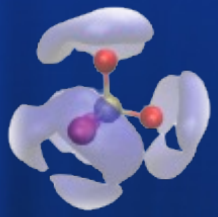
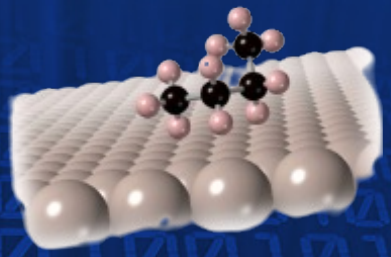
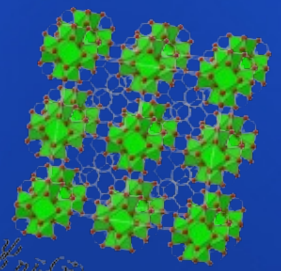
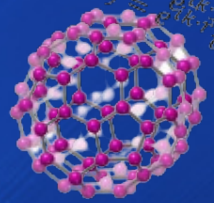
Machine Learning Models for Materials Discovery



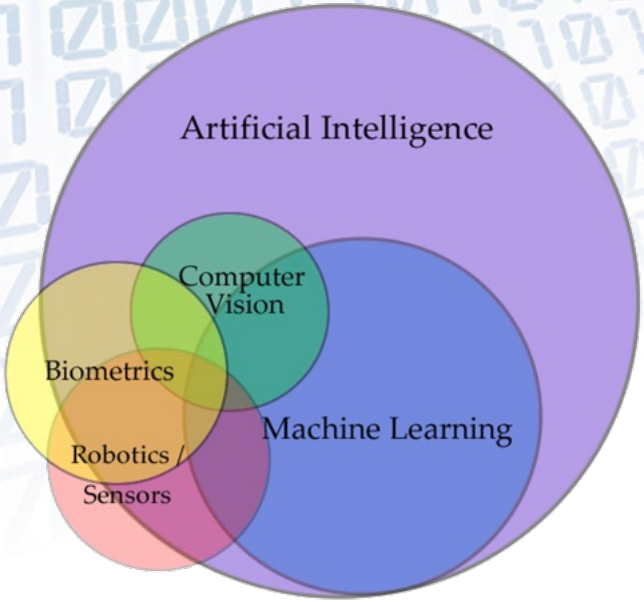






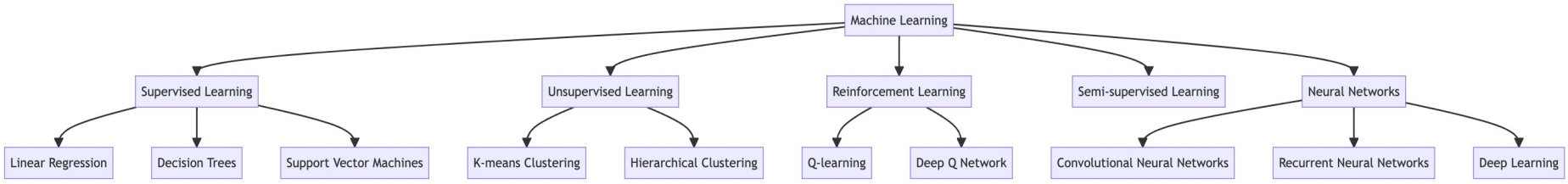
Marco Fronzi

Artificial Intelligence and and Machine learning



Machine Learning:

subset of artificial intelligence that focuses on **algorithms and models that enable computers to learn and make predictions or decisions without explicit programming**. It involves the development of computational systems that can automatically learn and improve from experience, without being explicitly programmed for every specific task.

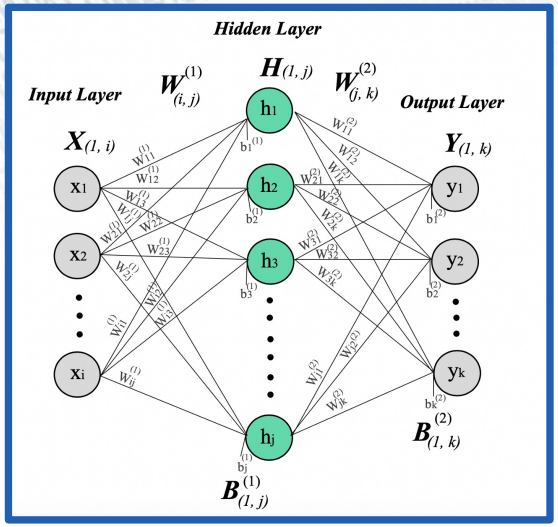


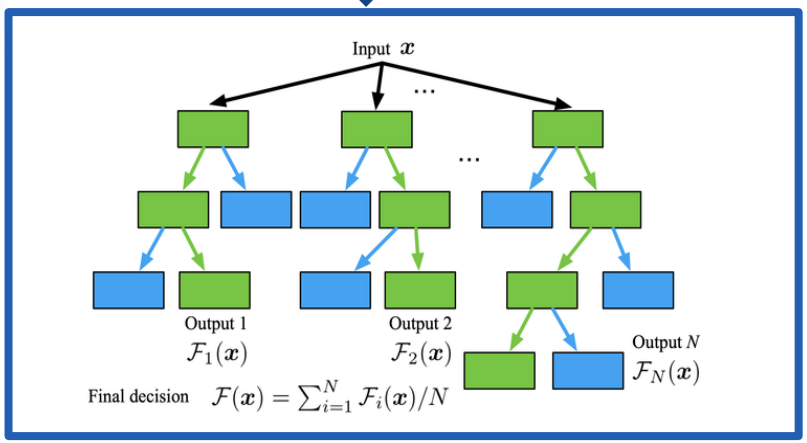
Second Slide Master

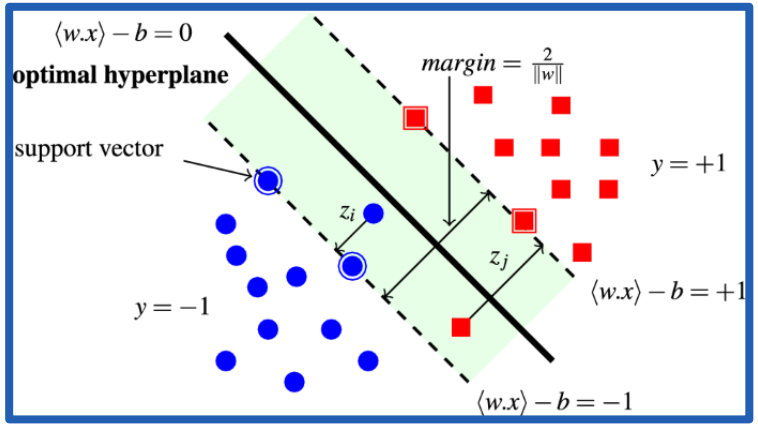
Neural Networks

Support Vector Machine

Random Forest







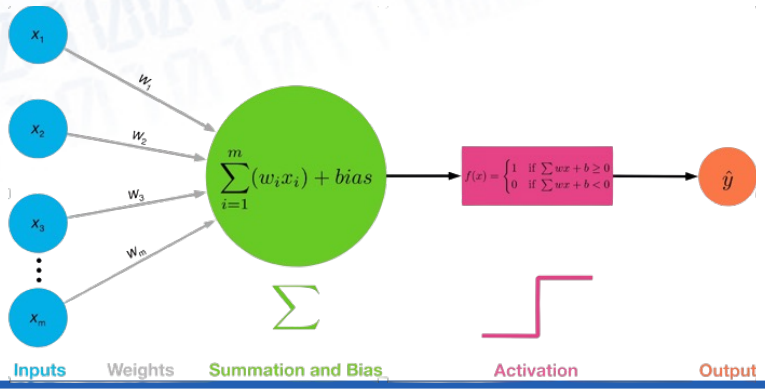
What is a Machine Learning Model?

Machine learning object is to find transfer function
To map structural parameters to target property

$\vec{x}$

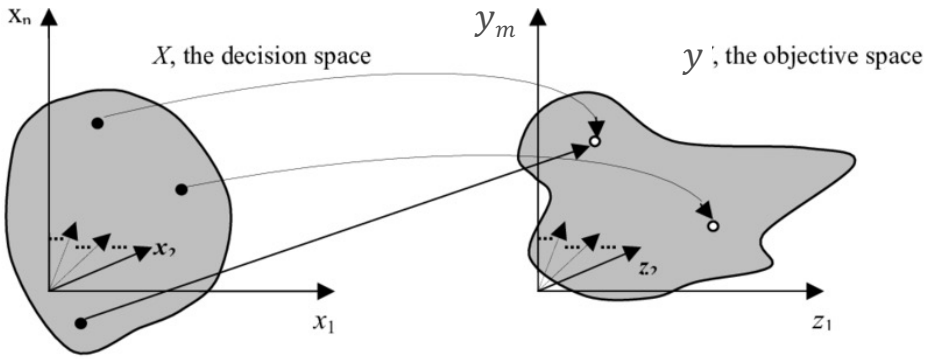
Parameter Space (Descriptors or Features)

$$f_{\lambda_1,\lambda_2,\ldots,\lambda_m}(x_1, x_2, \ldots, x_n) = y$$



y

Target Property



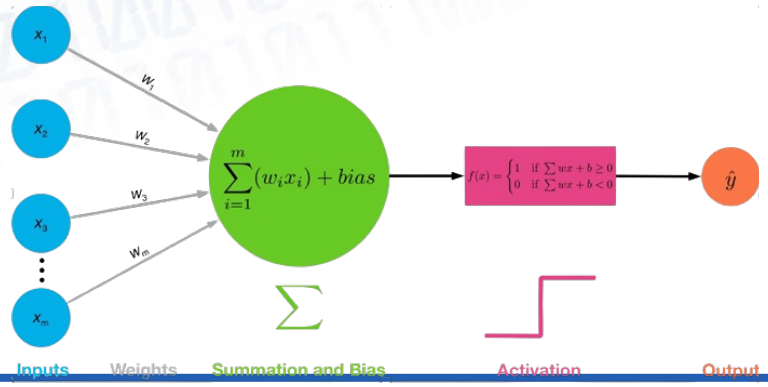
What is a Machine Learning Model?

Machine learning object is to find transfer function
To map structural parameters to target property

$\vec{x}$

Parameter Space (Descriptors or Features)

$$f_{\lambda_1,\lambda_2,\ldots,\lambda_m}(x_1, x_2, \ldots, x_n) = y$$



y

Target Property

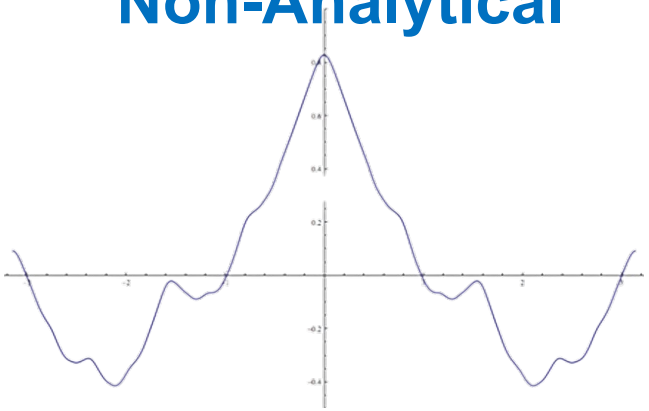
ML model is a function that maps n-dimensional space to a number
(regression = continue distribution)
(classification = discrete distribution)

If it maps to a discrete space (integer numbers) = classification
If it maps to a continuous space (real numbers) = regression

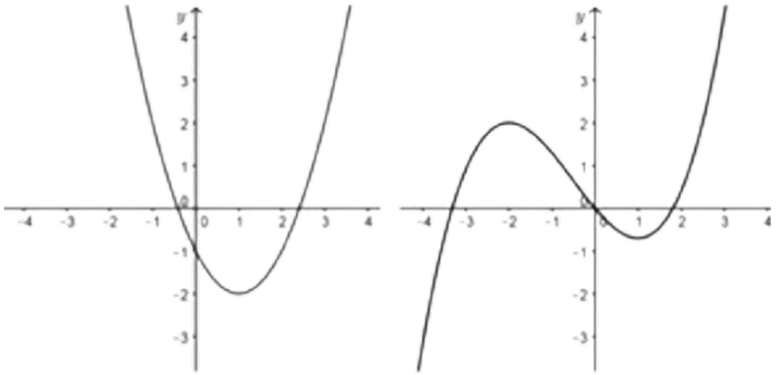
ML Models Properties

Machine learning object is to find transfer function
To map structural parameters to target property

Non-Analytical



Analytical



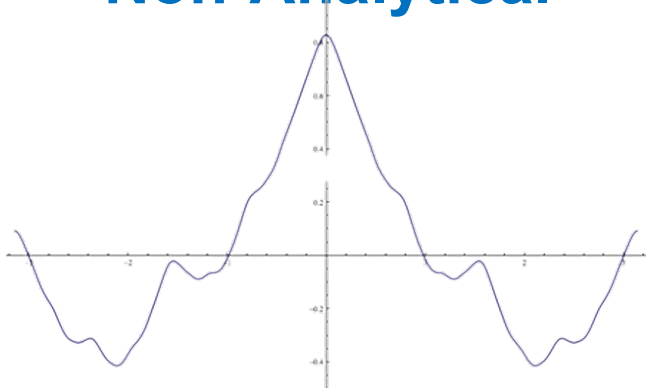
Properties= highly non linear; analytically unknown; numerically known;

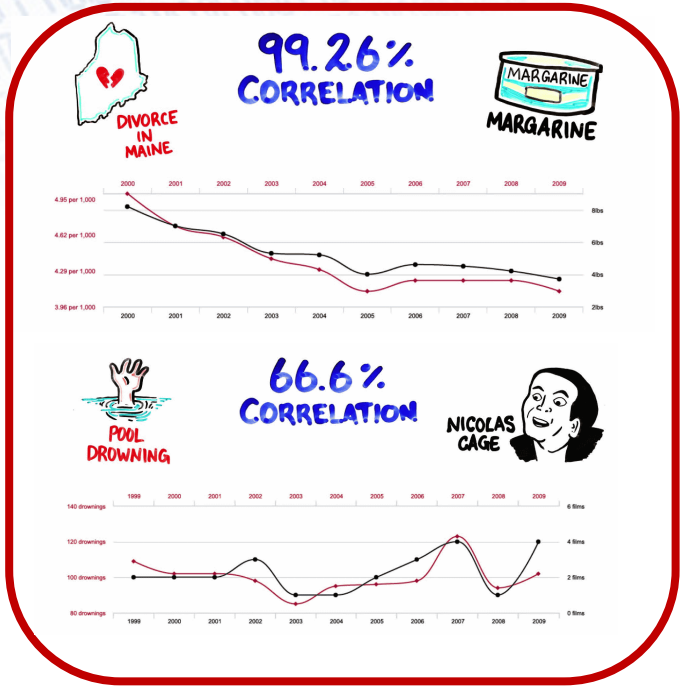
Depends on internal parameters (hyper-parameter)

ML Models Properties

Machine learning object is to find transfer function
To map structural parameters to target property

Non-Analytical





Machine learning models

Pro: very good at finding correlations

Cons: very bad at understanding causation

ML Models Properties

Pros:

1.Versatility: ML models can be applied to a vast array of problems, not just those in the physical sciences but across industries, such as finance, healthcare, marketing, etc.

2.Adaptability: ML models can learn from new data, improving their accuracy and effectiveness over time.

3.Efficiency: be faster and more computationally efficient than complex QM simulations.

Cons:

1.Data Dependence: The accuracy of ML models is heavily dependent on the quality and quantity of the training data.

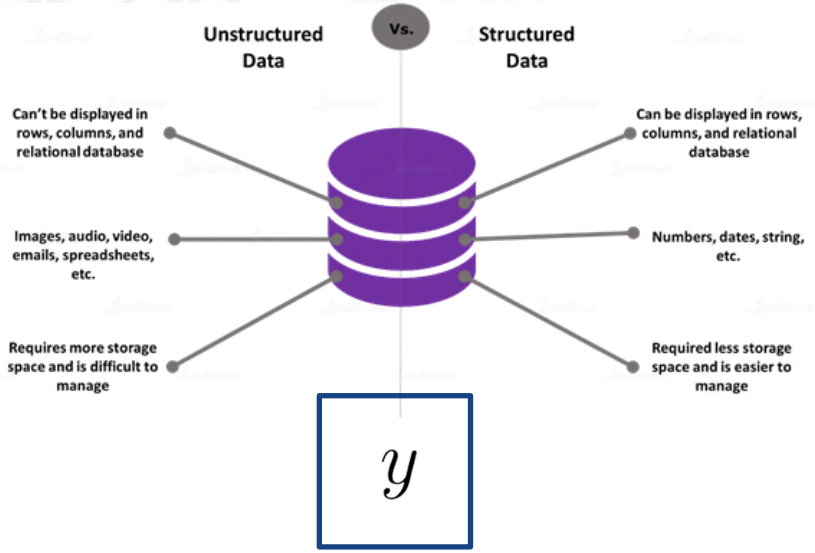
2.Black box: ML models can act as "black boxes," making it hard to understand why they make certain predictions Deep understanding of the science is needed

Building a ML Model

$$f_{\lambda_1,\lambda_2,\ldots,\lambda_m}(x_1, x_2, \ldots, x_n) = y$$

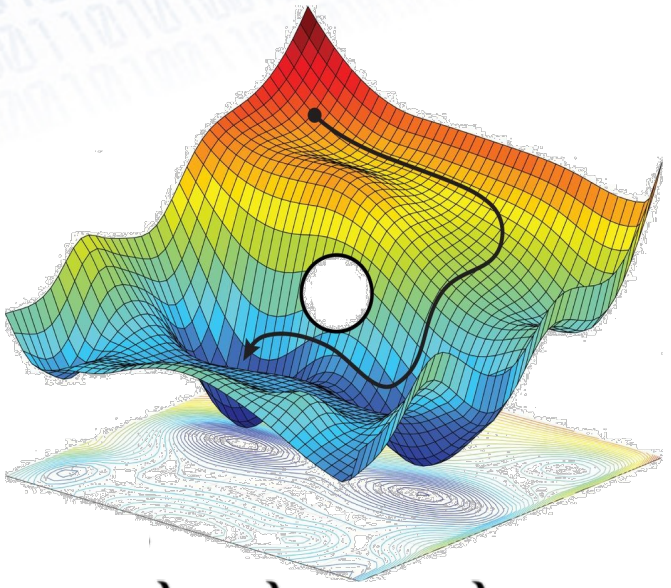
Data-set to provide as example (Comparable to Experience)

$$(x_1, x_2, \ldots, x_n)$$



$$y$$

Lot of work to do !!!

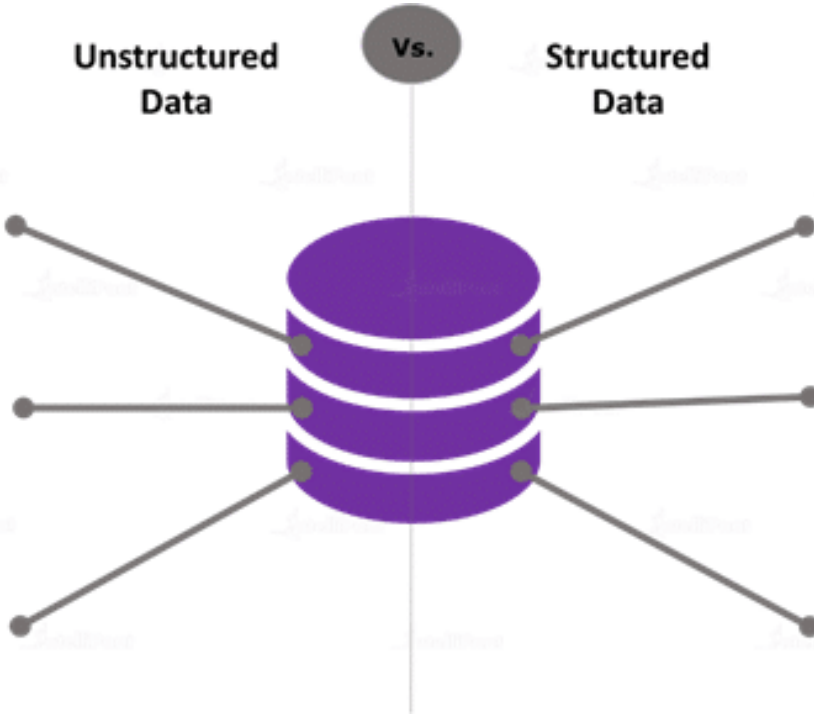


$$\lambda_1,\lambda_2,\ldots,\lambda_m$$

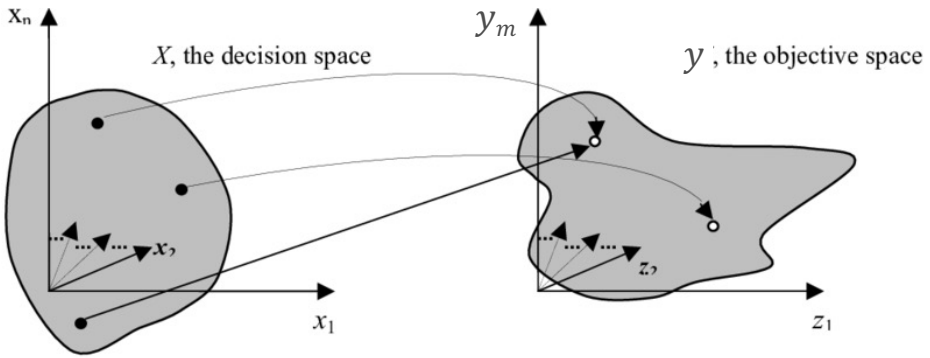
Easy task!
Many optimizers available

Building a Model Require a set of elements of known target property

$$f_{\lambda_1,\lambda_2,\ldots,\lambda_m}(x_1, x_2, \ldots, x_n) = y$$

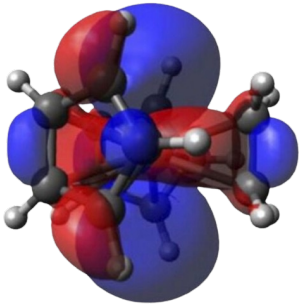


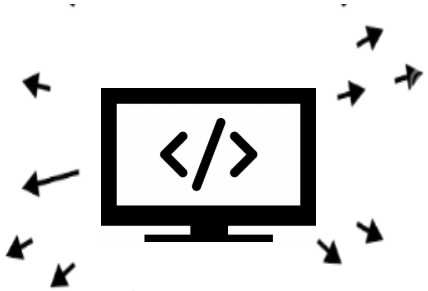
- Must provide a set of elements that ML can use to learn how to map the two spaces



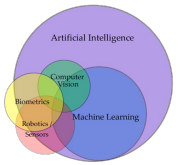
Quantum Mechanics and Machine Learning

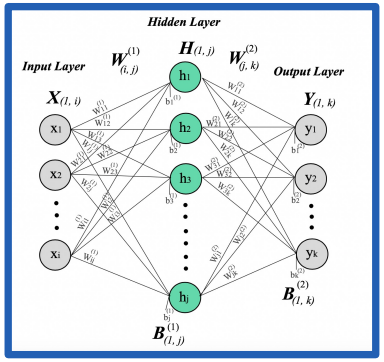
$$H(t)\,|\psi(t)\rangle = i\hbar\frac{d}{dt}\,|\psi(t)\rangle$$





$$R_{\text{learn}} = \arg\min_{R_\theta,\theta\in\Theta} \sum_{n=1}^{N} f\,(\,x_n\,, R_\theta(\,y_n\,)) + g\,(\theta)$$



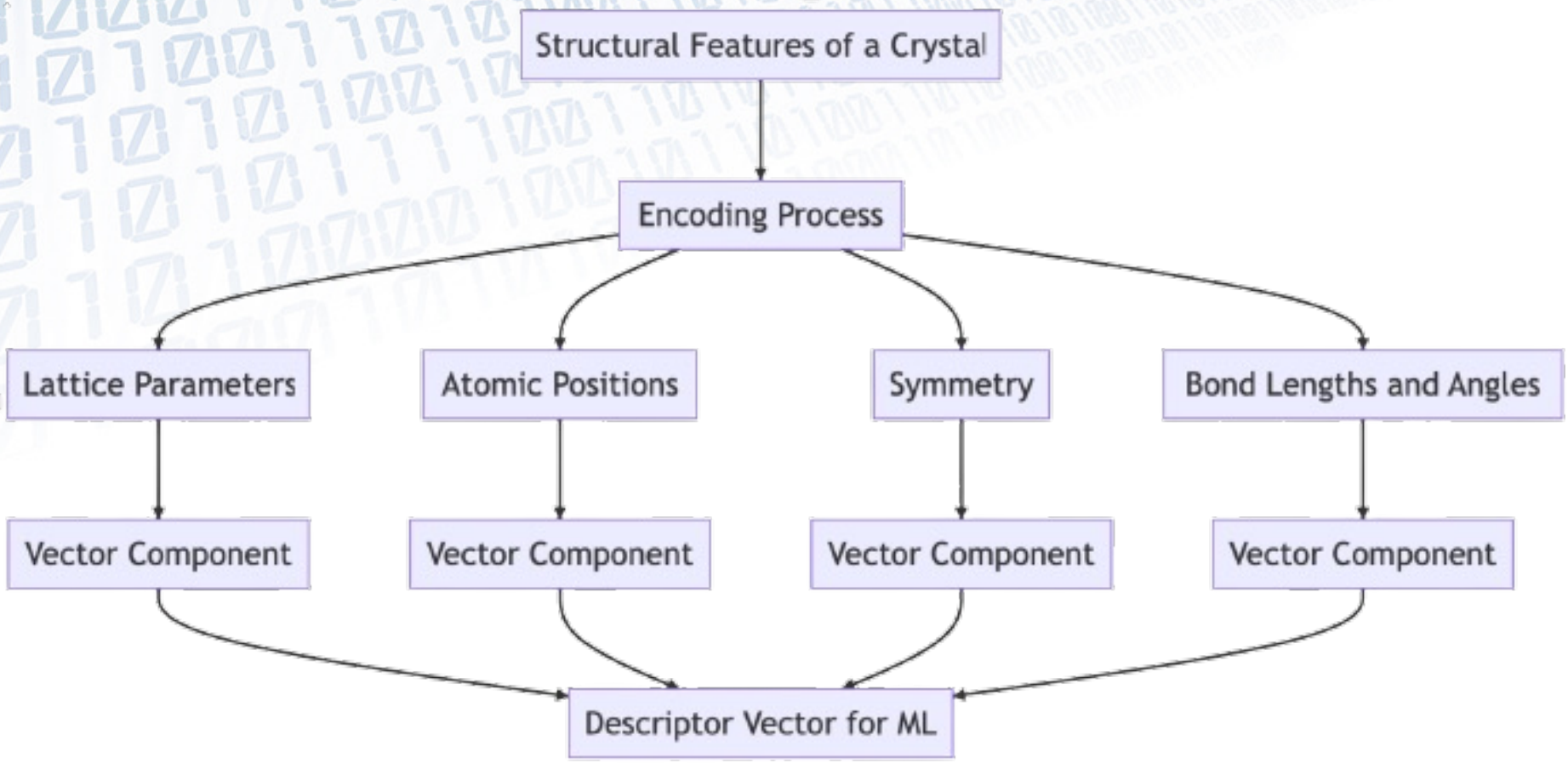


- Quantum mechanics provides a fundamental understanding of electron and lattice dynamics
- Allows prediction of material properties : electronic structure, phonon dispersion , etc.
- Ab initio calculations can guide the design of new materials with desired properties

- Machine learning models can predict material properties based on existing data .
- Faster than traditional quantum mechanical calculations
- Can identify patterns and relationships that are not immediately obvious

Featurization

Convert available material information (e.g. structural, elemental) into a numerical vector that uniquely describe the crystal

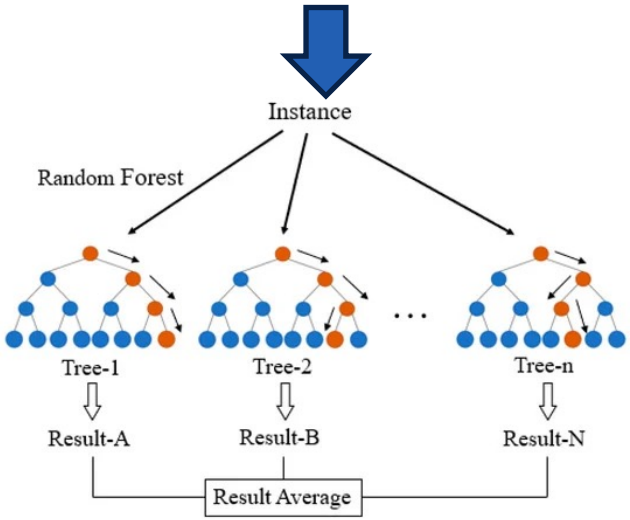


In this context, Feature and Descriptors are interchangeable words

Featurization

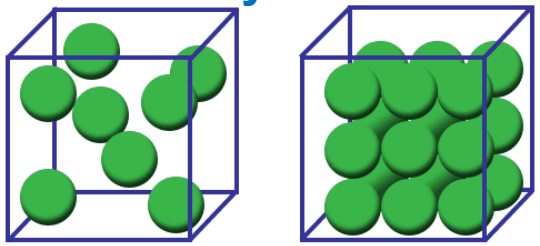
Features should have a weak correlation with each other and a significant correlation with the Target Function me

ElementProperty
DensityFeatures
CrystalNNFingerprint
GeneralizedRadialDistributionFunction
OPSiteFingerprint
ElectronicRadialDistributionFunction
ChemicalOrdering
VoronoiFingerprint
OxidationStates
Element, Composition

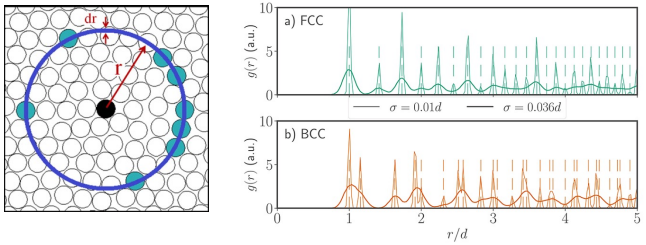


Random Forest Models can naturally indicate the most relevant features

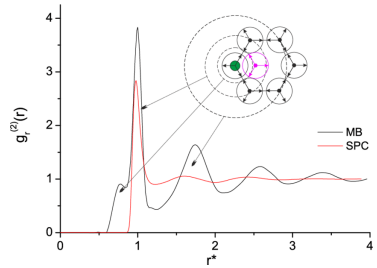
Density Features

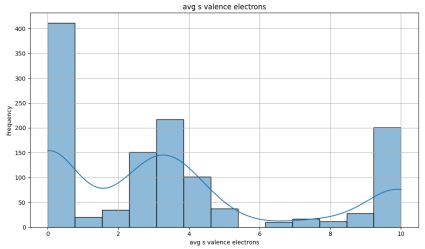


Generalized Radial Distribution Function



Electronic Radial Distribution Function

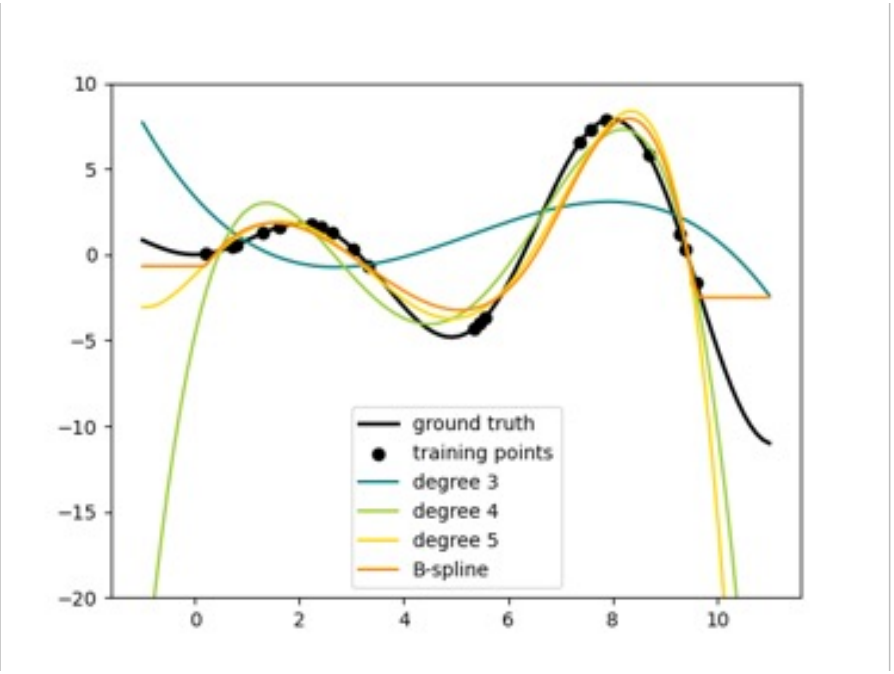




Polynomial Features

1.Purpose: Polynomial Features is used to create additional features by taking polynomial combinations of existing features.

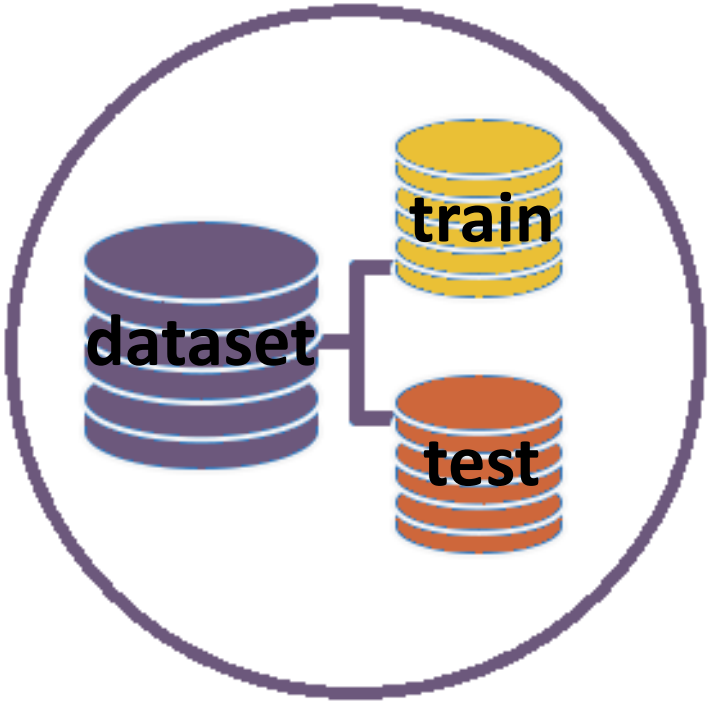
2.How It Works: Given an input dataset with n features (X1, X2, ..., Xn), and a specified polynomial degree d, Polynomial Features generates all possible combinations of features raised to powers up to d.



Select Samples

Two sets are necessary to build a model:
Training set used to optimize the model (tune the hyperparameters)

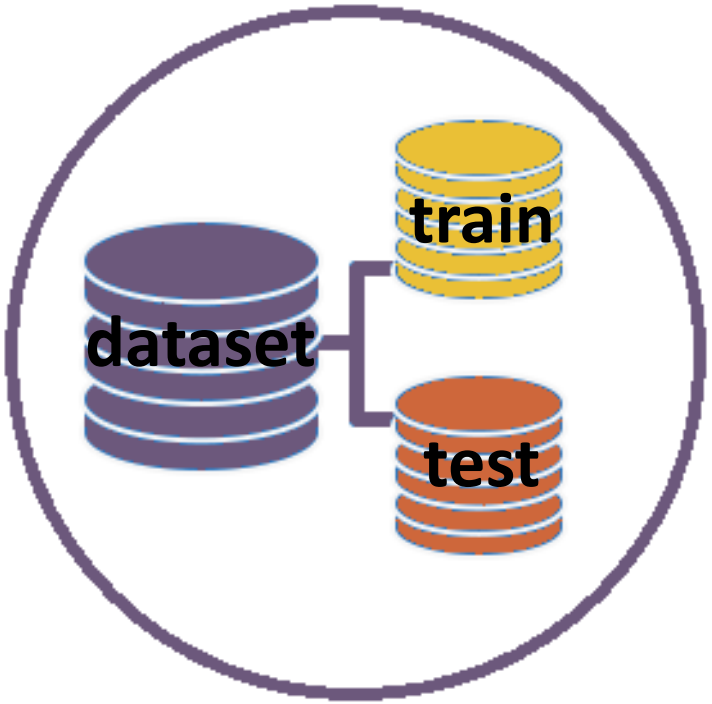
Test set: use to check quality of the model after optimization
Test if the model is able to generalize with reasonable error



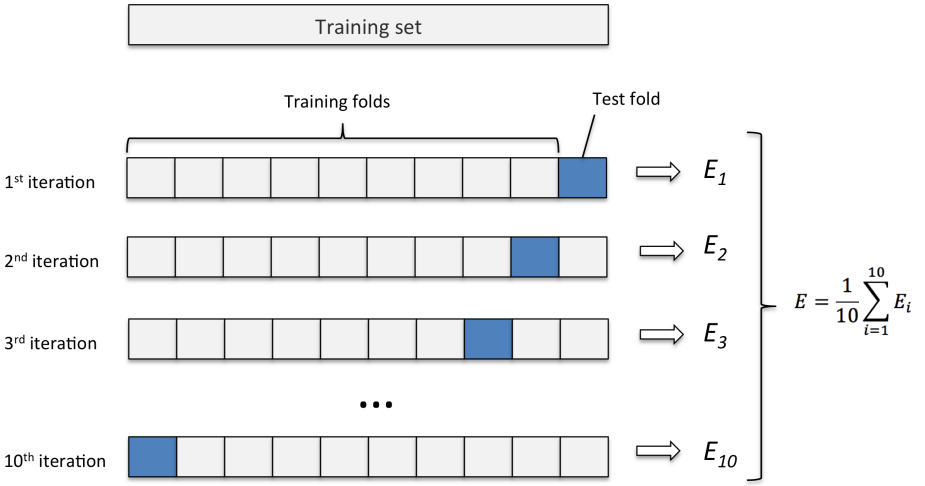
Train:test=80:20

Cross Validation

Perform multiple optimization choosing each time a different training test subset



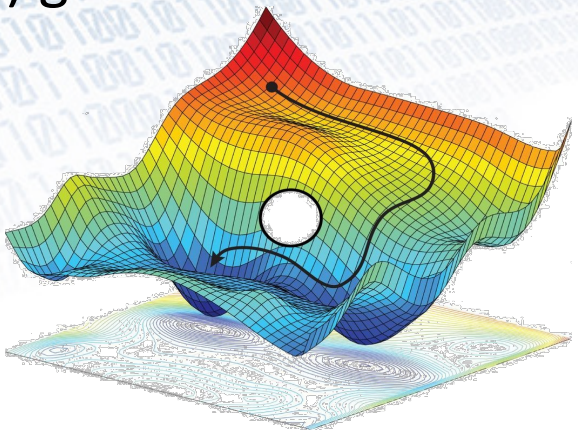
Train:test=80:20

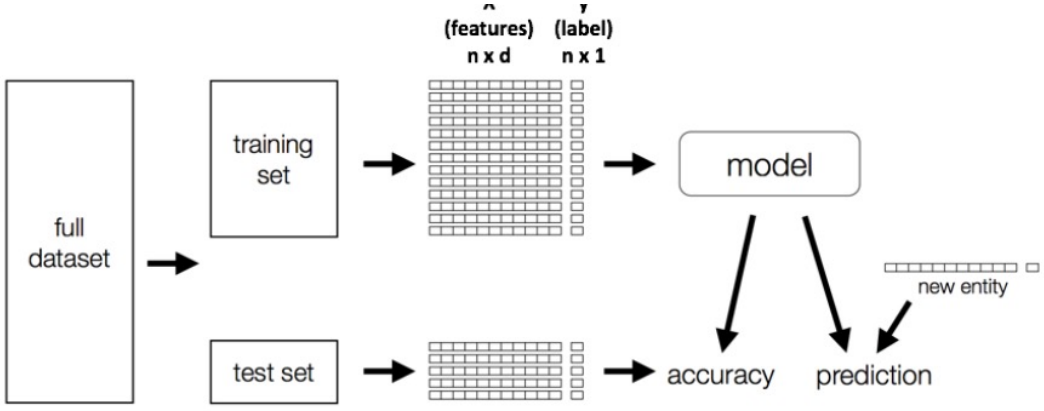


Optimization

Model training is a search for (energy) global minimum

Algorithms
=
provide M examples
and search for
global minimum
by changing hyper-parameters
values





In this context, Feature and Descriptors are interchangeable words

Metrics

Measures of prediction accuracy

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \qquad MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \qquad MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right|$$

Measure of the prediction of future outcomes, it measures how well observed outcomes are replicated by the model

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

y_i is the predicted data and x_i is the observed data

Loss Function

$$\text{Loss} = \sum_{i=1}^{n} x_i \log y_i$$

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|$$

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} (y_i - x_i)^2$$

} + Dropout regularization

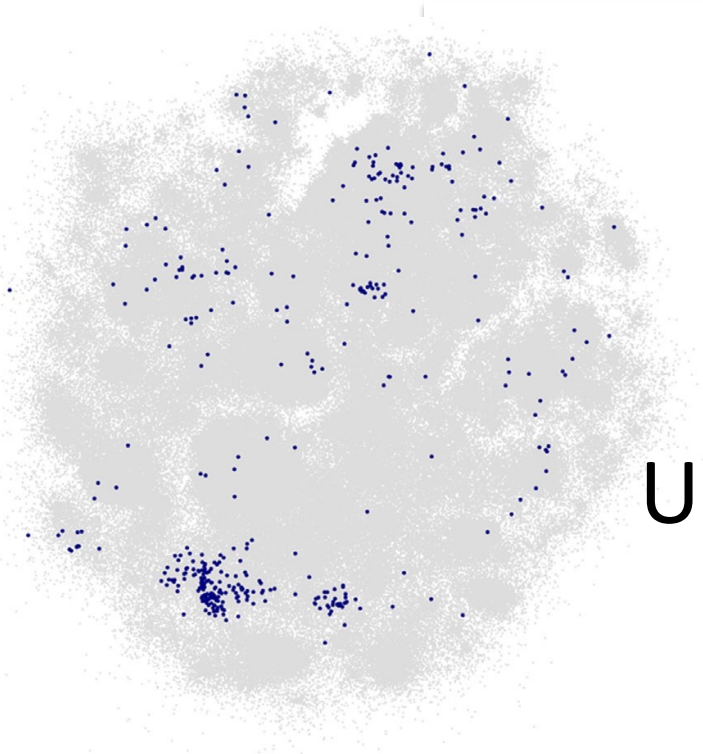
y_i is the predicted data and x_i is the observed data

Softmax
32 nodes

Select samples

Training and test set must be representative of the problem you want to solve

Must be uniformly distributed in the descriptors space

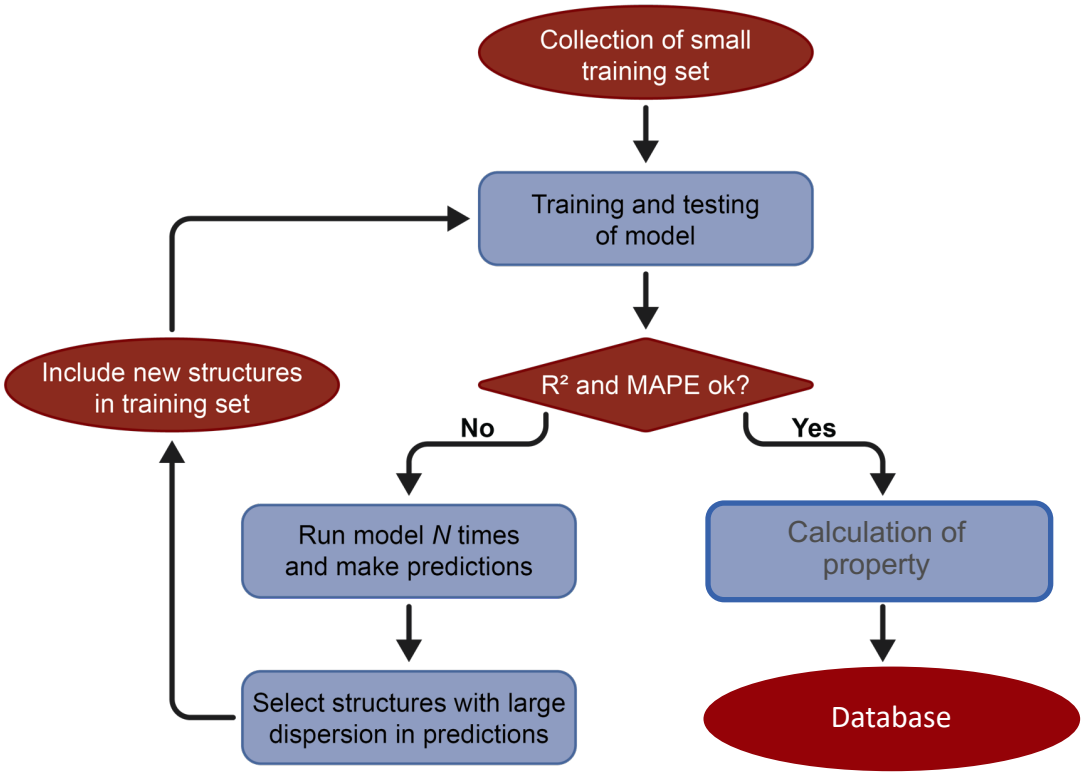


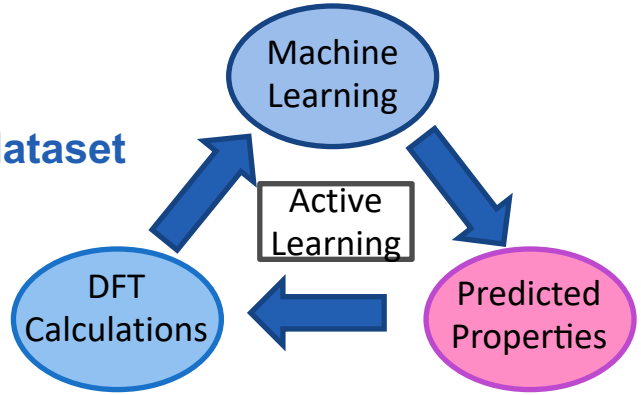
UMAP

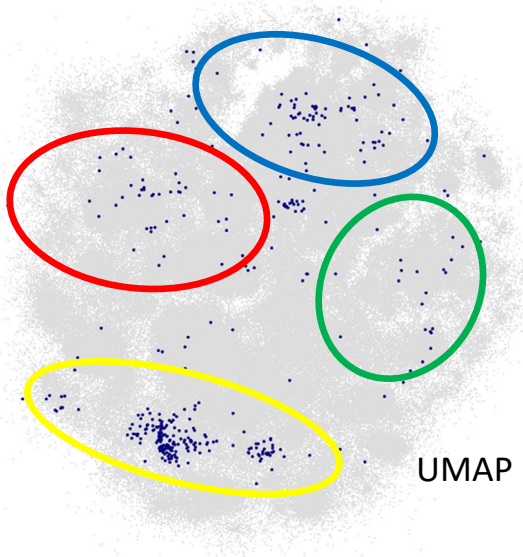
Active Machine Learning

1) Quality of dataset
2) Size of dataset
3) **Distribution of set in descriptor hyperspace**

Active learning is fundamental to select elements of DFT dataset



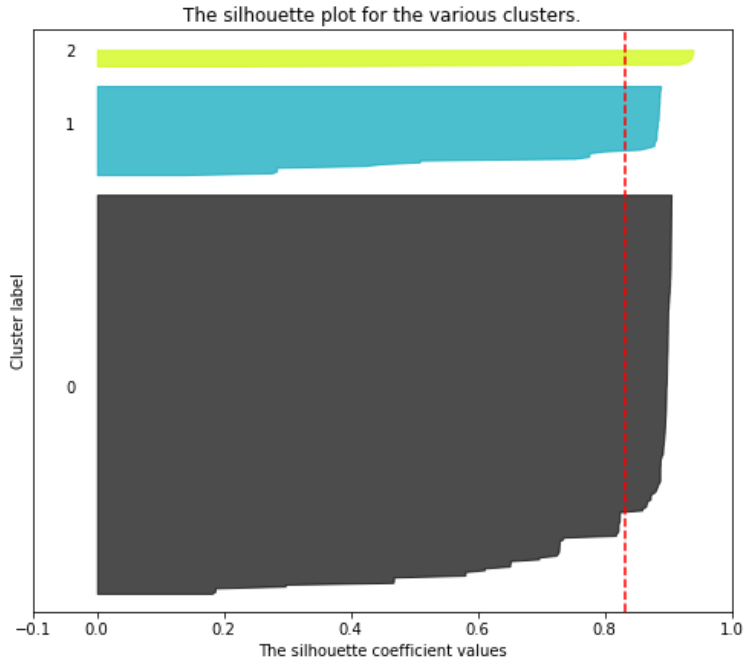


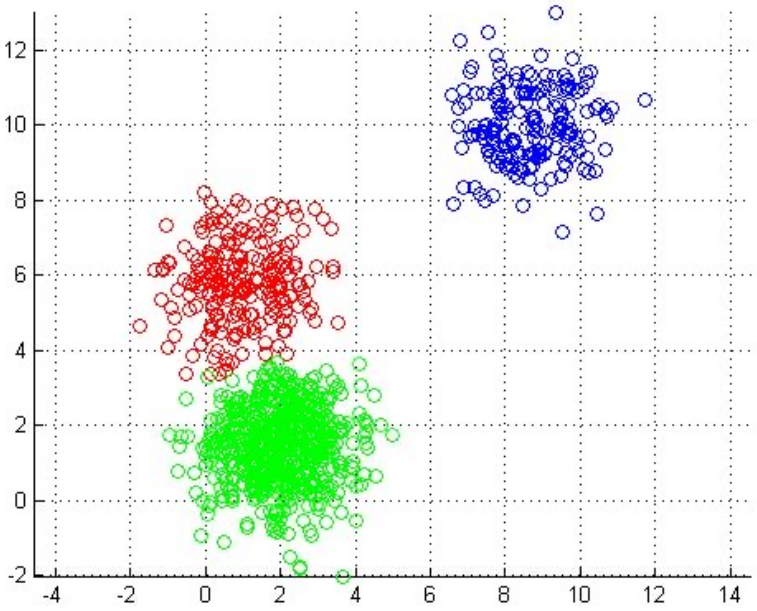


Cluster Analysis for Training Set Randomization

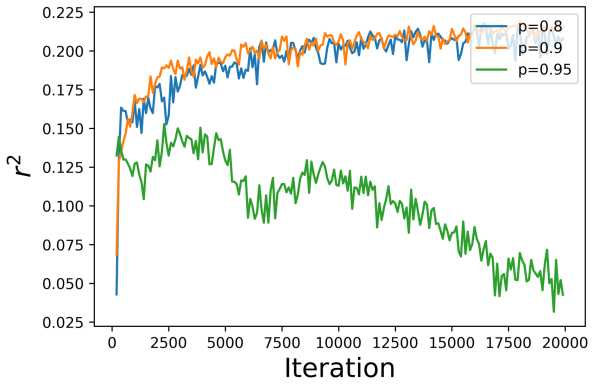
K-Means clustering has been carried out to ensure training samples are representative of the whole set

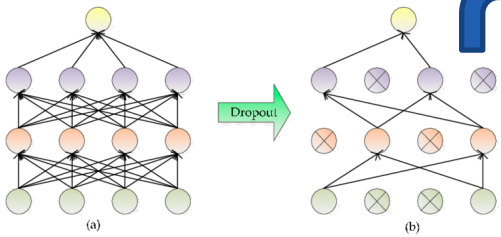
$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_j^i - c_j||^2$$



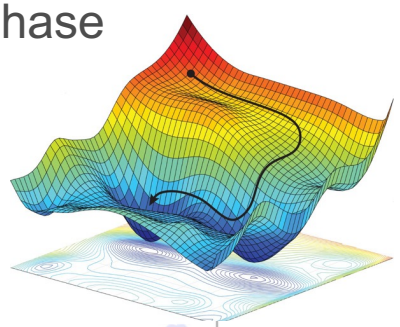


Setup ML dropout and loss function

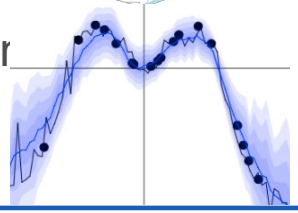




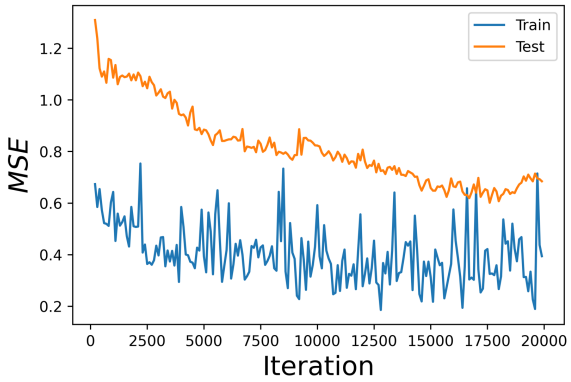
Optimization Phase



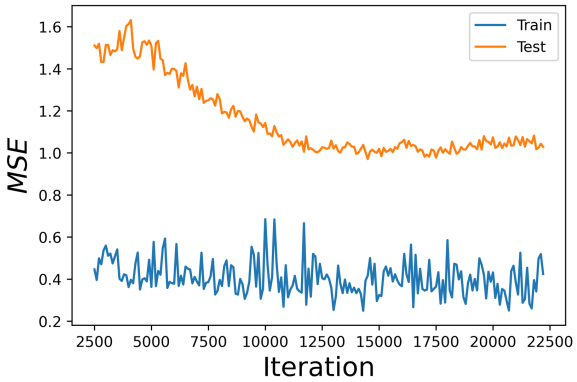
Statistical Distribution of Target



$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

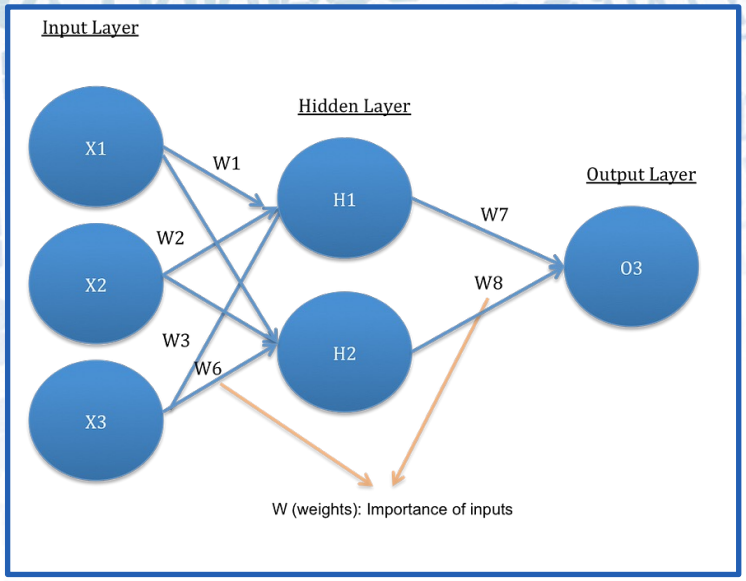


$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|$$



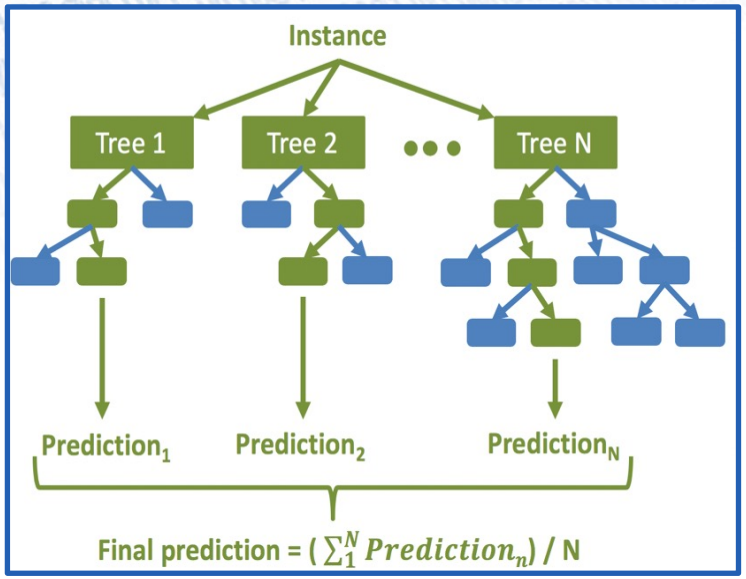
Neural Networks and Random Forest

Neural Networks



Neural networks are computational models inspired by the structure and functioning of the human brain. They consist of interconnected nodes, called neurons, organized in layers. Each neuron takes inputs, performs a computation, and produces an output. Neural networks can learn complex patterns and relationships in data through a process called training.

Random Forest



Random forests are ensemble learning models that combine multiple decision trees to make predictions. Each tree in the forest is built independently using a random subset of the training data and features. When making predictions, each tree or averages its individual output. A decision tree is a flowchart-like structure that uses rules based on features to make predictions. It starts with a root node, splits the data based on conditions, and reaches outcome predictions at the leaf nodes.

Neural Networks

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right) \tag{1}$$

Where:

- y is the output of the neuron,
- f is the activation function (e.g., sigmoid, ReLU, tanh),
- w_i are the weights,
- b is the bias,
- n is the number of inputs.

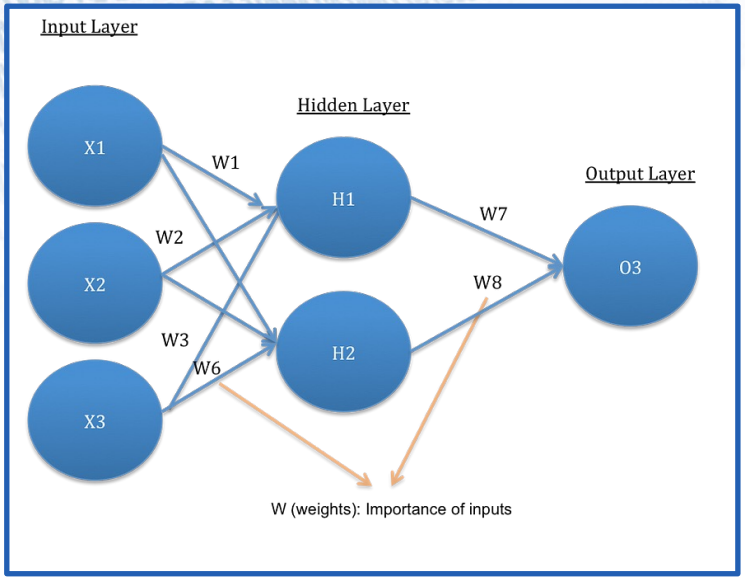
This equation represents the basic idea behind the weight update in neural networks. There are also advanced optimization algorithms like Adam, RMSprop, and others that have slightly different update rules, but they are all based on the concept of gradient descent.

During the optimization of Neural Networks, the weights w_i are adjusted using the gradient descent algorithm or its variants. The basic update rule for a weight w_i in gradient descent is:

$$w_i = w_i - \alpha \frac{\partial L}{\partial w_i} \tag{2}$$

Where:

- w_i is the weight being updated.
- α is the learning rate, a hyperparameter that determines the step size during the weight update.
- L is the loss function that measures how far the network's predictions are from the actual target values.
- $\frac{\partial L}{\partial w_i}$ is the partial derivative of the loss function with respect to the weight w_i, also known as the gradient. It indicates how the loss would change with respect to a small change in w_i.



– R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

– Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

– Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Random Forest

- Random Forests build multiple decision trees using the datasets.
- Each tree is built by recursively making binary splits in the dataset.
- The splits are made by minimizing a loss function. For regression, common loss functions are Mean Squared Error (MSE) and Mean Absolute Error (MAE). For classification, common measures are Gini Impurity and Entropy.

- R^2:

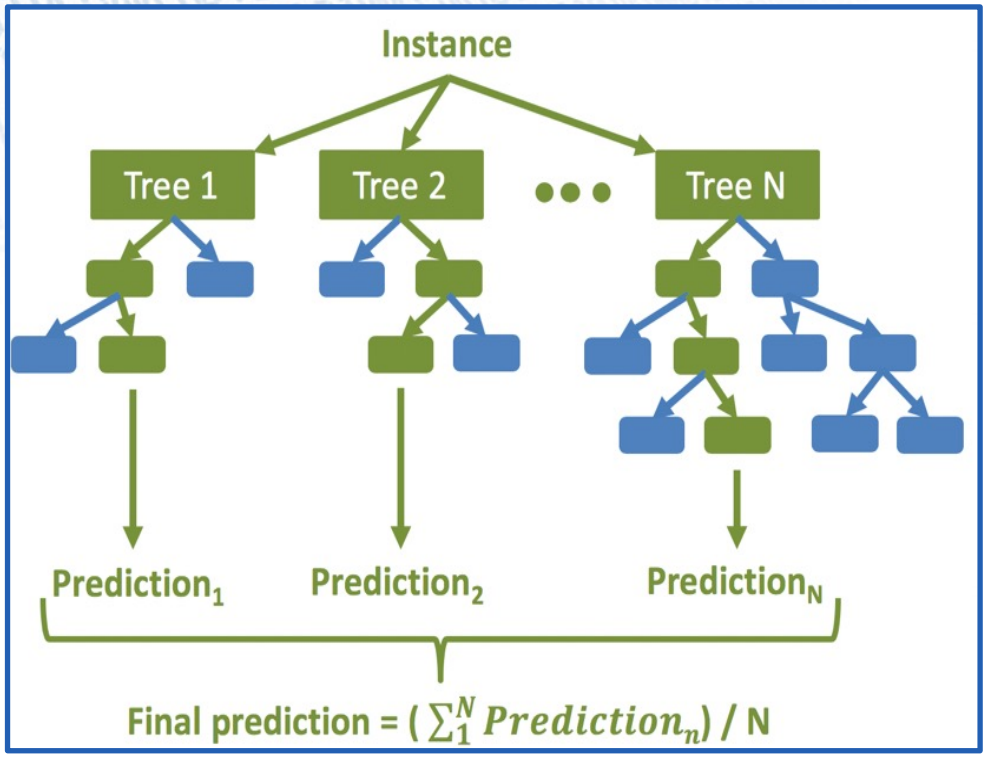
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

- Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$



Machine Learning Workflow