IMPERIAL
Department of Materials

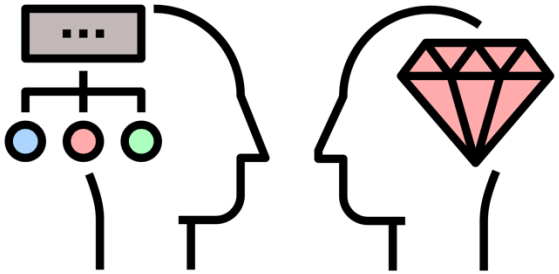
THE THOMAS YOUNG CENTRE TyC

Machine Learning for Materials

7. Building a Model from Scratch

Aron Walsh

Department of Materials
Centre for Processable Electronics



Module Contents

Class Outline

Building a Model from Scratch

A. Data Preparation

B. Model Choice

C. Training and Testing

Data Preparation

Data sets
in tutorials

Data sets
in the wild

Gremlins (1984); Adapted from @TowardsAI

Data Preparation

Data must be refined and structured
to build effective and robust statistical models

- Multiple sources
- Cleaning and pre-processing
- Feature engineering
- Feature scaling and normalisation

Data Sources

Primary choices are: (i) literature collection; (ii) databases; (iii) experiments or simulations

Data sets can be static (most common)

Data collection → Model training

Data sets can be dynamic (e.g. active learning)

Data collection → Model training → Data collection…

Data Sources

Primary choices are: (i) literature collection; (ii) databases; (iii) experiments or simulations

Data should be representative of your problem but does not need to be all-encompassing

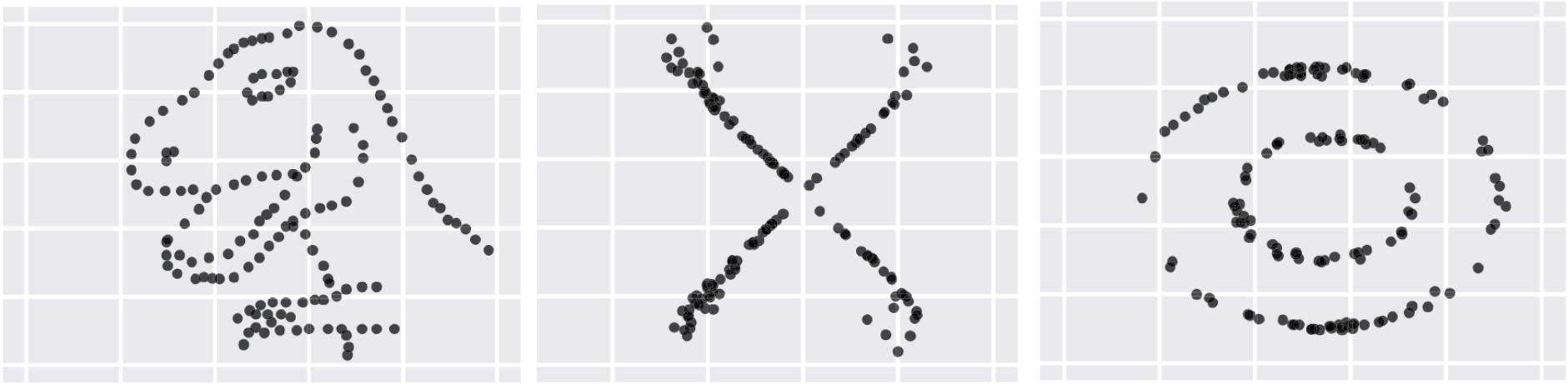
Effective local features are transferrable

Data size required depends on model complexity

Rule of thumb: 100-1000 data points per feature for classical ML (10 features → 10^2–10^4 training set)

Data Cleaning and Pre-processing

Beware of bias. Visualise data distributions as summary statistics don't tell the full story

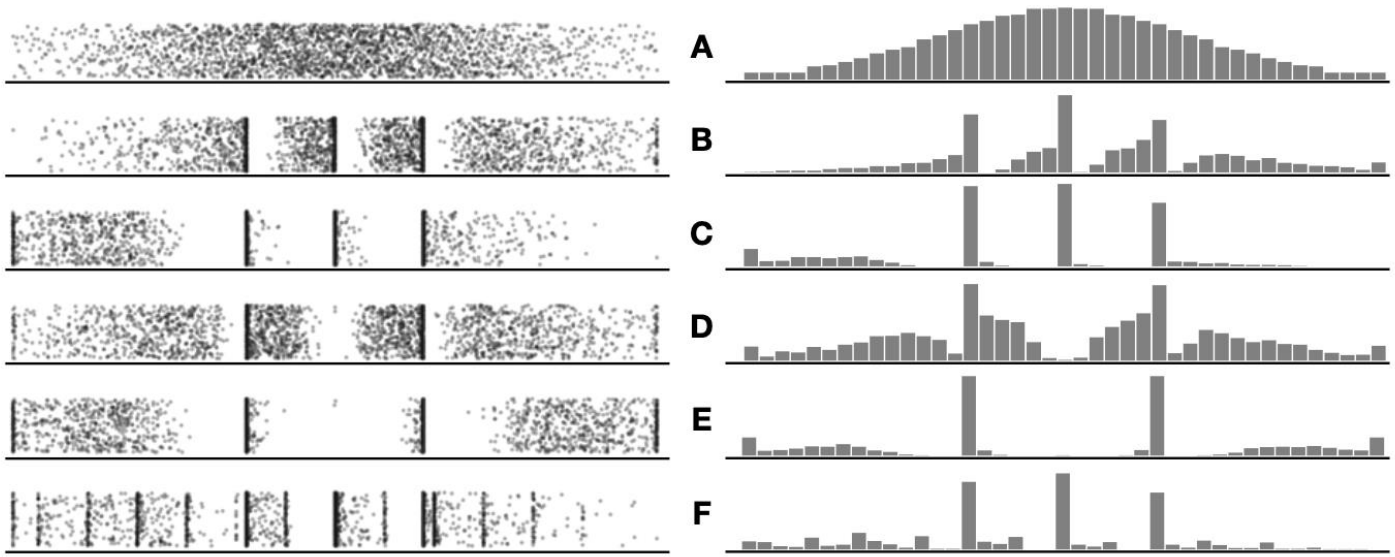


Each 2D dataset has the same summary statistics to two decimal places: $\bar{x}$ =54.26, $\bar{y}$ = 47.83, σ_x = 16.76, σ_y = 26.93, Pearson r = -0.06

"Same Stats, Different Graphs"; http://dx.doi.org/10.1145/3025453.3025912

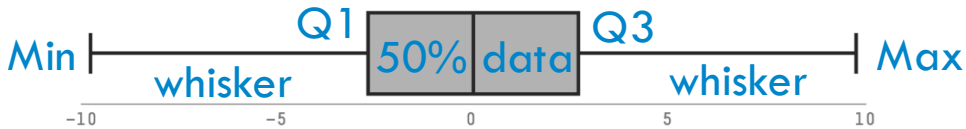
Data Cleaning and Pre-processing

Beware of bias. Visualise data distributions as summary statistics don't tell the full story



Six data distributions, each with the same 1st quartile, median, and 3rd quartile values (and the same box plot)

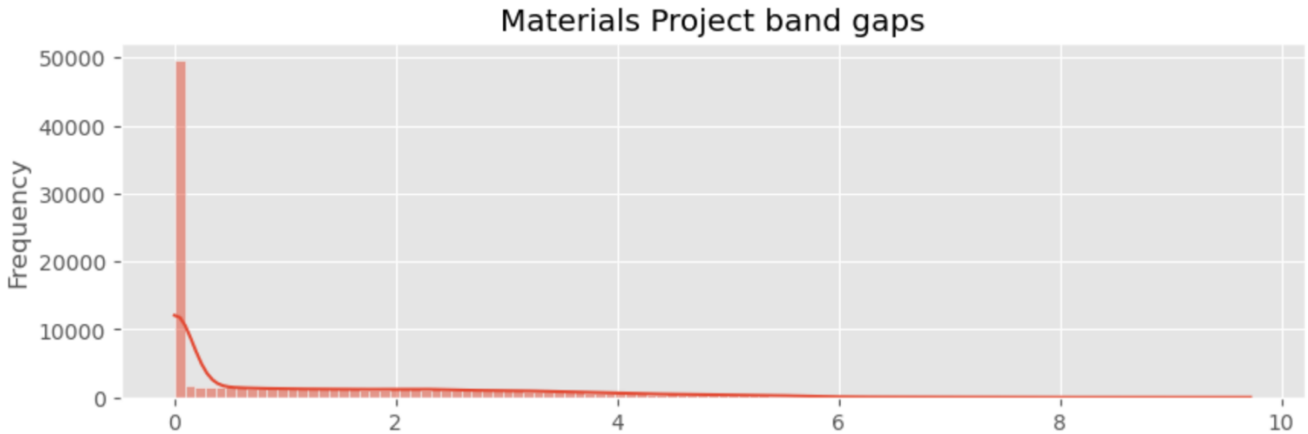
Box plot:

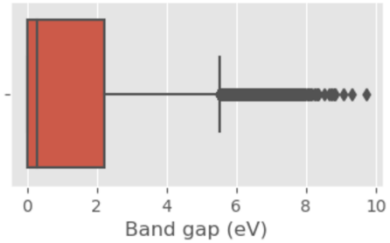


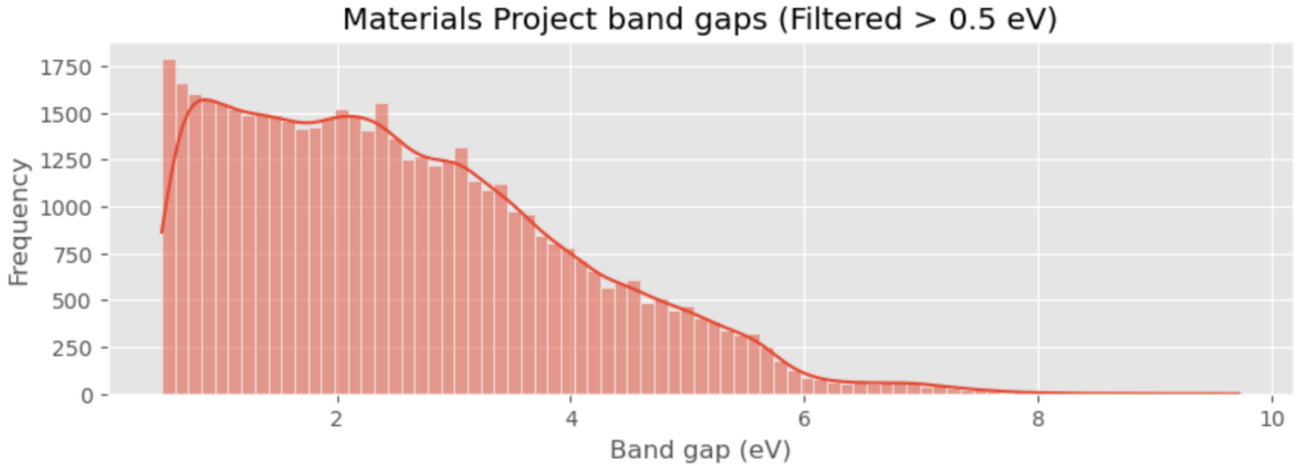
"Same Stats, Different Graphs"; http://dx.doi.org/10.1145/3025453.3025912

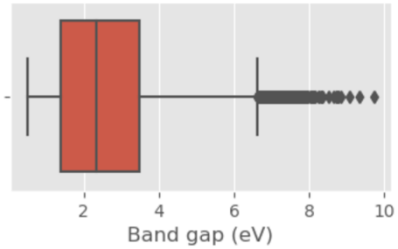
Data Cleaning and Pre-processing

Materials datasets are often biased with skewed property distributions









Calculated band gaps from density functional theory (PBE functional)

Data Cleaning and Pre-processing

Check for missing, outlier, and noisy data

Identify: use data exploration techniques

e.g. summary statistics, visualisation, profiling

Impute: fill in missing values

e.g. using mean imputation or regression

Cap: set thresholds for extreme values ("winsorising")

Remove: delete instances with missing/erroneous values from your dataset

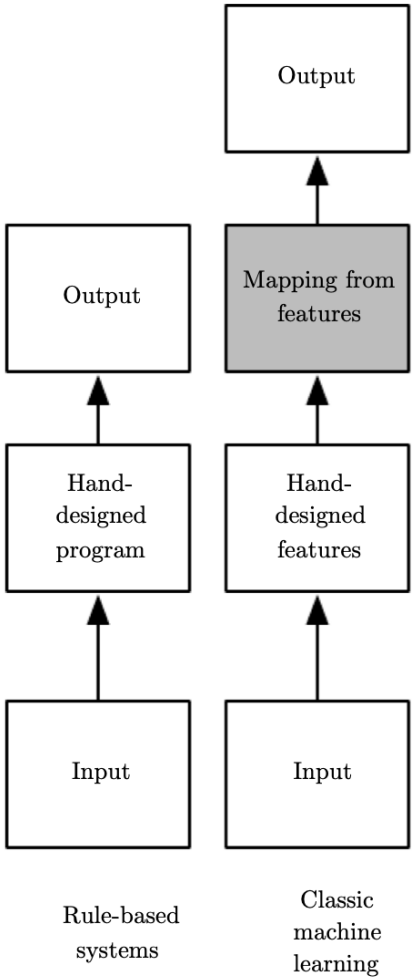
Other data transformations may be suitable (e.g. log, cube root)

```python
import pandas as pd

# Sample dataset with missing values
data = {'composition': [0.2, 0.3, 0.5, None, 0.6, 0.4],
        'hardness': [150, None, 180, 160, None, 170]}

# Create a DataFrame
df = pd.DataFrame(data)

# Impute missing values with the mean
df['composition'].fillna(df['composition'].mean(), inplace=True)
df['hardness'].fillna(df['hardness'].mean(), inplace=True)

# Display the cleaned dataset
print(df)
```

Feature Engineering

Classical ML

Crafting of features tailored to domain knowledge and problem specifics due to limitations of simpler models

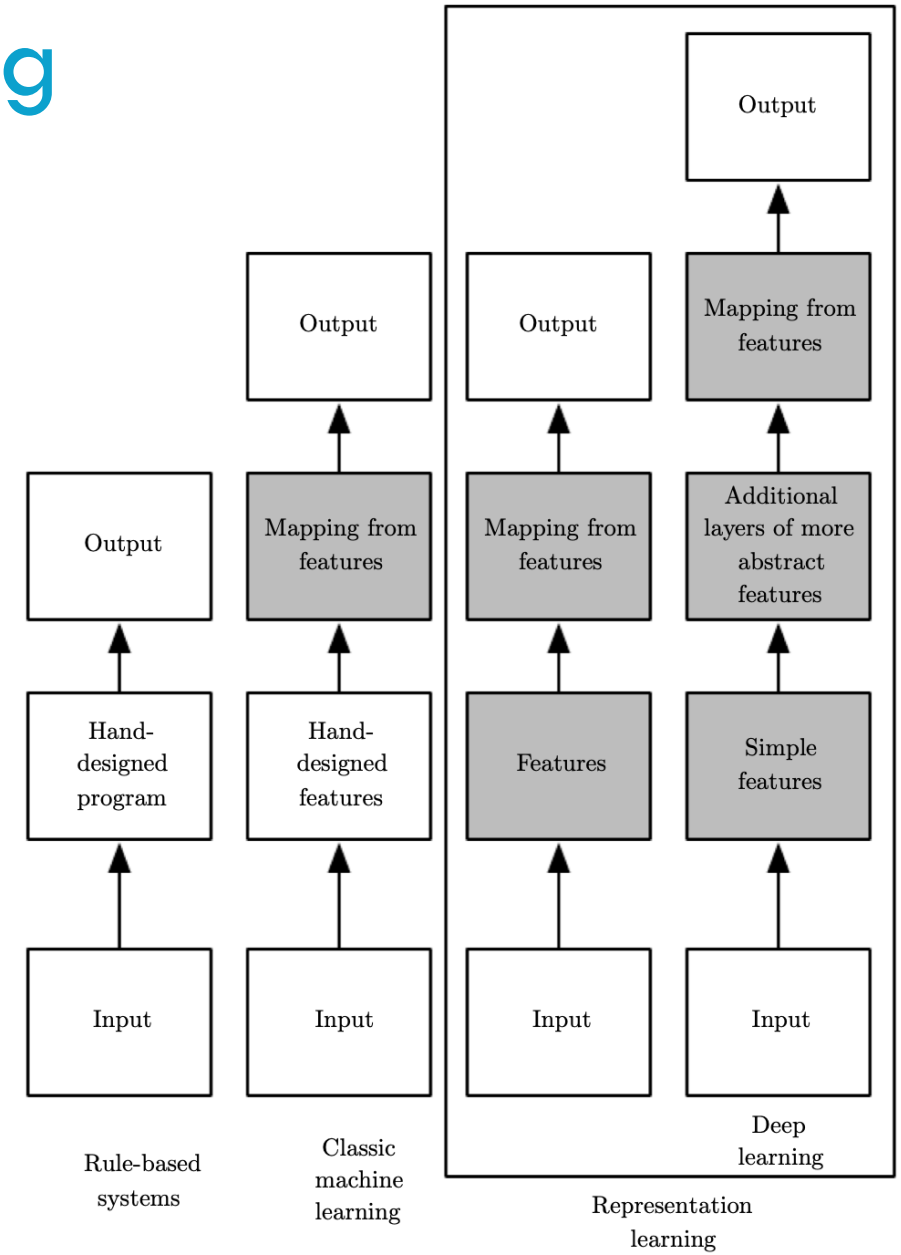


I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning"

Feature Engineering

Deep Learning

Use simple inputs and automatically learn features, benefiting from complex architectures (e.g. CNNs)



I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning"

Feature Engineering

Comment on "Predicting reaction performance in C–N cross-coupling using machine learning"

Kangway V. Chuang and Michael J. Keiser*

Ahneman *et al*. (Reports, 13 April 2018) applied machine learning models to predict C–N cross-coupling reaction yields. The models use atomic, electronic, and vibrational descriptors as input features. However, the experimental design is insufficient to distinguish models trained on chemical features from those trained solely on random-valued features in retrospective and prospective test scenarios, thus failing classical controls in machine learning.

		Lin. Reg.	*k*-NN	SVM	NN	RF
Ahneman et al.	R^2	0.66	0.64	0.64	0.93	0.92
	RMSE	15.6	16.1	16.1	7.0	7.5
Random Features	R^2	0.65	0.57	0.63	0.92	0.91
	RMSE	15.8	17.4	16.2	7.4	7.9
One-Hot Encoded	R^2	0.65	0.45	0.66	0.93	0.90
	RMSE	15.8	19.8	15.5	7.1	8.6

K. V. Chuang and M. J. Keiser, Science 362, 6416 (2018)

Feature Engineering

Choice of many of compositional, structural and property features for materials

Feature selection: chose the most relevant features to improve performance (iterative approach)

Dimensionality reduction: useful for high-dimensional data, e.g. principal component analysis (PCA)

Aggregation: combine data over dimension(s), e.g. mean value over space (r), time (t), wavelength (λ)

Feature Scaling and Normalisation

Uniformity in feature scales may enhance model stability and convergence

Standardisation: centre distribution around 0 with unit variance, e.g. $x_{standard} = (x-\bar{x})/std(x)$

Min-max scaling: rescale to a range (usually 0-1), e.g. $x_{scaled} = (x-min(x))/(max(x)-min(x))$

Robust scaling: adjust for outliers using median & interquartile range, e.g. $x_{rscaled} = (x-median(x))/IQR(x))$

```python
import numpy as np
from sklearn.preprocessing import MinMaxScaler

# Sample dataset
data = np.array([[2.0, 5.0],
                 [1.0, 3.0],
                 [4.0, 7.0],
                 [3.0, 6.0]])

# Create a MinMaxScaler instance
scaler = MinMaxScaler()

# Fit the scaler on the data and transform it
scaled_data = scaler.fit_transform(data)

print("Original Data:\n", data)
print("Scaled Data:\n", scaled_data)
```

Class Outline

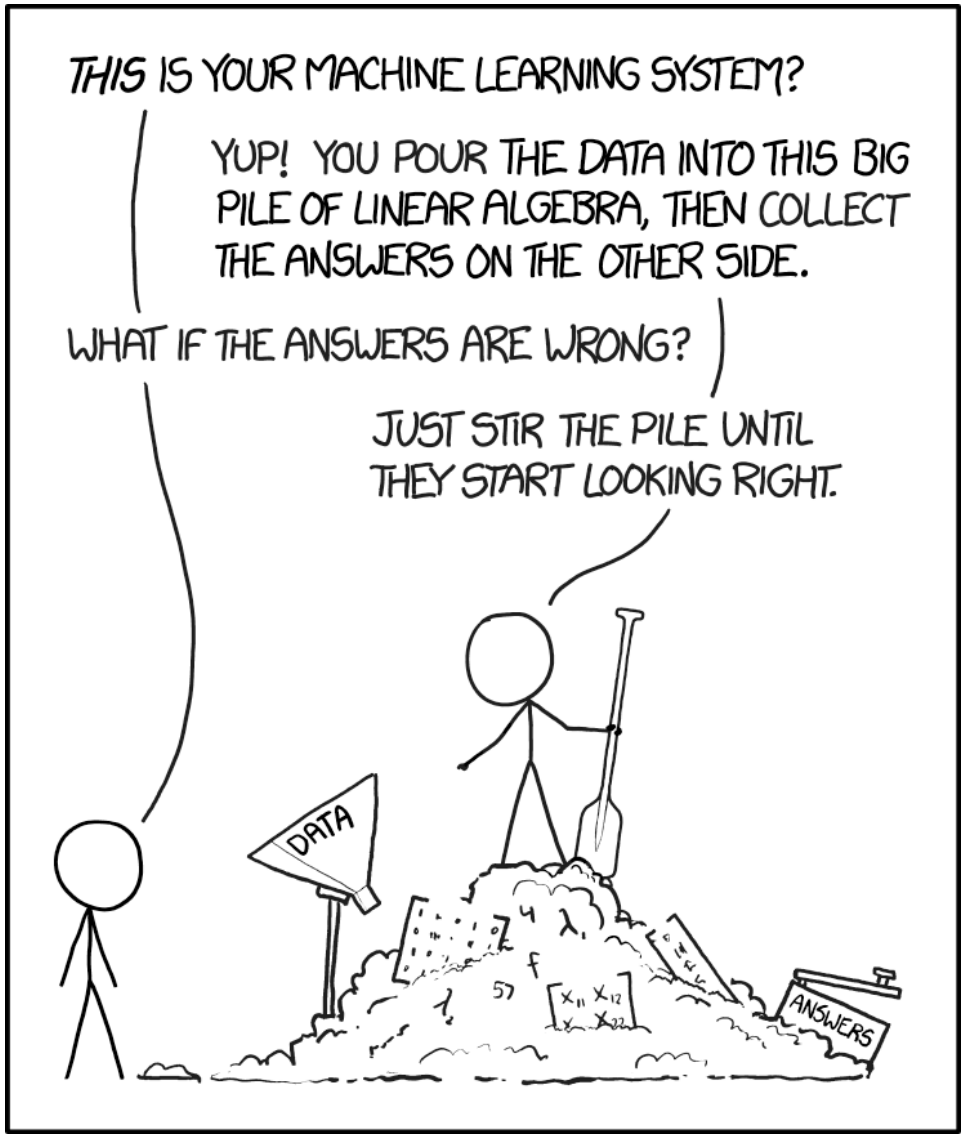
Building a Model from Scratch

A. Data Preparation

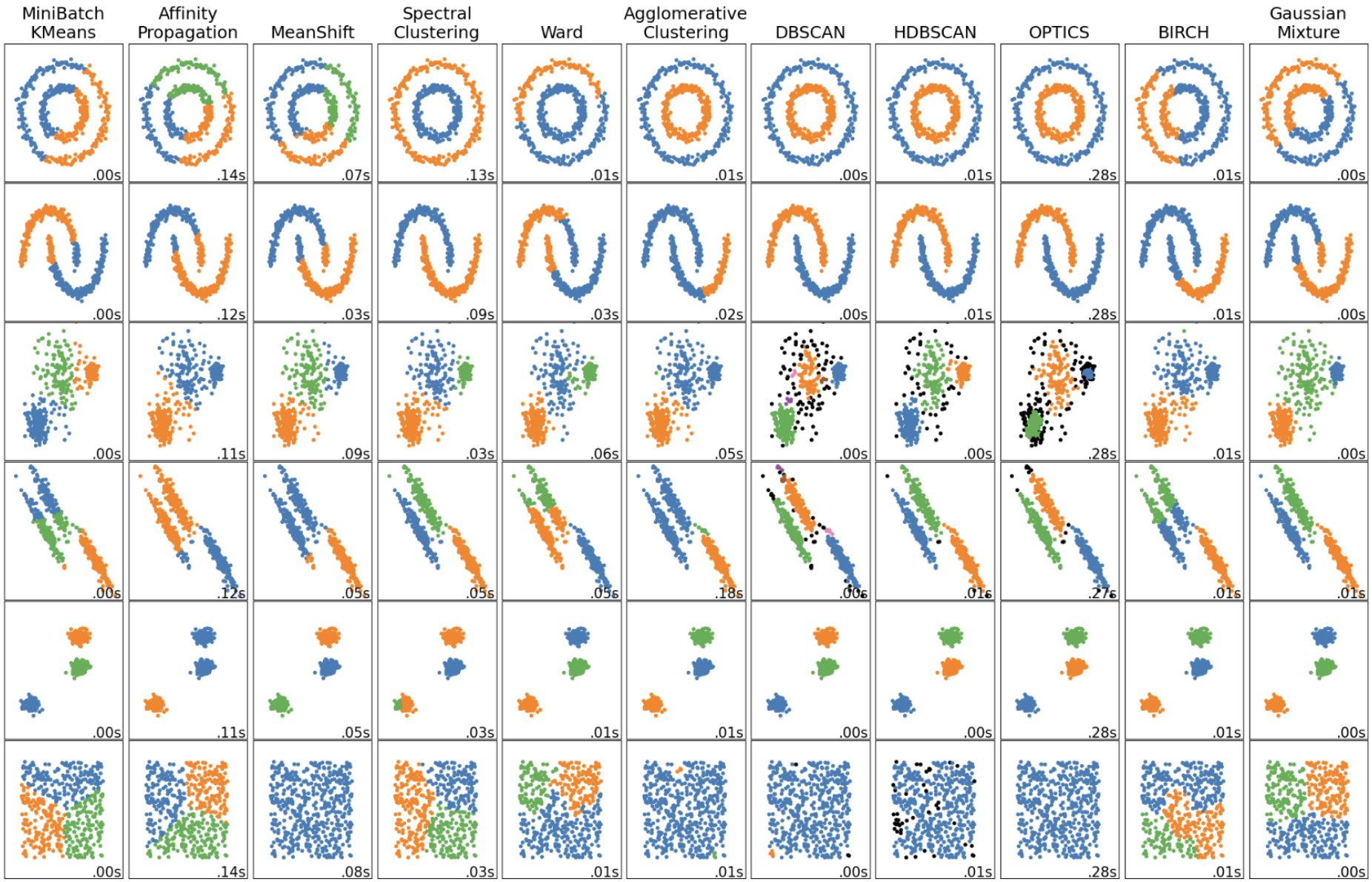
B. Model Choice

C. Training and Testing

Model Choice



https://xkcd.com/1838/



https://scikit-learn.org/stable/modules/clustering.html

Model Choice

Balance accuracy, generalisation, and transparency for materials predictions

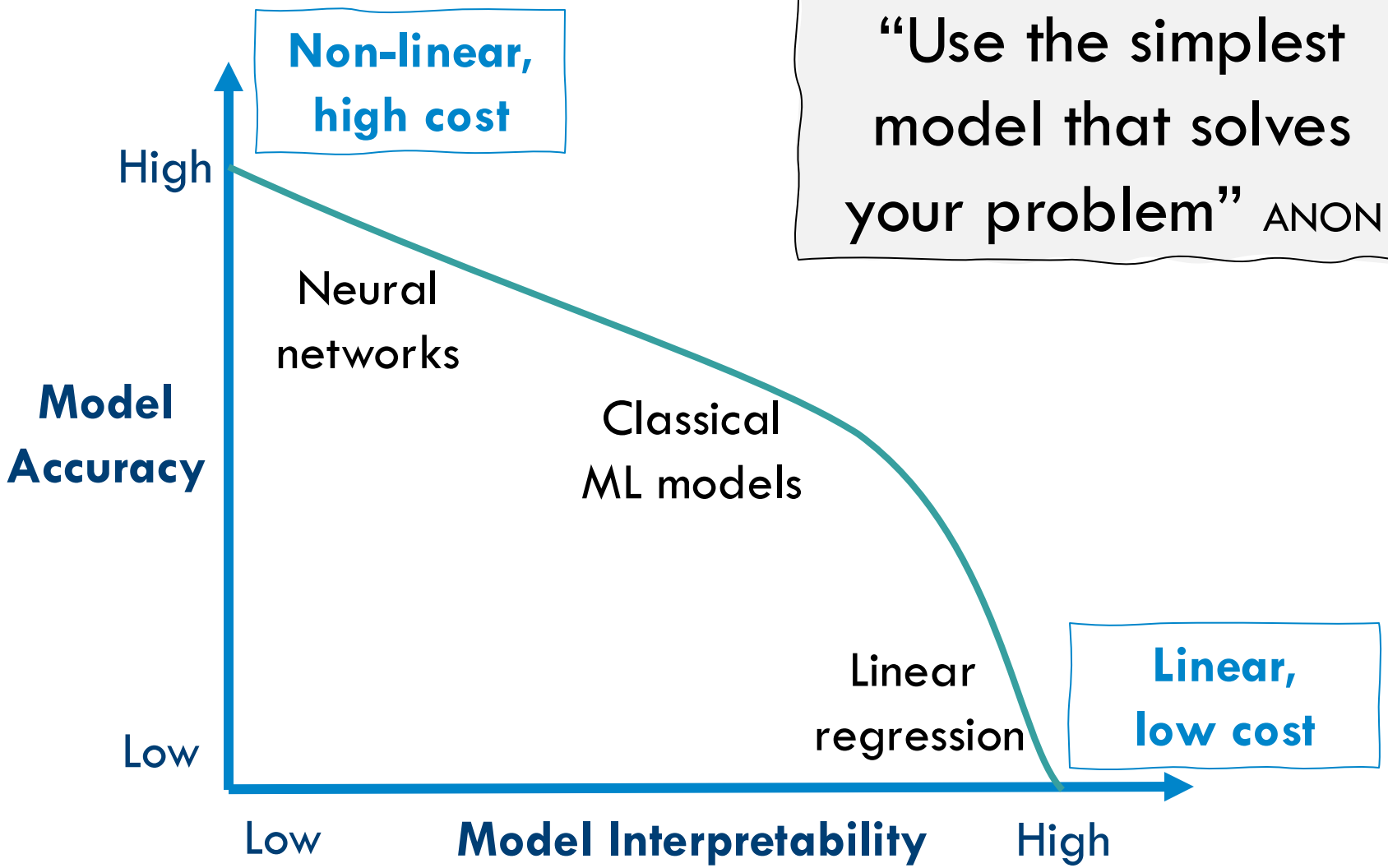
Goal: ensure the model is suitable for your task

e.g. property prediction, classification, clustering

Data size: for small datasets, simpler models with fewer parameters are preferable

Complexity: simpler models are more transparent; don't rush to the latest deep learning if not needed

Complexity Trade-off



"Use the simplest model that solves your problem" ANON

There are many model variants and exceptions to the schematic

Complexity Trade-off

One neuron versus deep learning in aftershock prediction

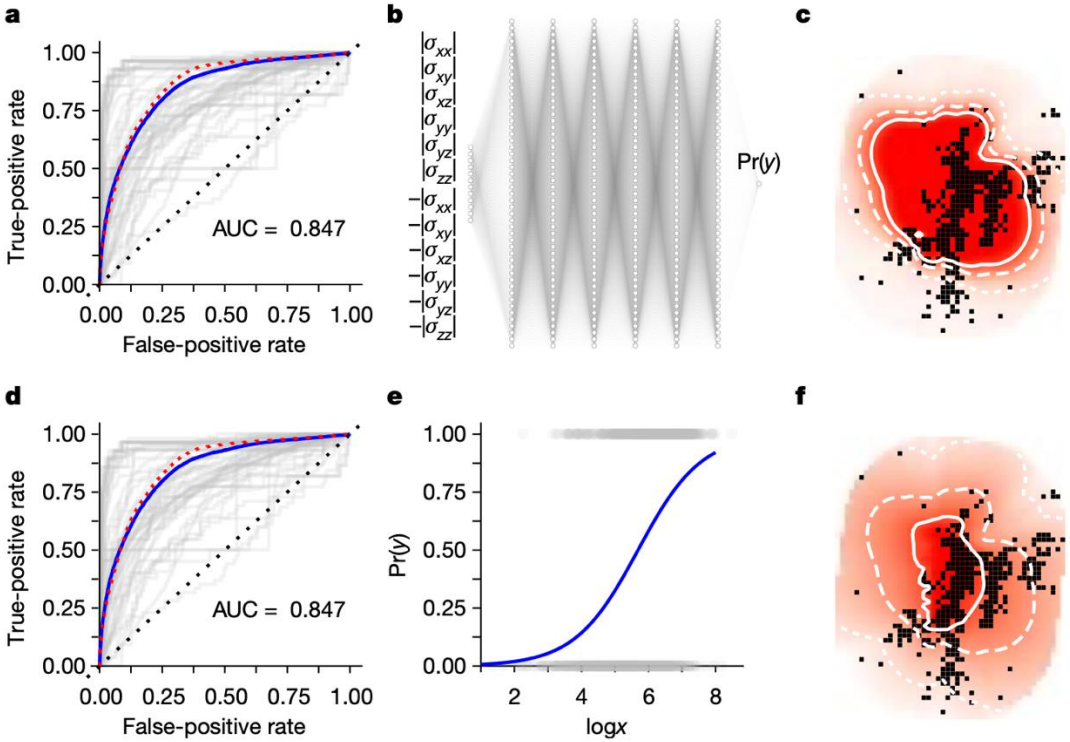
Arnaud Mignan[1,2,3]* & Marco Broccardo[2,4]*

Published deep learning model

13,451 parameters

One neuron

2 parameters



AUC (Area Under the Curve) = Classification metric [0,1]

A. Mignan and M. Broccardo, Nature 574, E1 (2019)

Model Architecture

The structure of a model influences its learning capability and complexity

Deep learning choices

Layers: input, hidden, output

Activation functions: sigmoid, ReLU…

Topology: feedforward, convolutional…

Optimal architecture should enhance feature extraction, model capacity, task suitability

Best practice is to compare to a baseline, e.g. *most frequent class* (classification) or *mean value* (regression)

```python
import torch.nn as nn

# Simple Linear Model
class LinearModel(nn.Module):
    def __init__(self, input_size, output_size):
        super().__init__()
        self.fc = nn.Linear(input_size, output_size)

# Complex Neural Network Model
class ComplexModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super().__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.fc2 = nn.Linear(hidden_size, output_size)

# Instantiate the models
simple_model = LinearModel(10, 1)        ←10 inputs, 1 output
complex_model = ComplexModel(10, 64, 1)        ←64 hidden neurons
```

Remember: fc = a regular fully connected layer in deep learning

Class Outline

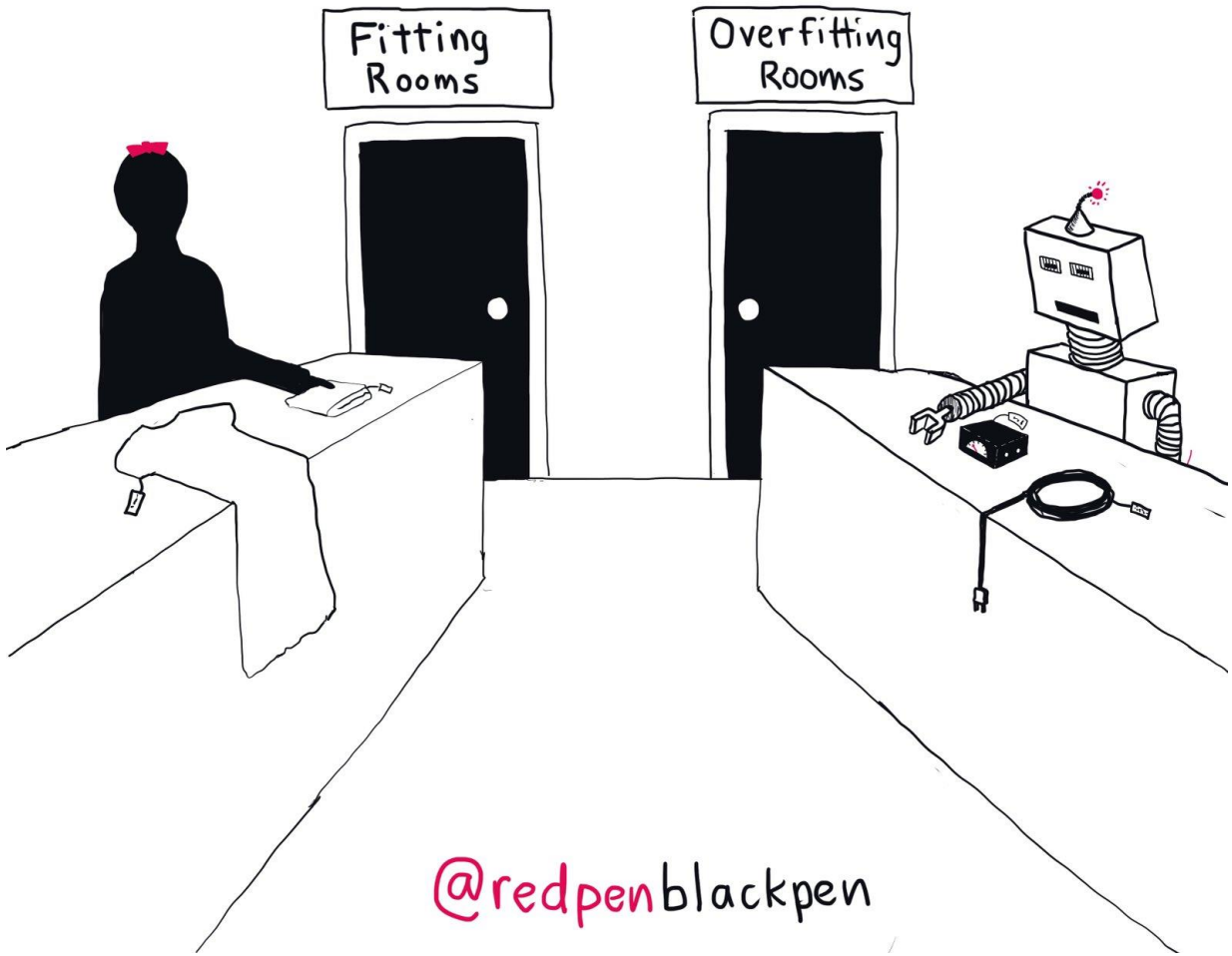
Building a Model from Scratch

A. Data Preparation

B. Model Choice

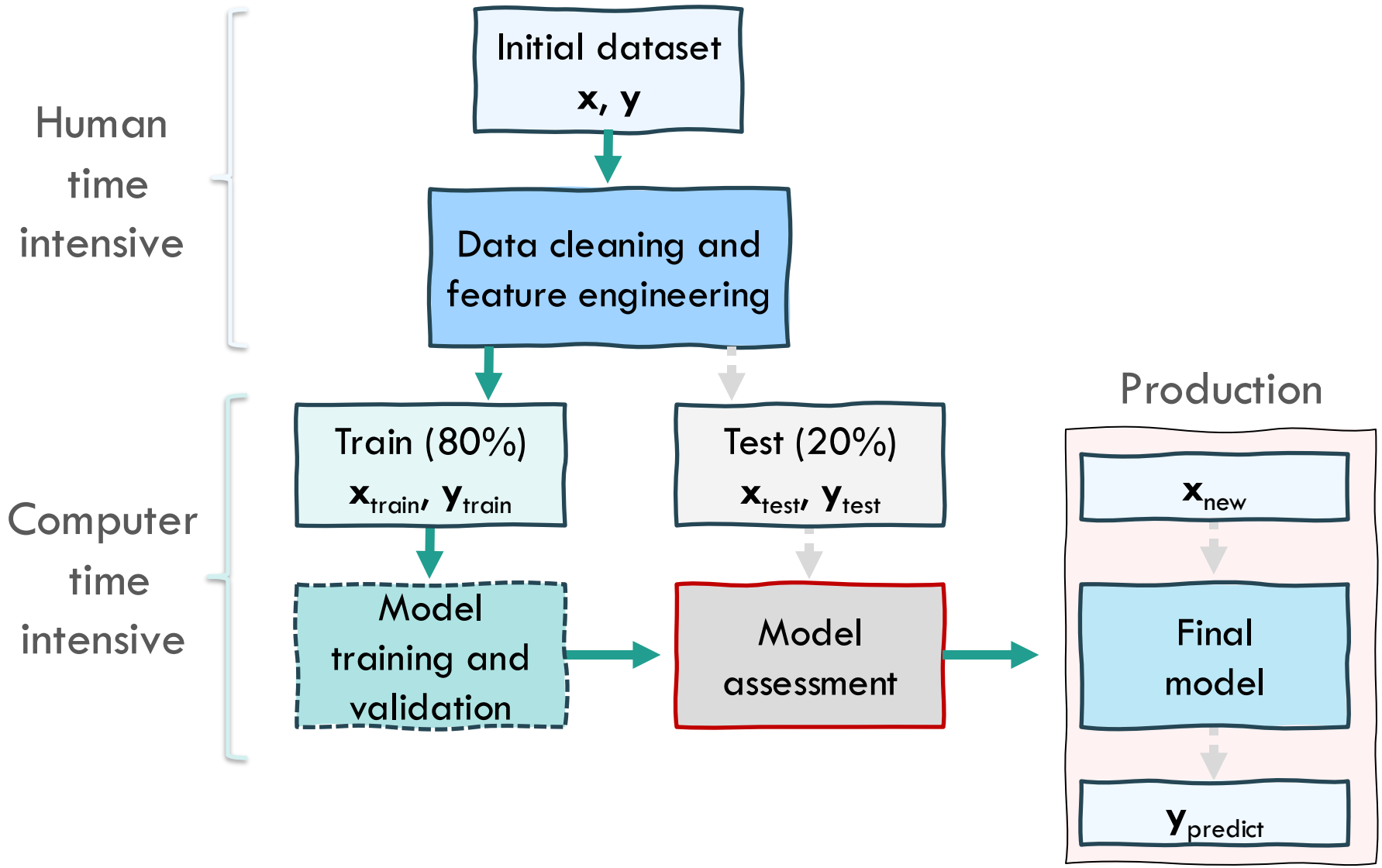
C. Training and Testing

Training and Testing



https://www.instagram.com/redpenblackpen

Supervised ML Model Workflow



The exact workflow depends on the type of problem and available data

Model Training

Iteratively optimise, validate, and fine-tune models for reliable and robust predictions

Key training choices

Loss function: quantify the difference between model predictions and target values, e.g. MSE

Optimisation algorithm: update model parameters to minimise the loss function, e.g. stochastic gradient descent (SGD), adaptive moment estimation (ADAM)

ADAM: D. P. Kingma and J. Ba, arXiv.1712.6980 (2014)

Model Evaluation

Evaluate models through data splitting for training (validation) & testing (final assessment)

Validation set: Subset of training data used to fine-tune hyperparameters and prevent overfitting

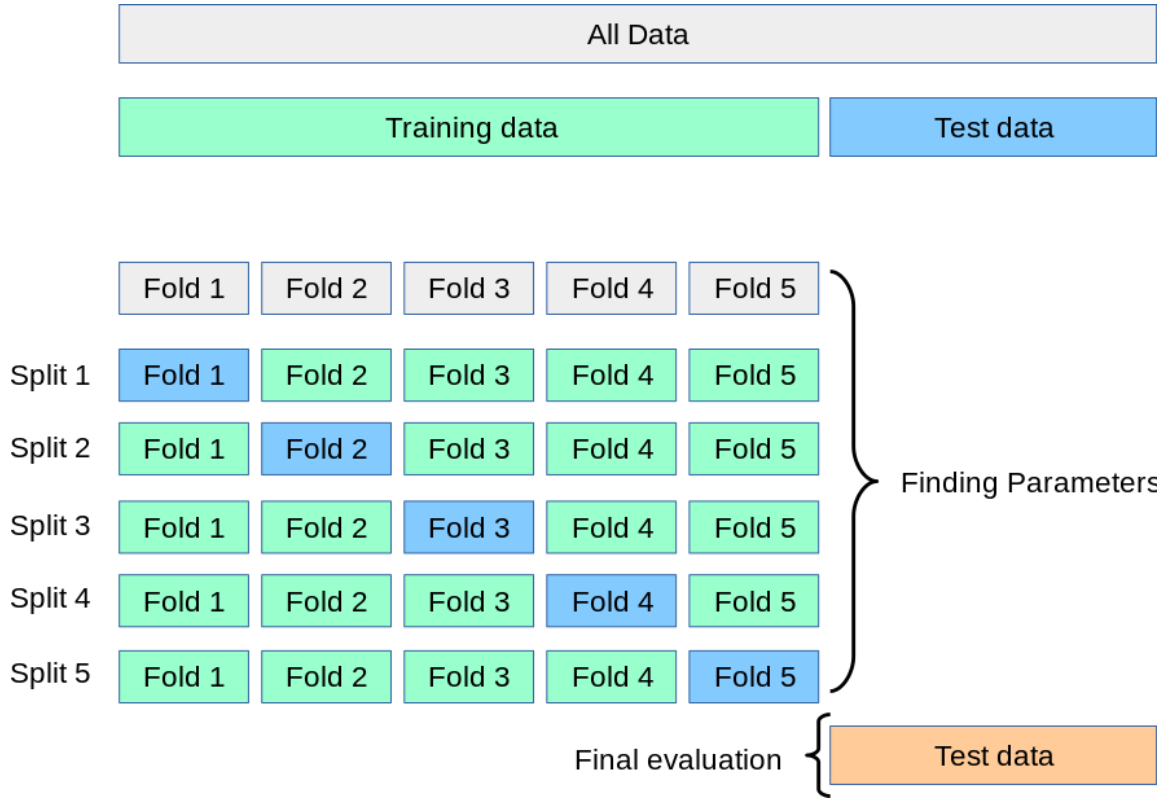
Cross-validation (CV): Divide training data into multiple subsets for training and validation

Test set: Separate "holdout" dataset used to evaluate final performance and predictive power

Warning: "test" and "validation" definitions may vary in some communities

Cross-Validation (CV)

Assess performance on multiple portions of the dataset. Choice in how the data is split



<u>*k*-fold CV</u>

Iteratively train on *k*-1 folds

<u>Stratified *k*-fold CV</u>

Ensure even class distribution

<u>Leave-one-out CV</u>

For small datasets

<u>Monte Carlo CV</u>

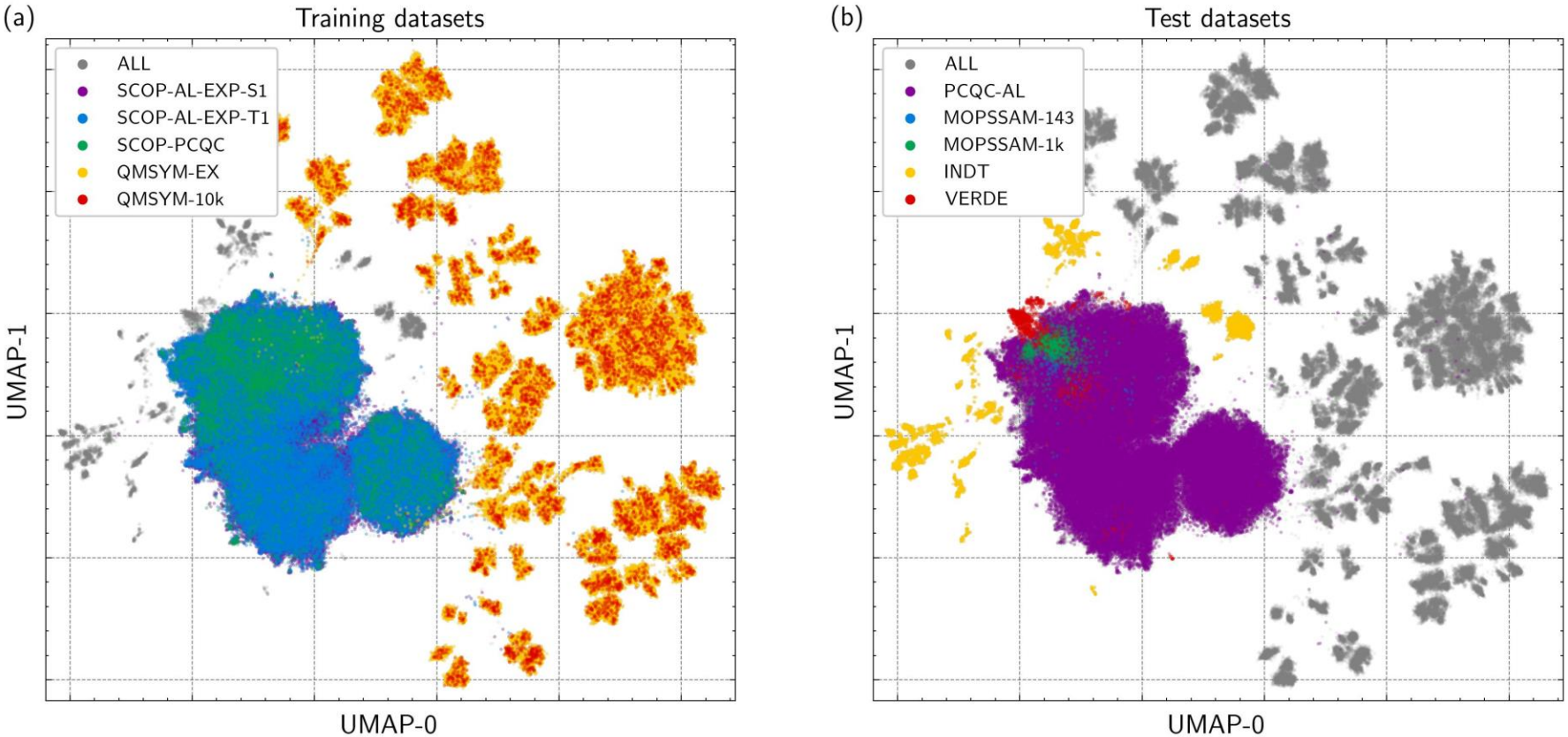
Random sampling

https://scikit-learn.org/stable/modules/cross_validation.html

Cross-Validation (CV)

For heterogeneous data, random splits are not ideal.

An alternative is to cluster the data first



Visualising molecular datasets in global chemical space using UMAP dimension reduction

S. Verma, M. Rivera, D. O. Scanlon and A. Walsh, J. Chem. Phys. 156, 134116 (2022)

Hyperparameter Tuning

Optimal choice of settings that impact model performance and learning during training

Tuning strategies

Grid search: exhaustive (within grid), but expensive

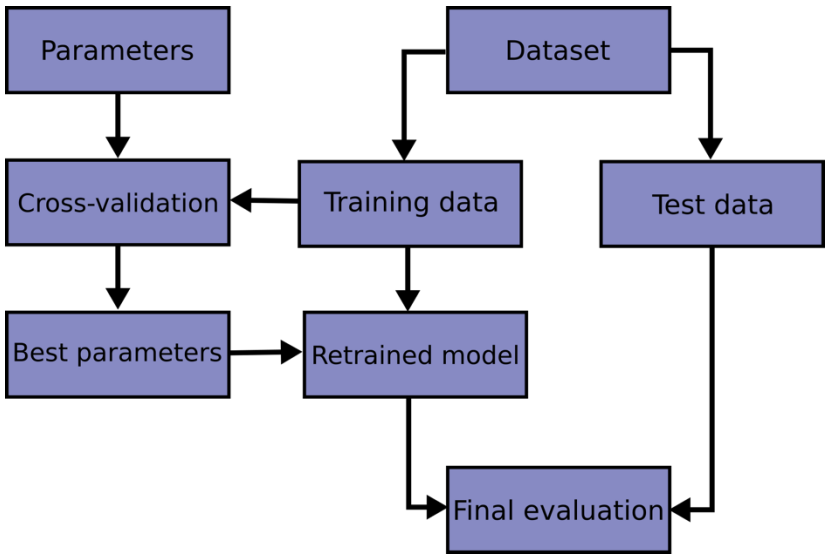
Random search: efficient, but may miss solutions

Optimisation: evolutionary, Bayesian… efficient, but complex (introduce their own parameters)

Well-tuned hyperparameters prevent overfitting, improve convergence, and enhance model generalisation

Grid Search CV

Cross-validation can be used to identify the optimal set of hyperparameters to <u>retrain</u> the best model



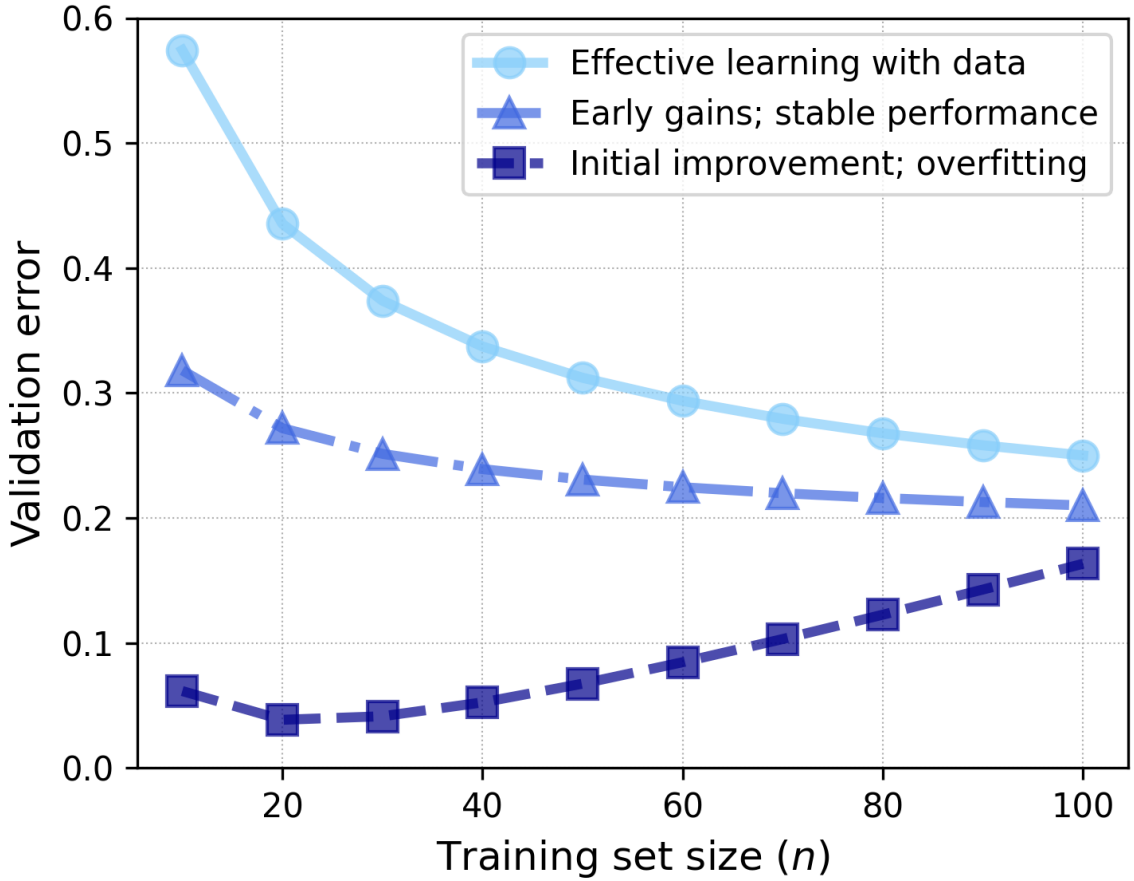
Final retraining step on all training data

```python
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

# Parameter grid for model hyperparameter tuning
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20]
}

# GridSearchCV with RandomForestClassifier
grid_search = GridSearchCV(
    estimator=RandomForestClassifier(),
    param_grid=param_grid,
    cv=10,  # 10-fold cross-validation
    scoring='accuracy'  # Use accuracy as the metric
)

# Fit the model on the training data
grid_search.fit(X_train, y_train)

# Retrieve the best model after cross-validation
best_model = grid_search.best_estimator_
print(f"Best Parameters: {grid_search.best_params_}")
print(f"Best Mean Validation Score: {grid_search.best_score_:.4f}")

# Make predictions using the best model
predictions = best_model.predict(X_test)
```

https://scikit-learn.org/stable/modules/cross_validation.html

Learning Curves

Learning curves can visualise how model performance metrics change with dataset size



Single number model comparisons overlook data size dependence

A plateau in validation error can indicate a stable model

Avoid "p-hacking" (Data Dredging)

Manipulation of data and analysis methods to achieve statistically significant results

Term comes from hacking the *p*-value:

p-value = P(observed result | null hypothesis is true)

No statistical relationship between variables

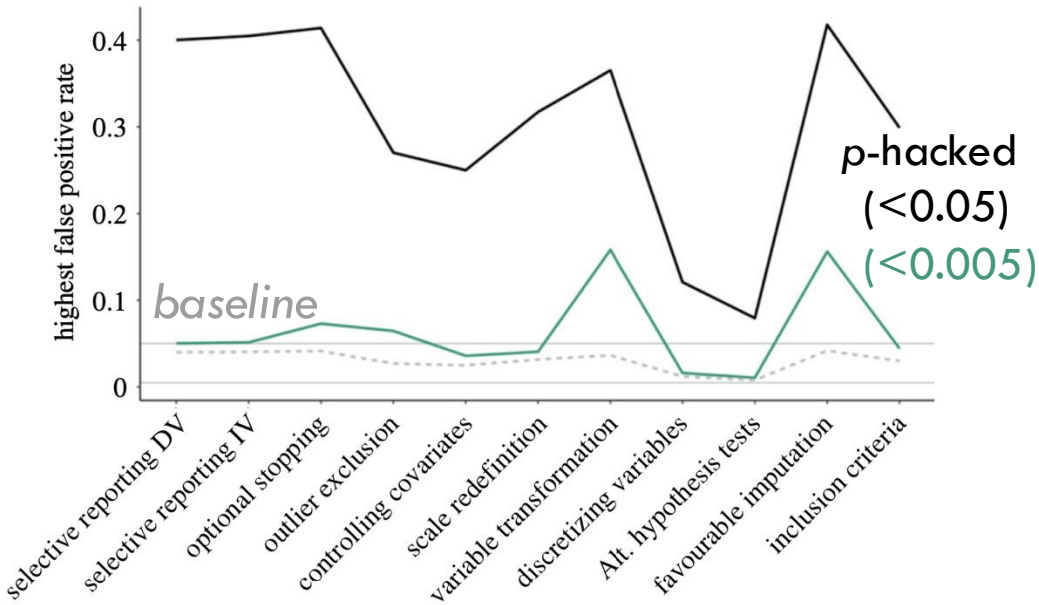
<u>Misuse of ML methods</u>

Selective train & test sets

Data leakage

Deliberate outlier exclusion

Improper rounding



"Big little lies" A. M. Stefan and F. D. Schönbrodt, R. Open Soc. Sci. 10, 220346 (2023)

Checklist for ML Research Reports

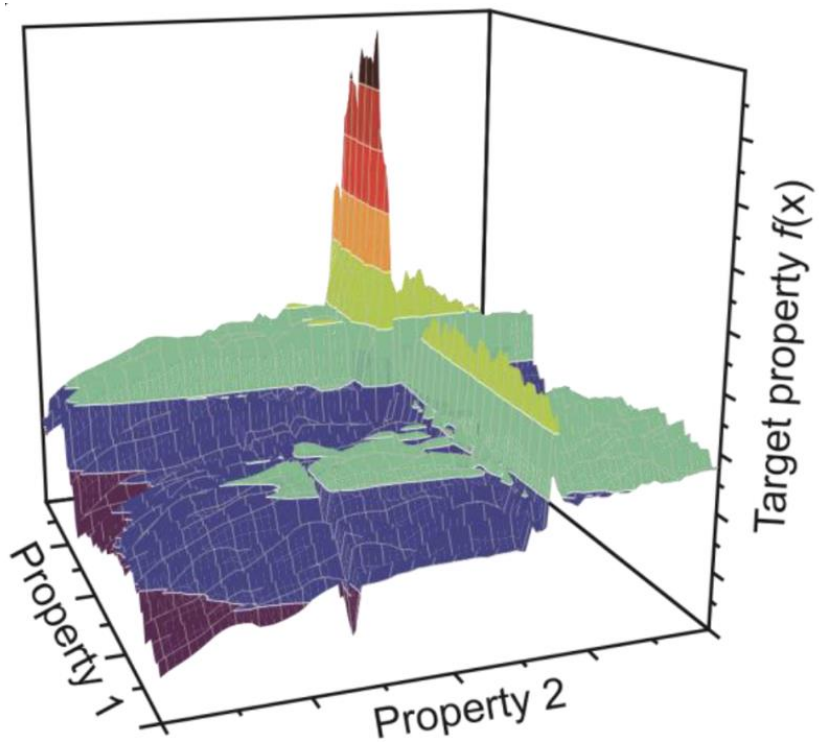
Checklist for reporting and evaluating machine learning models	
1. Data sources	
1a. Are all data sources listed and publicly available?	
1b. If using an external database, is an access date or version number provided?	
1c. Are any potential biases in the source dataset reported and/or mitigated?	
2. Data cleaning	
2a. Are the data cleaning steps clearly and fully described, either in text or as a code pipeline?	
2b. Is an evaluation of the amount of removed source data presented?	
2c. Are instances of combining data from multiple sources clearly identified, and potential issues mitigated?	
3. Data representations	
3a. Are methods for representing data as features or descriptors clearly articulated, ideally with software implementations?	
3b. Are comparisons against standard feature sets provided?	
4. Model choice	
4a. Is a software implementation of the model provided such that it can be trained and tested with new data?	
4b. Are baseline comparisons to simple/trivial models (for example, 1-nearest neighbour, random forest, most frequent class) provided?	
4c. Are baseline comparisons to current state-of-the-art provided?	
5. Model training and validation	
5a. Does the model clearly split data into different sets for training (model selection), validation (hyperparameter opimization), and testing (final evaluation)?	
5b. Is the method of data splitting (for example, random, cluster- or time-based splitting, forward cross-validation) clearly stated? Does it mimic anticipated real-world application?	
5c. Does the data splitting procedure avoid data leakage (for example, is the same composition present in training and test sets)?	
6. Code and reproducibility	
6a. Is the code or workflow available in a public repository?	
6b. Are scripts to reproduce the findings in the paper provided?	

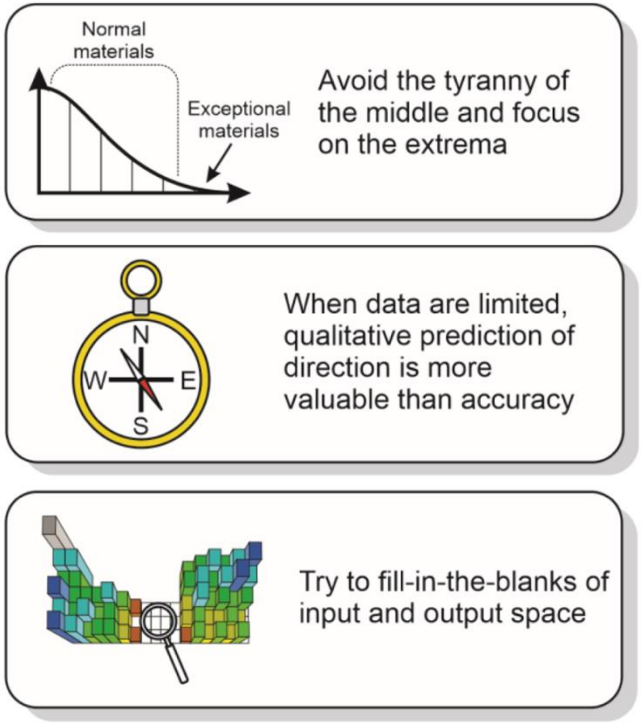
Useful for project planning too: https://www.nature.com/articles/s41557-021-00716-z

N. Artrith et al, Nature Chemistry 13, 505 (2021)

Beyond (Average) Supervised Models

The most interesting materials, and the emergence of unexpected properties, are often outliers





J. Schrier et al, J. Am. Chem. Soc. 40, 21699 (2023)

Class Outcomes

1. Knowledge of ML model development process
2. Identify and mitigate overfit and underfit regimes in model training
3. Selection of appropriate performance model evaluation techniques

Activity:

Crystal hardness revisited