OAK RIDGE National Laboratory

Artificial Intelligence Methods for Microscopy Analysis and Knowledge Extraction

IMC-20 Pre-Congress Workshops

Rama K. Vasudevan

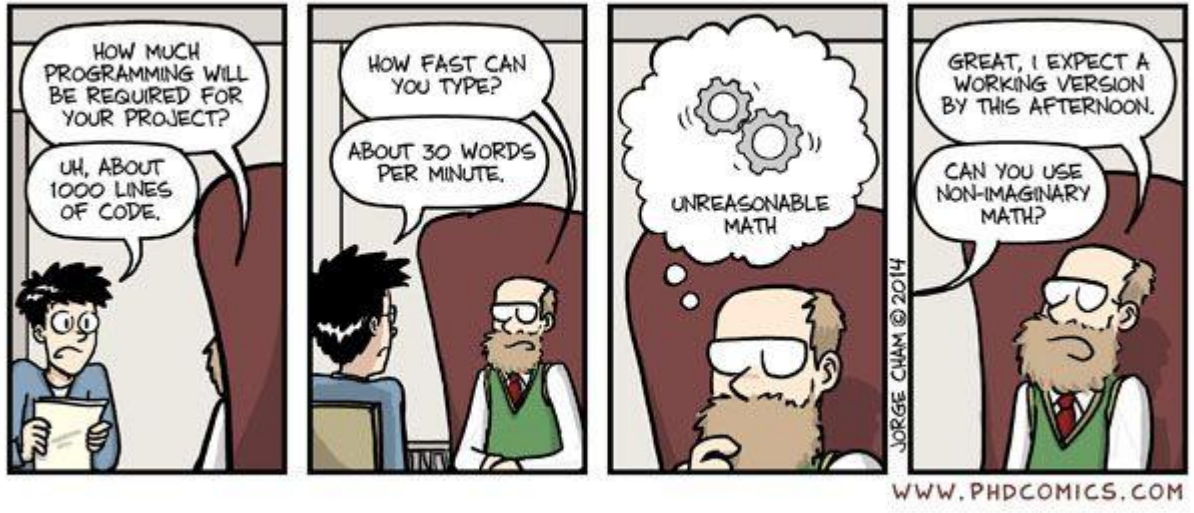
Center for Nanophase Materials Sciences,
Oak Ridge National Laboratory

10th September 2023
Busan, Korea

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

What you need for today



- An understanding of python and the python scientific software stack is desirable but not essential
- Laptop that has internet access
- Time (all of today!)

Logistics

Organizers

Rama Vasudevan
Oak Ridge National Lab

Yunseok Kim
SungKyunKwan University

Speakers

Joshua Agar
Drexel University

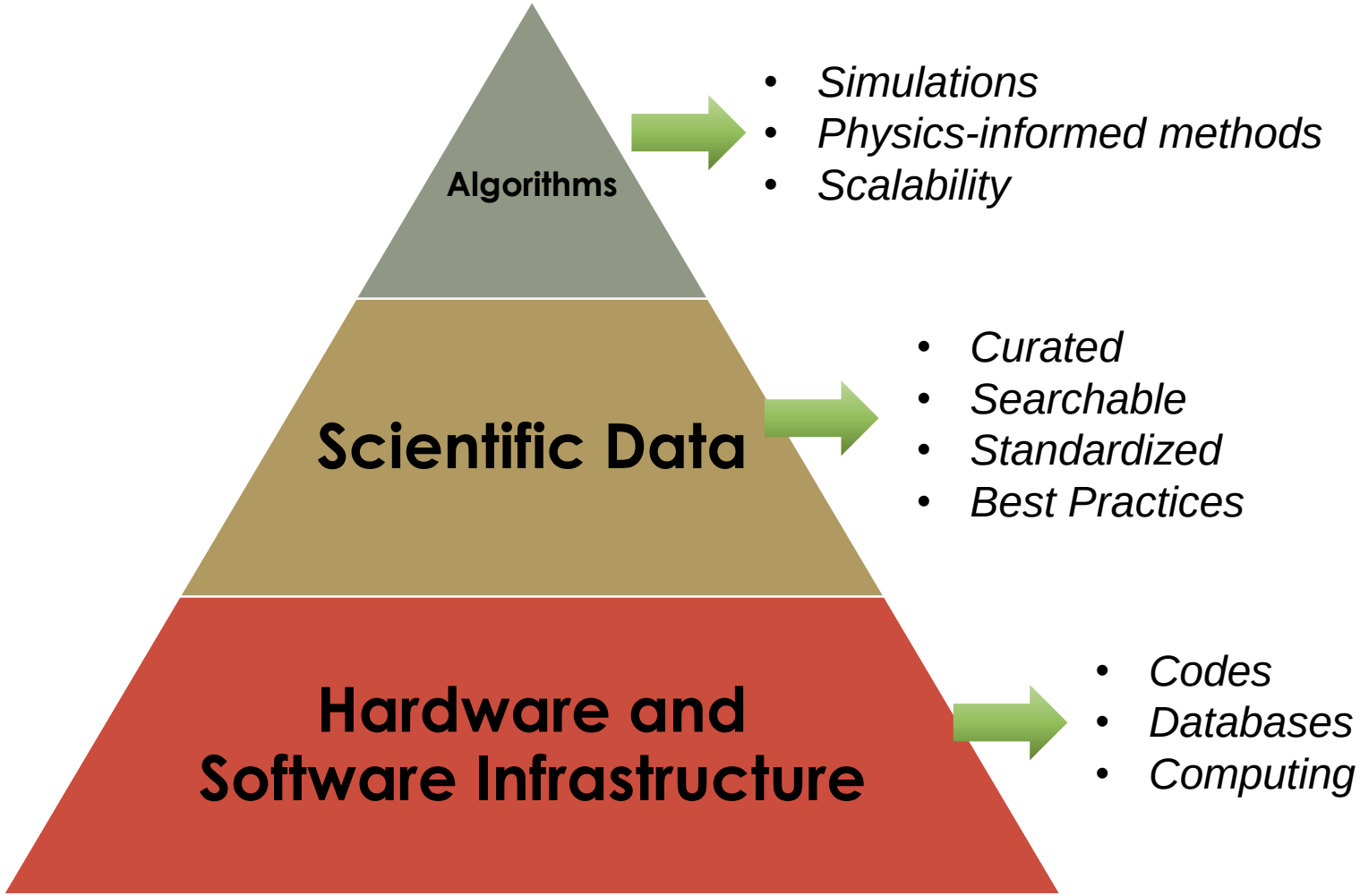
Sergei Kalinin
University of Tennessee

Ayana Ghosh
Oak Ridge National Lab

Schedule

Presenter	Time Slot	Topic
Rama Vasudevan	9:00-10:30AM	Introduction to Machine Learning
Coffee Break	***10:30-11:00***	Best coffee roasting methods
Joshua Agar	11:00-11:45AM	Image and Spectral Analytics for Microscopy
Rama Vasudevan	11:45-12:30PM	Spectral unmixing and image processing
Lunch Break	***12:30-2:00PM***	
Joshua Agar	2:00-2:45PM	Deep learning: introduction and applications to microscopy
Ayana Ghosh	2:45PM-3:30PM	Advanced machine learning and causal ML
Coffee Break	3:30PM-4:00PM	Is coffee or tea better in the afternoon?
Sergei Kalinin	4:00PM-4:45PM	Automated and autonomous microscopy
Ayana Ghosh	4:45PM-5:30PM	Hypothesis Learning for autonomous physics discovery
Close of day	5:30PM	

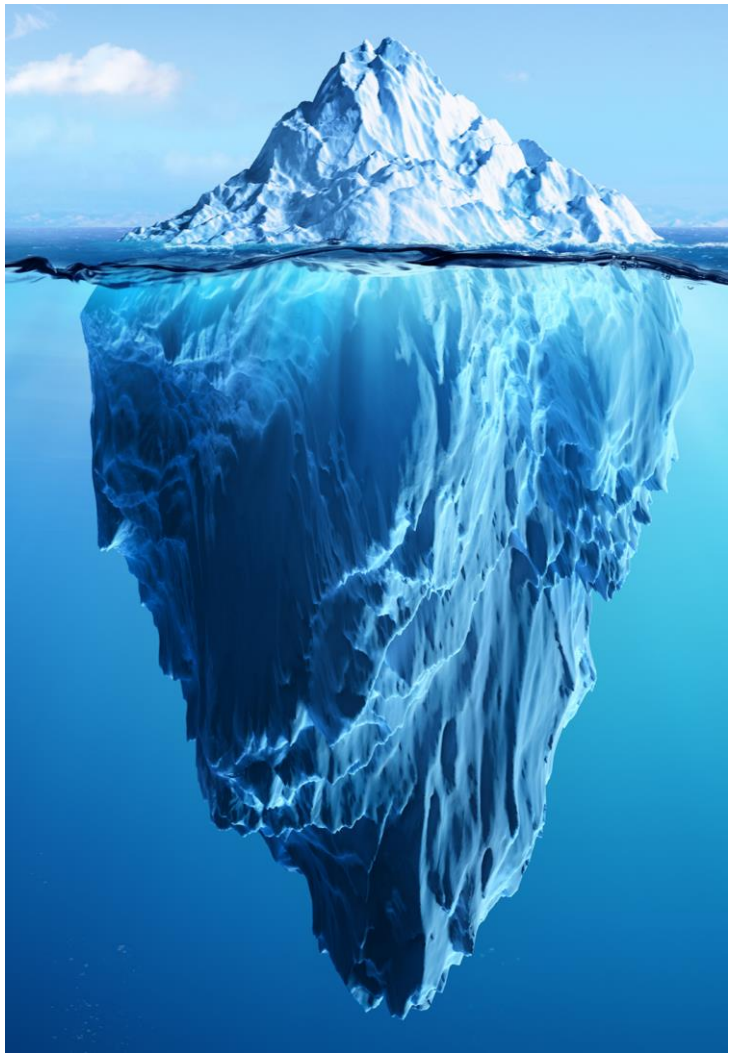
The pyramid of machine learning



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

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

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



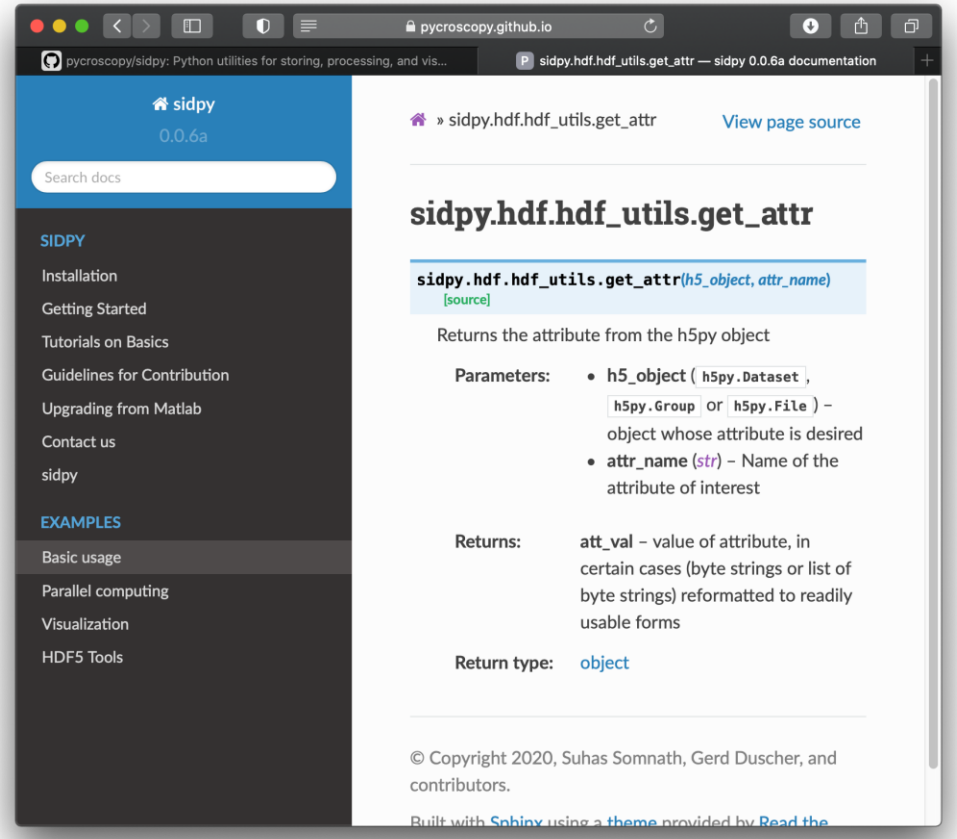
A word on data infrastructure

- 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.
- Pycroscopy is our method at attempting to alleviate this problem

pycroscopy

github.com/pycroscopy

An ecosystem for microscopy data ingestion, analytics and visualization



pycroscopy

A general-purpose package for microscopy imaging and spectroscopy data analytics, including regisytration, image cleaning, unmixing, ect.

scifireaders

For ingesting a variety of microscopy f les for output to sidpy dataset objects

pyusid

Python package for reading and visualizing our universal spectral imaging dataset format

pynsid

Python package for reading writing and visualizing our N-dimensional spctral imaging datase format

sidpy

Python utilities for storing, visualizing, and Spectroscopic and Imaging Data (SID)

bglib

Utilities to analyze, f t and visualize Band-excitation and G-mode imaging and spectroscopy data promarily for SNMS SPM users

atomai

Deep learning toolkit for analysis of atomiclly resolved imageing and spectroscopy datasets

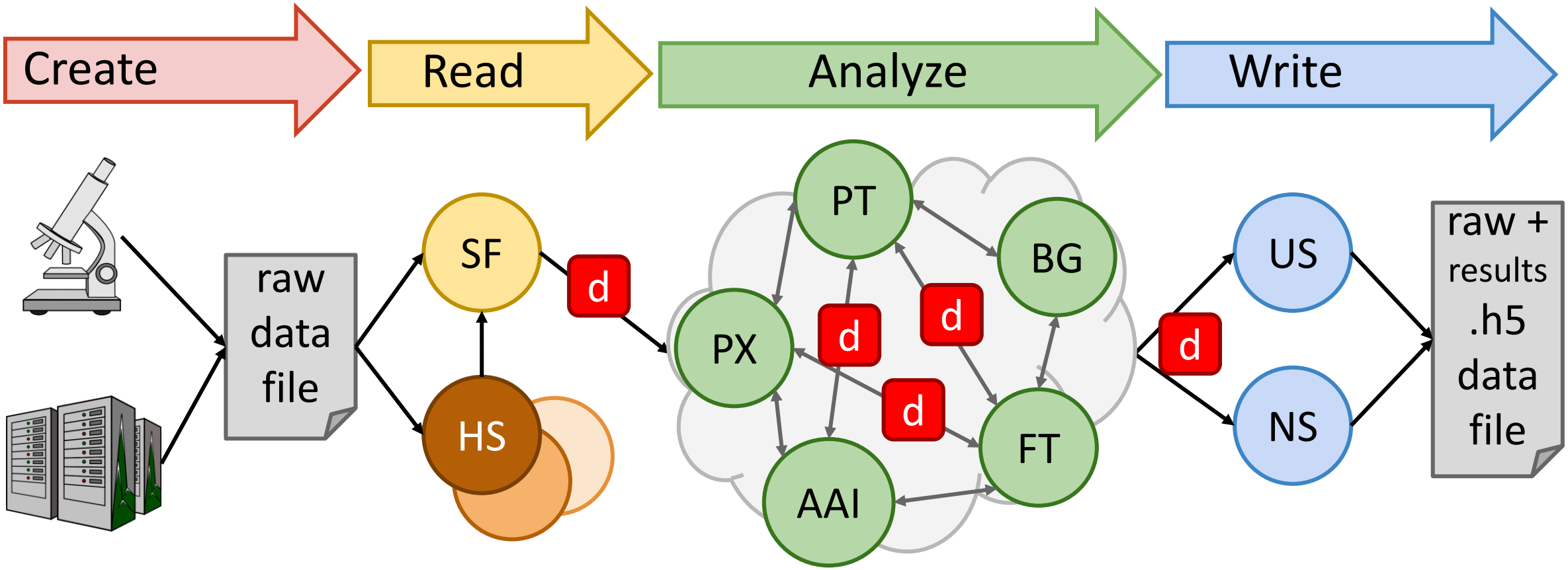
stemtools

Python based codes for analysis of 4D-STEM and abreeation corrected vanilla STEM datasets

pytemlib

Python tools for simulation, registraion, analysis and visualization of TEM datasets

Pycroscopy philosophy

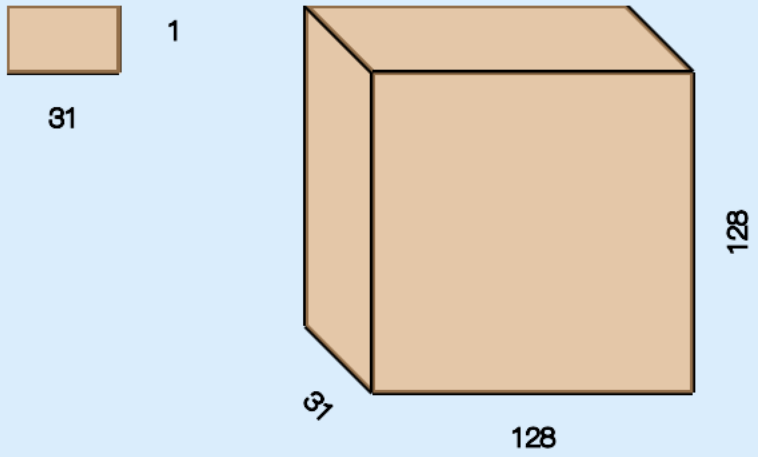


Data from measurements or simulations are read into **`sidpy.Dataset`** (d) objects directly by **`SciFiReaders`** (SF). Data are processed using multiple science packages in the Pycroscopy ecosystem that interoperate via **`Dataset`** objects. **`Dataset`** objects are written to HDF5 files via **`pyUSID`**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

NSID Model (implemented as sidpy.Dataset)

Dataset Object built on top of dask arrays

	Array	Chunk
Bytes	120.12 MiB	120.12 MiB
Shape	(31, 31, 128, 128)	(31, 31, 128, 128)
Count	1 Tasks	1 Chunks
Type	float64	numpy.ndarray



Benefits of the model:

- Easy to understand
- sidpy takes care of plotting (dataset.plot())
- Can easily perform parallel computations
- Easy to push to file including metadata
- Useful for data pipelining

- Maintain N-dimensional form
- All of the advantages of dask (large sizes, parallel compute)
- Additional data given for each dimension of dataset, such as name, quantity, units
- Metadata stored in dictionary
- Can be readily pushed to hdf5 files

Introduction to Machine Learning

But what's it used for in microscopy?

- If you have repetitive tasks, it can be used to automate them
- If you need to fit functions but don't know the function, you can use an ML method
- If you need to identify and track objects in images or movies
- If you need to understand spectral datasets
- Predicting properties from structure or processing

Machine Learning

Supervised

“Give me some examples”

Unsupervised

“I don’t need any examples”

In both cases: Machine learning models learn from the data at hand

Pre-processing

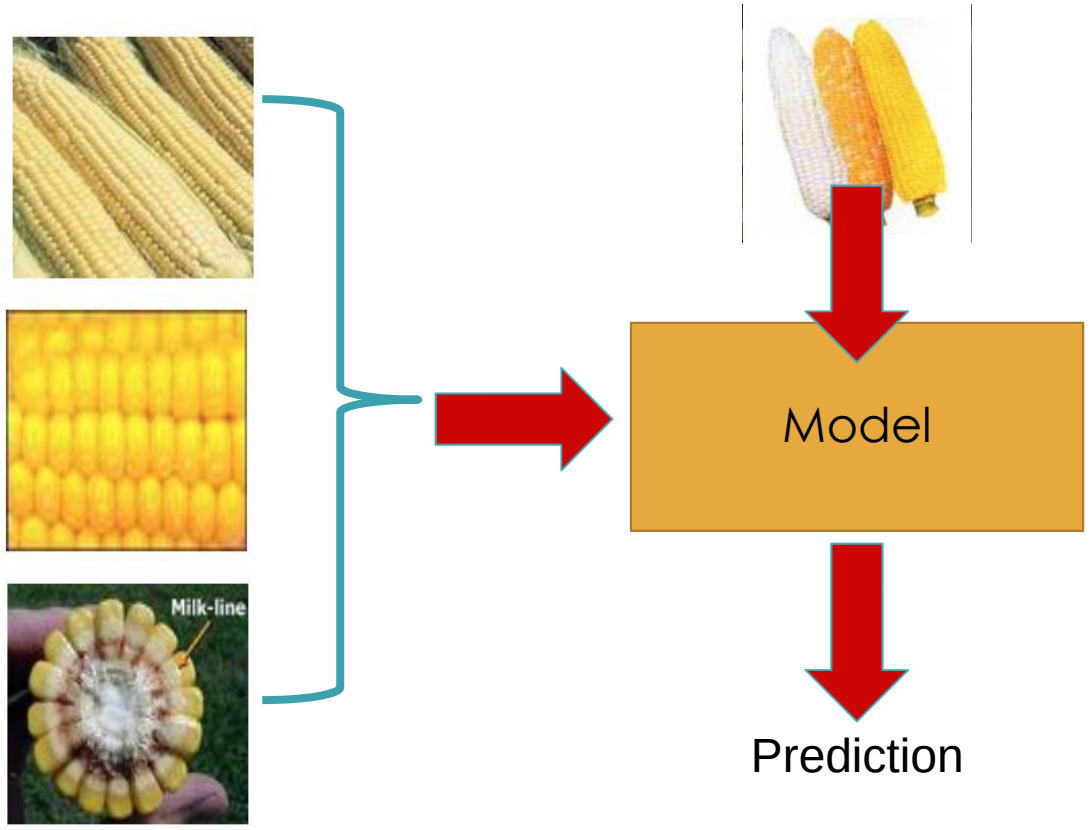
- Machine learning learns from data, without respect to what is an artifact and what is a real signal.

- As a result, care and appropriate pre-processing is required. Examples include step-changes in response, shifts of the signal in time, changes to intensity from artefacts (e.g. topographic), etc.

- Old adage: garbage in, garbage out.

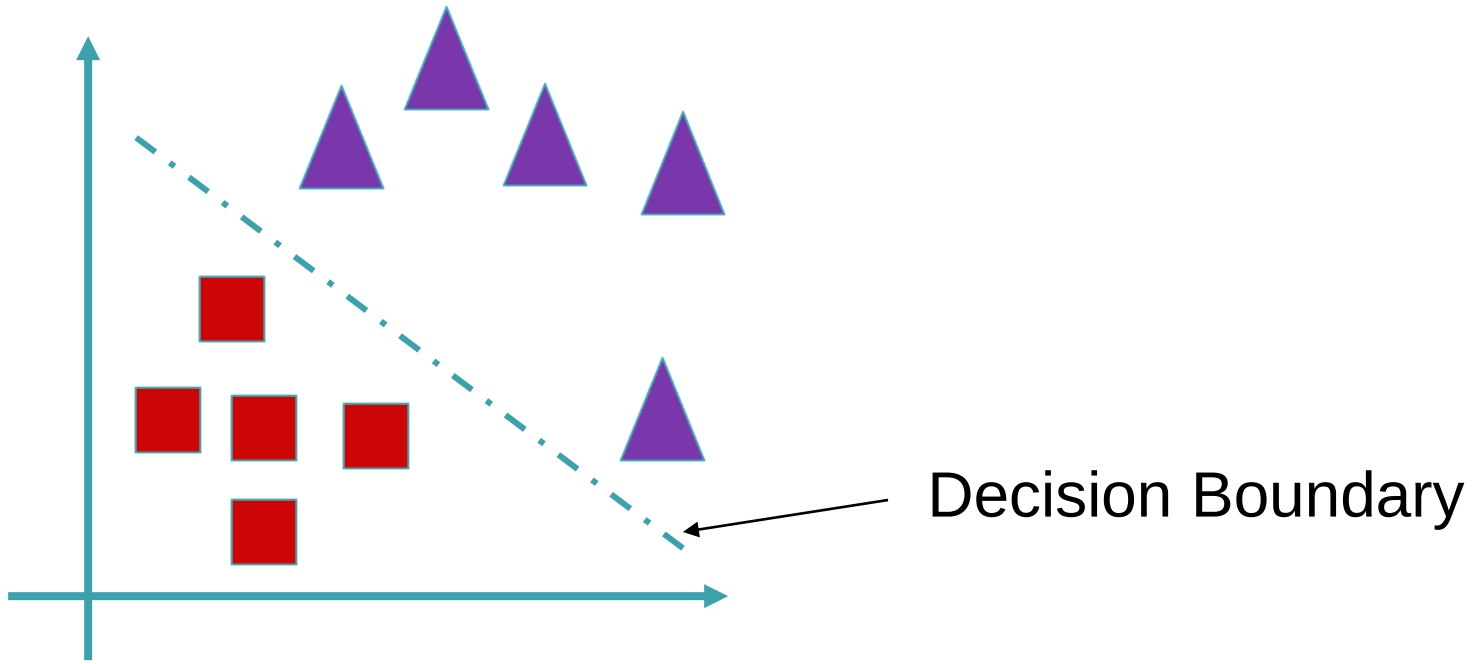
Supervised Classification

- Supervised classifiers can automatically classify data once the model has been trained on a ‘training set’.



Supervised Classification: Support Vector Machines

- Support Vector Machines are one common method, and is conceptually easy to understand.

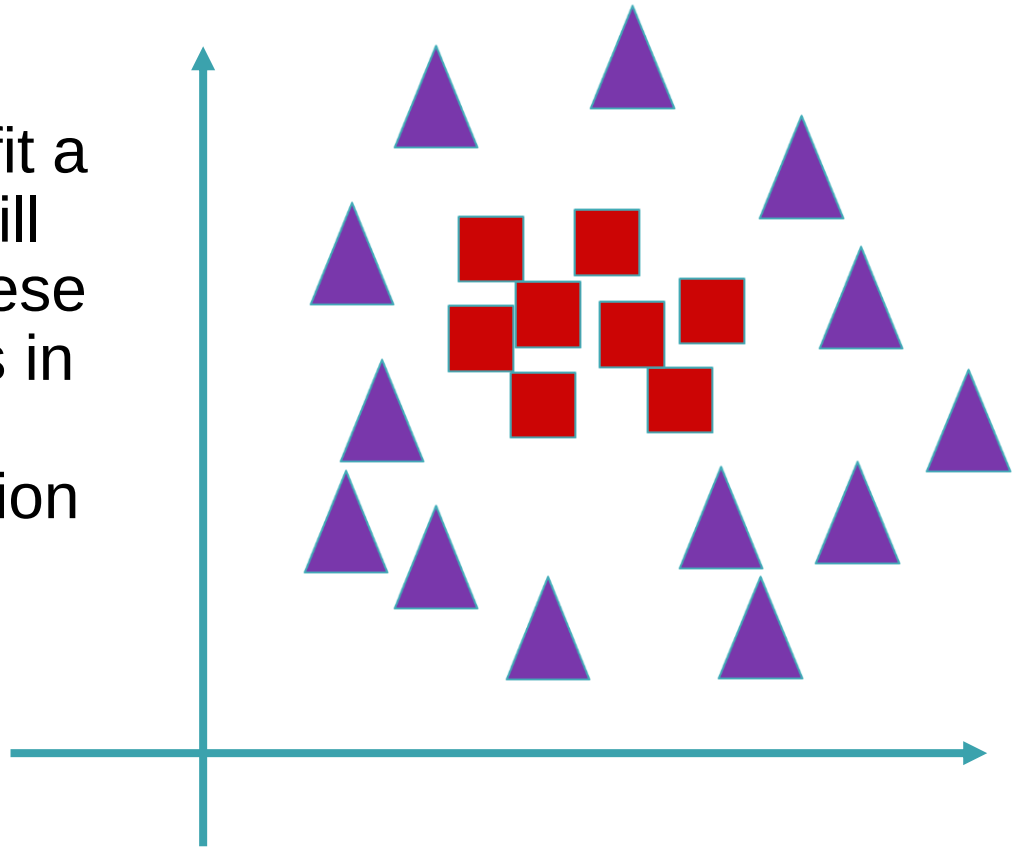


SVM are linear classifiers that fits lines (planes/hyperplanes in higher dimensions), such that the margin (separation between the line and the nearest training point) is maximized.

Supervised Classification

- But what to do when it cannot be linearly separated? E.g.,

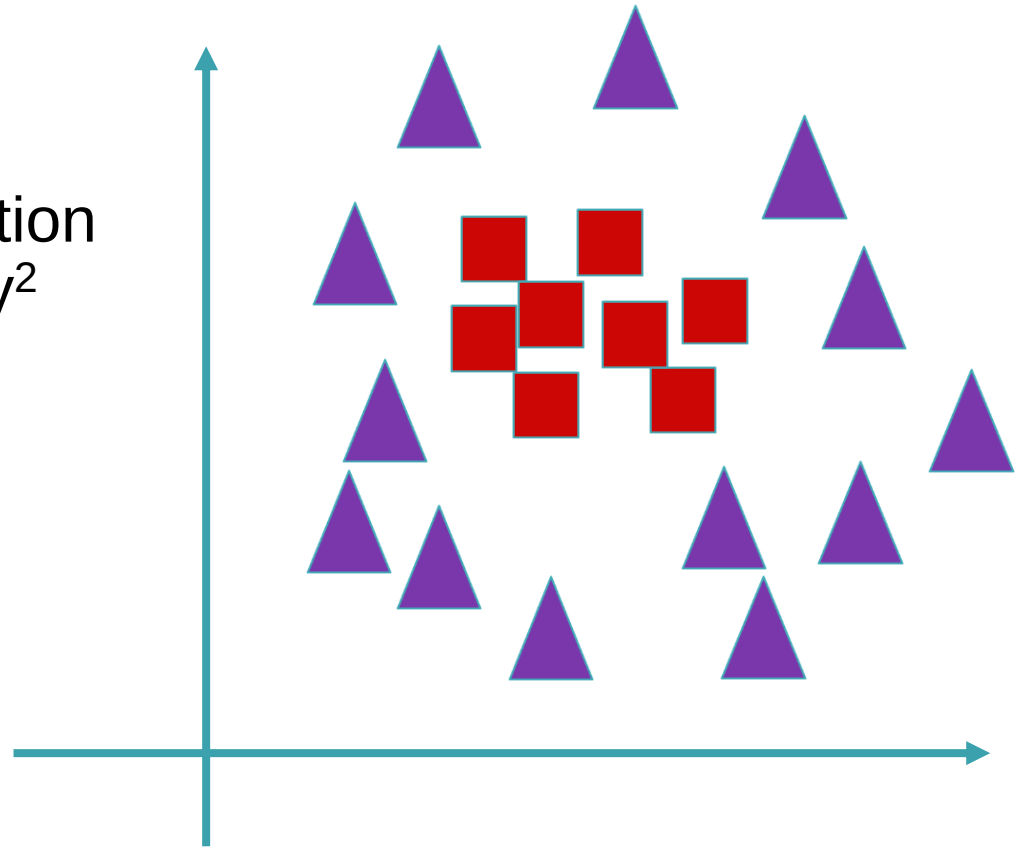
We cannot fit a line that will separate these two classes in this representation



Supervised Classification

- Solution: Transform to a space where they can be separated.

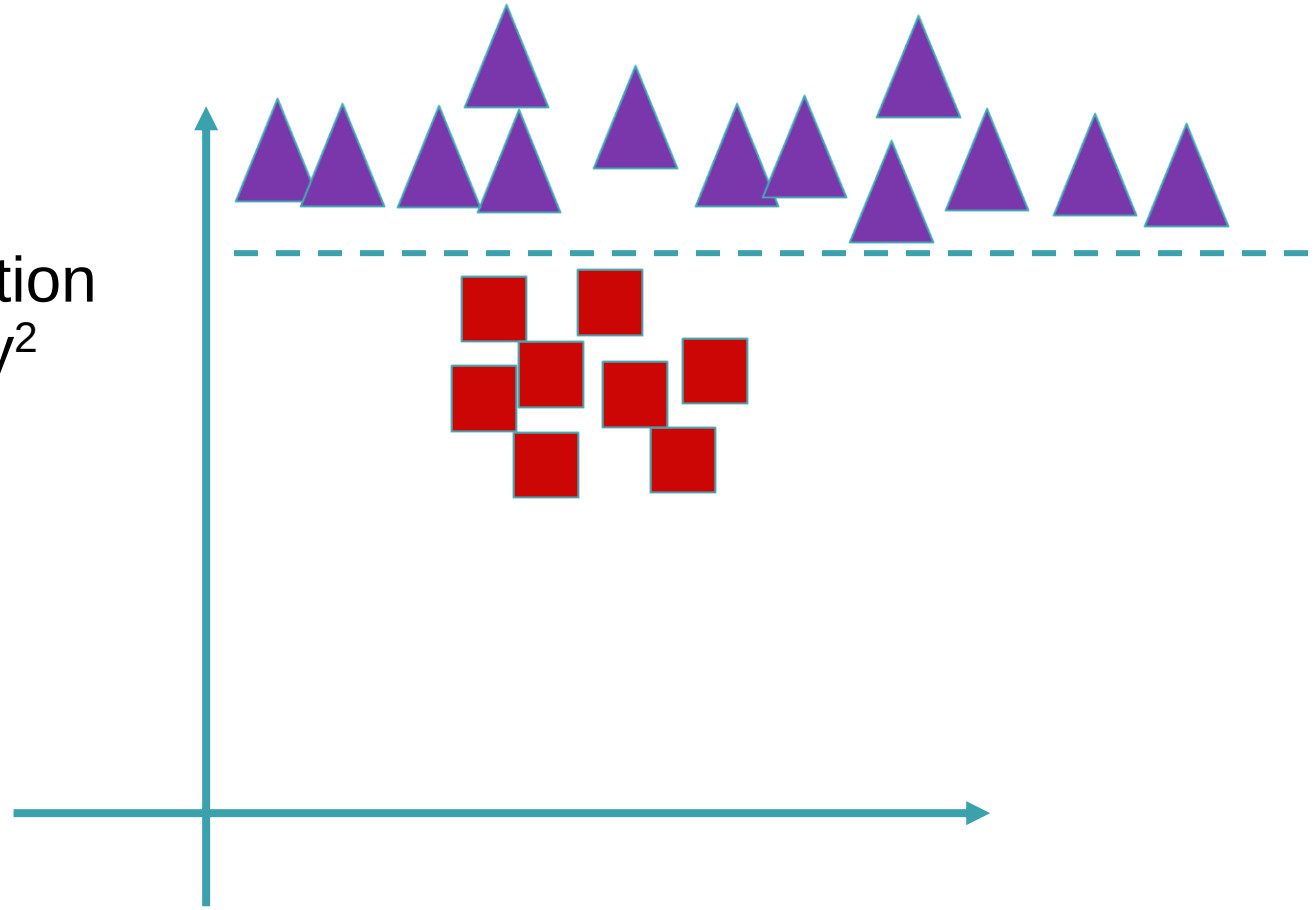
Apply a transformation
$Z = x^2 + y^2$



Supervised Classification

- Solution: Transform to a space where they can be separated.

Apply a transformation
$Z = x^2 + y^2$



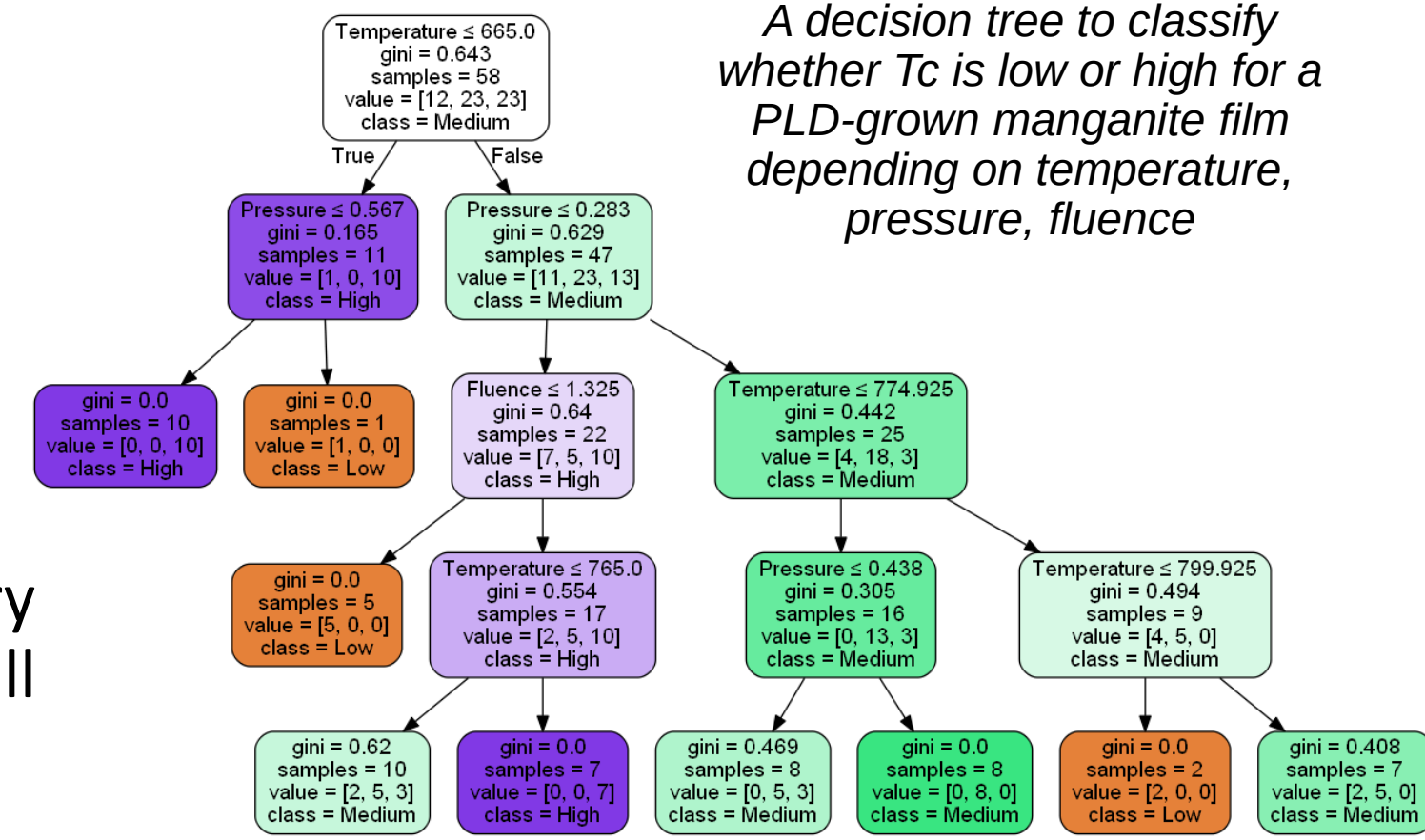
Supervised Classification: Kernels

- This is called the kernel trick, where a kernel function is applied to enable the separation to occur in the transformed space.

- This is a very common method used in ML models to enable learning of nonlinear functions

Supervised Models: Decision Trees

- A very common method that's used to classify data is the decision tree
- The decision tree is easily interpretable, and essentially learns to split data based on values of features.
- Advantages: will ignore featureses that are unnecessary to the classification, when small they are easy to interpret, and have often been shown to be extremely good (esp. with random forests, bagging, etc.)



A decision tree to classify whether Tc is low or high for a PLD-grown manganite film depending on temperature, pressure, fluence

Young et al., J. Appl. Phys. **123** 115303 (2018)

Decision Trees

- The decision tree is learned by using the concept of information entropy. Each split is chosen to maximize the purity of each daughter node of the tree.
- Numerous types of metrics can be used here, including KL divergence, or Gini Impurity, or Variance Reduction
- The main issue with decision trees is the tendency to overfit to the training data. Ensemble methods can reduce this tendency.

Unsupervised Classification

- In many cases, we do not have labeled examples. In this case, we can turn towards unsupervised methods
- A common method is k-means clustering, but there are numerous others
- Unsupervised methods can also be used to automatically learn features in the dataset; this will be explored later in the presentation by Dr. Agar.

K-Means Clustering: Unsupervised learning

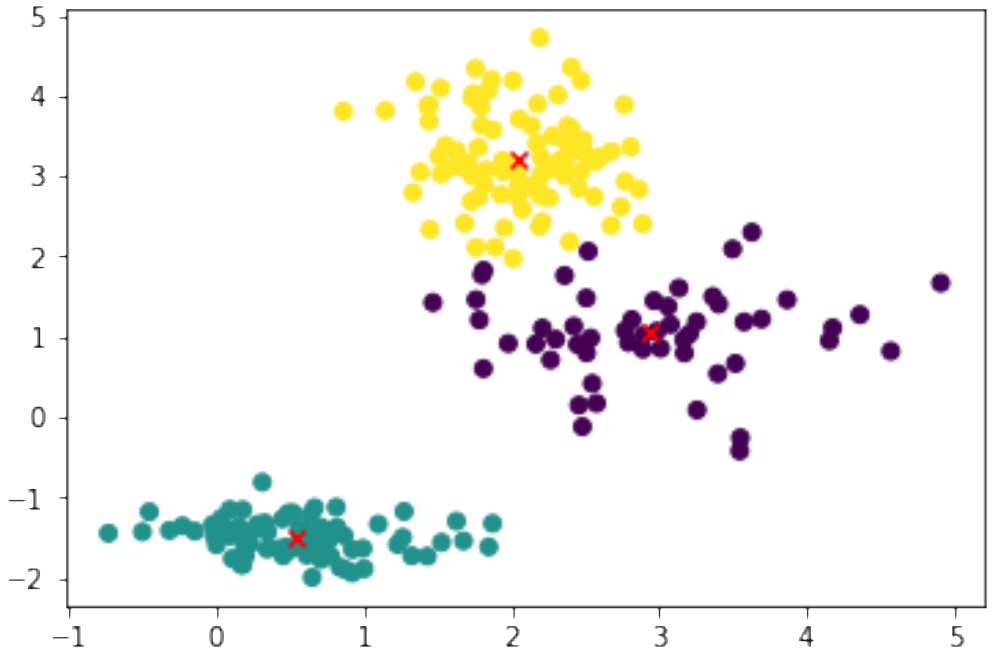
- One example: we would like to quickly a visualize large dataset, i.e. see what types of responses are most prevalent and group each response point accordingly.
- This is termed ‘clustering’. The easiest and most widely used method is the k-means algorithm

K-means Clustering algorithm, to separate data ($x_1, x_2, \ldots, x_n$) into *k* clusters

$$\arg\min_{\boldsymbol{s}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2$$

where μ_i is the mean of points in S_i

(Determine **S** = $\{S_1, S_2, \ldots S_k\}$, such that within cluster sum of squares is minimized)



Colab Notebook

- Let's look at the notebook

IMC20-AI-Tutorial/jupyterbook/Intro_to_Machine_Learning at main · m3-learning/IMC20-AI-Tutorial (github.com)