The pyramid of machine learning



- *Simulations*
- *Physics-informed methods*
- *Scalability*

- *Curated*
- *Searchable*
- *Standardized*
- *Best Practices*

- *Codes*
- *Databases*
- *Computing*



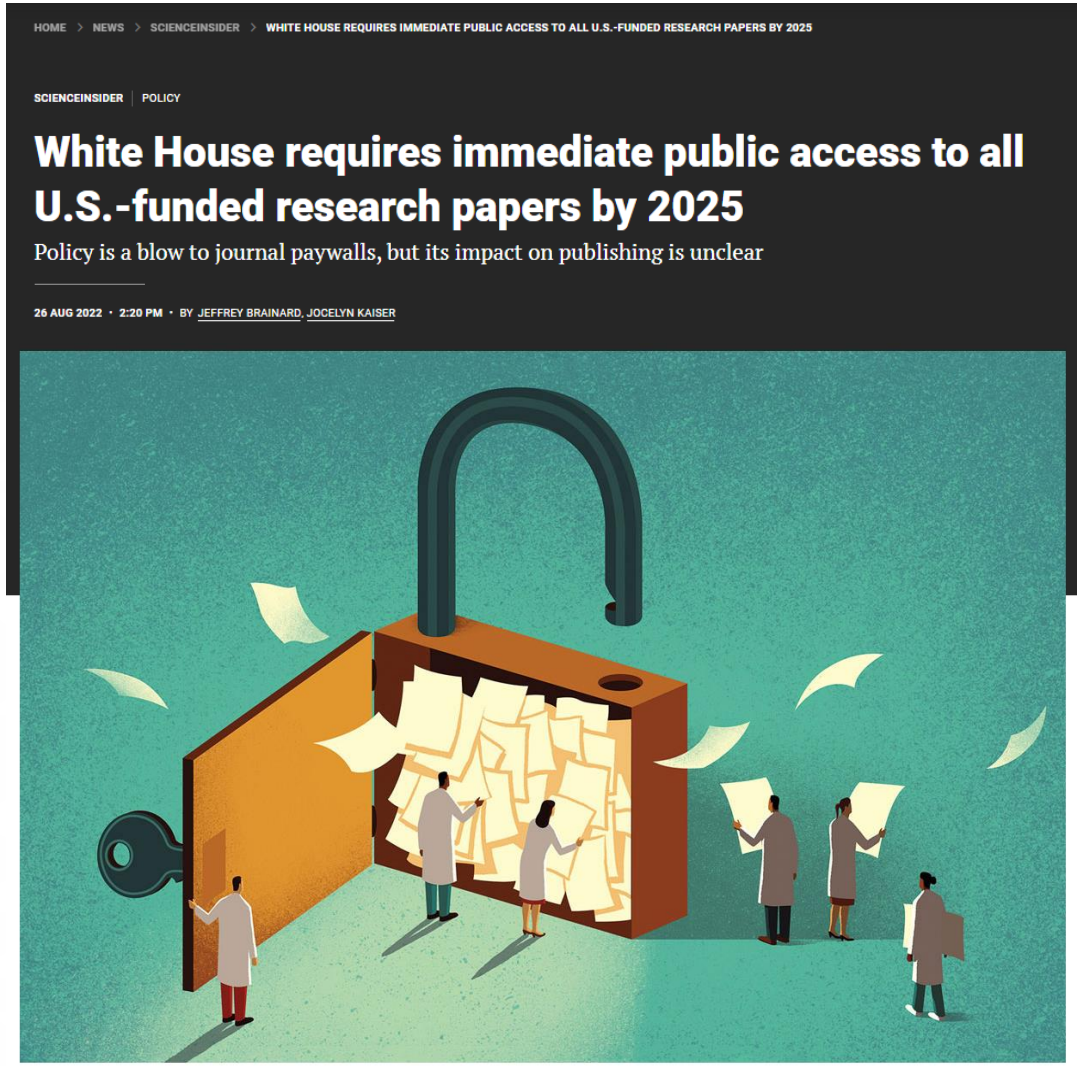
Slide courtesy of R. Vasudevan

Why bother with infrastructure?

- Almost all of machine learning relies extensively on having access to good quality data
- In most laboratories, this data is acquired via multiple instruments in different formats, and not findable or accessible, and often lacks necessary metadata for ML labeling
- As such, in many cases, ML in science is impossible especially in the experimental domains, without the necessary investments in data standardization and storage
- Similarly, reproducibility of workflows relies on strongly tested codebases, not one-off scripts.

Slide courtesy of R. Vasudevan

Soon to be a requirement!

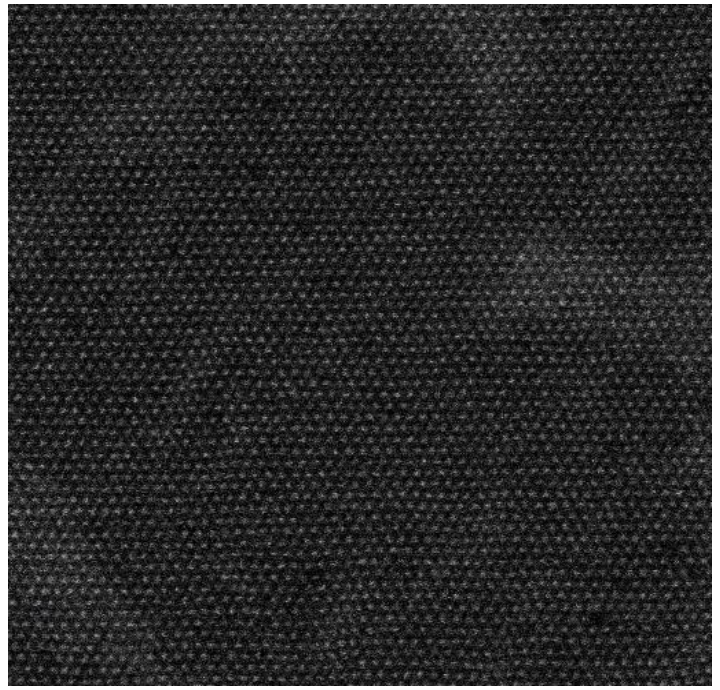


- Data management plans in proposals
- Repositories of data and codes associated with publications
- Good to be ready!

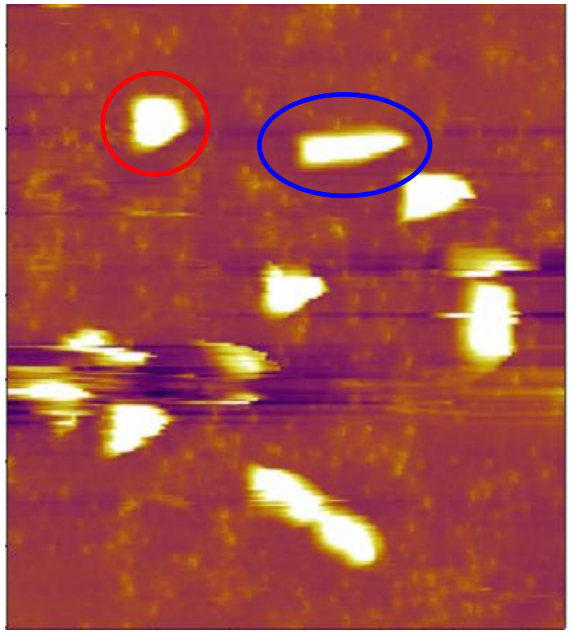
https://www.science.org/content/article/white-house-requires-immediate-public-access-all-u-s--funded-research-papers-2025

Get data – from scientific tools

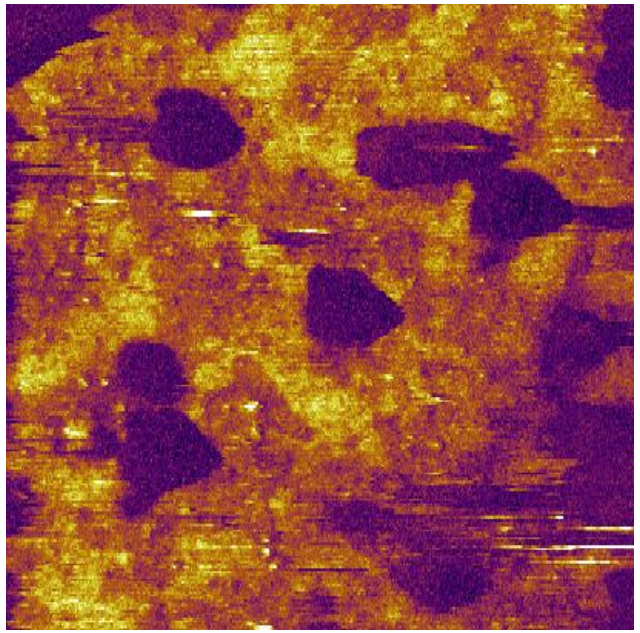
- Spectra
- (Multimodal) Images
- Hyperspectral images
- Videos
- Time traces
- …

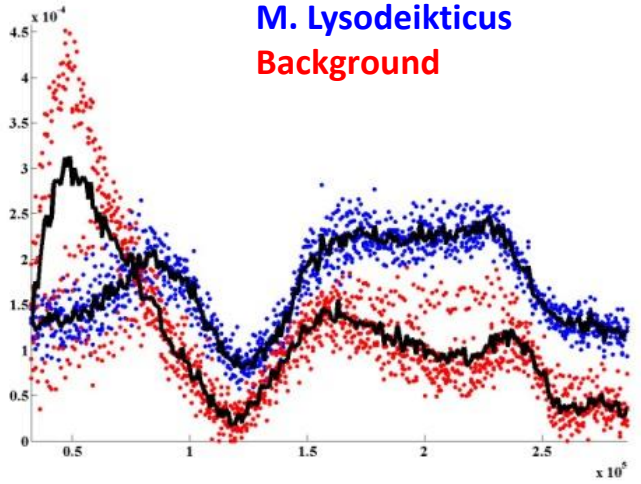


Topography



PFM Amplitude





As scientists, we rarely have to deal with the classical ELT (Extract-Load Transform, aka Data Wrangling) problems. But….

Standardization of Microscope Data

Slide by S. Somnath

Micro Raman Microscope

Atomic Force Microscope (AFM)

AFM with Infrared spectroscopy (AFM-IR)

Scanning Tunneling Microscope (STM)

Scanning Transmission Electron Microscope (STEM)

AFM with Raman spectroscopy

Multitude of File Formats

.wdf

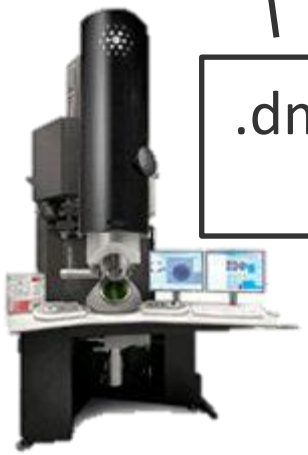
.ibw

.cdb

- Proprietary
- Incompatible

≠

.asc



.dm3

.mat

Slide by S. Somnath

Disjointed communities….

- Clustering
- Fit spectra …

- Filter Image
- Register Image …

- Fit Spectra
- SVD Filtering …

- FFT Filtering
- SVD Filtering …

- FFT Filtering
- Classify Images …

- Register Images
- Clustering

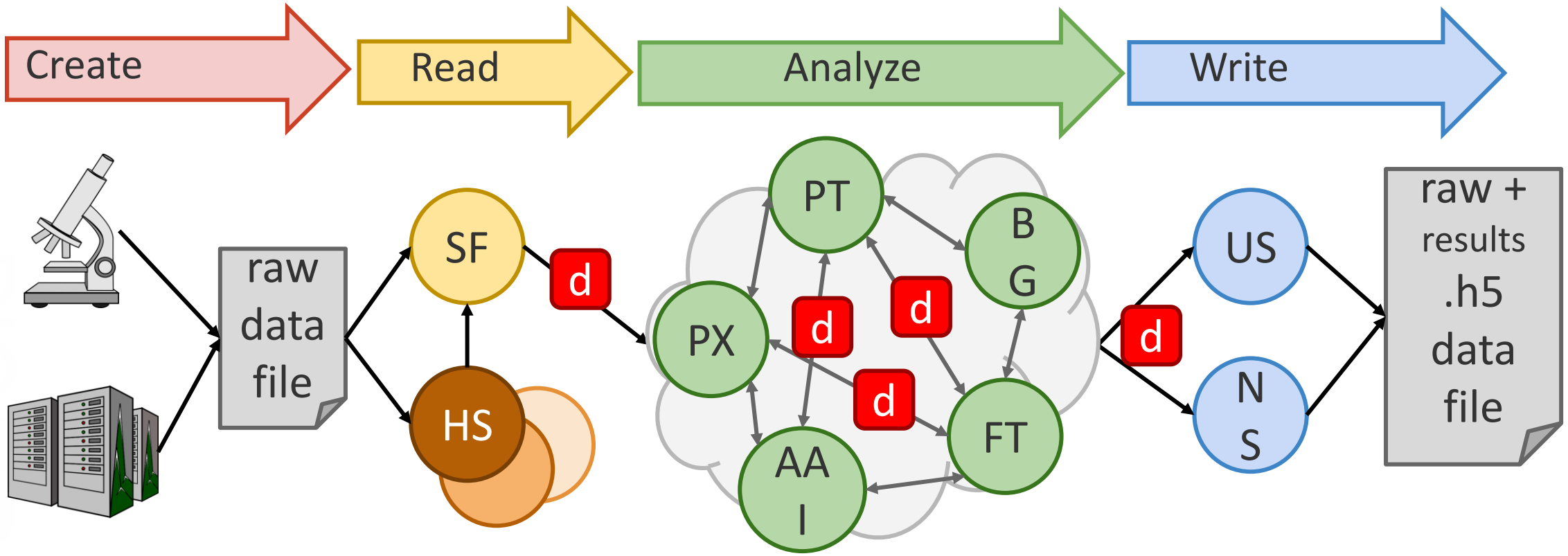
Slide by S. Somnath

… cannot share code efficiently

- HIGHLY instrument-specific code
- Different programming languages
- Often licensed / costly software like Matlab
- Most popular sharing method = email!
- No centralized repository

Slide by S. Somnath

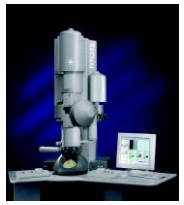
Solutions: Integrated Ecosystems



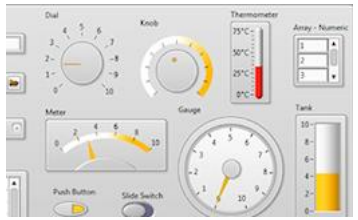
Data from measurements or simulations are read into **`sidpy.Dataset`** (d) objects directly by **`SciFiReaders`** (SF). Data are processed using multiple science packages in the Pycroscopy ecosystem that interoperate via **`Dataset`** objects. **`Dataset`** objects are written to HDF5 files via **`pyUSID`** (US) or **`pyNSID`** (NS).

Slide courtesy of R. Vasudevan

Solutions: Integrated Ecosystems



Instrument Tier



Automated, standardized, modularized data acquisition

Instrument-agnostic, self-describing, model in an open friendly file format

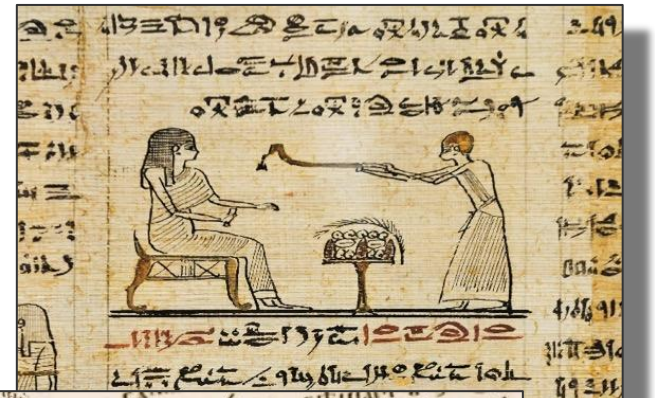
Centralized repository for data processing, analysis

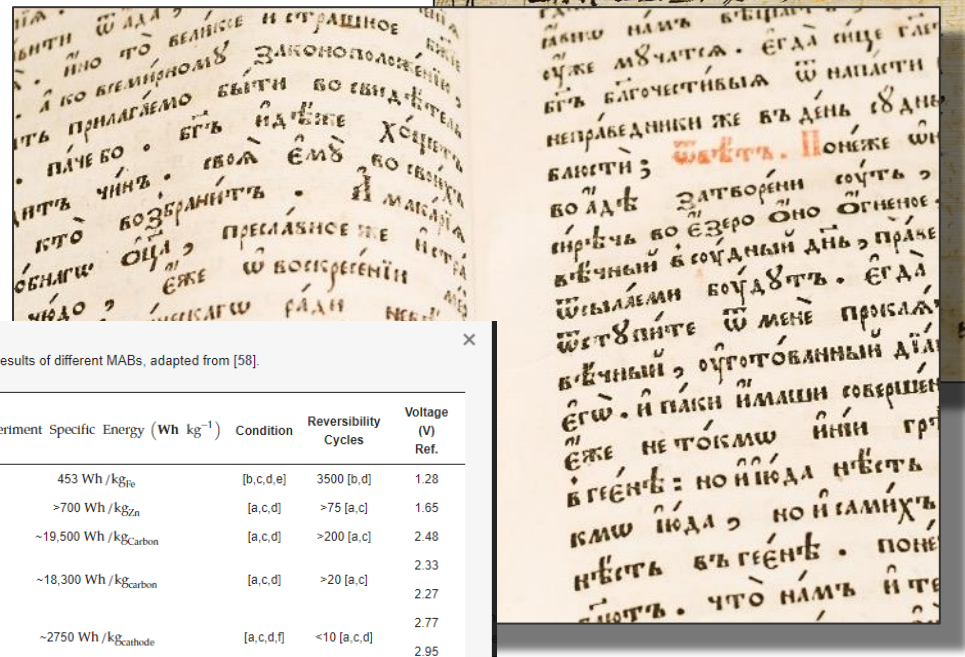
Interactive visualization + analysis + storage on the cloud

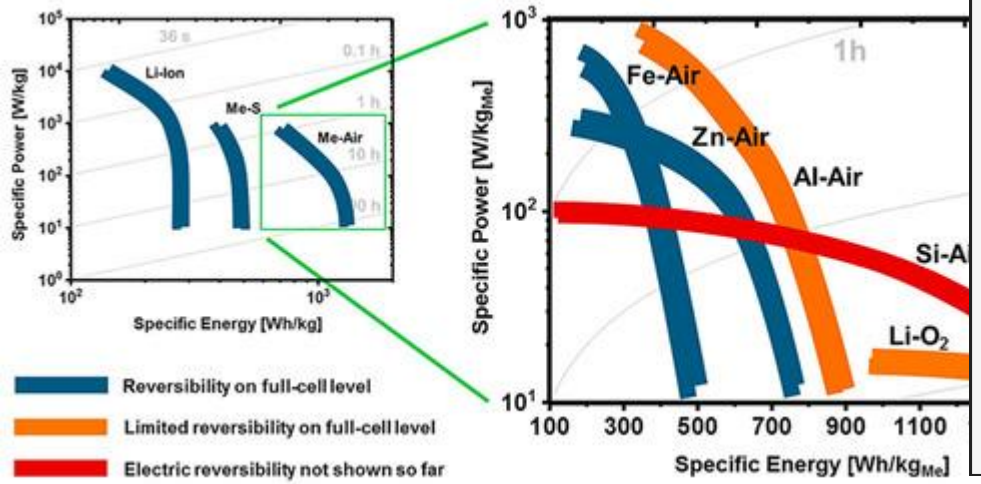
Slide courtesy of R. Vasudevan

Get data – from publications

- Printed matter
- Pdf files with figures and tables
- Deposited data files
- (Rarely) workflows







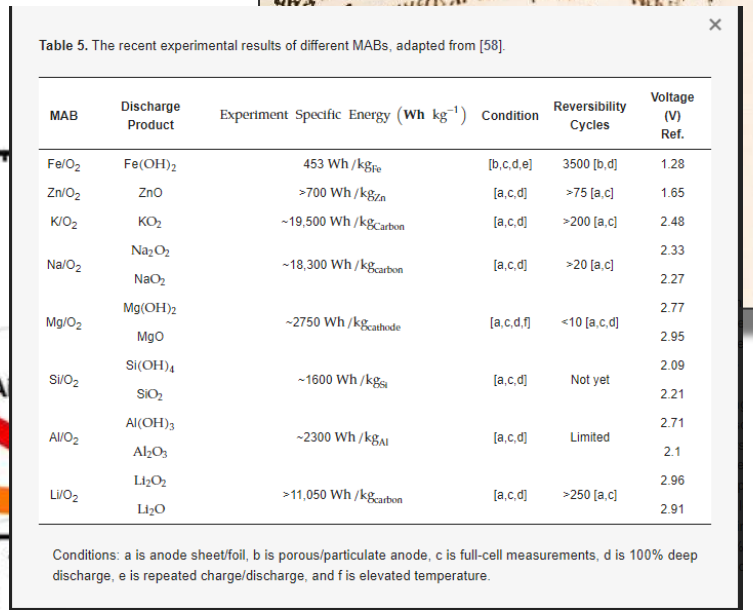
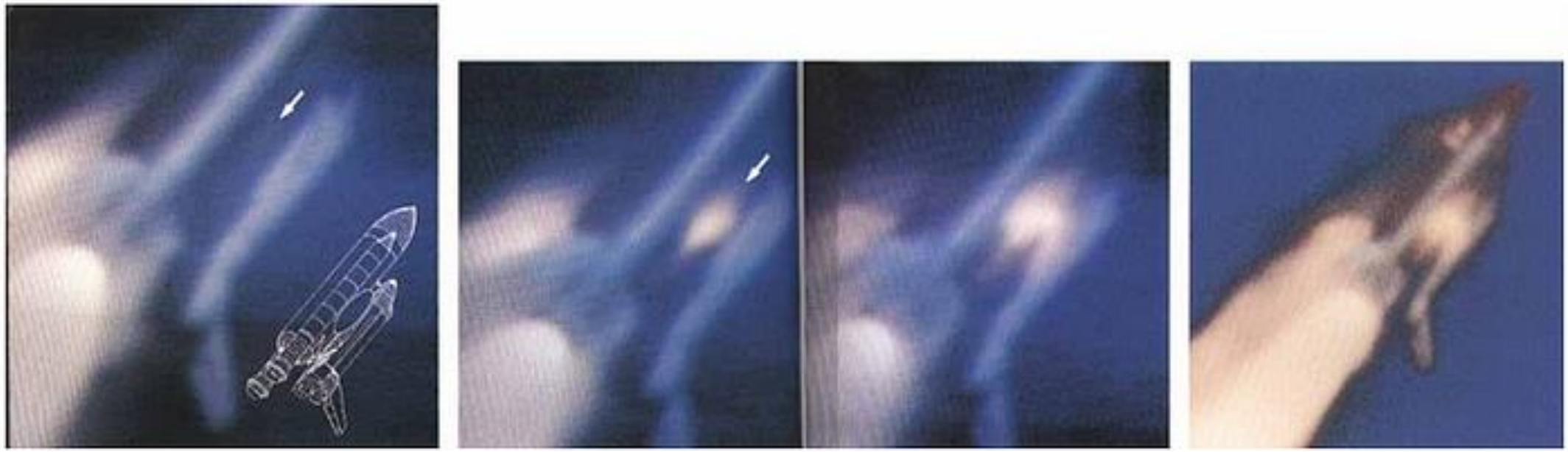


Table 5. The recent experimental results of different MABs, adapted from [58].

MAB	Discharge Product	Experiment Specific Energy (Wh kg^{-1})	Condition	Reversibility Cycles	Voltage (V) Ref.
Fe/O_2	$Fe(OH)_2$	453 Wh /kg$_{Fe}$	[b,c,d,e]	3500 [b,d]	1.28
Zn/O_2	ZnO	>700 Wh /kg$_{Zn}$	[a,c,d]	>75 [a,c]	1.65
K/O_2	KO_2	~19,500 Wh /kg$_{Carbon}$	[a,c,d]	>200 [a,c]	2.48
Na/O_2	Na_2O_2 NaO_2	~18,300 Wh /kg$_{carbon}$	[a,c,d]	>20 [a,c]	2.33 2.27
Mg/O_2	$Mg(OH)_2$ MgO	~2750 Wh /kg$_{cathode}$	[a,c,d,f]	<10 [a,c,d]	2.77 2.95
Si/O_2	$Si(OH)_4$ SiO_2	~1600 Wh /kg$_{Si}$	[a,c,d]	Not yet	2.09 2.21
Al/O_2	$Al(OH)_3$ Al_2O_3	~2300 Wh /kg$_{Al}$	[a,c,d]	Limited	2.71 2.1
Li/O_2	Li_2O_2 Li_2O	>11,050 Wh /kg$_{carbon}$	[a,c,d]	>250 [a,c]	2.96 2.91

Conditions: a is anode sheet/foil, b is porous/particulate anode, c is full-cell measurements, d is 100% deep discharge, e is repeated charge/discharge, and f is elevated temperature.

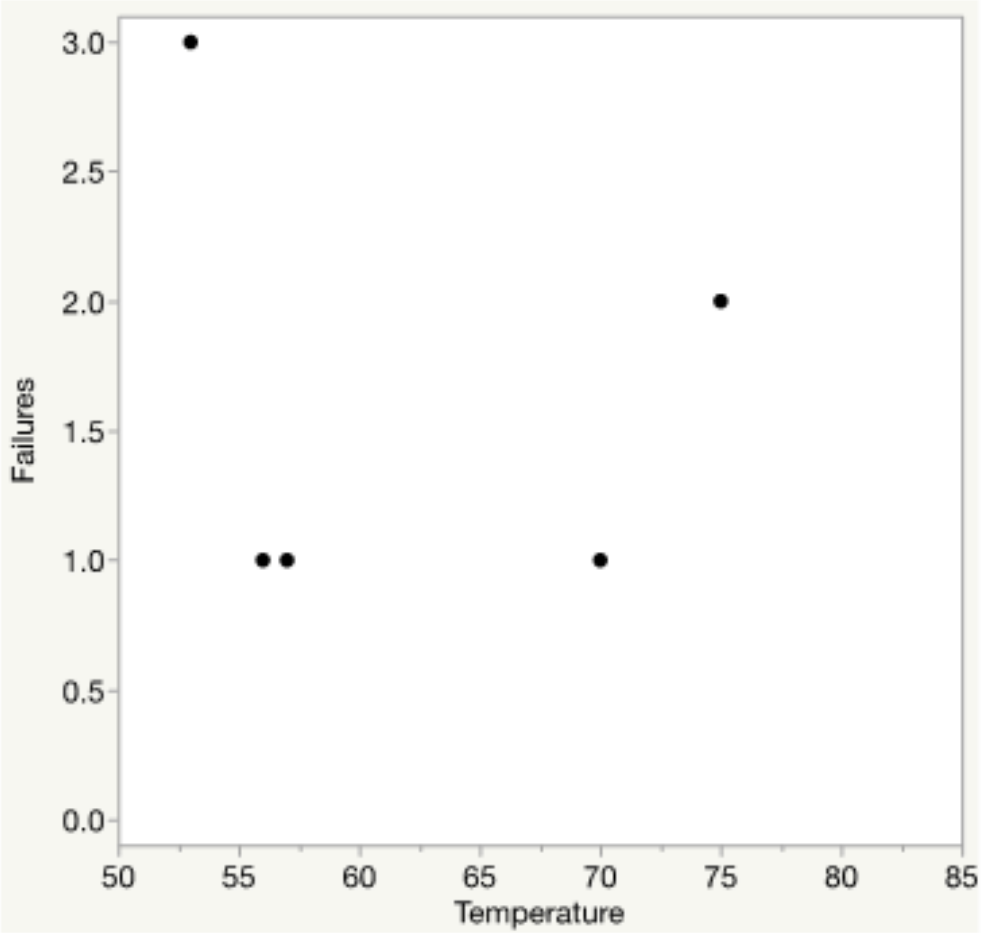
Challenger disaster



https://medium.com/habits-for-success/the-challenger-disaster-a-lesson-in-the-power-of-data-analysis-and-visualization-398d2ac8f59b

https://www.youtube.com/watch?v=hOEt1MOuYX4

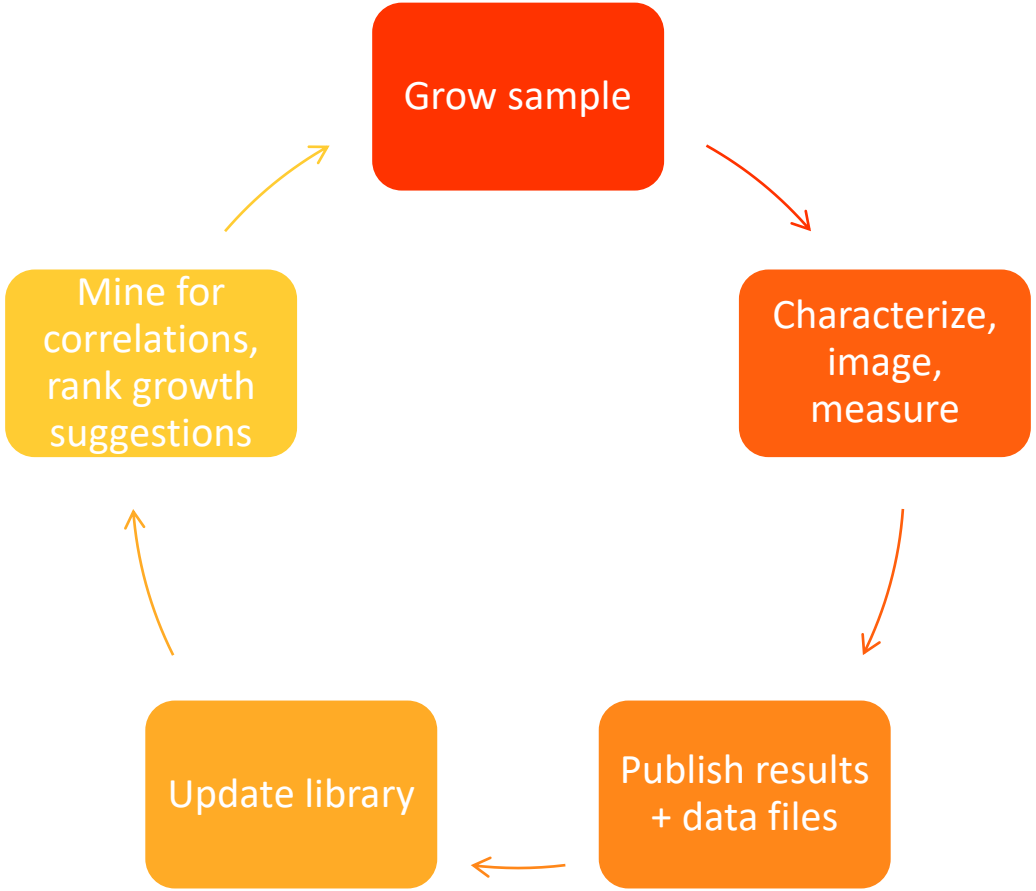
Challenger disaster

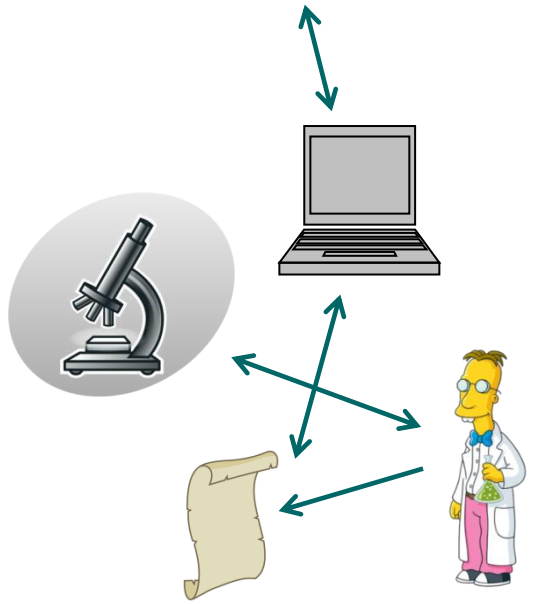


https://www.researchgate.net/figure/Challenger-Data-Number-of-O-Ring-Failures-Launch-Versus-TemperatureLeft-Panel-Five_fig2_344257416

Conclusion: requires community involvement

- Relevant sample preparation conditions
- Unique sample identifier
- Measured properties in appropriate format





Slide courtesy of R. Vasudevan

Get data – data bases!

- **On-line data bases**
  - Materials Project, https://next-gen.materialsproject.org/
  - Materials Data Facility: https://materialsdatafacility.org/
  - Papers with code: https://paperswithcode.com/datasets
  - Standard data sets for molecular properties, e.g. http://quantum-machine.org/datasets/
  - Integrators, e.g.: https://github.com/sedaoturak/data-resources-for-materials-science

- **Enterprise data bases**

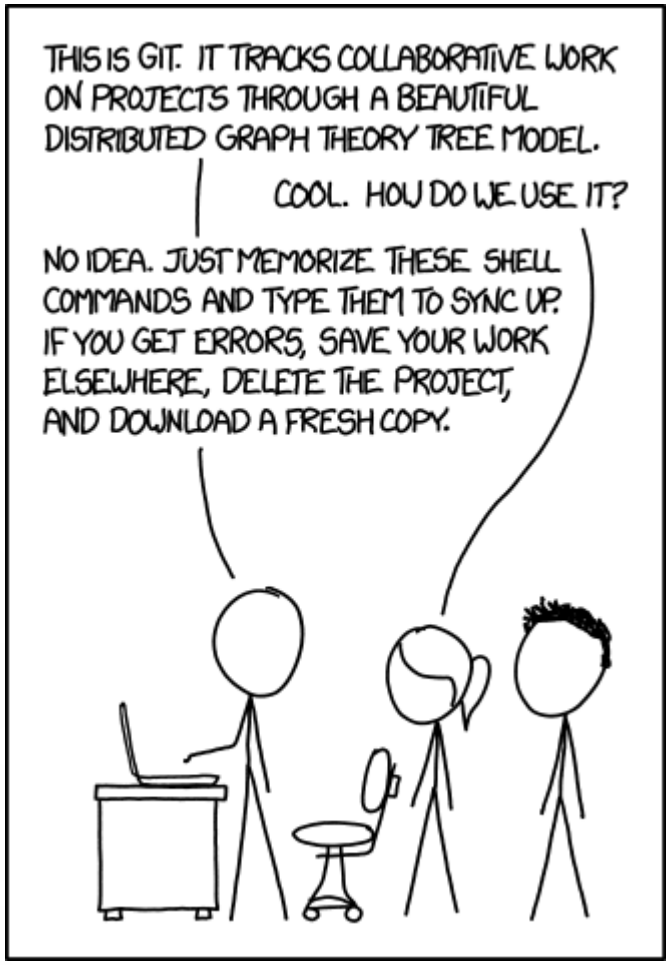
How we can run code:

- Google Colabs
- AWS SageMaker notebooks
- IDE: Spyder, PyCharm, etc.
- Command line interface

Code Repositories and Version Control

- Sharing scripts between users can be workable for immediate or short-term needs, but is not scalable nor lasting
- For reproducibility, it is better to have codes that reside in packages that are documented and well tested
- Most of you are familiar with python packages; but many are probably new to version control
- Version control systems such as git enable multiple people to work on a single software project at the same time to speed up development and ensure consistency
- Git is an open-source distributed version control system. It maintains a history of changes that have occurred in the project and allows for updates as well as reversions to older ‘commits’.

How we can share code:



Git would take a significant amount of time to explain in detail. However, there are plenty of online tutorials, e.g.

https://www.atlassian.com/git/tutorials

We will test some of the basic git functionality as demo.

1. Make sure you have a GitHub account

2. Example:
https://github.com/ramav87/ML_Summer_Course

3. Head on over to
https://github.com/SergeiVKalinin/MSE_2023_Git

We will practice forking a repository, making a change to the code and submitting a pull request

What language do we use:

Main Python libraries we will use:

1. NumPy
2. MatPlotlib
3. Scikit-learn
4. Keras

Other libraries we may use:

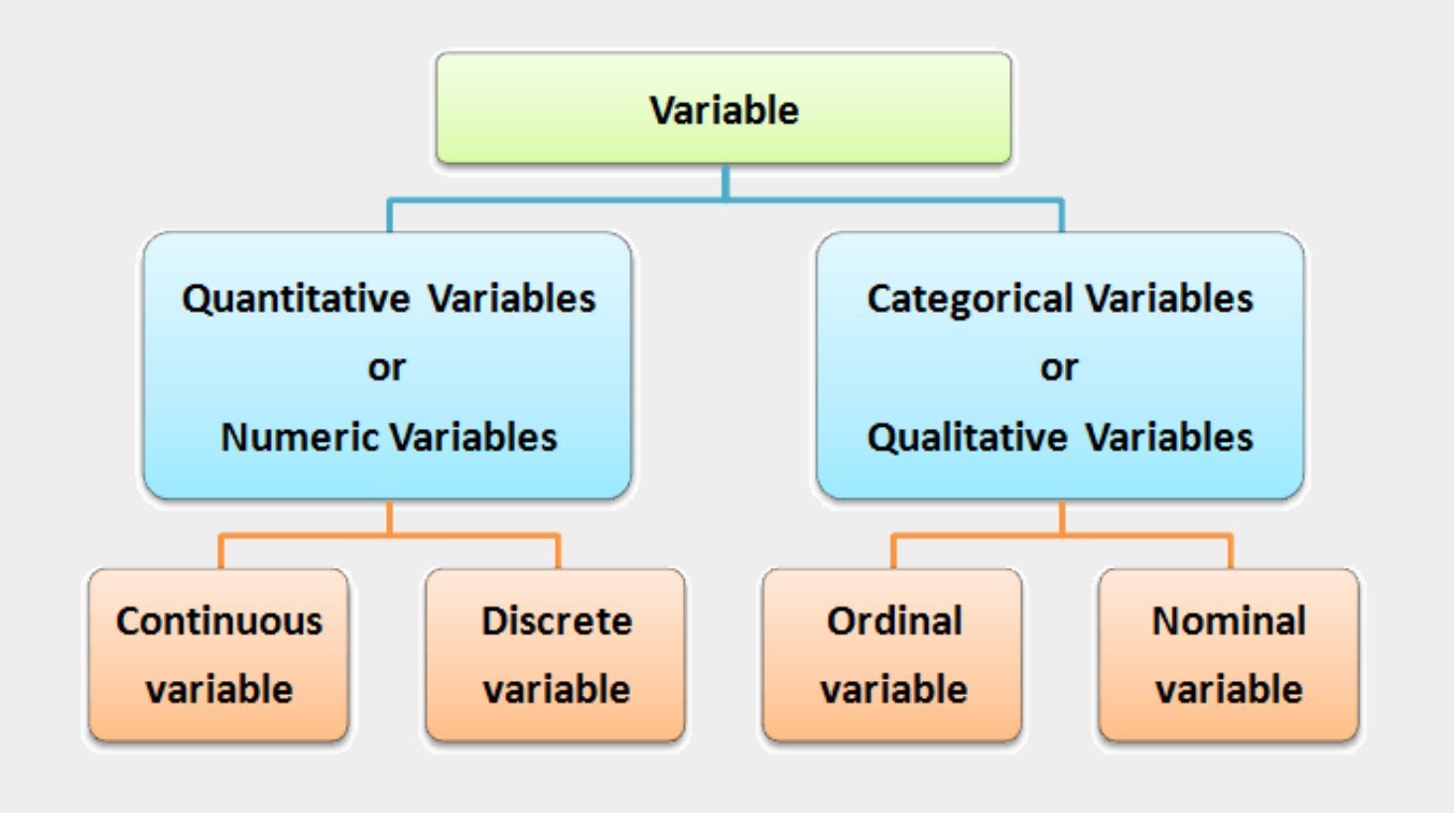
1. Seaborn
2. GPax
3. SciPy

We will learn these as we need them!

Learning programming:

1. Learn to program per se
2. Writing code used in project
3. Making code that others will use
4. Using codes written by others
5. Being a part of the team developing code
6. Leading the team developing code

Type of variables



Nominal variable: no order

Colors: blue, red, yellow, white

Dog breed: Labrador, German shepherd, poodle

Ordinal variable: there is order

Letter grades: A, B, C, …

Severity of the software bug: mild, serious, critical

https://prinsli.com/categorical-variables/

Classification Workflow

1. Selecting features and collecting labeled training examples
2. Choosing a performance metric
3. Choosing a learning algorithm and training a model
    - Can we understand how it works?
    - Does it have any hyperparameters
    - How well does it generalize to new data?
    - How expensive is it?
4. Evaluating the performance of the model
5. Changing the settings of the algorithm and tuning the model.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

There are many classifiers:

Choosing an appropriate classification algorithm for a particular problem task requires practice and experience; each algorithm has its own quirks and is based on certain assumptions.

To paraphrase the **no free lunch theorem** by David H. Wolpert, no single classifier works best across all possible scenarios (*The Lack of A Priori Distinctions Between Learning Algorithms*, *Wolpert, David H*, *Neural Computation 8.7* (1996): 1341-1390).

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Hello, I am Akinator

Think about a real or fictional character.
I will try to guess who it is

akinator®

Inactive

Sensitive content

Download on the App Store

PLAY

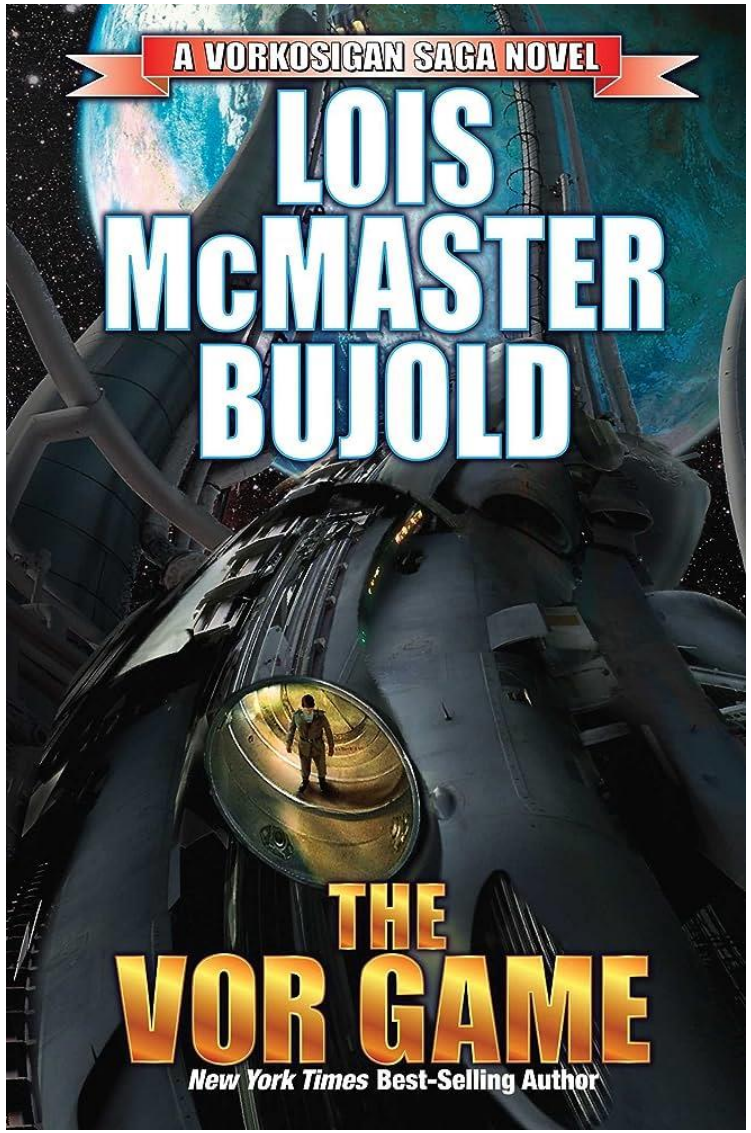
GET IT ON Google Play

824 people are playing right now.
631563588 games played 30599 today.

Famous sportsmen Akinator can guess

Playing a classifier game:

1. Is your character a pokemon: No
2. Does your character have legs: Yes
3. Is your character a girl: No
4. Is your character famous youtuber: No
5. Is your character real: No
6. Does your character have human head: Yes
7. Is your character from Japanese anime: No
8. Wear a mask: No
9. Originally from video game: No
10. Animated: No
11. Played in superhero movie: No
12. From a book: Yes
13. Have been in movie: No
14. Have powers: No
15. Popular tv show: No
16. Have phone: yes
17. Adult: Yes
18. Killed people: Yes
19. Live in future: Yes
20. Been in space: Yes
21. Wife dead: No
22. Linked with metal suit: No
23. Short: Yes
24. Father famous: Yes



Answer: Miles Vorkosigan

Tennis Player Example

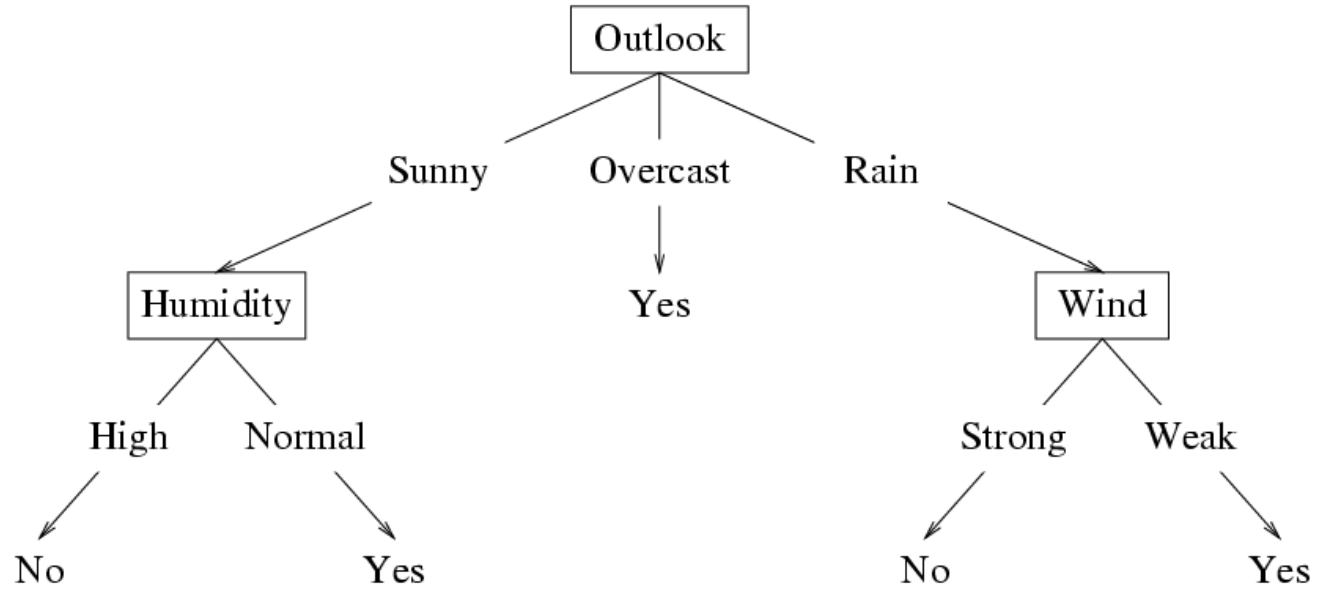
PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Decision Tree Hypothesis Space

- **Internal nodes** test the value of particular features x_j and branch according to the results of the test.
- **Leaf nodes** specify the class $h(\mathbf{x})$.

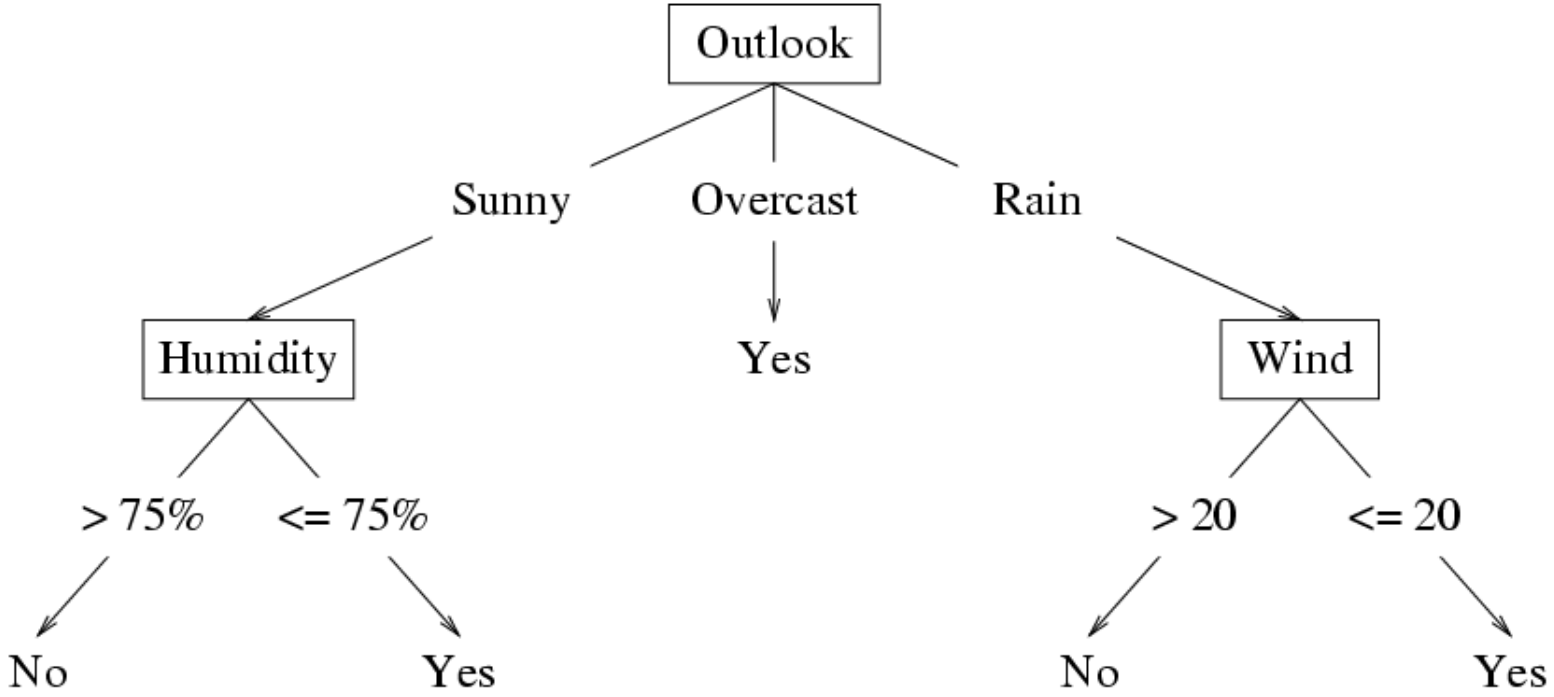


Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

What if we have continuous variables?

If the features are continuous, internal nodes may test the value of a feature against a threshold.

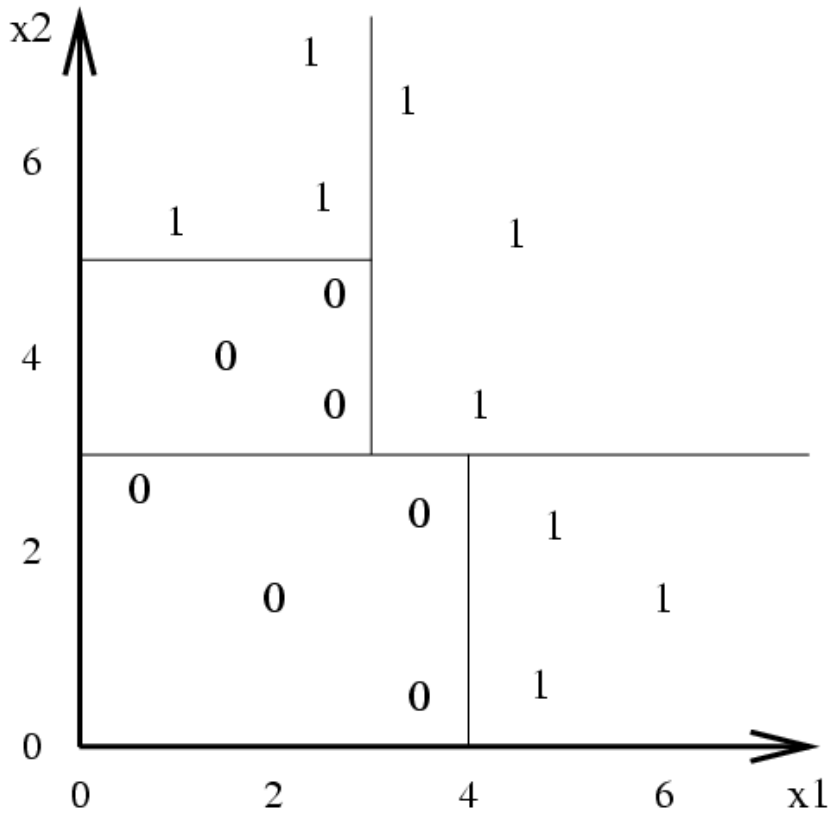


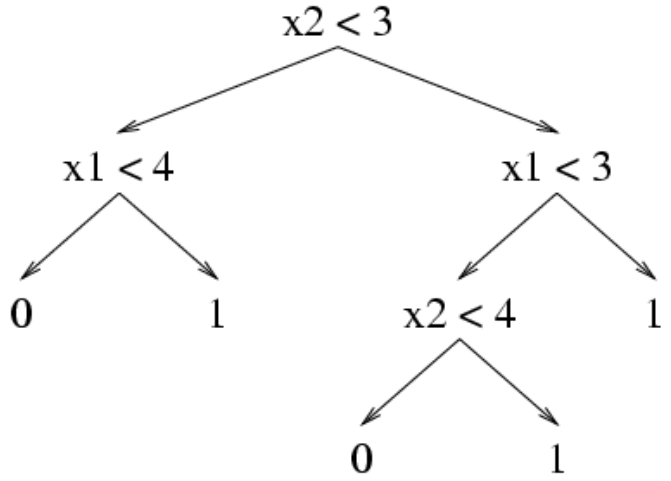
Also: regression trees

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Decision Tree Boundaries

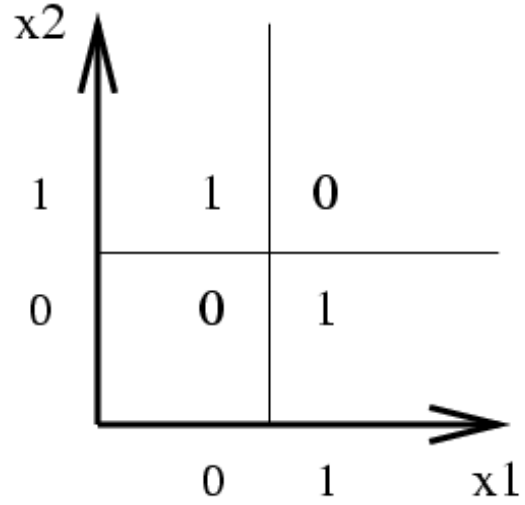
Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.

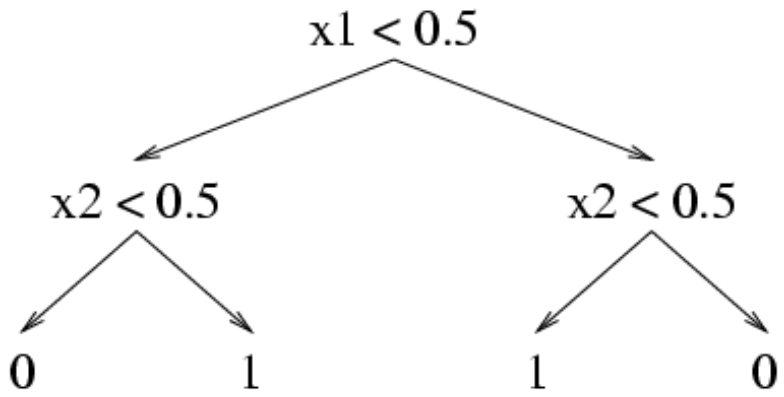




Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Can Represent Any Boolean Function





The tree will in the worst case require exponentially many nodes, however.

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Classification and Regression Tree (CART)

- Create a set of questions that consists of all possible questions about the measured variables

- Select a splitting criterion (likelihood).
  - Initialization: create a tree with one node containing all the training data.
  - Splitting: find the best question for splitting each terminal node. Split the one terminal node that results in the greatest increase in the likelihood.
  - Stopping: if each leaf node contains data samples from the same class, or some pre-set threshold is not satisfied, stop. Otherwise, continue splitting.
  - Pruning: use an independent test set or cross-validation to prune the tree.

- There are ways to estimate and incorporate priors into the decision tree (though these methods somewhat predate Bayesian methods).

- Computational complexity is very low for both evaluation and training.

How do we split?

Information gain:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

- f is the feature to perform the split
- D_p and D_j are the dataset of the parent and jth child node
- I is our **impurity** measure
- N_p is the total number of training examples at the parent node
- N_j is the number of examples in the j-th child node

Binary split:

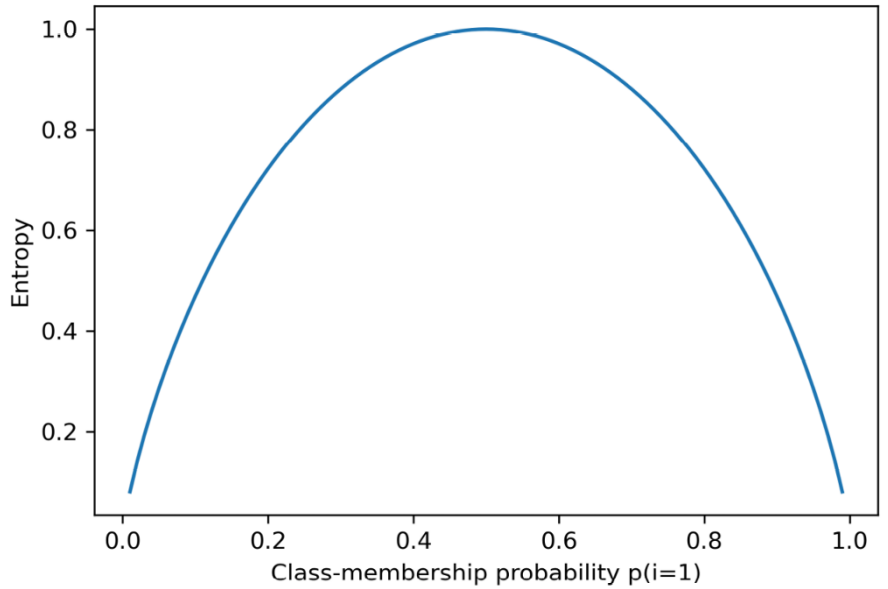
$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

What are impurity measures?

Entropy:

$$I_H(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t)$$



- $p(i|t)$ is the proportion of the examples that belong to class i for a particular node, t.
- The entropy is 0 if all examples at a node belong to the same class, and entropy is maximal if we have a uniform class distribution.
- For binary class setting, the entropy is 0 if $p(i=1|t) = 1$ or $p(i=0|t) = 0$. If the classes are distributed uniformly with $p(i=1|t) = 0.5$ and $p(i=0|t) = 0.5$, the entropy is 1.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

What are impurity measures?

Gini impurity:

$$I_G(t) = \sum_{i=1}^{c} p(i|t)\,(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

Classification error:

$$I_E(t) = 1 - \max\{p(i|t)\}$$

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Computing Information Gain

- Let's begin with the root node of the DT and compute IG of each feature
- Consider feature "wind" $\in$ {weak,strong} and its IG w.r.t. the root node

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

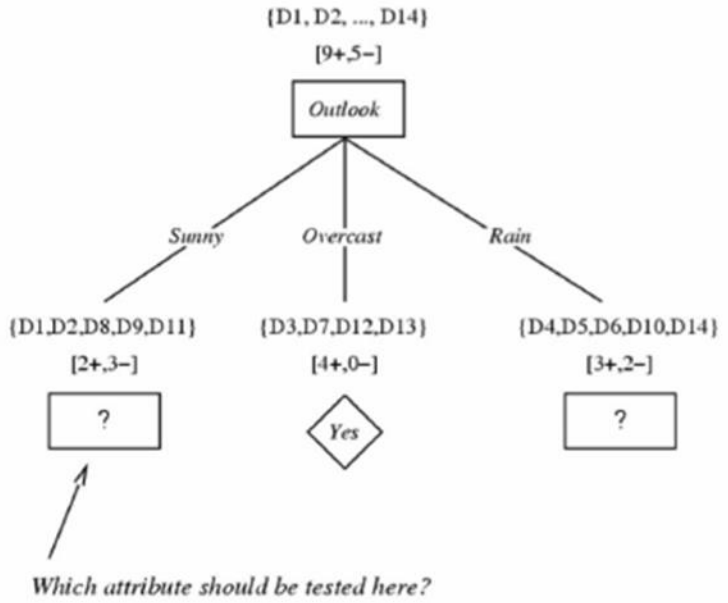
- Root node: $S = [9+, 5-]$ (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$
- $S_{weak} = [6+, 2-] \Longrightarrow H(S_{weak}) = 0.811$
- $S_{strong} = [3+, 3-] \Longrightarrow H(S_{strong}) = 1$

$$\begin{aligned} IG(S, \text{wind}) &= H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong}) \\ &= 0.94 - 8/14 * 0.811 - 6/14 * 1 \\ &= 0.048 \end{aligned}$$

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

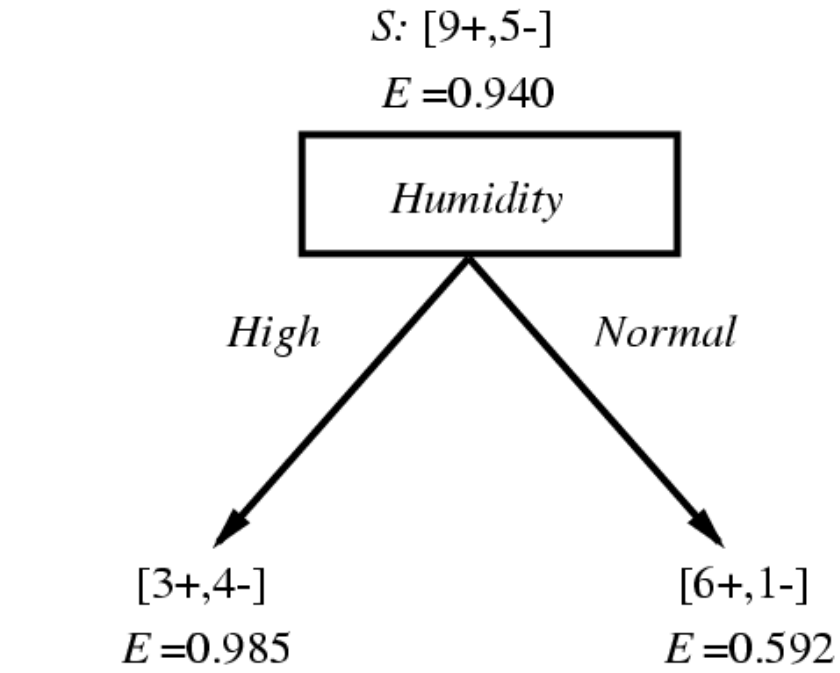
Choosing the Most Informative Feature

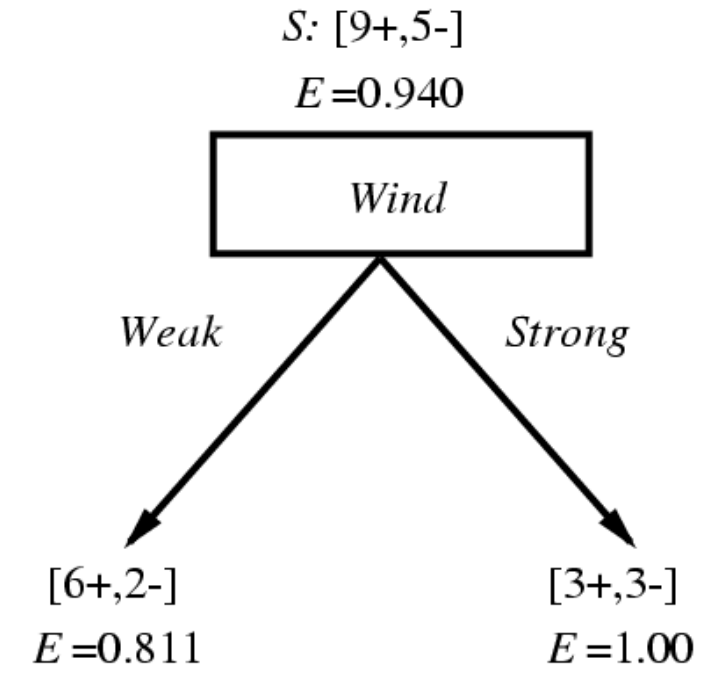
- At the root node, the information gains are:
  - $IG(S, \text{wind}) = 0.048$ (we already saw)
  - $IG(S, \text{outlook}) = 0.246$
  - $IG(S, \text{humidity}) = 0.151$
  - $IG(S, \text{temperature}) = 0.029$
- "outlook" has the maximum $IG \Longrightarrow$ chosen as the root node
- Growing the tree:
  - Iteratively select the feature with the highest information gain for each child of the previous node



Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Selecting the Next Attribute





Gain (S, Humidity)

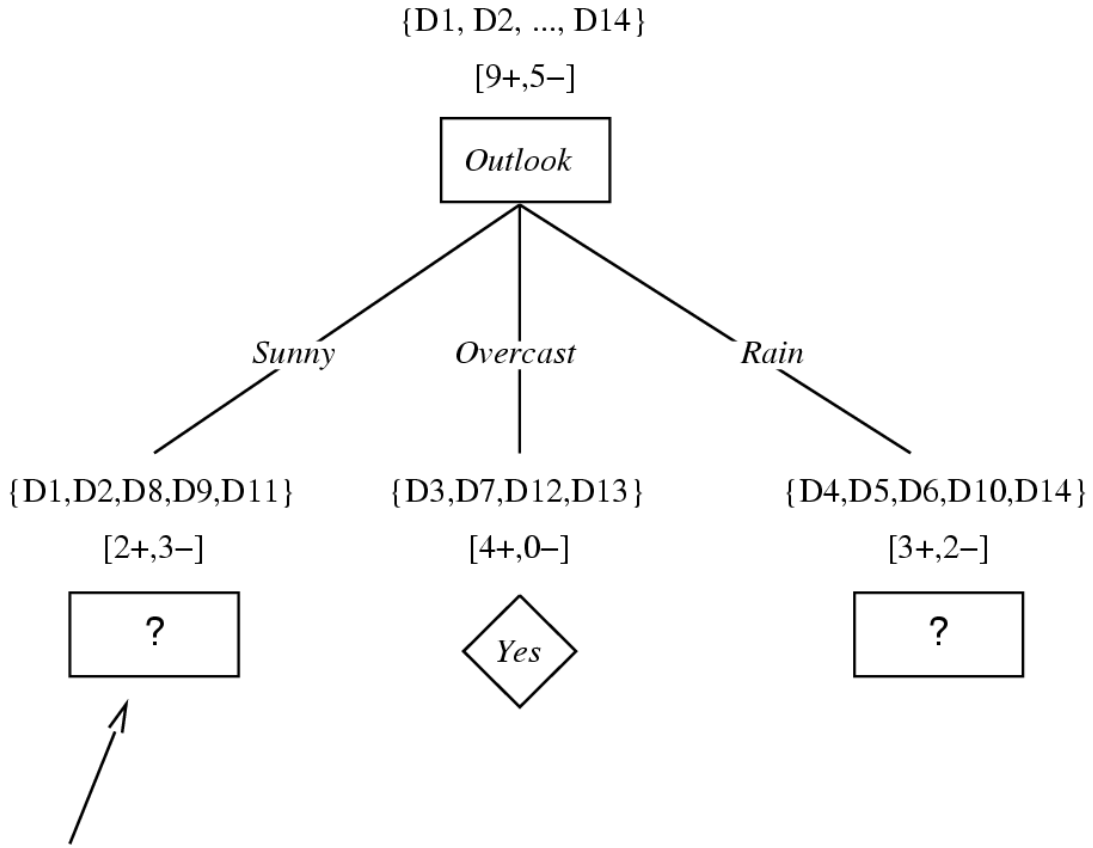
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

And so on...



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,9,D11}

Gain (S_{sunny} , *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny} , *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny} , *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Decision Tree are Adaptable:

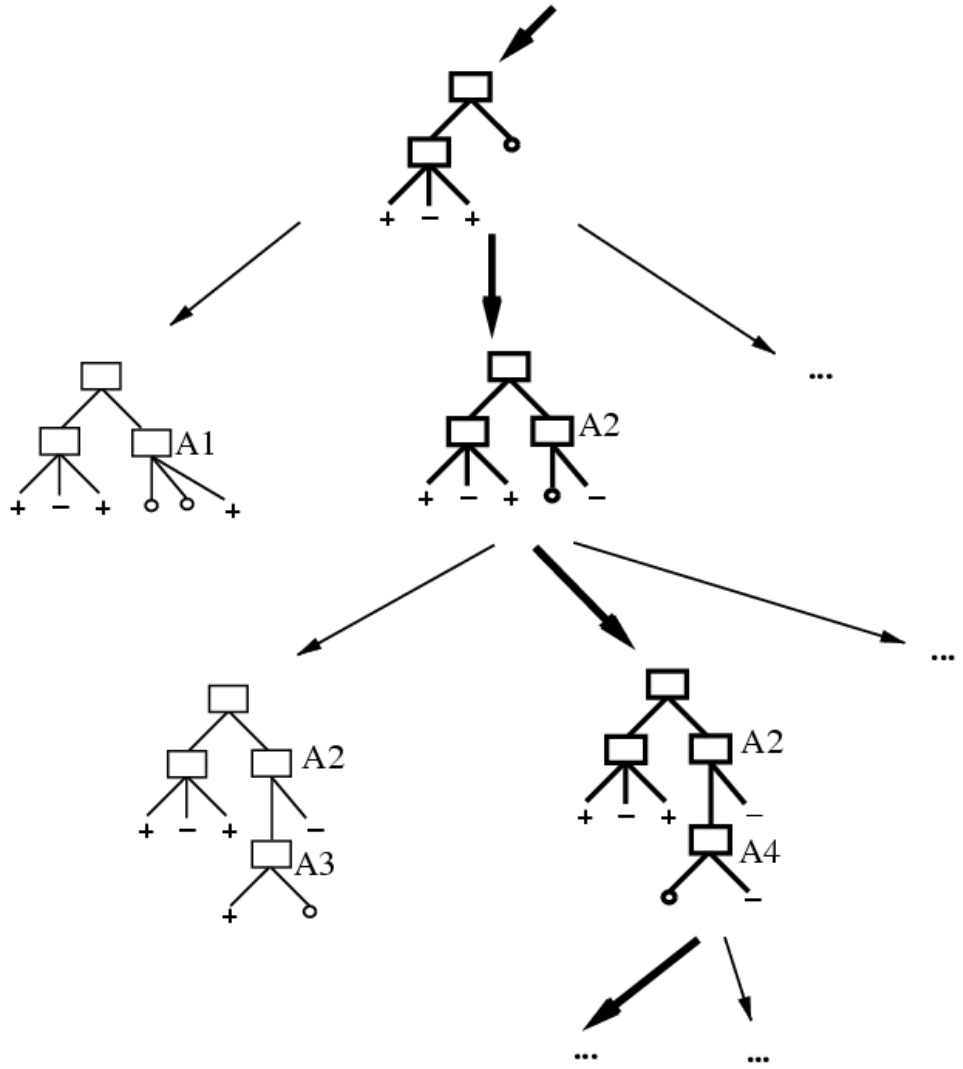
- Features with multiple discrete values
  - Multi-way splits
  - Test for one value versus the rest
  - Group values into disjoint sets

- Real-valued features
  - Use thresholds

- Regression
  - Splits based on mean squared error metric

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Hypothesis Space Search



You do not get the globally optimal tree!

Search space is exponential.

Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Solution: Decision Tree Forest

1. Draw a random **bootstrap** sample of size *n* (randomly choose *n* examples from the training dataset with replacement).

2. Grow a decision tree from the bootstrap sample. At each node:
   a. Randomly select *d* features without replacement.
   b. Split the node using the feature that provides the best split according to the objective function, for instance, maximizing the information gain.

3. Repeat *steps 1-2 k* times.

4. Aggregate the prediction by each tree to assign the class label by **majority vote**.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Overfitting

Consider error of hypothesis h over

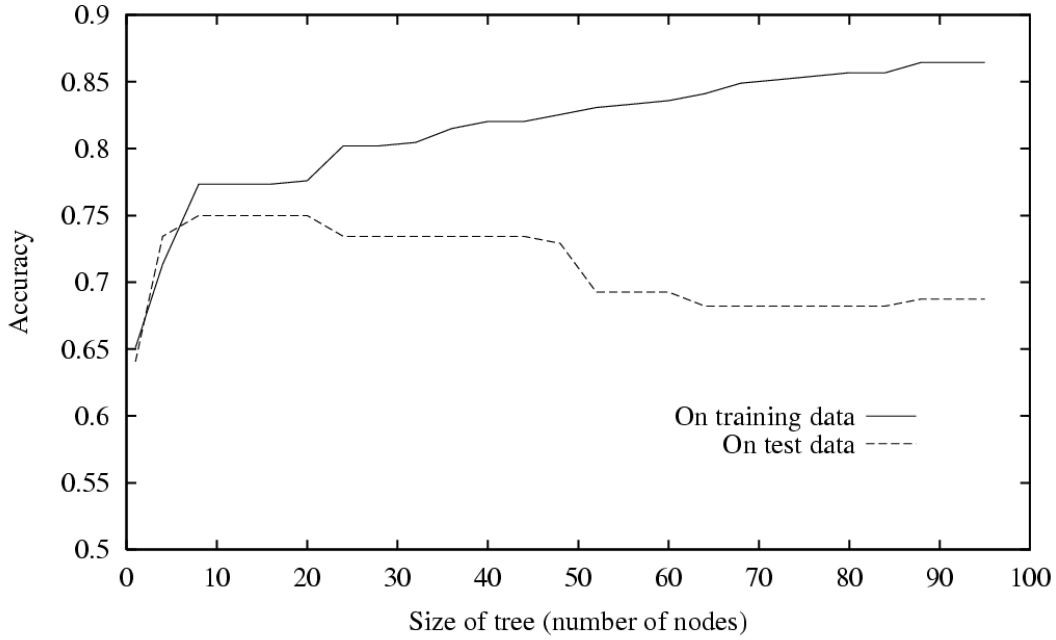
- training data: $error_{train}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$



- Prune tree to reduce error on validation set

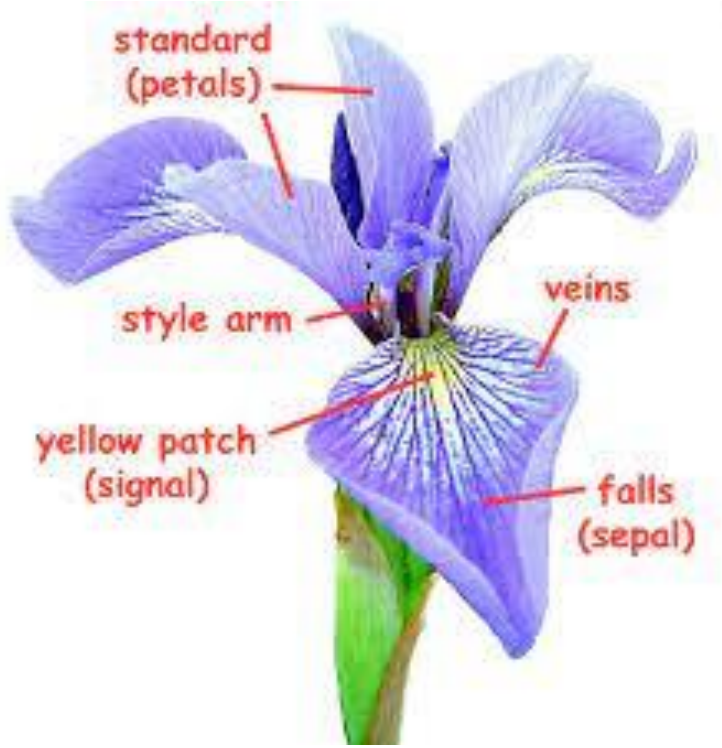
Adapted from Greg Grudic, Decision Trees (Notes borrowed from Thomas G. Dietterich and Tom Mitchell)

Let’s do some classification!

http://www.lac.inpe.br/~rafael.santos/Docs/CAP394/WholeStory-Iris.html

Let's do some classification!

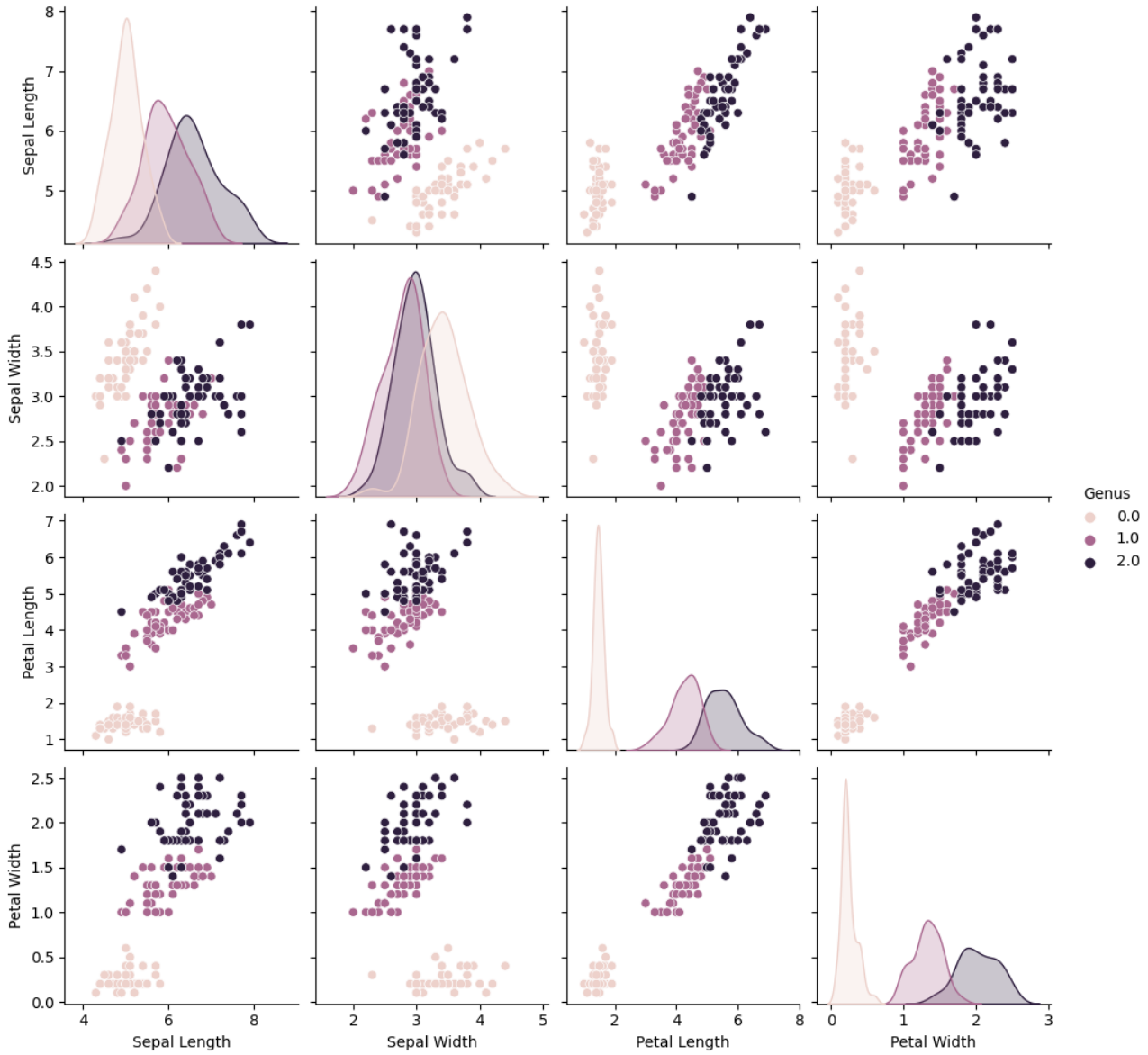
	Sepal length	Sepal width	Petal length	Petal width	
	0	1	2	3	4
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica



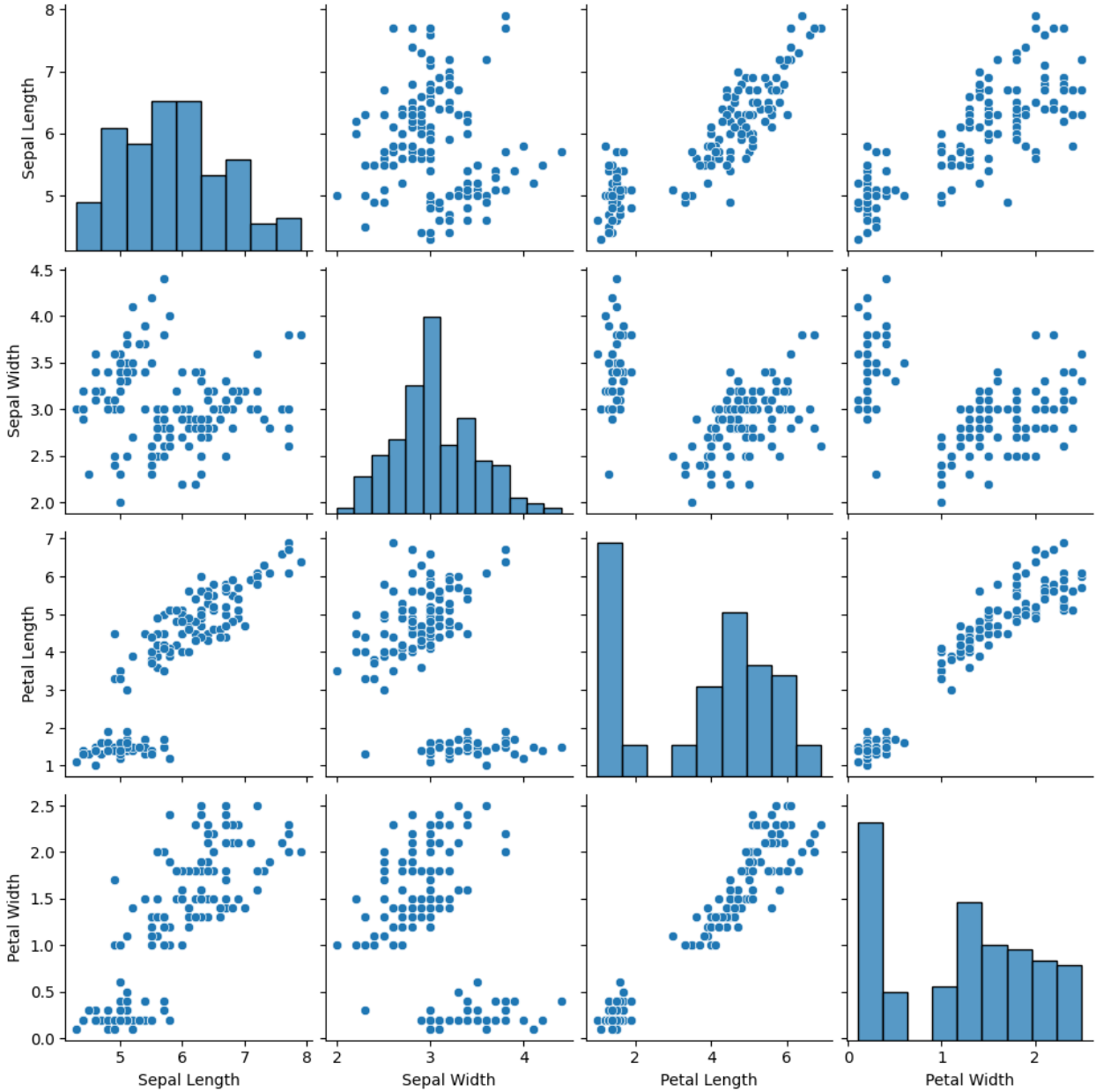
In scikit-learn, Iris-setosa, Iris-versicolor, and Iris-virginica, are already stored as integers

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

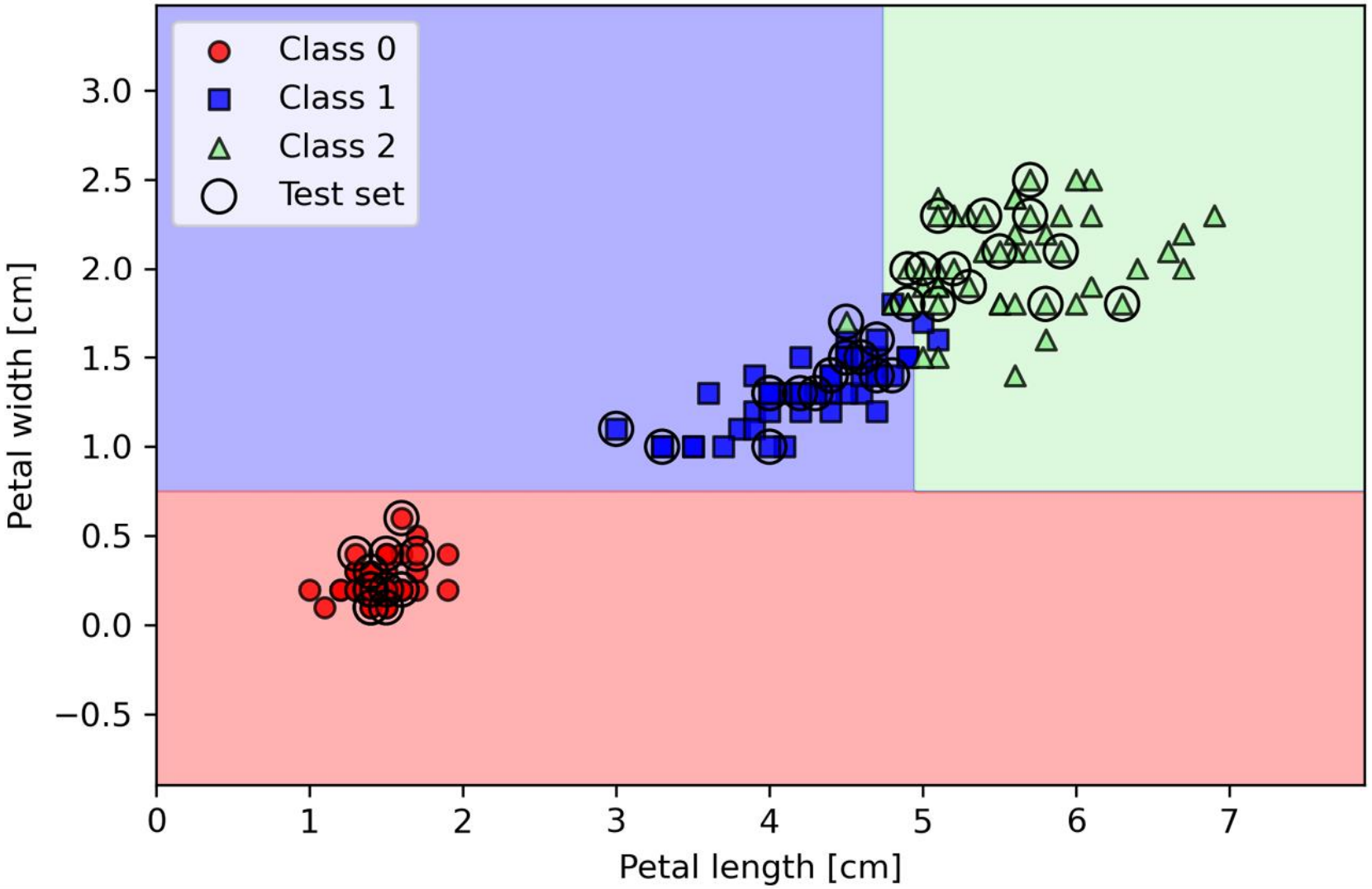
But first, exploratory data analysis…



Can we do it if there is no labels?

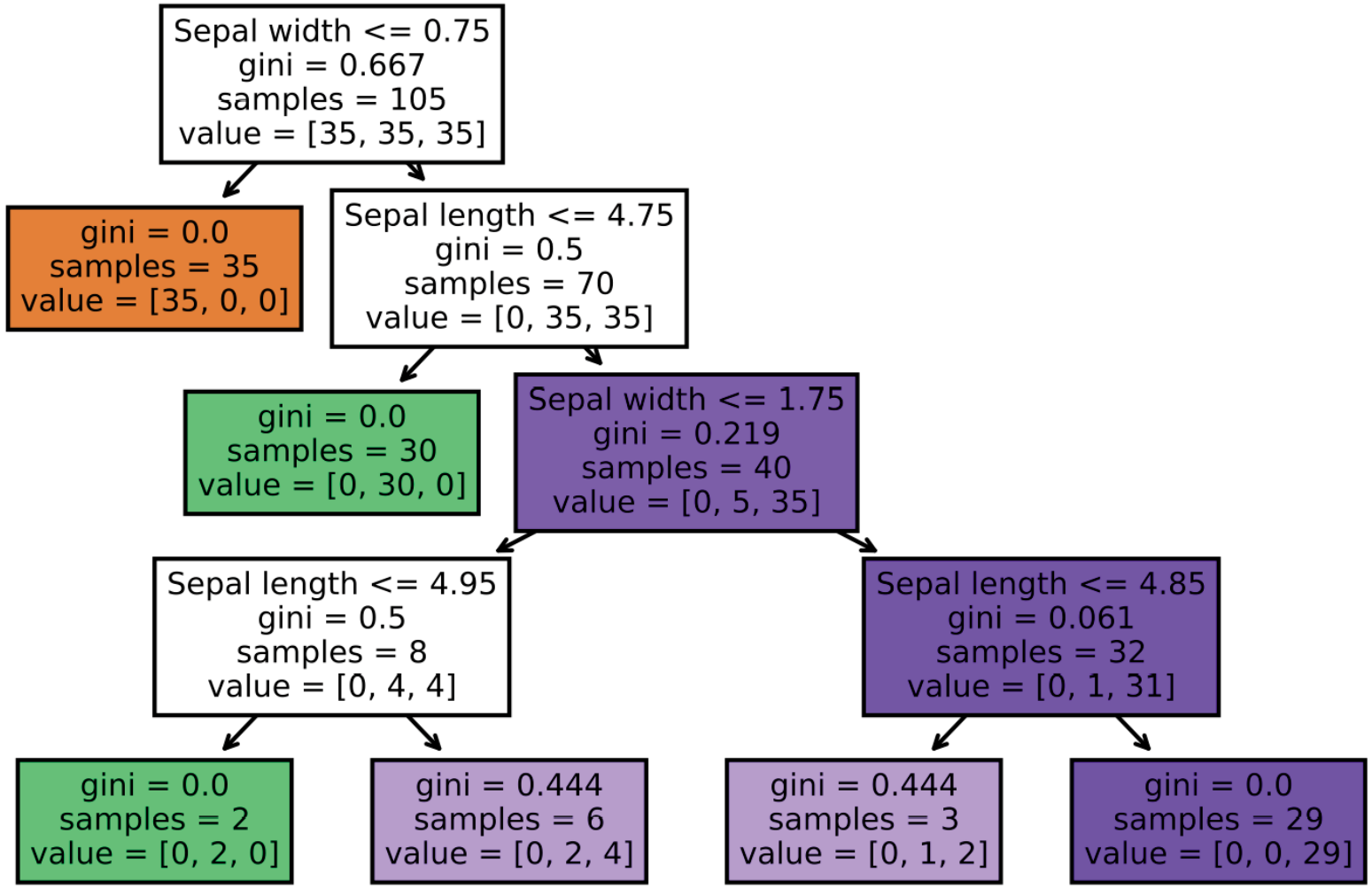


Visualization



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Running Decision Tree Algorithm



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Conclusion:

- Decision trees are the single most popular data mining tool
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- It's possible to get in trouble with overfitting
- They do classification: predict a categorical output from categorical and/or real inputs

Advantages of Decision Trees - I

- Simple to understand and interpret. People are able to understand decision tree models after a brief explanation. Trees can also be displayed graphically in a way that is easy for non-experts to interpret.
- Able to handle both numerical and categorical data.
- Requires little data preparation. Other techniques often require data normalization. Since trees can handle qualitative predictors, there is no need to create dummy variables.
- Uses a white box or open-box model. If a given situation is observable in a model the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model, the explanation for the results is typically difficult to understand, for example with an artificial neural network.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

https://en.wikipedia.org/wiki/Decision_tree_learning

Advantages of Decision Trees - II

- Non-parametric approach that makes no assumptions of the training data or prediction residuals; e.g., no distributional, independence, or constant variance assumptions
- Performs well with large datasets. Large amounts of data can be analyzed using standard computing resources in reasonable time.
- Mirrors human decision making more closely than other approaches.
- Robust against co-linearity, particularly boosting.
- In built feature selection. Additional irrelevant feature will be less used so that they can be removed on subsequent runs. The hierarchy of attributes in a decision tree reflects the importance of attributes. It means that the features on top are the most informative.
- Decision trees can approximate any Boolean function e.g. XOR.

https://en.wikipedia.org/wiki/Decision_tree_learning

Disadvantages of Decision Trees

- Trees can be very non-robust. A small change in the training data can result in a large change in the tree and consequently the final predictions.
- The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree.
- Decision-tree learners can create over-complex trees that do not generalize well from the training data (overfitting.) Mechanisms such as pruning are necessary to avoid this problem
- The average depth of the tree that is defined by the number of nodes or tests till classification is not guaranteed to be minimal or small under various splitting criteria.
- For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of attributes with more levels.

https://en.wikipedia.org/wiki/Decision_tree_learning