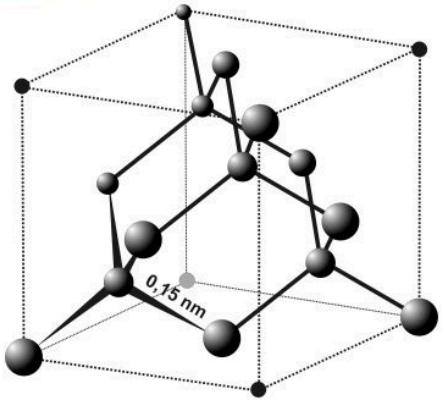
Lecture #9: Course wrap up

Slides by Artem Dembitskiy

Materials science

- studies the relationship between the structure and properties of materials
- develops new materials to enable new technolgies, which improve our lives (hopefully)



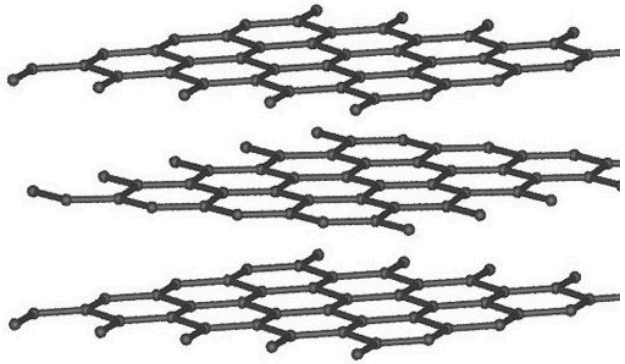


Image source

Tasks that materials science solves:

- Design
- Imrove
- Understand reasons of a failure
- ...

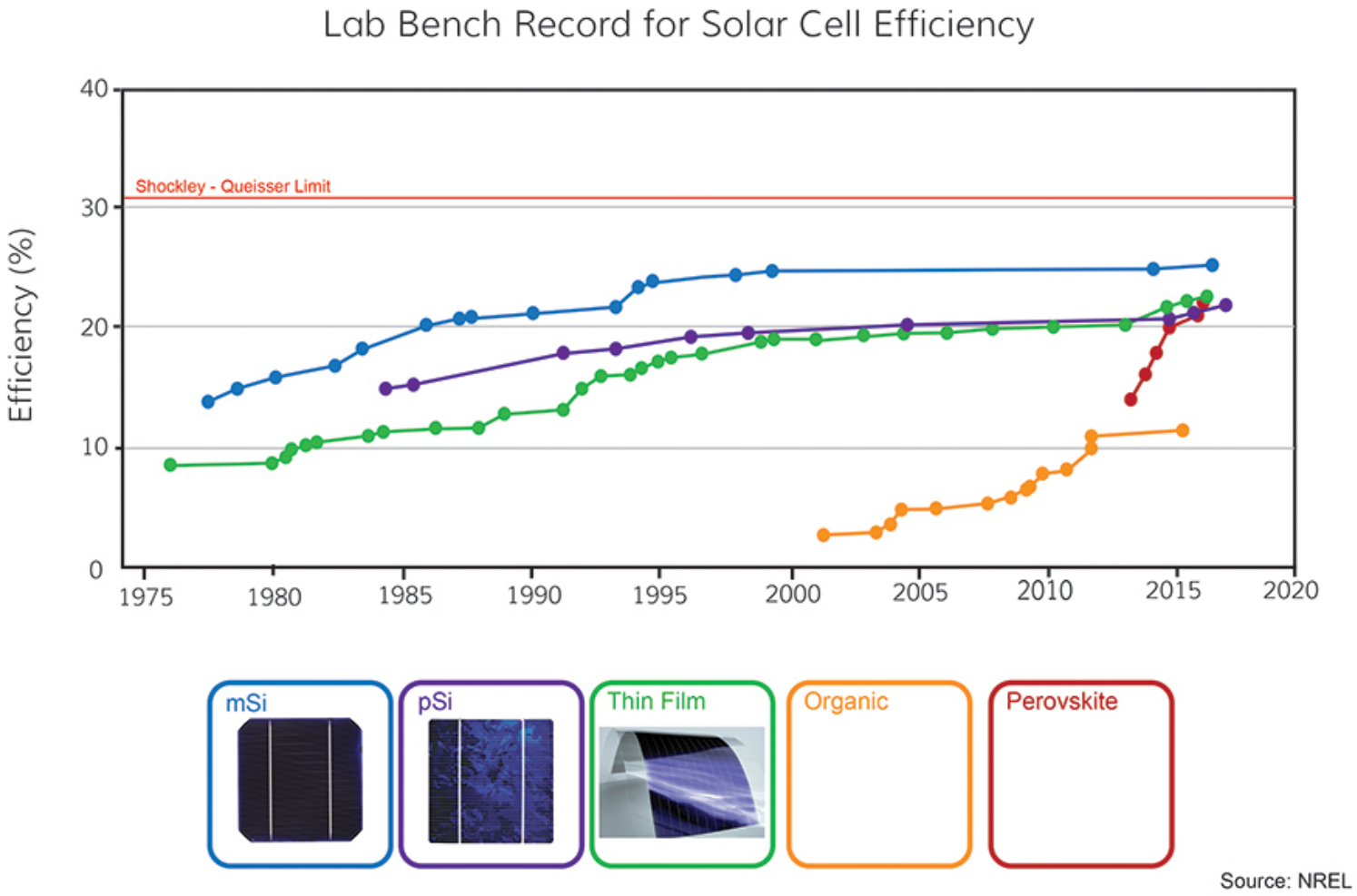
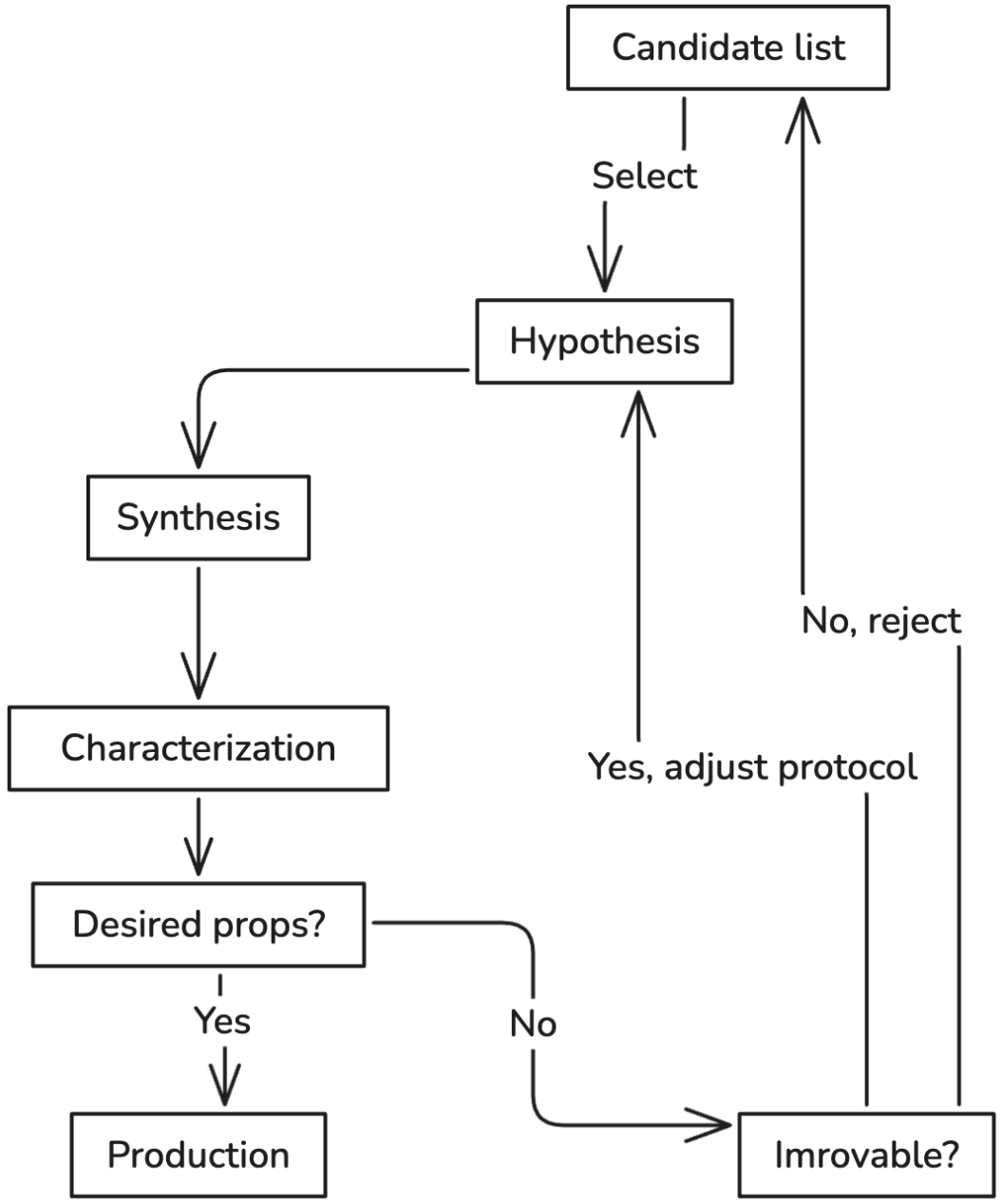


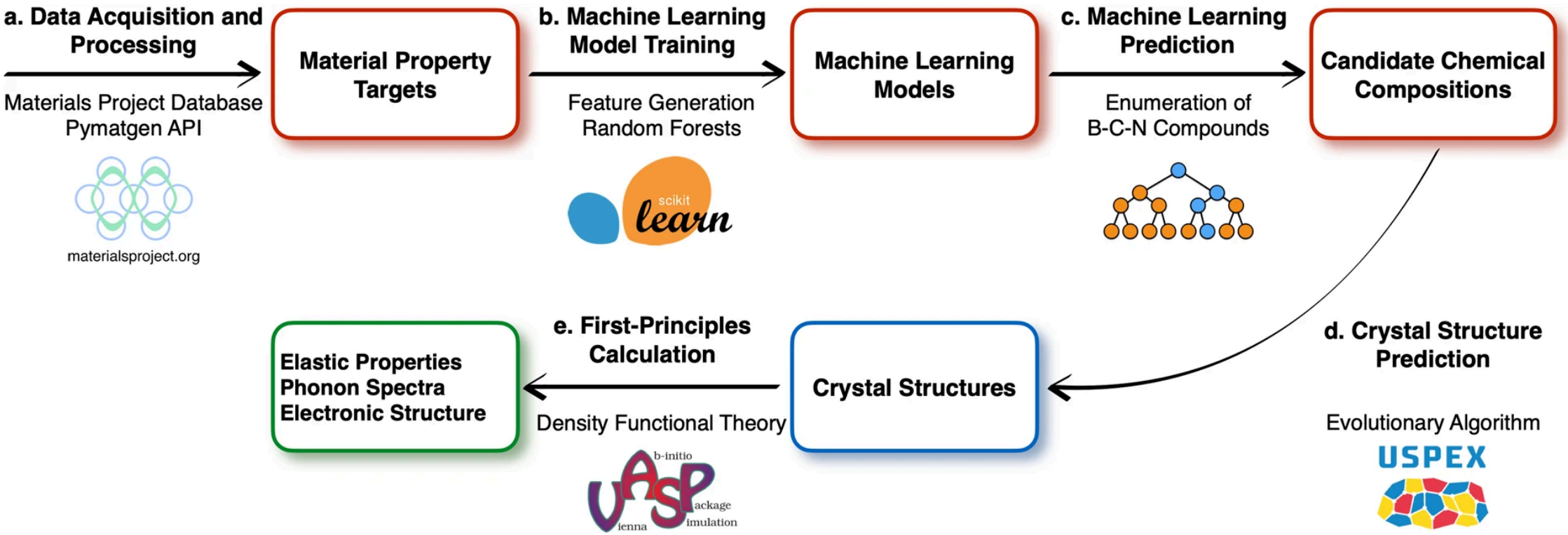
Image source

Trial-and-error



Materirals informatics

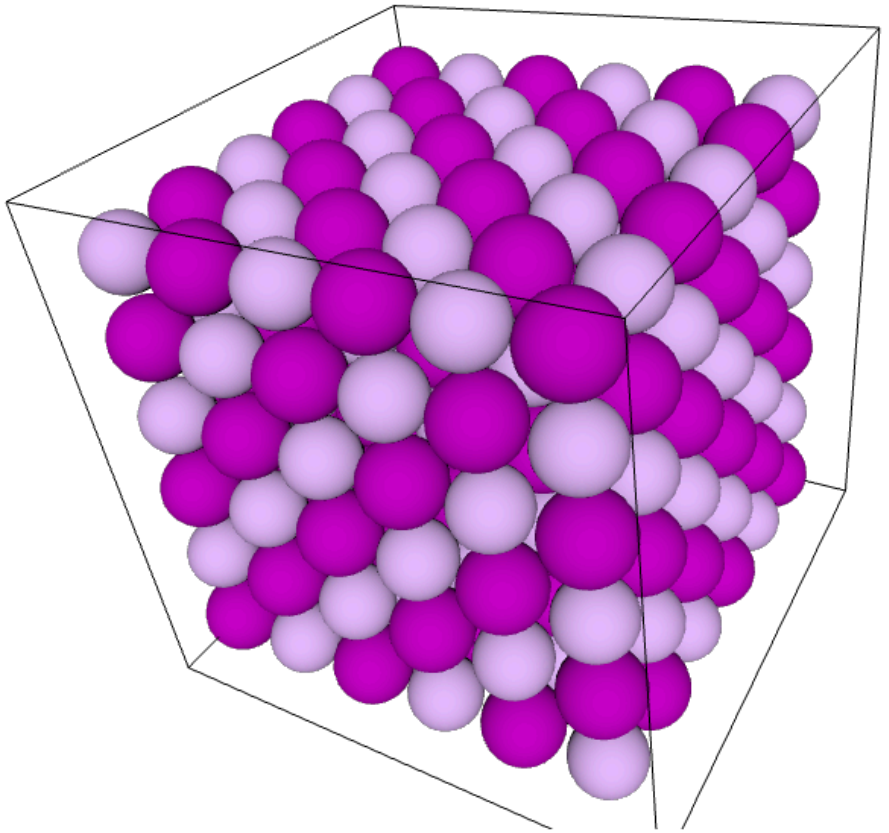
"... is a field of study that applies the principles of informatics and data science to materials science and engineering to improve the understanding, use, selection, development, and discovery of materials." (wiki)



Python tools for atomistic modeling

```
from ase.io import read, write
from ase.visualize import view
from ase.build import make_supercell

atoms = read('data/LiI.cif')
sc = make_supercell(atoms,
  [[3, 0, 0 ],
   [0, 3, 0],
   [0, 0, 3]]
   )
view(sc, viewer='x3d')
write('data/LiI_3x3x3.cif', sc)
```



Automation:

- saves your time
- gives you the opportunity (tools) to realize your ideas
- reduces number of mistakes (typos) made

Data

Why use someone else's data?

- guide to your research objective
- reference
- baseline
- insight
- explanation
- enrichment
- time

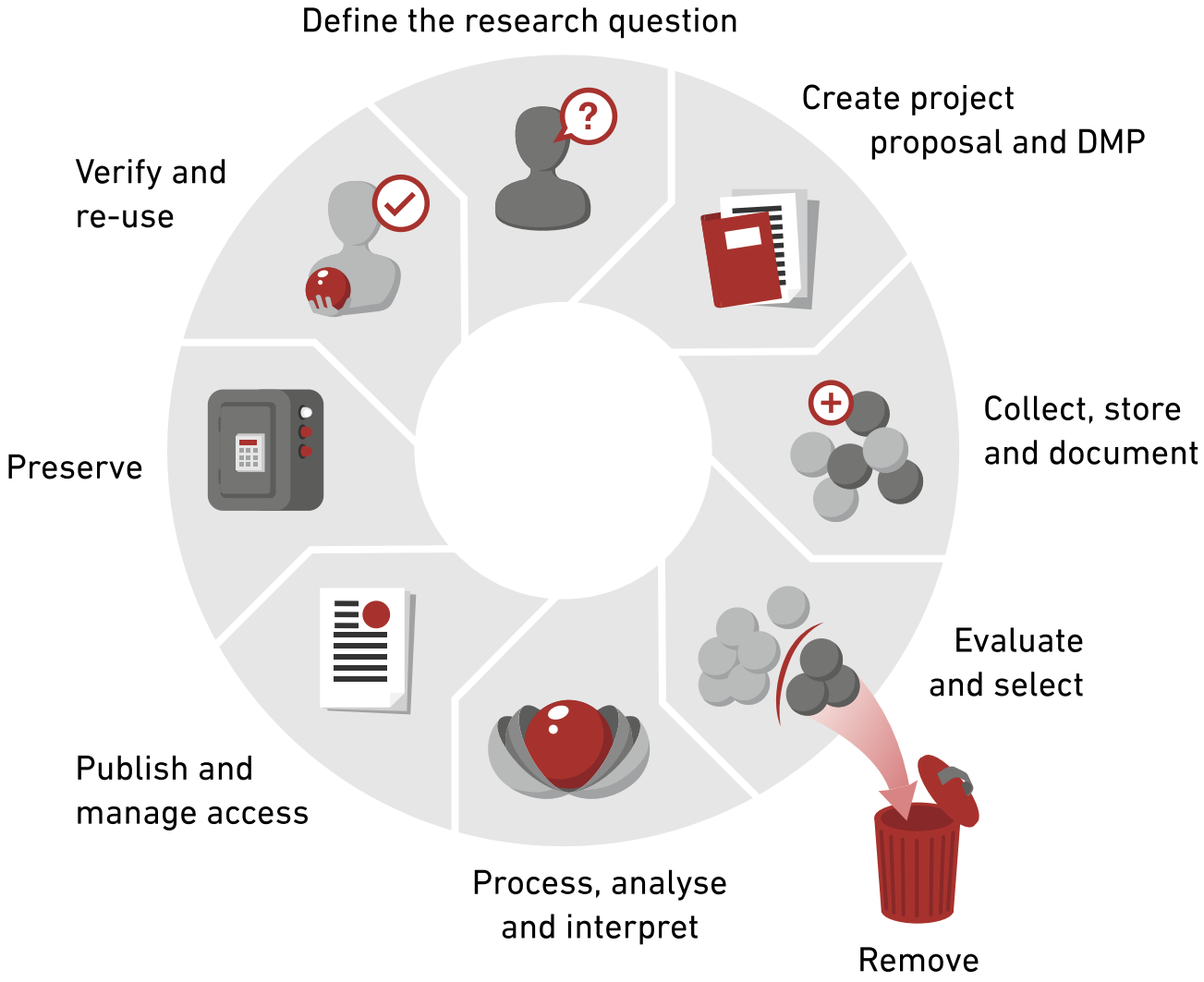
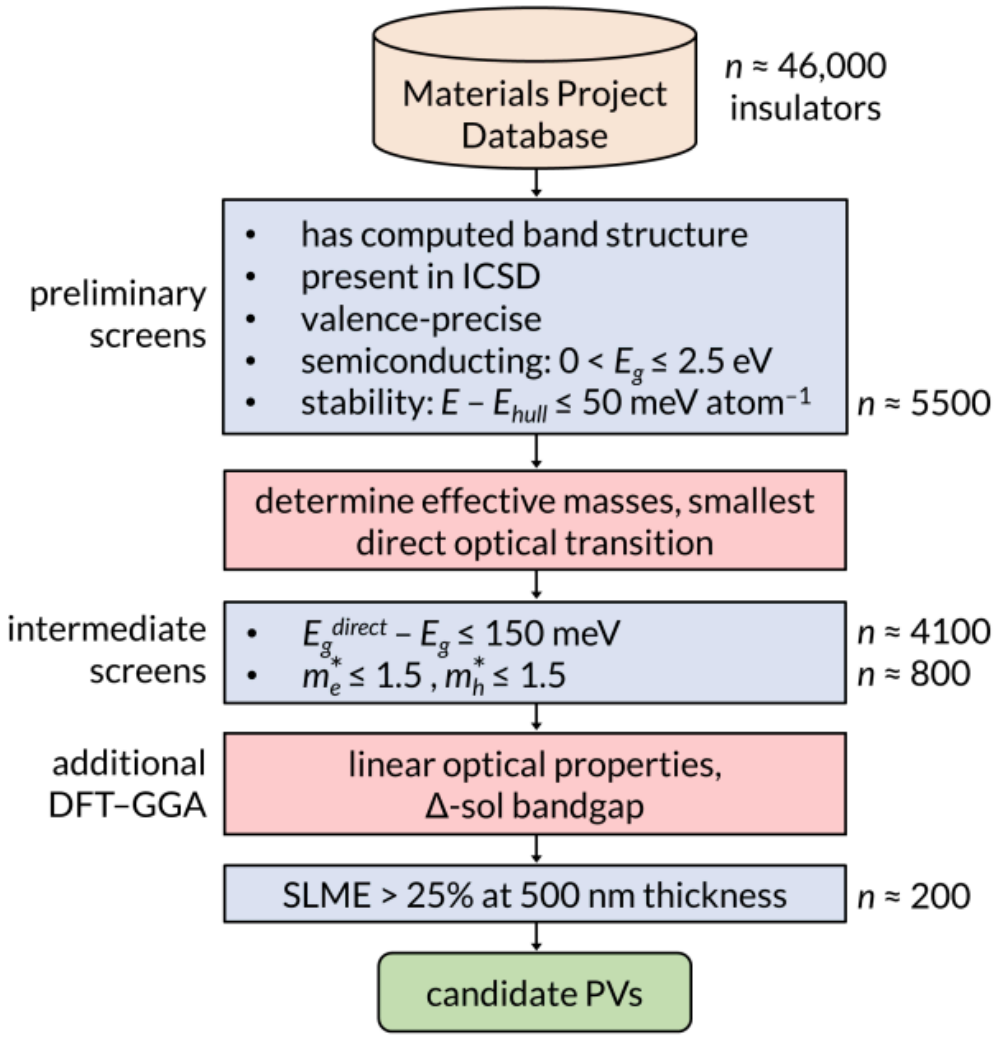


Image source

The MP's data usage example

Screening inorganic PVs

- Screened database
- Identified candidates
- Calculated properties of interest for ~800 compounds
- Shared the data



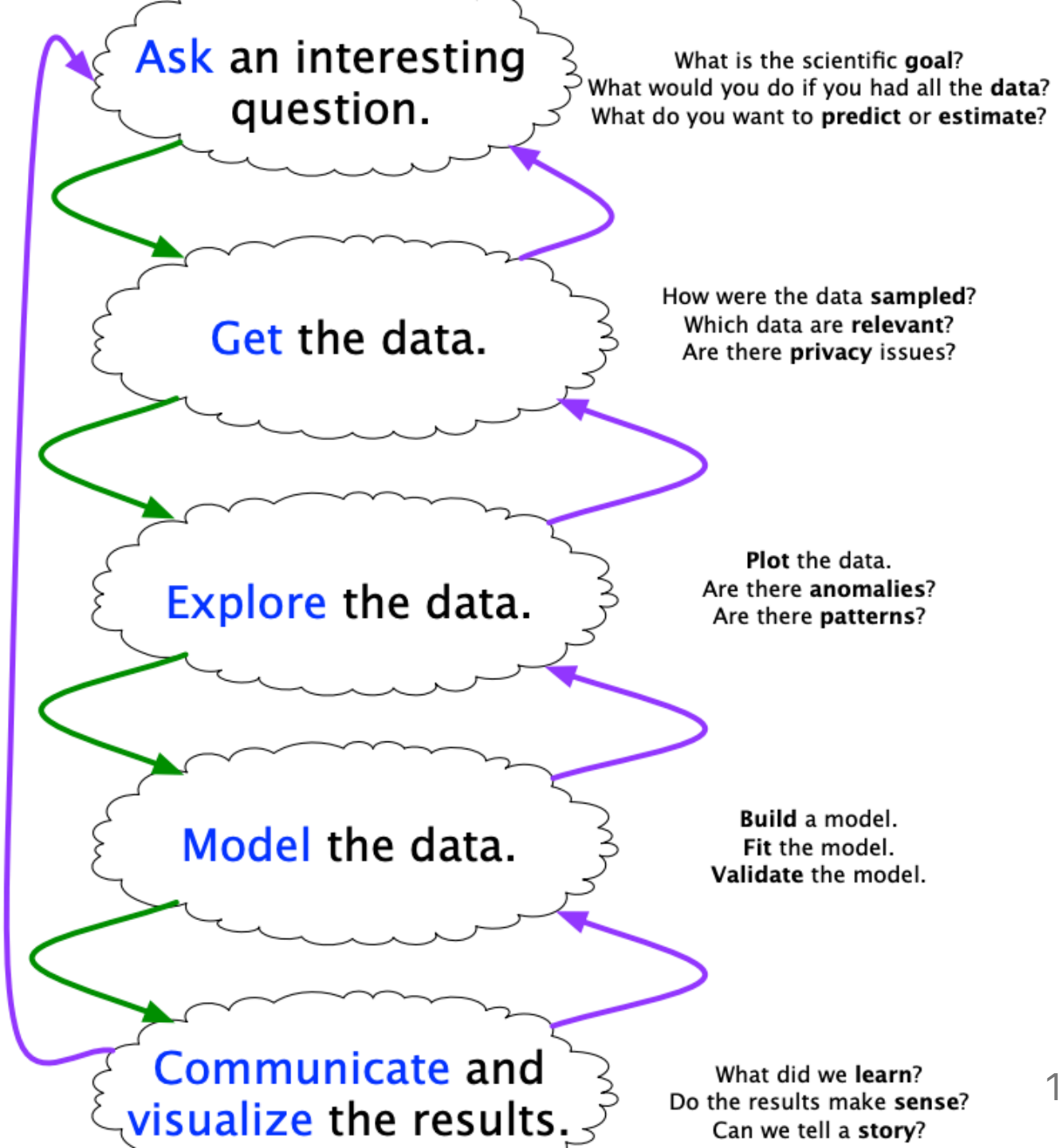
Candidate Inorganic Photovoltaic Materials from Electronic Structure-Based Optical Absorption and Charge Transport Proxies

The FAIR Guiding principles

... for scientific data management and stewardship

Image source

The data science workflow

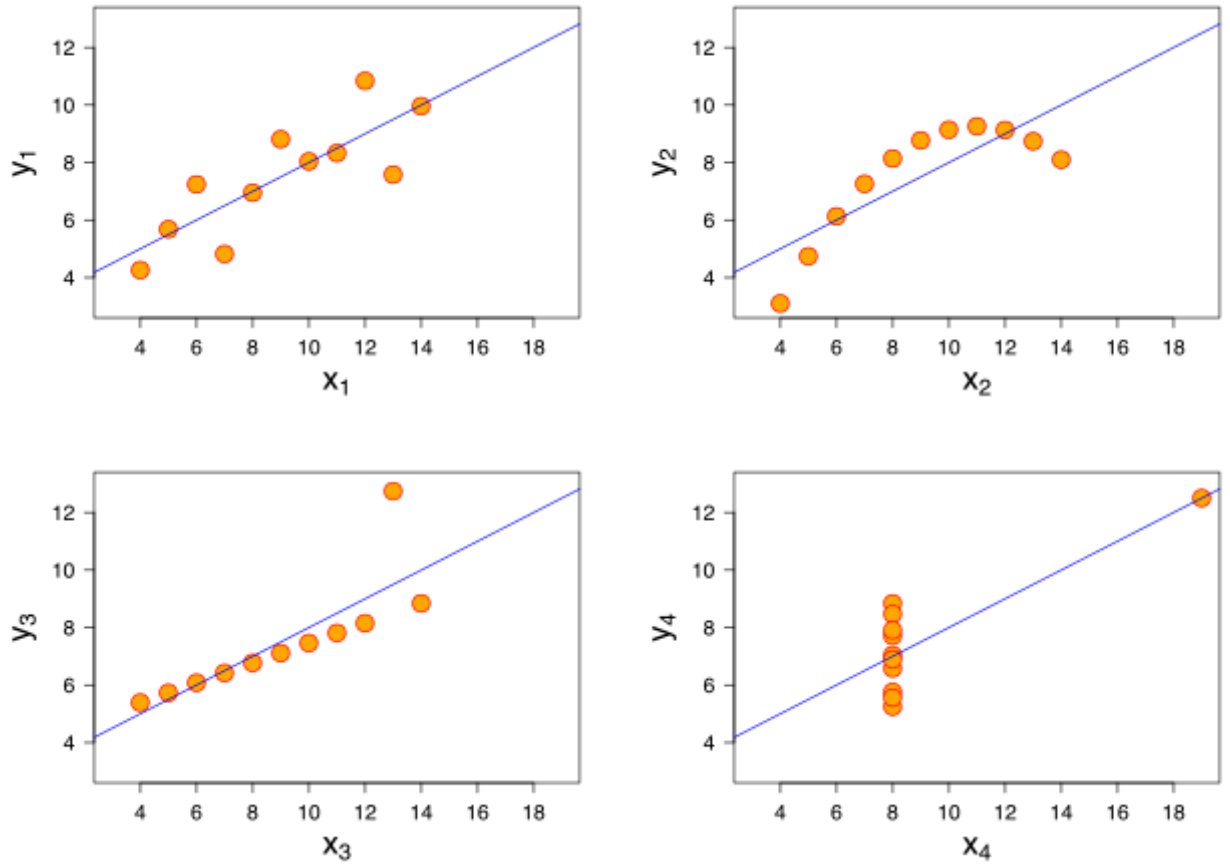


From CS 109a: Data Science, Effective Exploratory Data Analysis and Visualization by Pavlos Protopapas & Kevin Rader slide #2

Why is visual inspection of data important?

- Same descriptive statistics
- Very different distributions

https://en.wikipedia.org/wiki/Anscombe's_quartet



Exploratory data analysis pipeline

- Build data
- Clean data
- Explore global features
- Explore group features

Visulaization goals

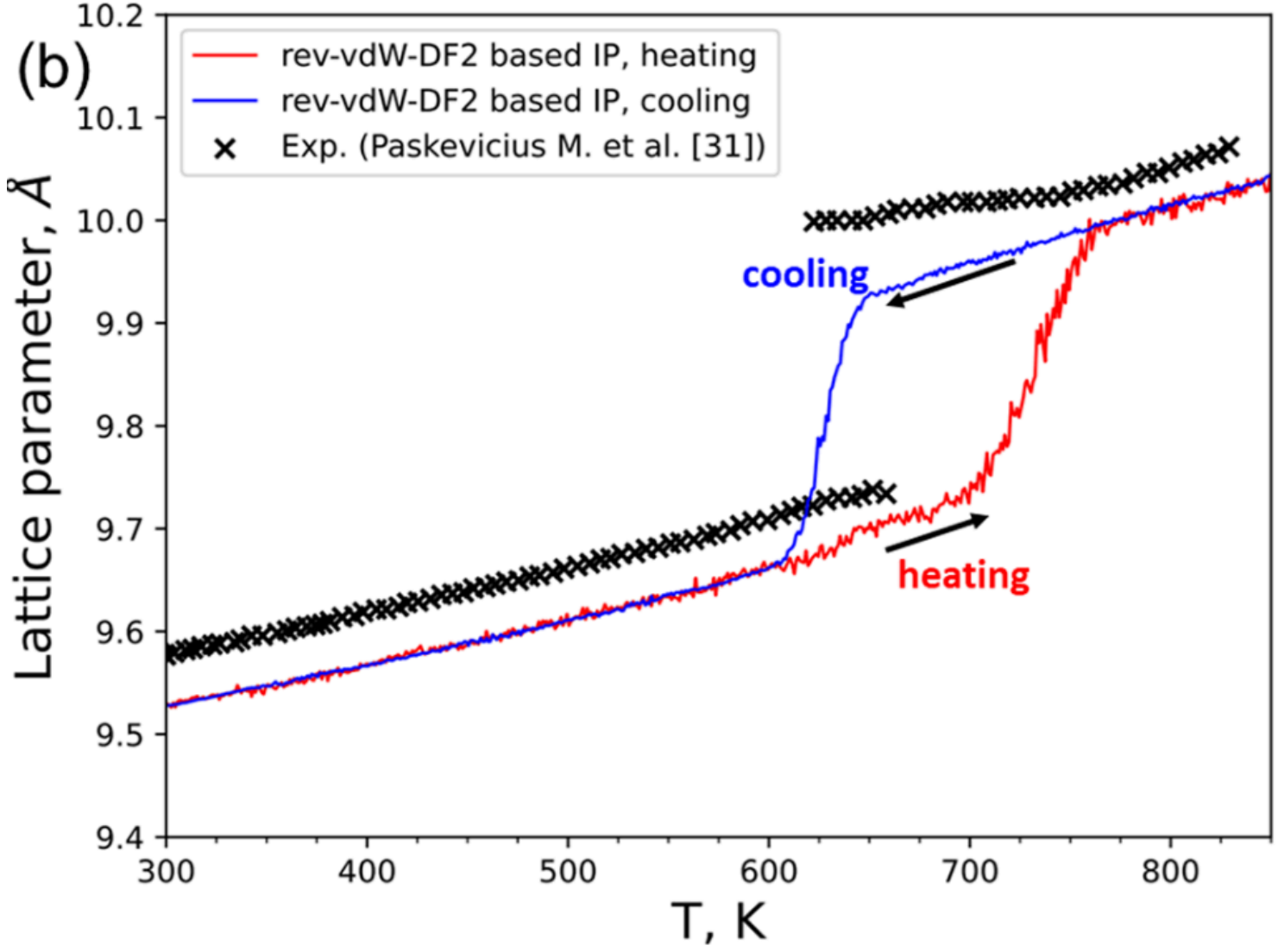
Communicate (Explanatory)

- Present data and ideas
- Explain and inform
- Provide evidence and support
- Influence and persuade

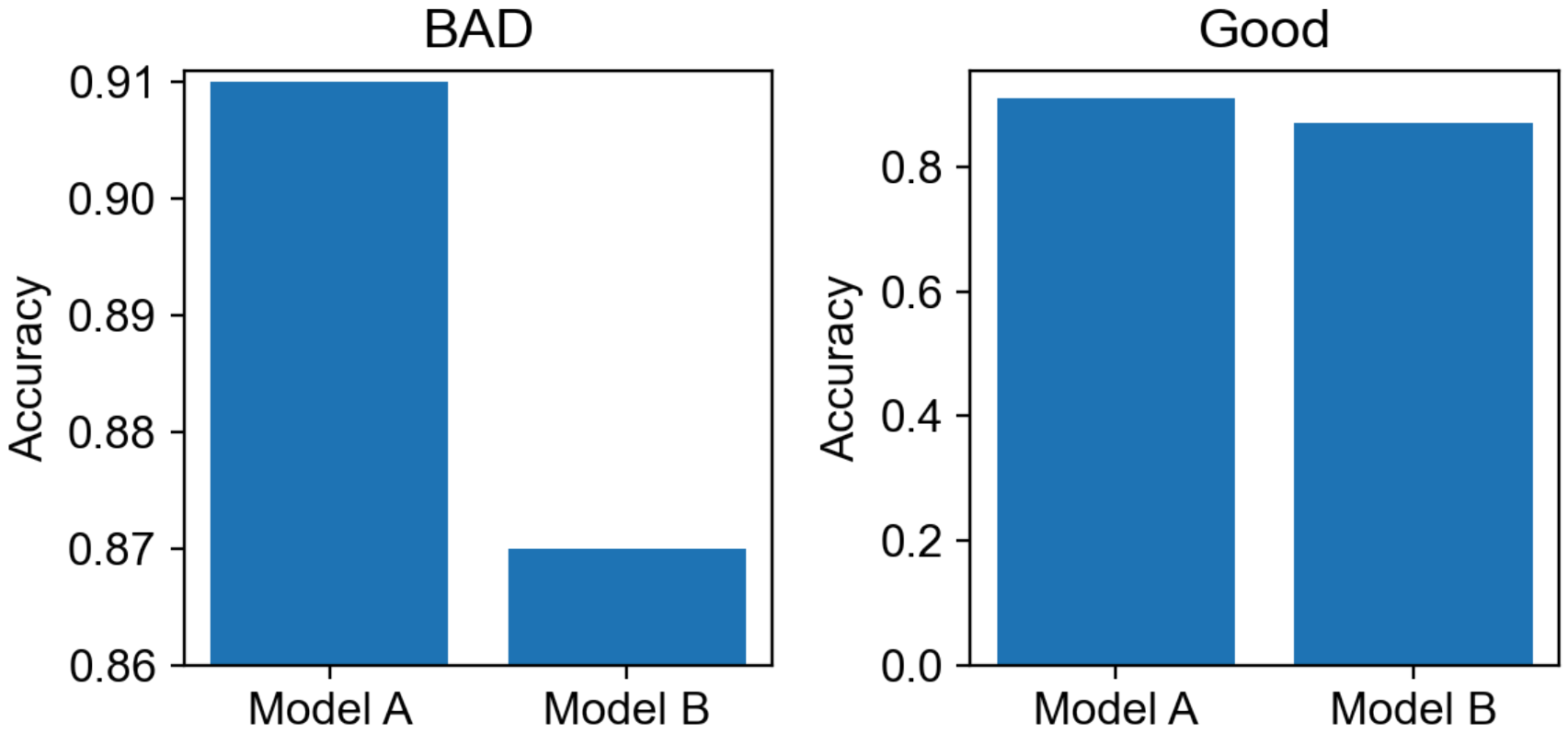
Analyze (Exploratory)

- Explore the data
- Assess a situation
- Determine how to proceed
- Decide what to do

Explore

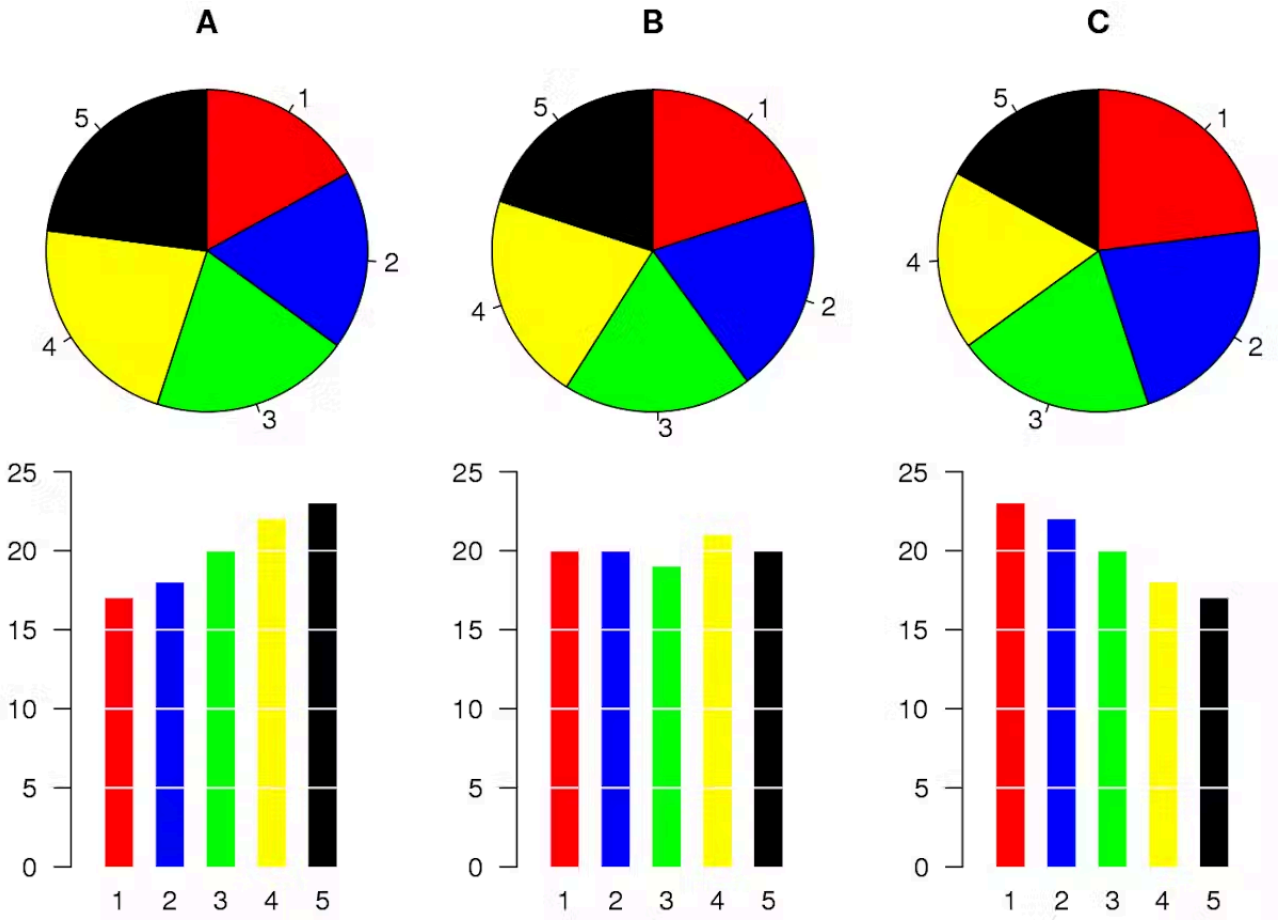

(b)
rev-vdW-DF2 based IP, heating
rev-vdW-DF2 based IP, cooling
Exp. (Paskevicius M. et al. [31])
cooling
heating
Lattice parameter, Å
T, K
10.2
10.1
10.0
9.9
9.8
9.7
9.6
9.5
9.4
300
400
500
600
700
800

Lie factor


BAD
Accuracy
0.91
0.90
0.89
0.88
0.87
0.86
Model A
Model B
Good
Accuracy
0.8
0.6
0.4
0.2
0.0
Model A
Model B

Don't use pie charts

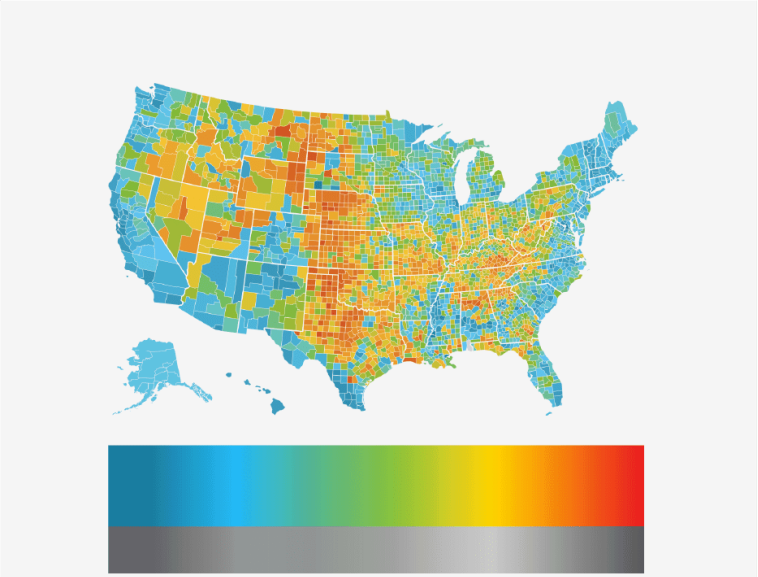
Barplots are easier to compare



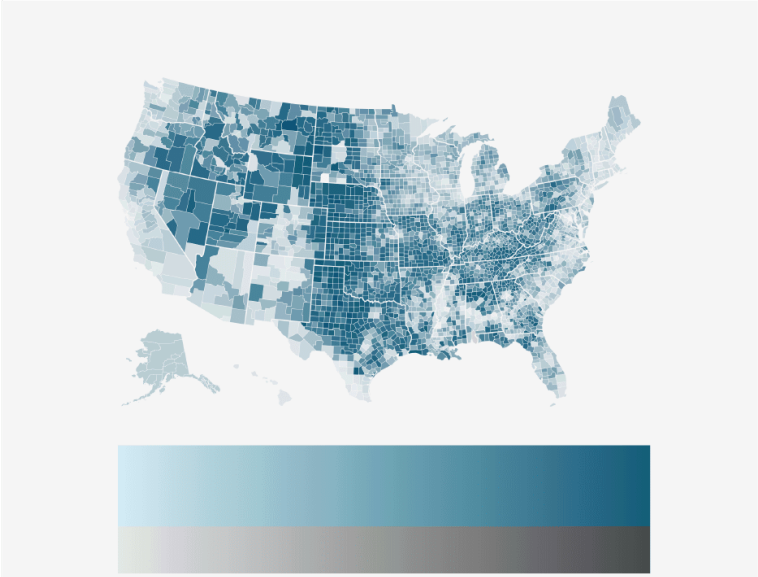
Here's why you should (almost) never use a pie chart for your data

Use color

Have a look at this page:
https://blog.datawrapper.de/colors/



NOT IDEAL

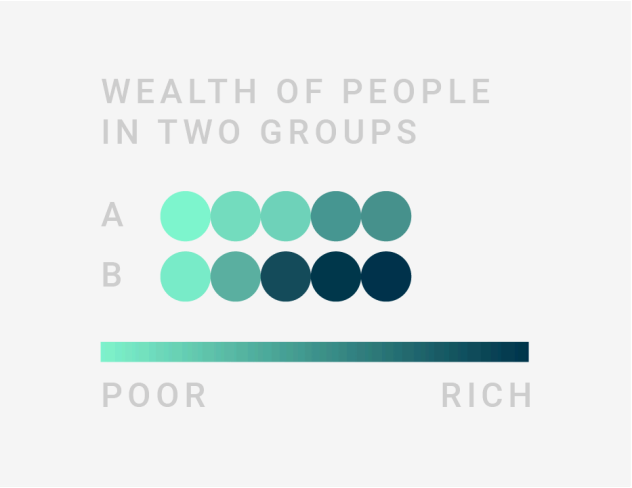


BETTER

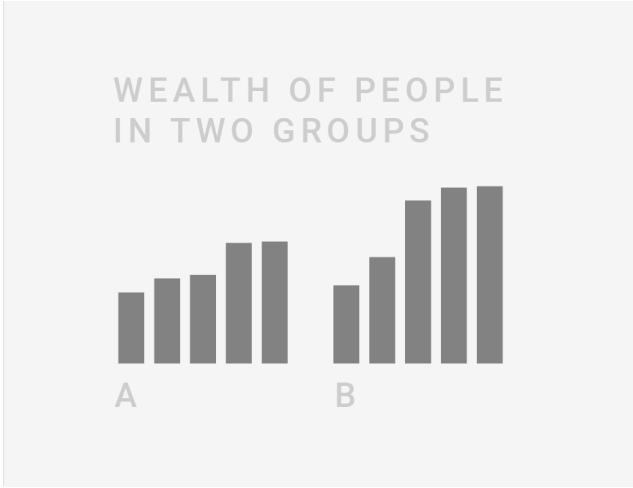
But consider a better alternative if possible

- the simpler the better

Have a look at this page:
https://blog.datawrapper.de/colors/

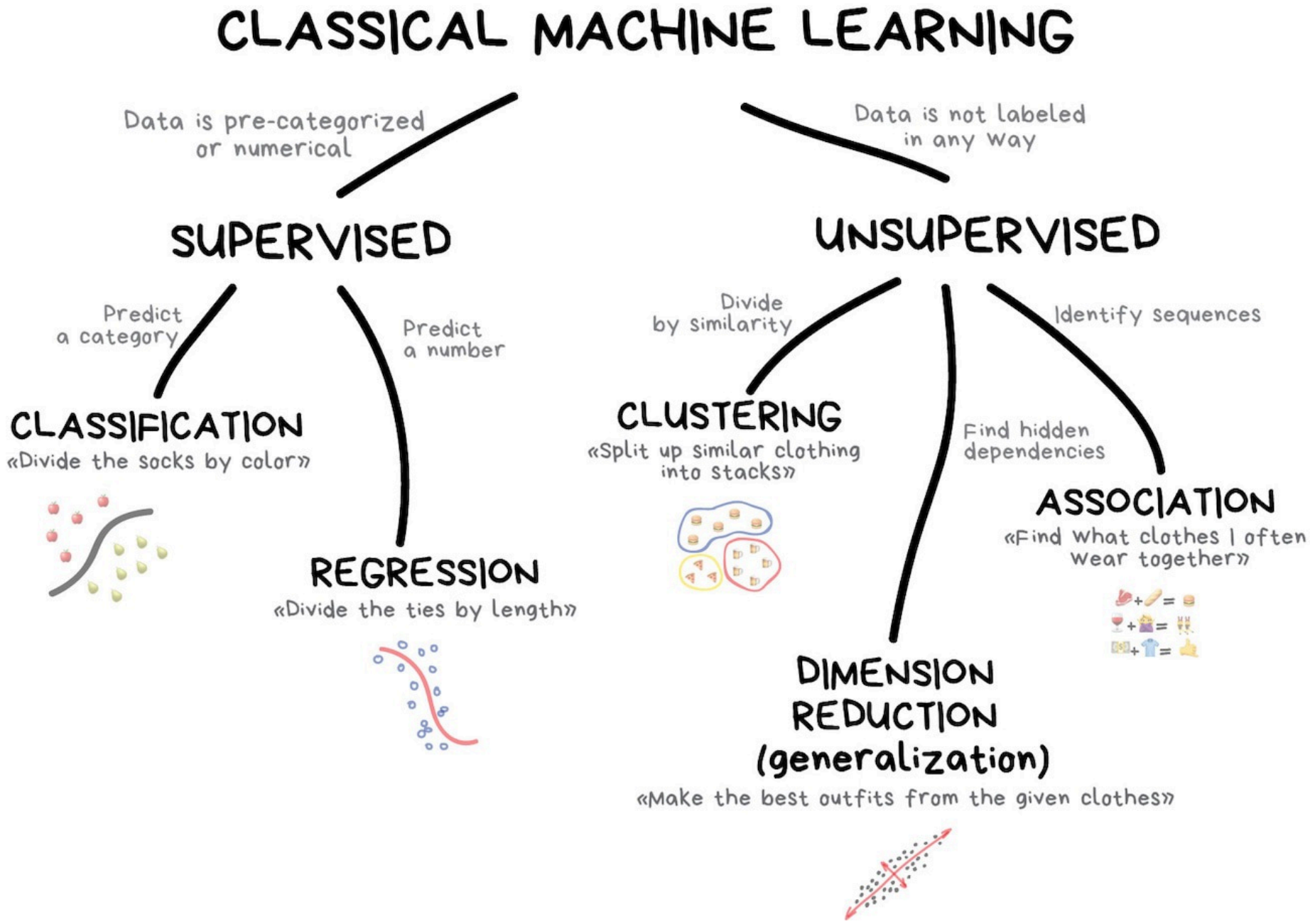


NOT IDEAL



BETTER

Classical ML tasks



Machine Learning for Everyone. In simple words. With real-world examples. Yes, again

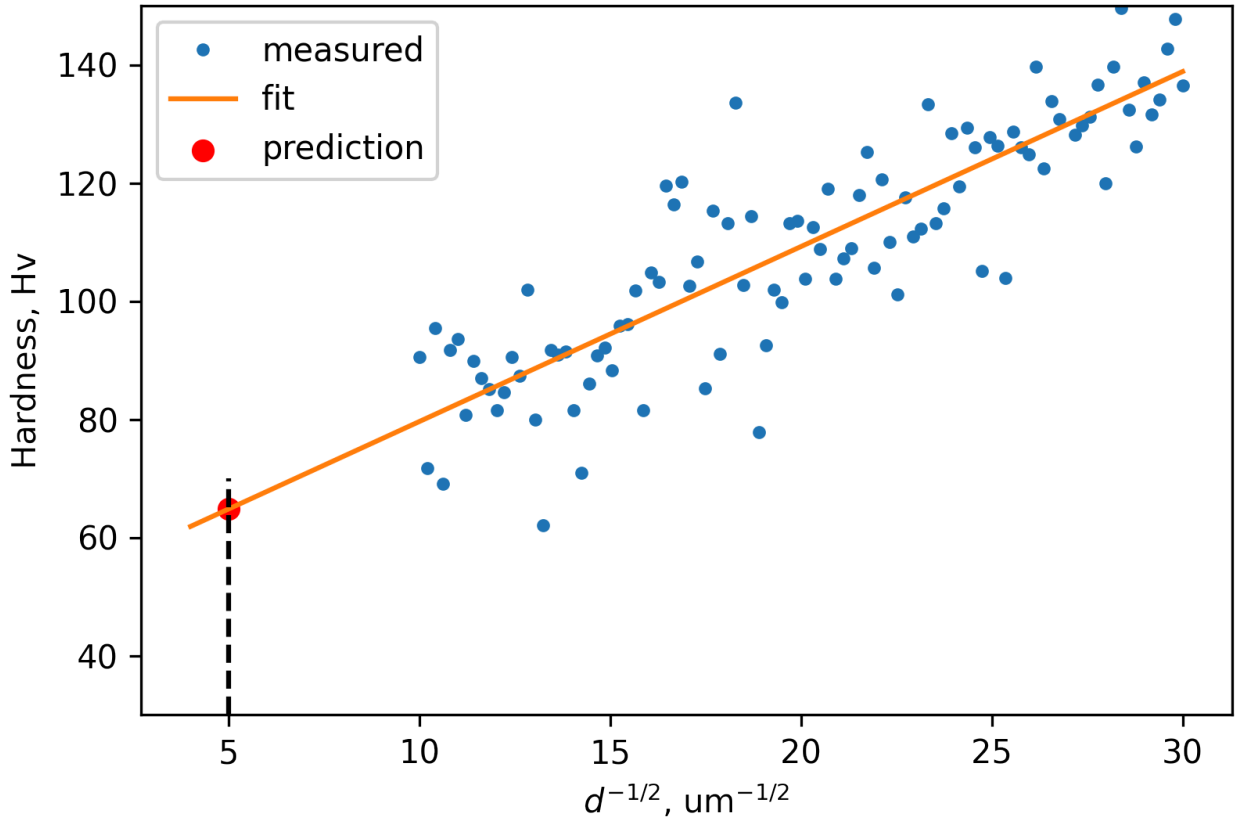
Supervised learning

"Tries to find the unknown function that connects known inputs to unknown outputs"

Model prediction:

Hardness($d^{-1/2}$ = 5 um$^{-1/2}$)?

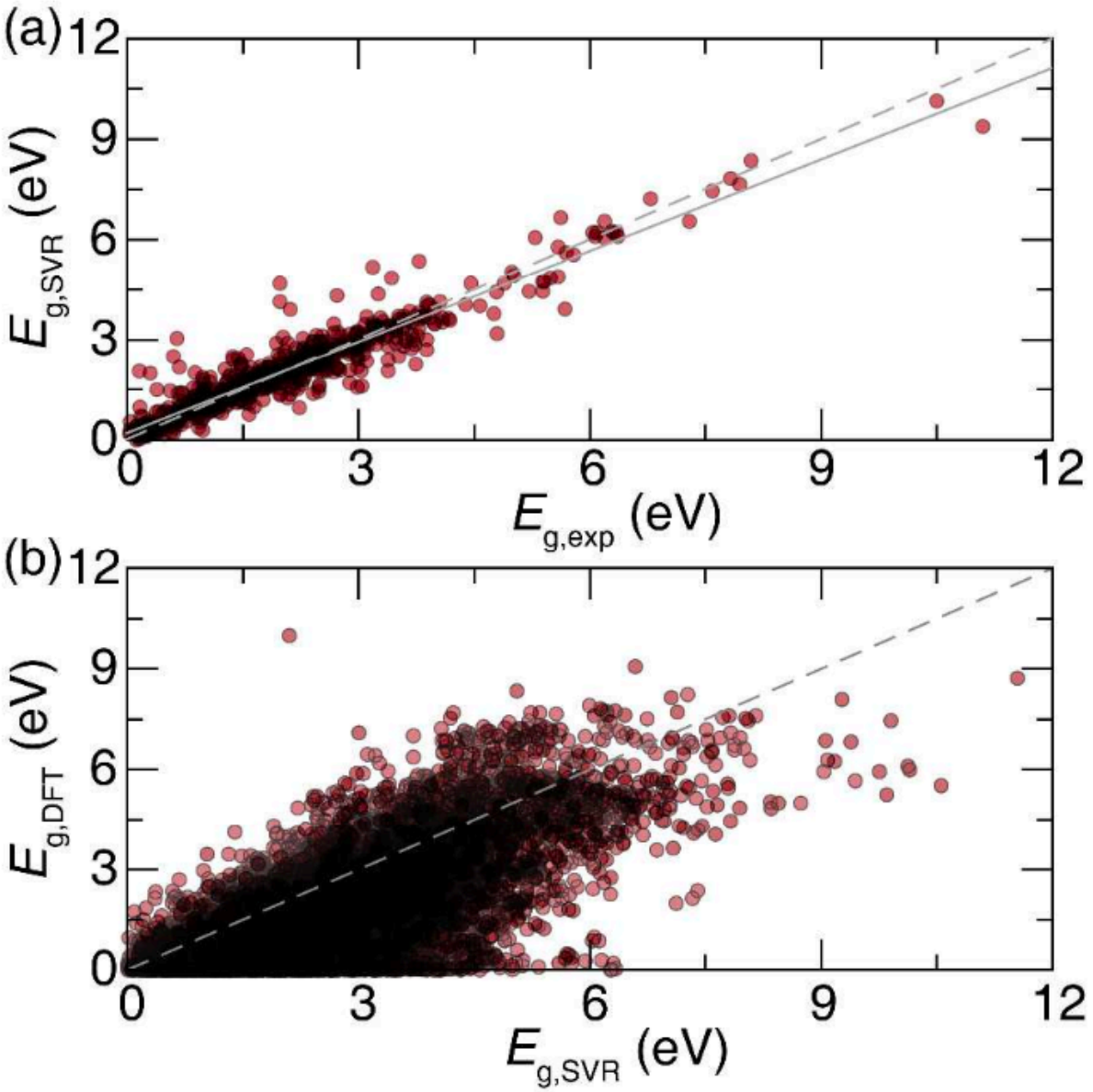
- input: grain size (known)
- output: hardness (unkown)



Surrogate model for predicting a band gap

- takes < 1 second to predict the property
- trained on existing data
- chemical composition only
- simple laptop is sufficient

SVR - Support Vector Regression

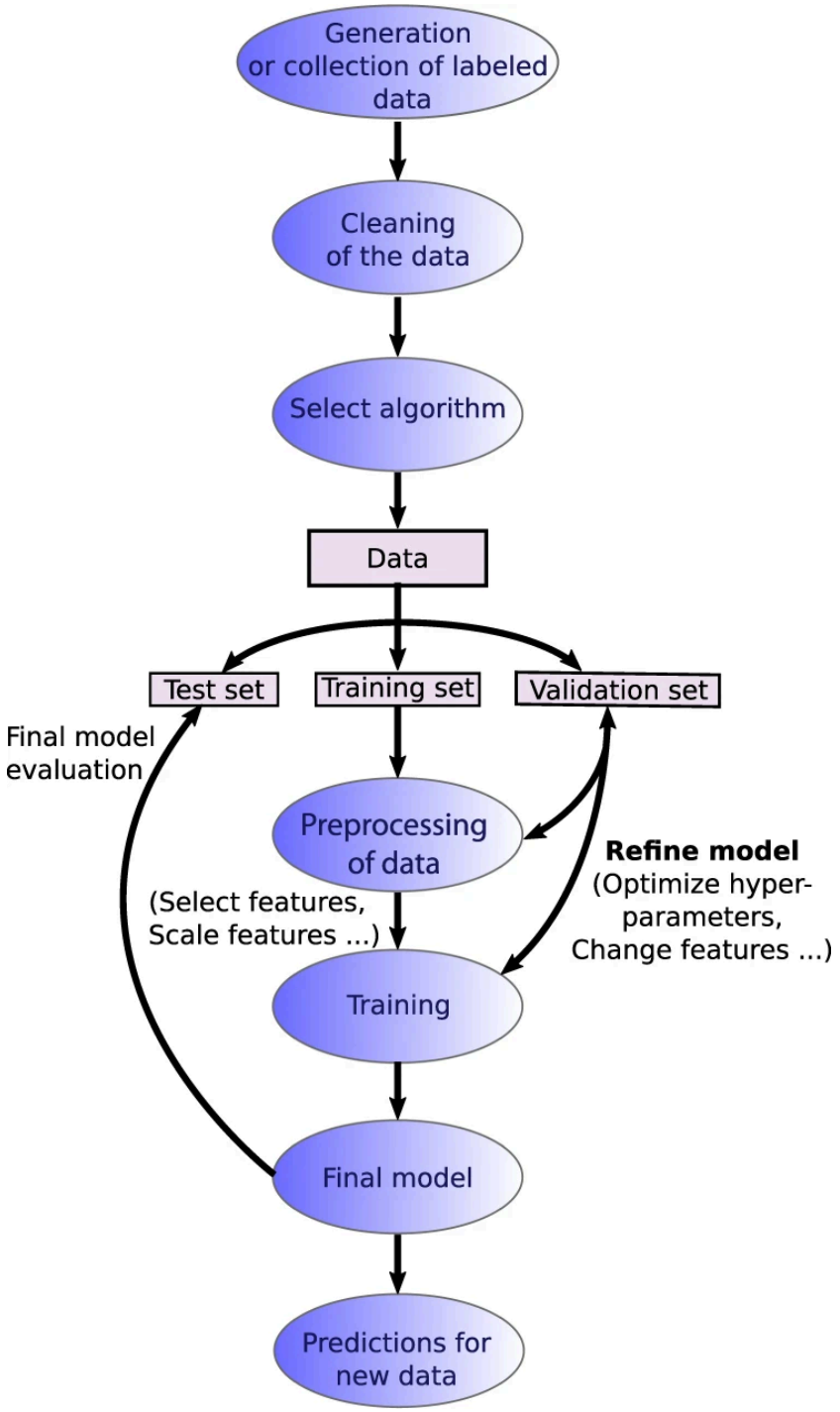


Limitations

- High quality data is required to fit the model
- Accuracy is good for screening stages (RMSE = 0.45 eV)
    - but poor for a more specific tasks

There is a trade off between speed and accuracy

Basic supervised learning workflow



Linear regression

$$J(\boldsymbol{w}, b) = \frac{1}{n} \sum_{i}^{n} (\boldsymbol{w}\boldsymbol{x}_{\boldsymbol{i}}^{T} + b - y_i)^2$$

Optimization problem

$$J(\boldsymbol{w}, b) \rightarrow \min_{w,b}$$

Hyperparameters?

- During the training (learning) process we find the best **parameters** (weights) of our model
- In addition we have **hyperparameters** which are usually fixed before the actual training process begins

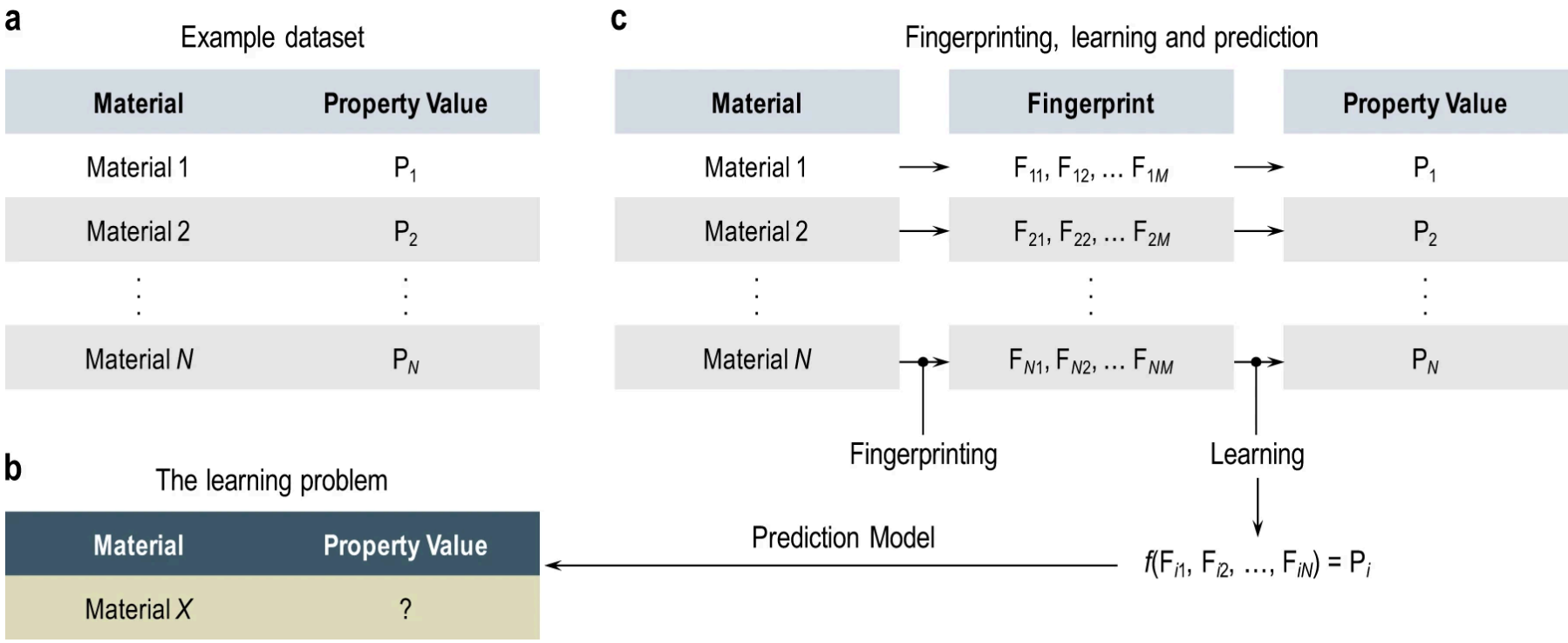
Example:

- number of trees in random forest, or depth of a decision tree
- λ in Ridge Regression
- They control the learning process itself rather than being learned from the data.

These parameters should be tuned to get better performance of the model

Features

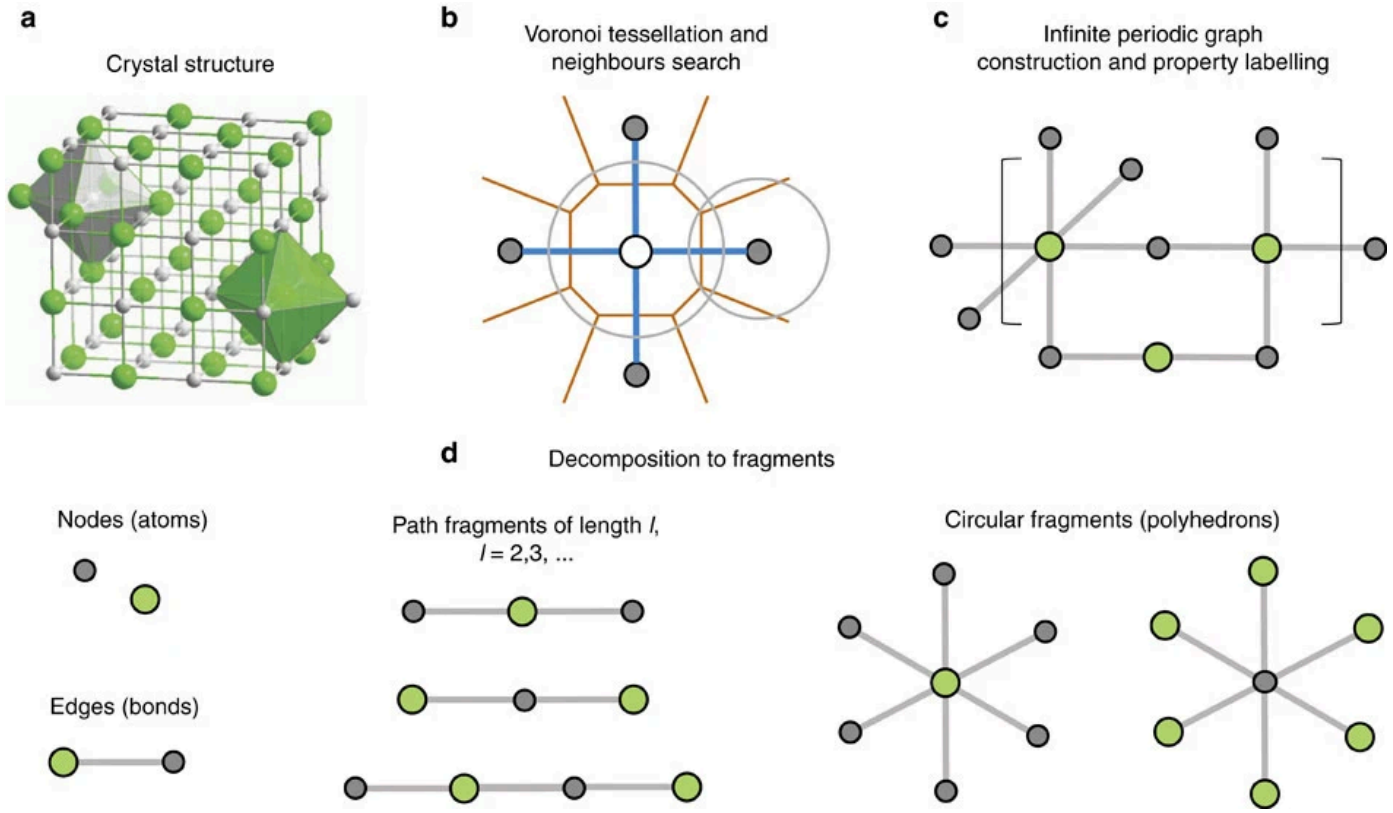
- In classical ML we use features
- to represent materials in a machine-readable format
- Think of it like a fingerprint



Heirarchy of features

Depending on the resolution we have:

- Local descriptors
  - site
- Fragment descriptors
  - bond
  - polyhedron
- Global descriptors
  - chemical family
  - structural type
  - density

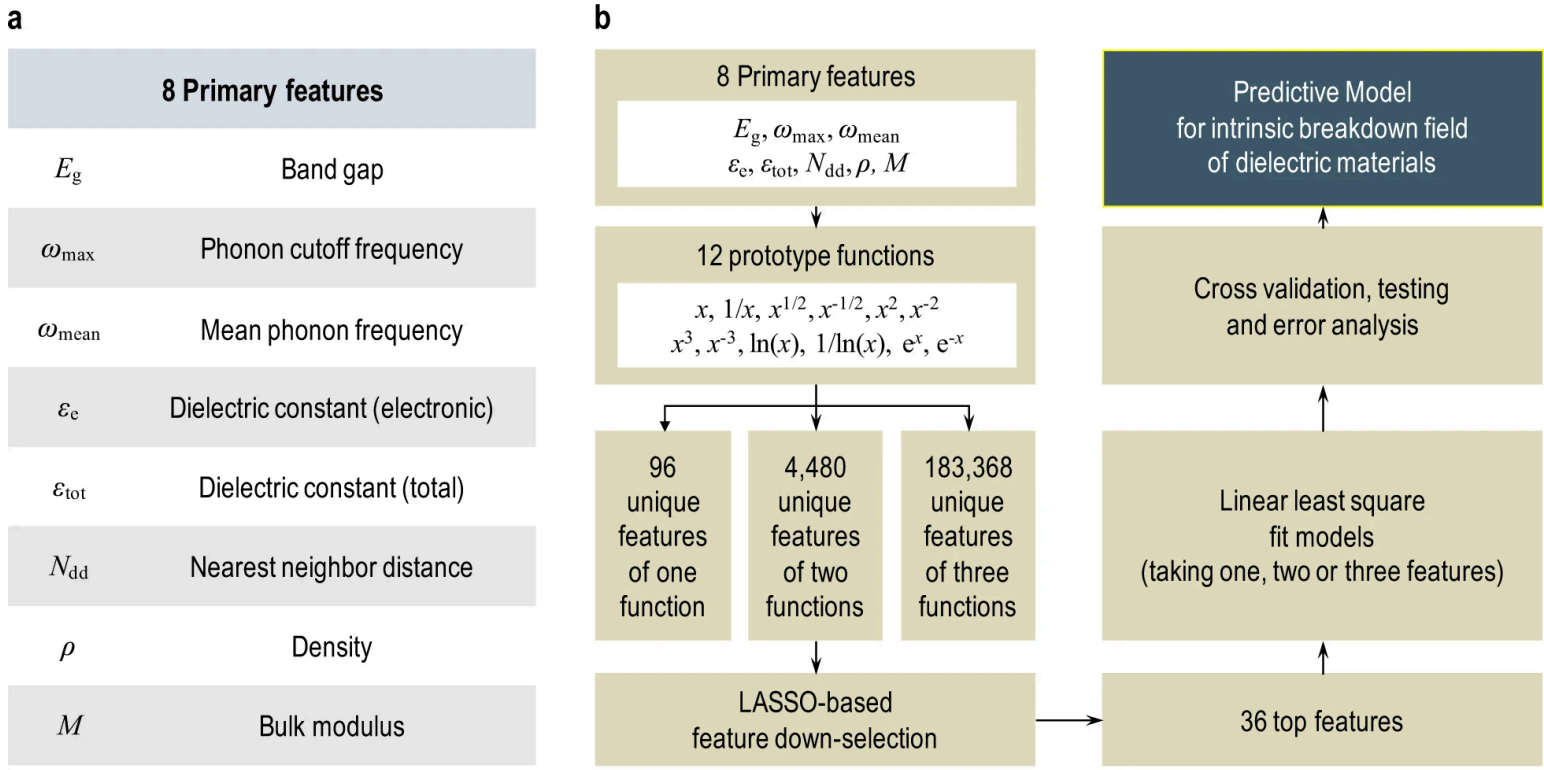


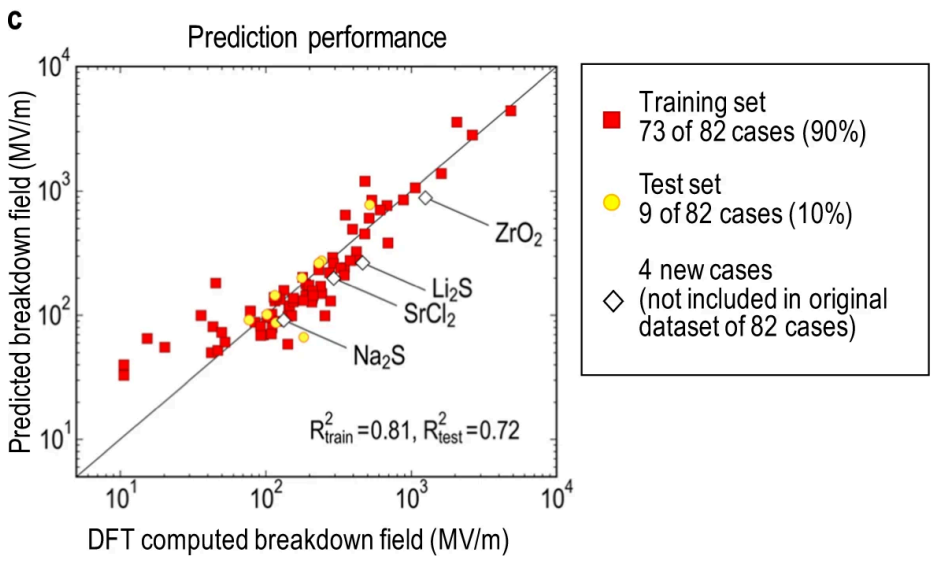
Universal fragment descriptors for predicting properties of inorganic crystals

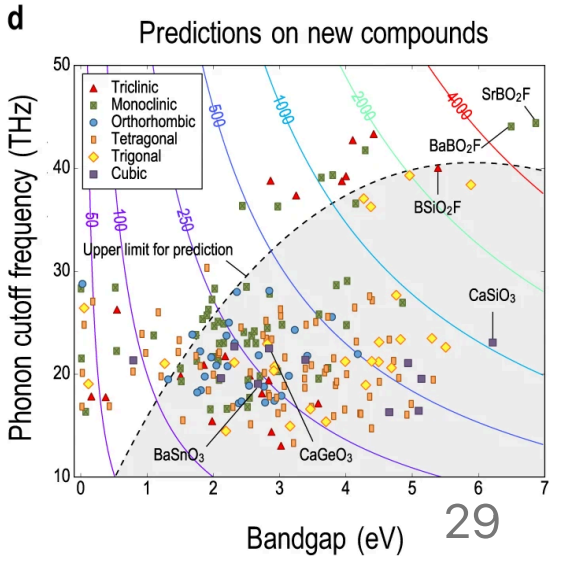
Feature engineering

- Primary descriptors are used to design more complex features
- e.g. by applying set of mathematical operators

See SISSO paper







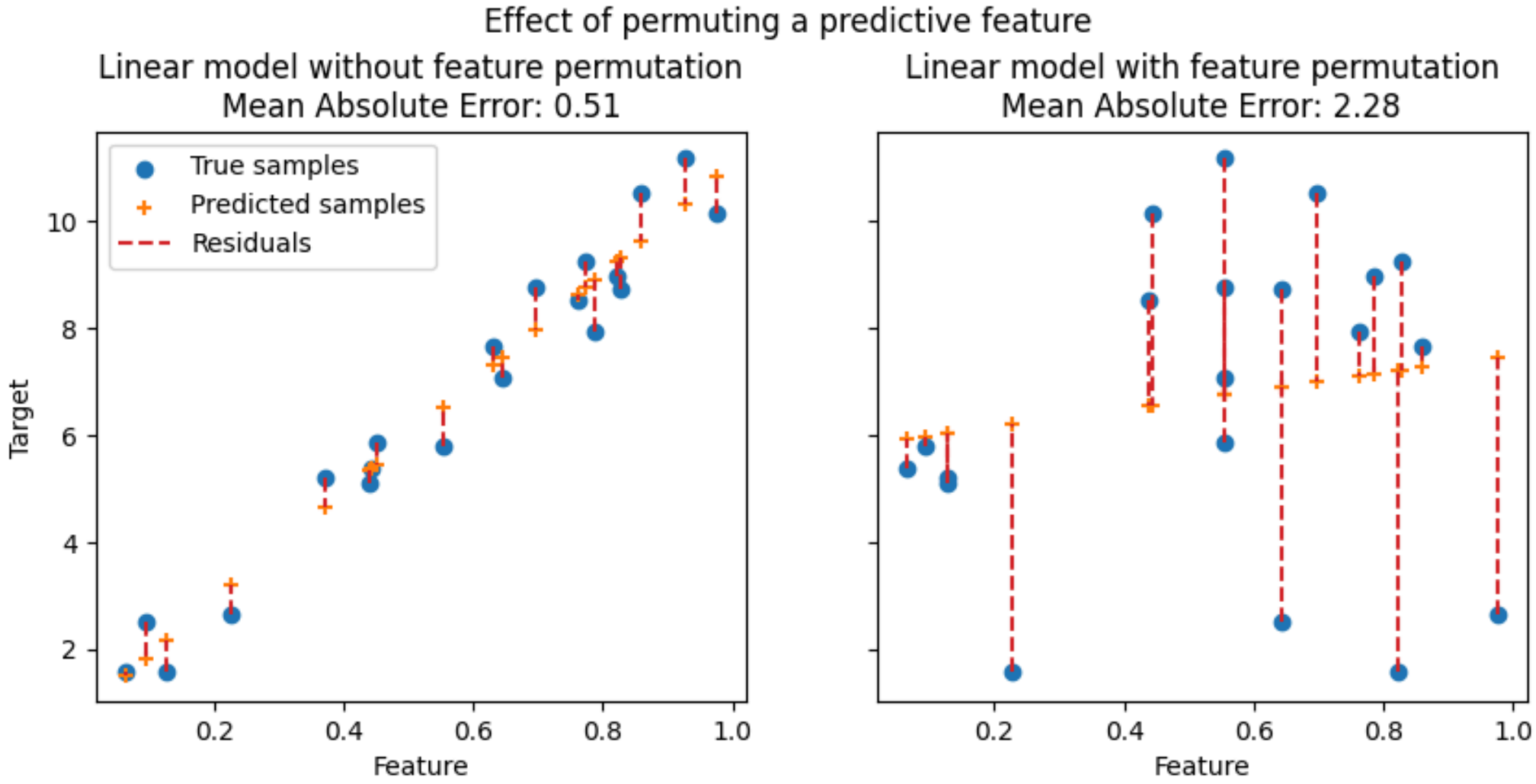
Permutation feature importance

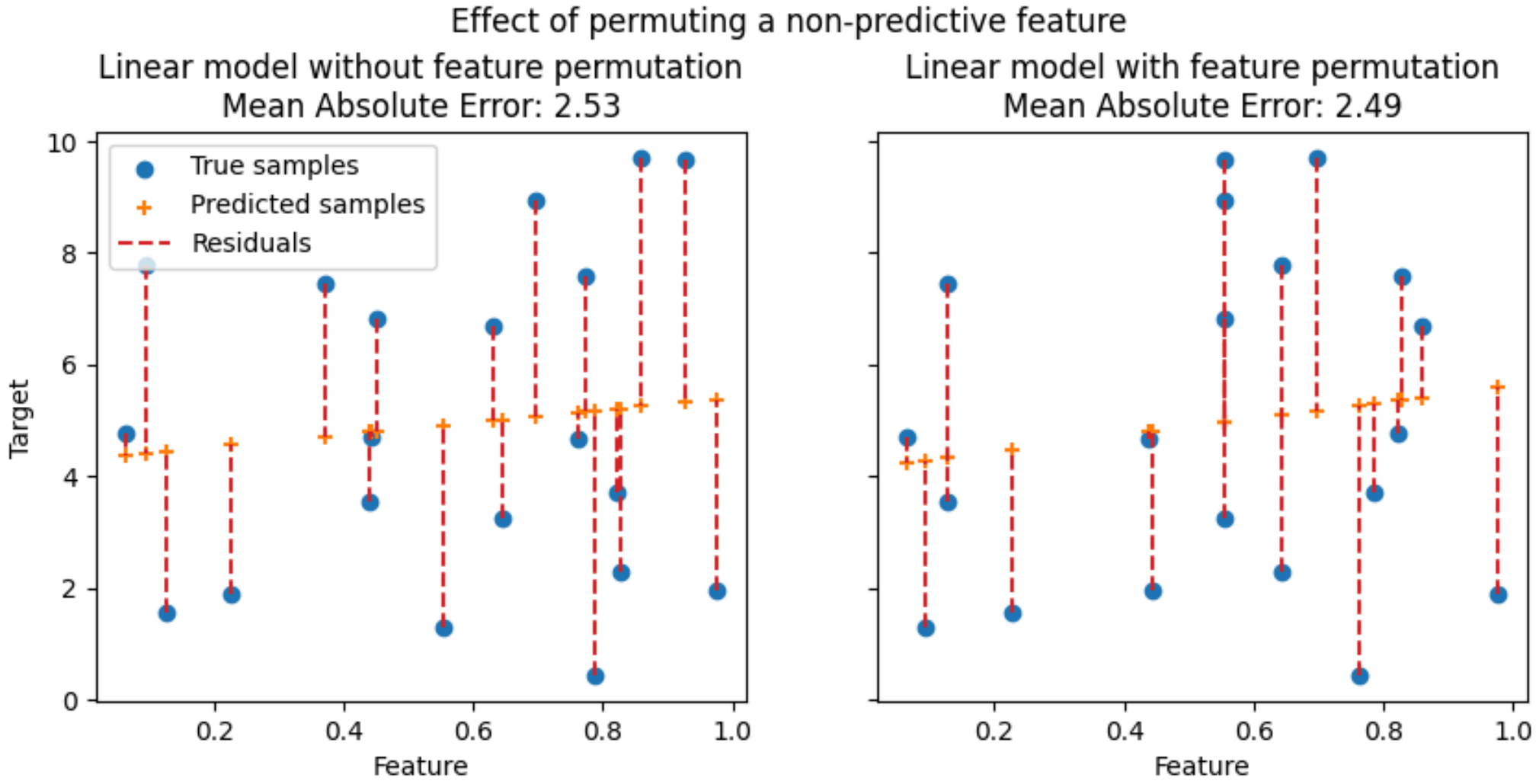
We want to know which features most influence model prediction

Given dataset $\{X, y\}$

- Fit the model
- Get scores
- Randomly shuffle one of the feature vectors x_i
- Refit model
- Get scores
- The higher degradation of the model performance the more important the feature

Permutation importance in sklearn


Effect of permuting a predictive feature
Linear model without feature permutation
Mean Absolute Error: 0.51
Linear model with feature permutation
Mean Absolute Error: 2.28
True samples
Predicted samples
Residuals
Target
Feature
Feature
2
4
6
8
10
0.2
0.4
0.6
0.8
1.0


Effect of permuting a non-predictive feature
Linear model without feature permutation
Mean Absolute Error: 2.53
Linear model with feature permutation
Mean Absolute Error: 2.49
True samples
Predicted samples
Residuals
Target
Feature
Feature
0
2
4
6
8
10
0.2
0.4
0.6
0.8
1.0

The most important properties of an ideal descriptor:

- Invariant with respect to translation of the coordinate system
- Invariant with respect to rotation of the coordinate system
- Invariant with respect to permutation of atomic indices: changing the enumeration of atoms does not affect the target
- Unique: single way to construct a descriptor and the descriptor itself corresponds to a single property
- Continuous: small changes in the atomic structure -> small changes in the descriptor
- Compact
- Computationally cheap

From DScribe: Library of descriptors for machine learning in materials science by Lauri Himanen at al. (Computer Physics Communications, 2020)

Machine learning for molecular simulations

Finite difference method

The shape of $V(R)$ is known at the given moment t

The acceleration for each particle is given by

$a_i = \frac{F_i}{m_i} = -\frac{1}{m_i}\frac{\partial V}{\partial r_i}$ eq. (1)

0. Initialize velocities
1. Select a small timestep δt (it should correctly describe the changes in your sysrtem)
2. Calculate forces
3. Integrate eq. (1) within $(t, t+\delta t)$ window to get velocities $\boldsymbol{v_i}(t+\delta t)$
4. Update coordinates using calculated velocities for each particle
   $\boldsymbol{r_i} = \boldsymbol{r_i} + \delta t \boldsymbol{v_i}$
5. Repeat 2-4 steps N times
6. Analyse the discrete trajectory obtained

Energy calculations with interatomic potentials

- the potential energy is calculated as the sum of atomic potential energies

$$E_{pot} = \sum_{i \in N_{atoms}} E_{i,atomic}$$

- forces are calculated as follows

$$\vec{F}_i = -\nabla_i E_{pot}$$

Fitting (training) the potentials

We run AIMD at elevated temperatures to sample atomic configurations

- **for a given chemical system**

The dataset:
Geometries: $\{R_n\}$
Energies: $\{E_n(R_n)\}$
Forces: $\{F_{i,\alpha}(R_n)\}$

Loss function

$\mathcal{L} = \lambda_E ||\hat{E} - E||^2 + \lambda_F \frac{1}{3N} \sum_{i=1}^{N} \sum_{\alpha=1}^{3} \left|\left| -\frac{\partial \hat{E}}{\partial r_{i,\alpha}} - F_{i,\alpha} \right|\right|^2$, where the hat symbol (e.g., $\hat{E}$) denotes predictions

Universal interatomic potentials

- Once trained on large datasets covering a wide chemical and structural configuration space
  - For example, the Materials Project dataset
- Can be applied to the unknown systems
  - To perform downstream tasks

Pros:

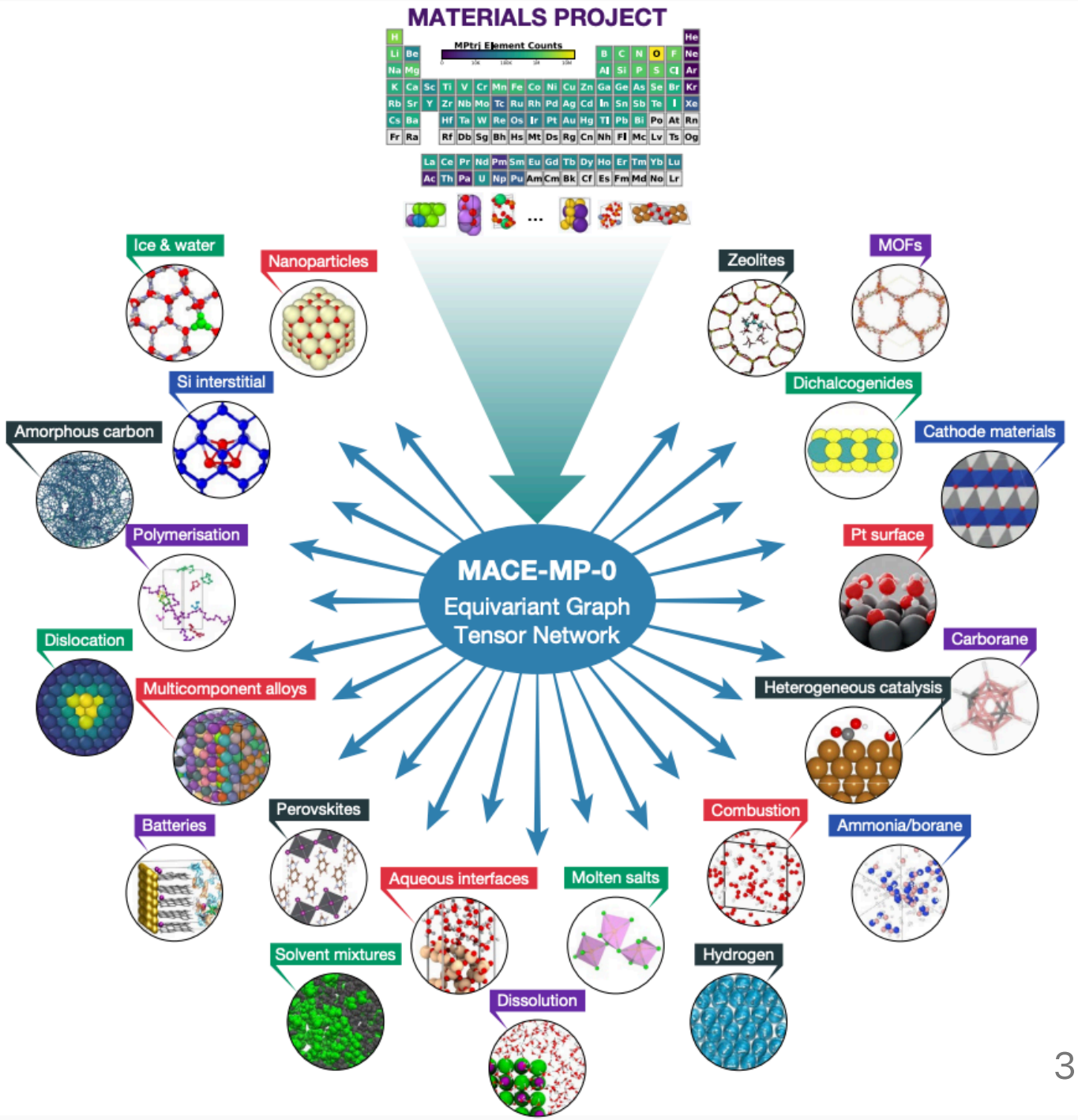
- Works out of the box
- Can be refitted (finetuned) with a small amount of data

Cons:

- Universality is debatable. Should be carefully validated before using
- A lot of weights -> slower compared to ML potentials developed for a specific system

Example

MACE-MP-0 graph neural network



Take home message

Materials informatics

- is about effectively using data, statistics, and data science tools for accelerating the trial-and-error process in materials science
- is not always about training ML models

Python (or another programming language) these days

- is support for your development as a successful scientist
- saves your time

Data

- respect/explore/use/share data

ML

- there is a trade-off between accuracy and speed
- expertise is essential for reliable results
- don't use test data split for model training