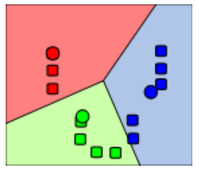
From (chemical) data to information

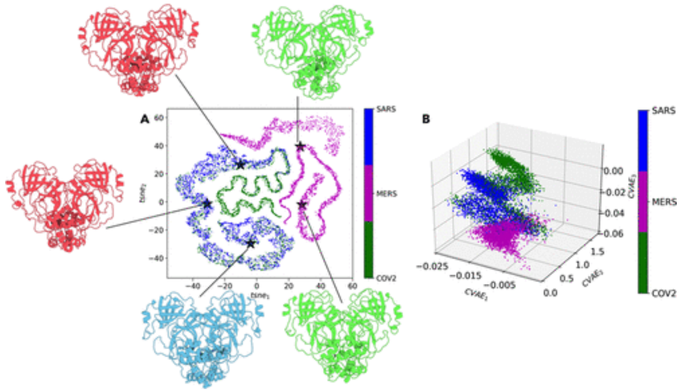
Introduction to Machine Learning Lecture 1

Dr Antonia (Toni) Mey

antonia.mey@ed.ac.uk

Room B23 JBB





Learning outcomes

- Understand the main pillars of machine learning
- Know about different **clustering** techniques as part of unsupervised learning
- Be able to use common nomenclature used in machine learning
- Use PCA to **reduce the dimensions** a data set
- Understand how to formulate a **regression** problem in machine learning
- Understand how to formulate a **classification** problem in machine learning
- A broad idea of different machine learning architectures for deep learning

Topics overview

Lecture 1

- What is machine learning?
- Examples of machine learning (in Chemistry)
- Introduction to **unsupervised learning**:
  - Clustering (k-means and others)
- Linear Regressions

Lecture 2

- Chemistry data is high dimensional
- **Dimensionality reduction**:
  - Principle component analysis
  - Time lagged component analysis
  - t-SNE and why not to use it

Lecture 3

- **Classification** problems
- Classifications in practice:
  - Random Forests
  - Support vector machine

Lecture 4

- Shallow Learning
- Deep Learning part I
  - Multilayer perceptron

Lecture 5

- Deep Learning part II
  - Transformers
  - Graph Neural Networks

Topics overview

Lecture 1

- What is machine learning?
- Examples of machine learning (in Chemistry)
- Introduction to **unsupervised learning**:
  - Clustering (k-means and others)
- Linear Regressions

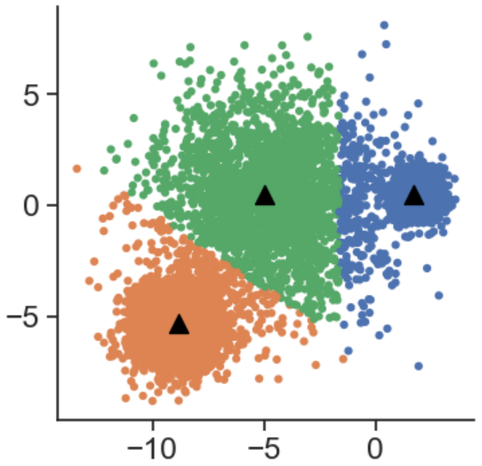
Lecture 2

- Chemistry data is high dimensional
- Dimensionality reduction:
  - Principle component analysis
  - Time lagged component analysis
  - t-SNE and why not to use it

Lecture 3

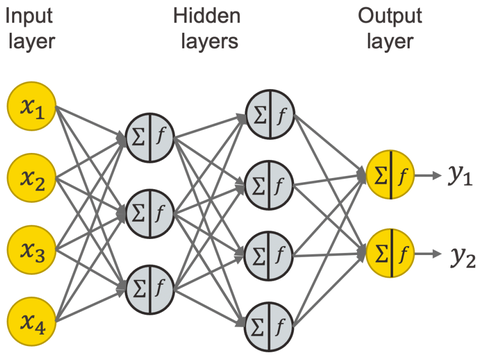
- Classification problems
- Classifications in practice:
  - Random Forests
  - Support vector machine

Workshop 1



Workshop 2

Workshop 3



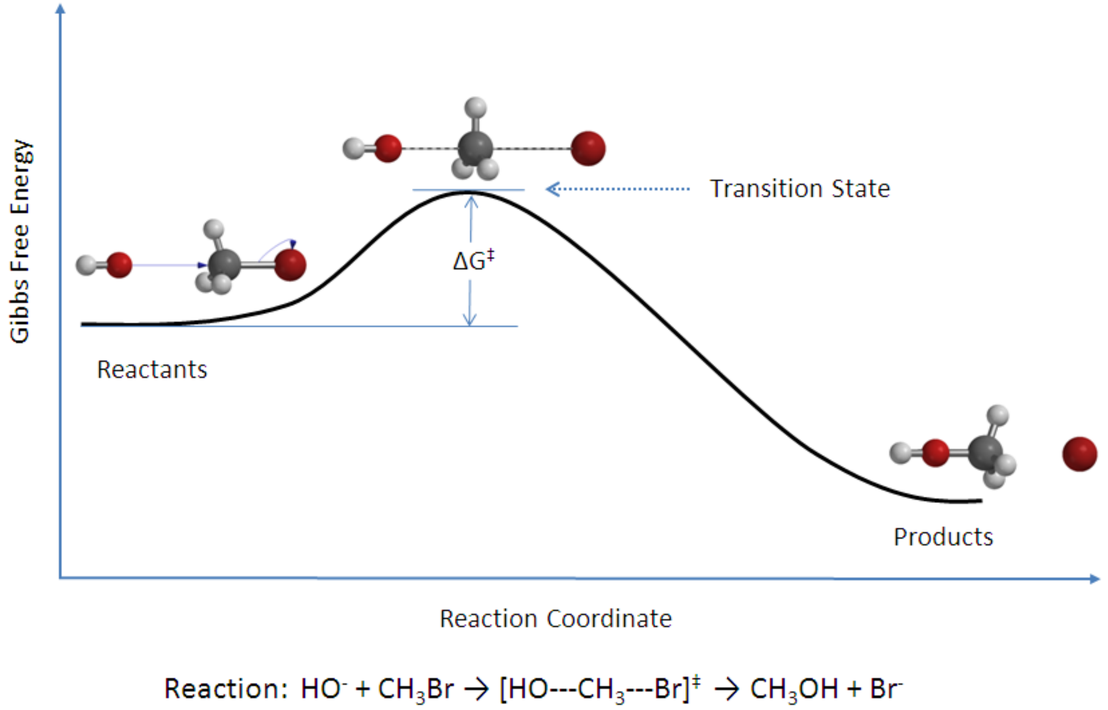
Assessment

Jupyter notebook: Classifying SN2 reactions

Noteable™

Released after workshop 3

Due:TBC



Reaction: $HO^- + CH_3Br \rightarrow [HO{-}{-}{-}CH_3{-}{-}{-}Br]^‡ \rightarrow CH_3OH + Br^-$

Further resources

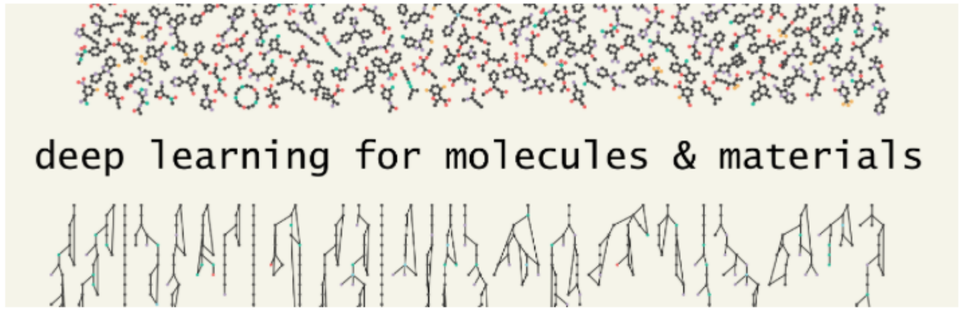
Best practices in machine learning for chemistry

Statistical tools based on machine learning are becoming integrated into chemistry research workflows. We discuss the elements necessary to train reliable, repeatable and reproducible models, and recommend a set of guidelines for machine learning reports.

Nongnuch Artrith, Keith T. Butler, François-Xavier Coudert, Seungwu Han, Olexandr Isayev Anubhav Jain and Aron Walsh

Artificial Intelligence in Drug Discovery

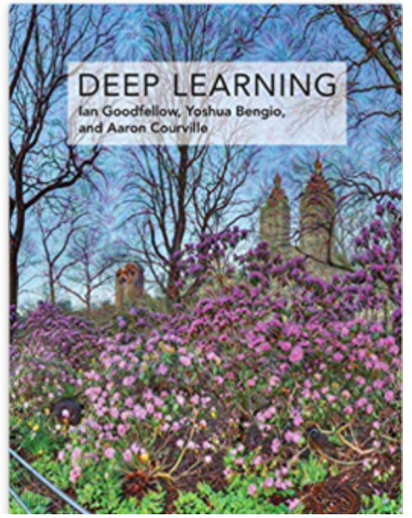
Editor: Nathan Brown

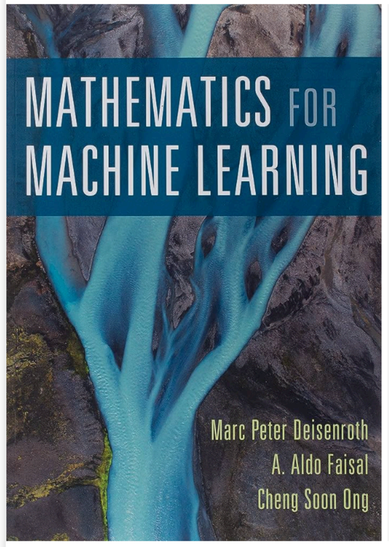


https://dmol.pub/index.html

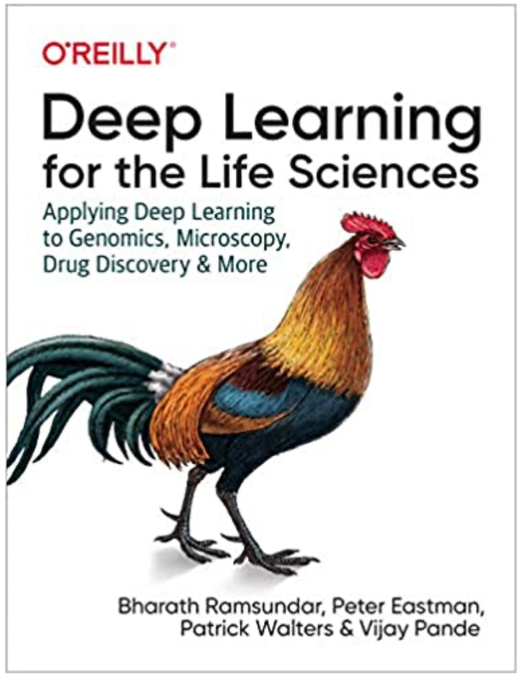
Machine Learning in Chemistry: The Impact of Artificial Intelligence

Editor: Hugh M Cartwright





https://mml-book.github.io/book/mml-book.pdf



What is machine learning?

Artificial intelligence

Design an intelligent agent that perceives its environment and makes decisions to maximise chances of achieving its goal.

Machine learning

Gives computers the ability to learn without specifically being programmed (Arthur Samuel 1959)

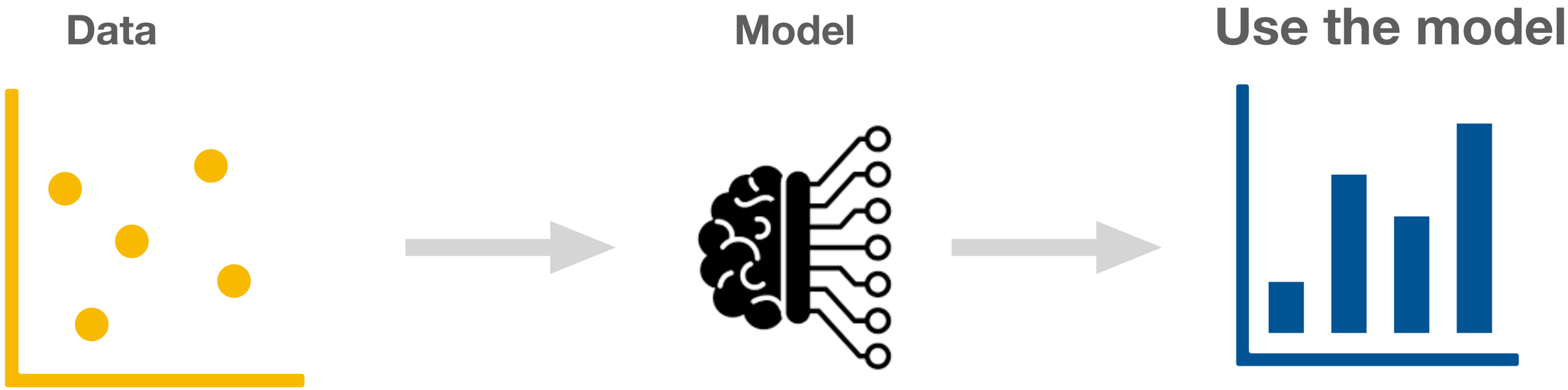
Supervised learning

Unsupervised learning

Reinforcement learning

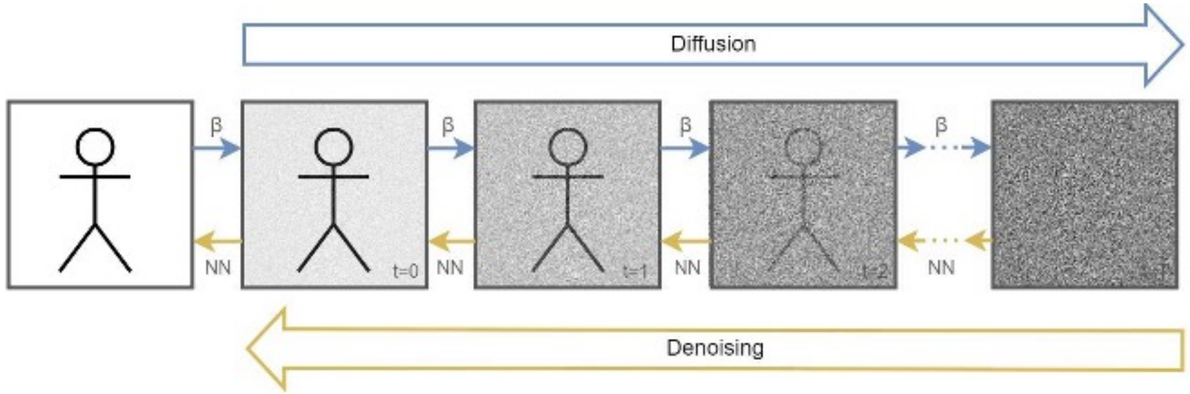
https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12

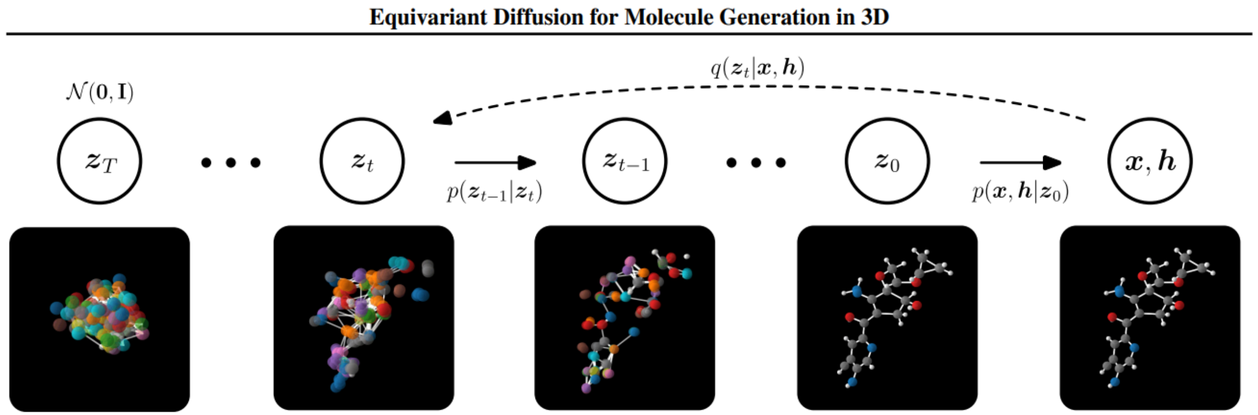
You generally learn a model from data you want to use



Diffusion model are generative models for cats and molecules

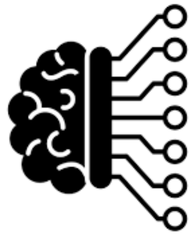
Train



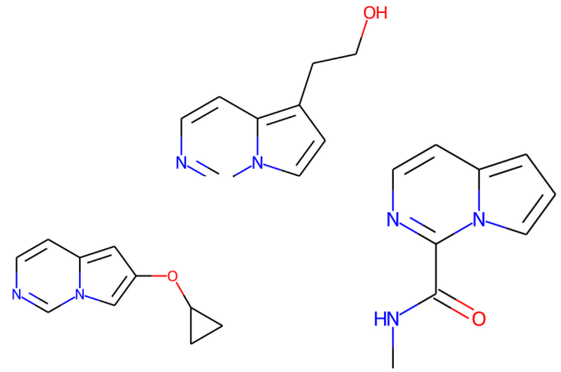


Hoogeboom et al. **arXiv: 2203.17003**

Model



Sample



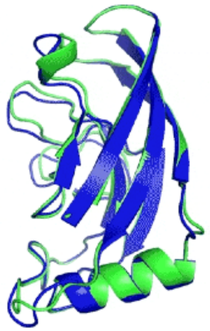
Choi et al. **arXiv:2108.02938**

Examples of Machine Learning (in Chemistry)

Dall-E 2

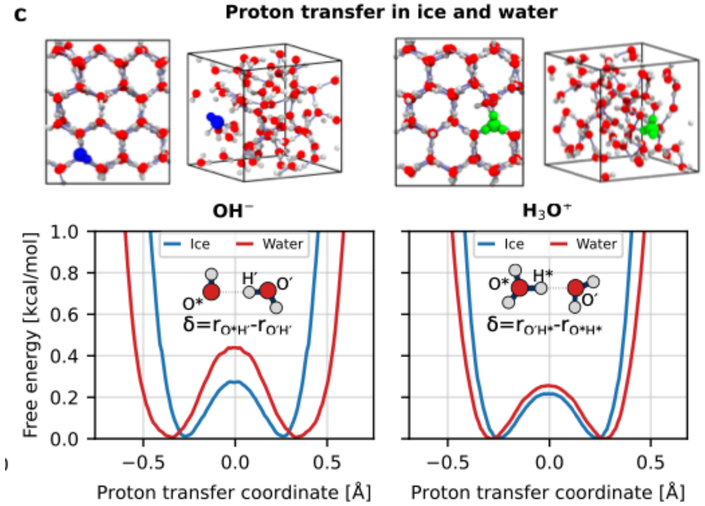
AlphaFold 2

T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)



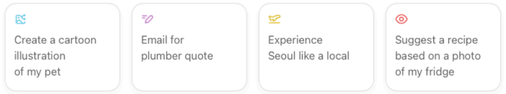
T1049 / 6y4f
93.3 GDT
(adhesin tip)

MACE

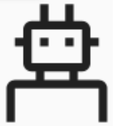


2024

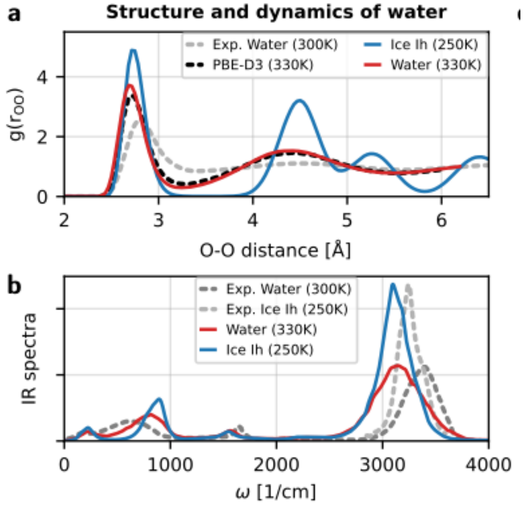
MSc Student in Chemistry learning about ML
Stable Diffusion



ELM uses Generative AI which carries with it potential ethical concerns and risks. These might include e.g. the creation of deepfakes, the possibility of privacy violation, the spread of disinformation, the generation of fallacious or misleading content and discriminatory or biased outputs.

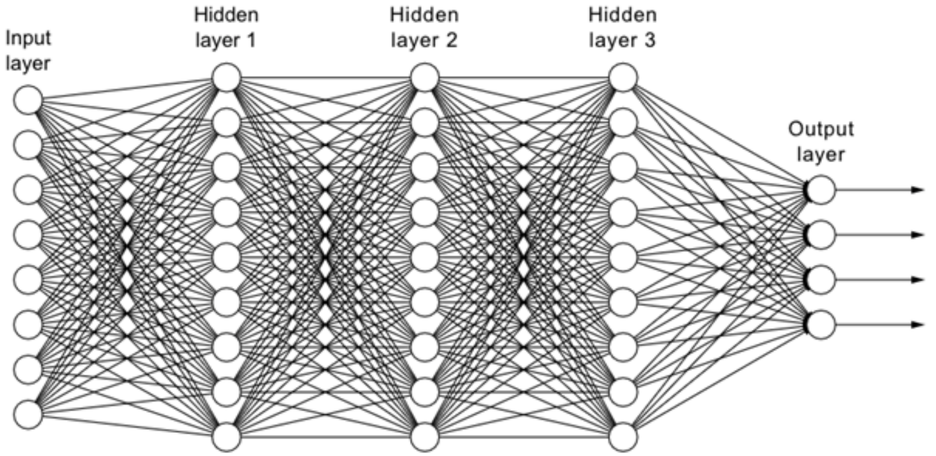


elm.edina.ac.uk



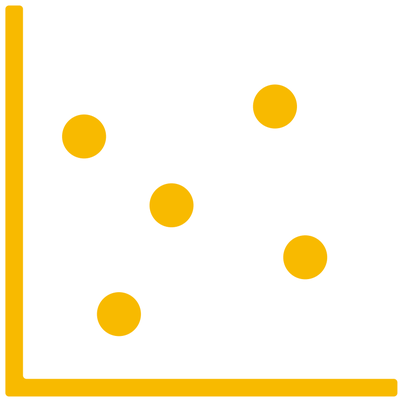
https://doi.org/10.48550/arXiv.2401.00096

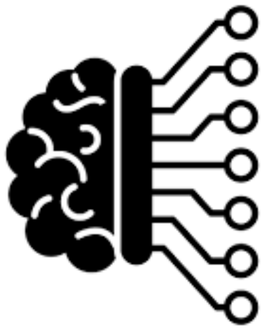
What you might think of as machine learning

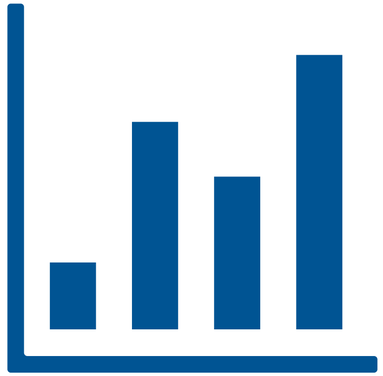


Model

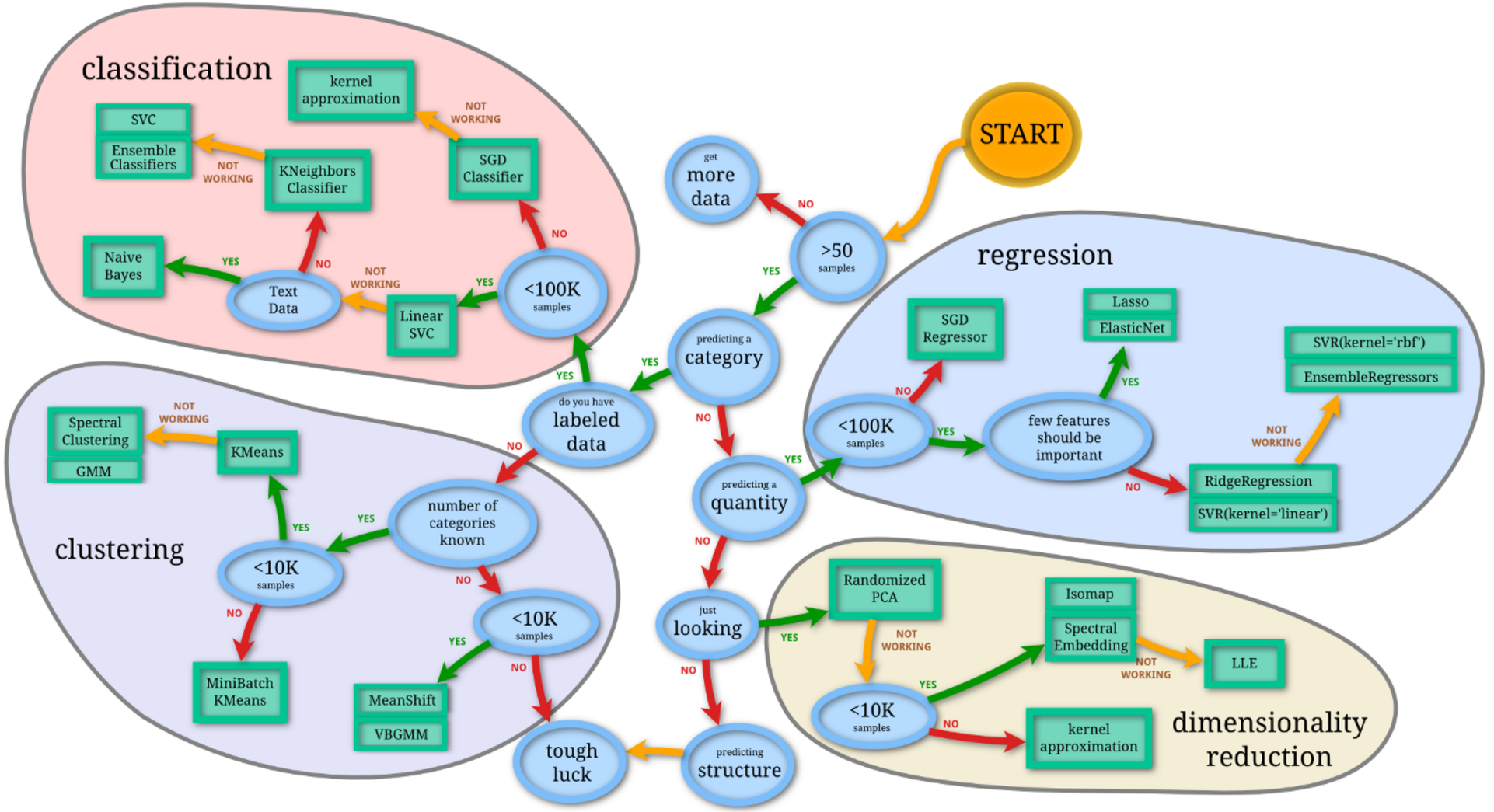
Use the model





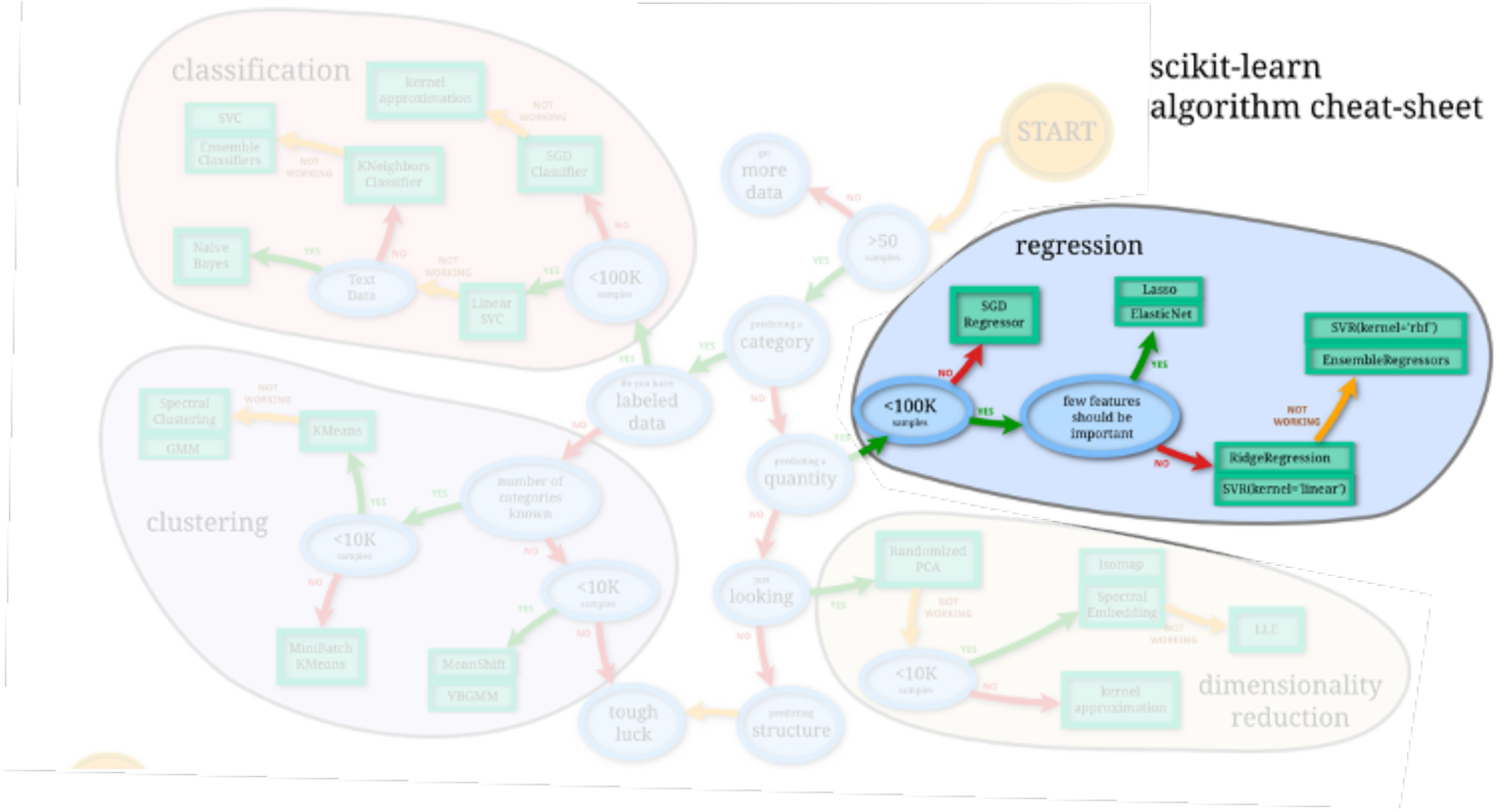


The Data Mining World

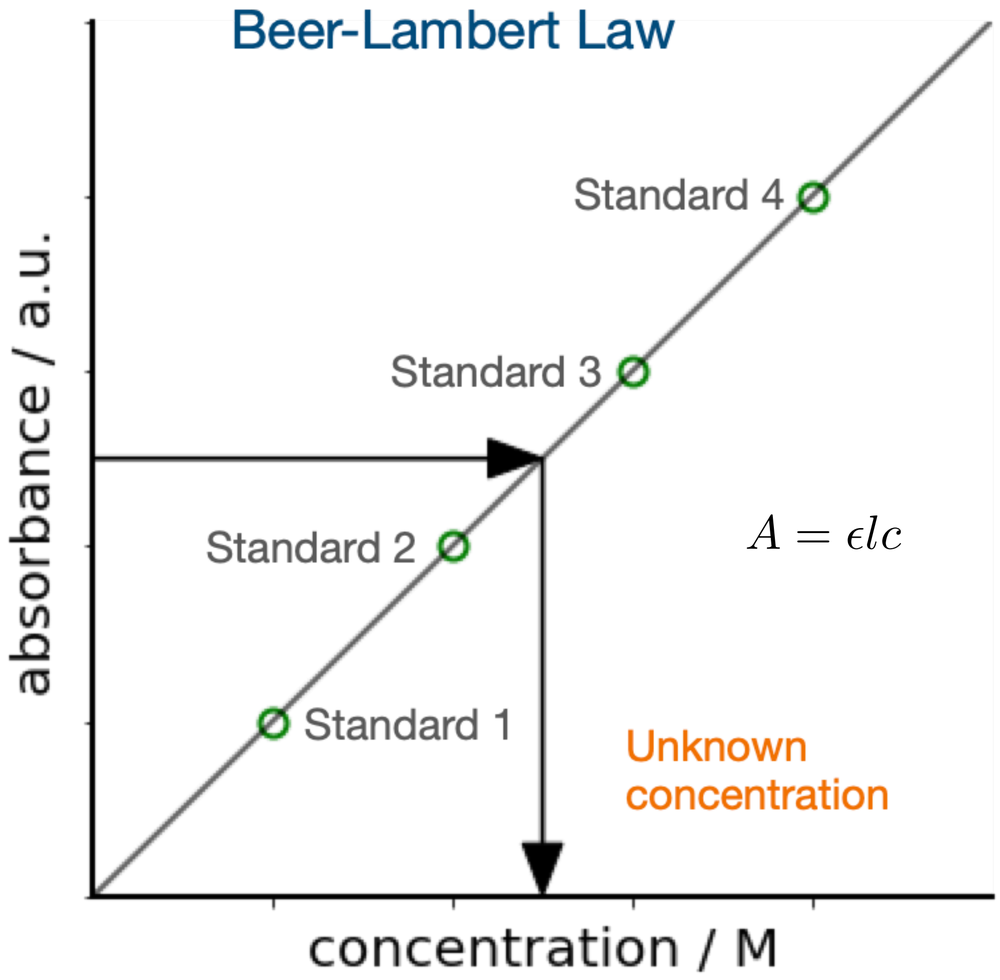


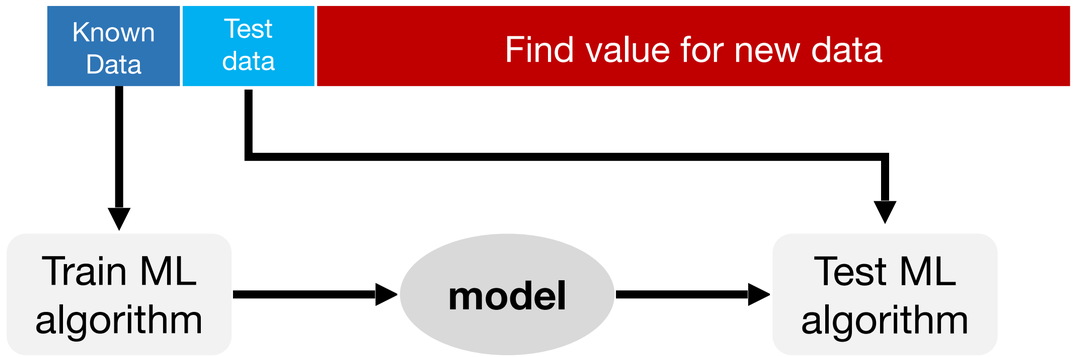
From **scikit-learn.org**

The Data Mining World



Linear regression: an example





Now the model is the line of best fit that will allow us to identify an unknown concentration

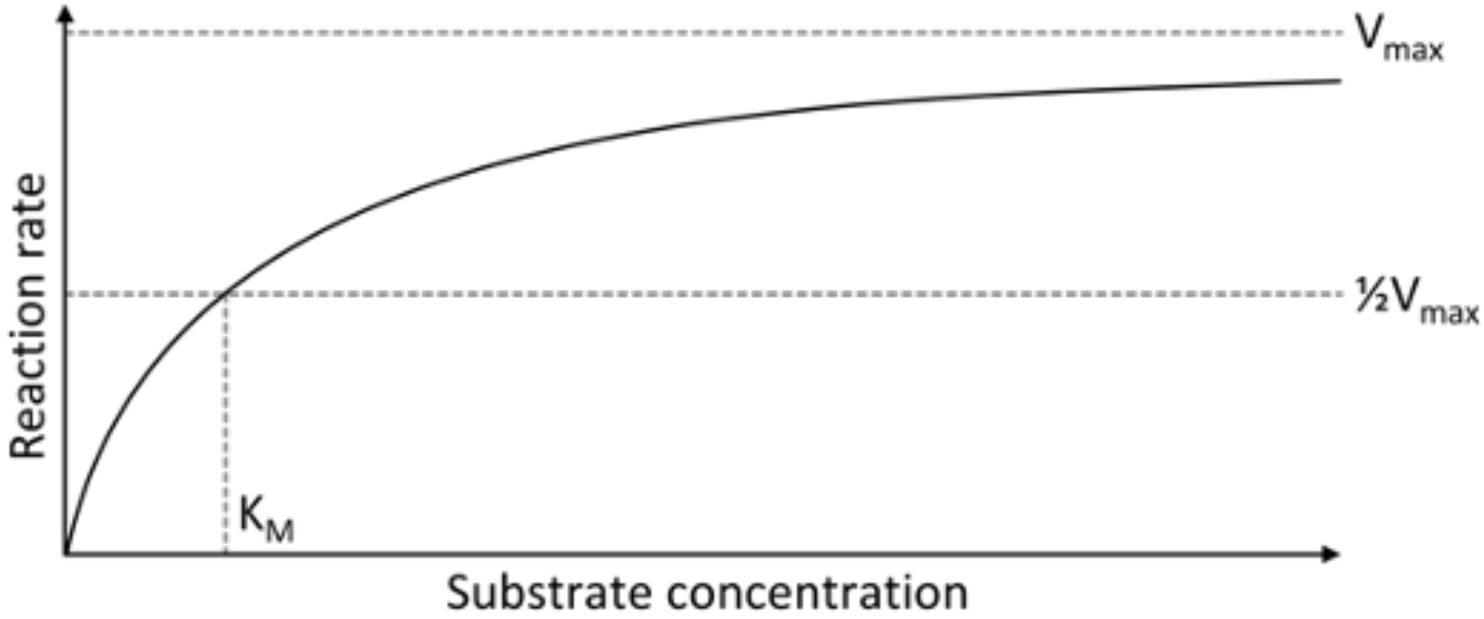
How do we find the line of best fit?

Non-linear Least squares

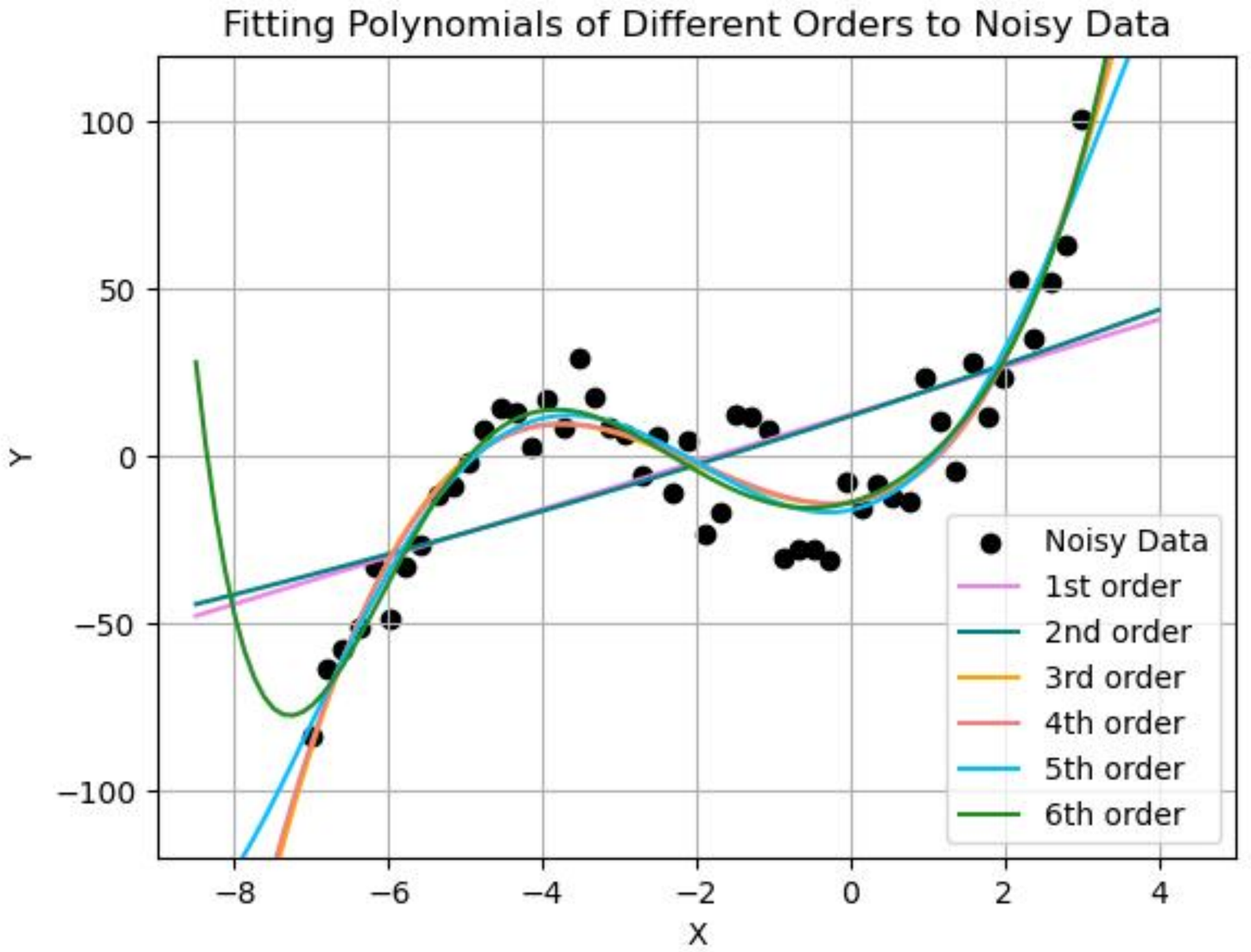
Non-linear combinations of model parameters, e.g. Michaelis Menten

$$\frac{d[P]}{dt} = \frac{V_{\max}[S]}{K_{\mathrm{M}} + [S]}$$

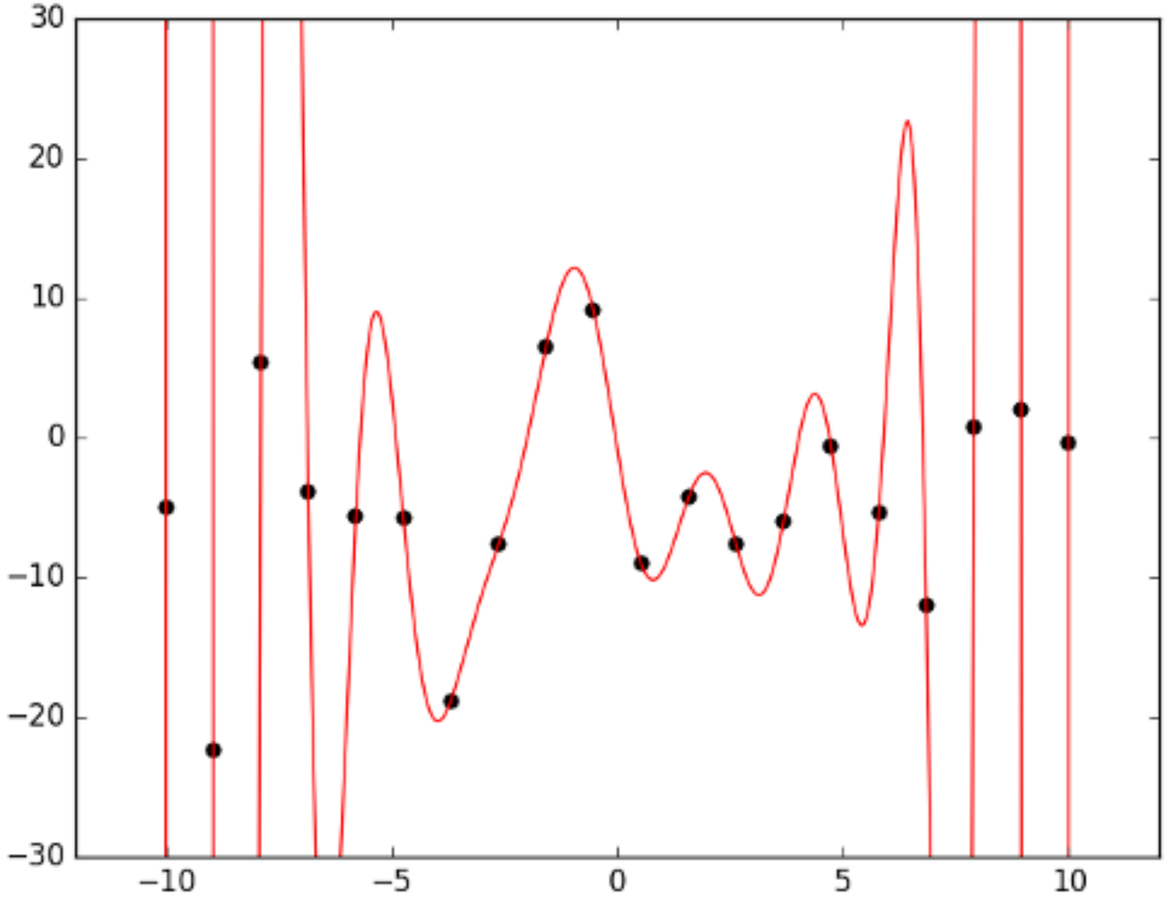
$$f(x, a, b) = \frac{ax}{b + x}$$



The more parameters the better the fit?



The more parameters the better the fit?



"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk"

John von Neumann

N points can be perfectly fitted with an N-1 order polynomial

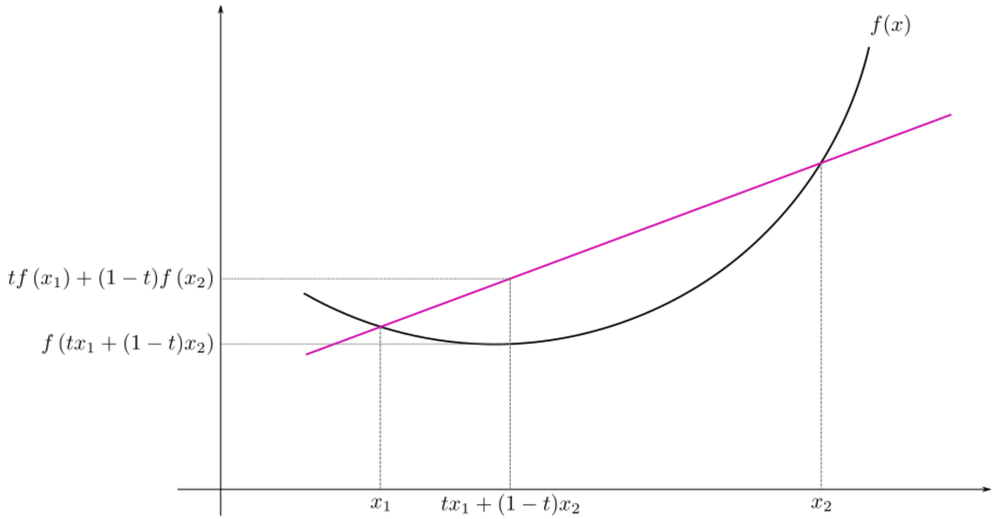
Optimisation problem mathematically

1) Define the parameters **x** of your problem

2) Let y=f(**x**) a function associating a score to every parameter combination in a subset of Euclidean space $\mathbf{R}^n$

3) find **x*** such that f(**x***) < f(**x**) for all **x** (a.k.a. *minimization*)

Convex function: iff the segments connecting any two points on the function, lie above it



Definitions:

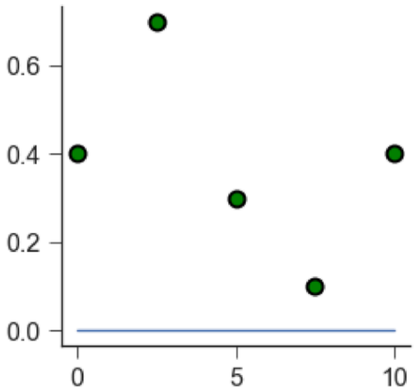
- *parameters **x***: (sometimes) degrees of freedom
- *subset of Euclidean space*: search space
- *f(x)*: fitness / scoring / objective / cost function

the function has *only one* minimum

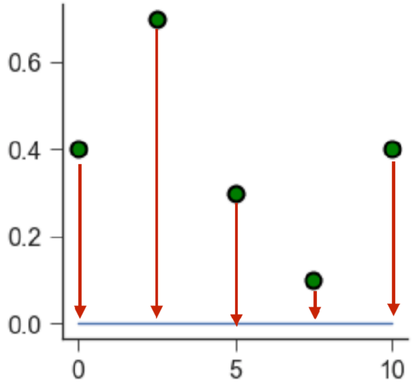
demonstrating that a problem is convex is a *big* deal!

What is a good objective function?

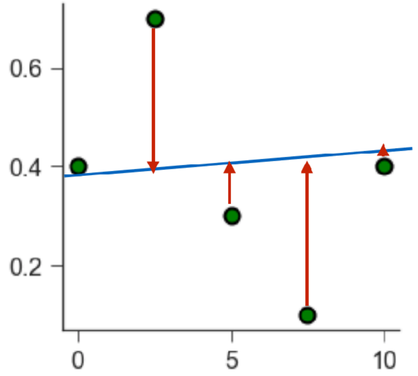
Linear regression: in ML language



The blue line is meant to **fit** the green data points.



Red arrows are a called **residuals** and small arrows mean a good fit.



Mean square error is a good **loss function,** because it is **convex.**

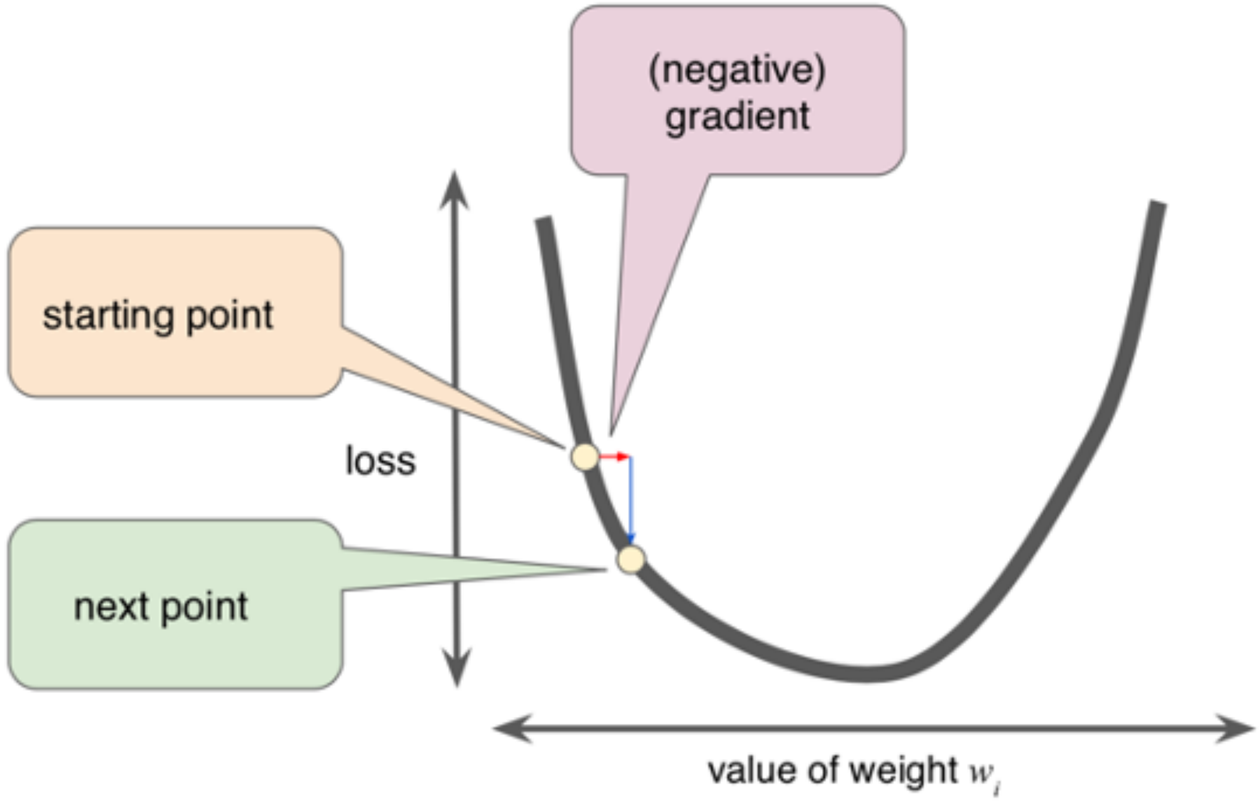
$$\mathrm{MSE} = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$$

n is the number of data points,
Y_i is the observed value (i.e. the measured data point),
$\hat{Y}_i$ is the predicted value (i.e. the value that lies on the line of best fit).

Gradient Descent

$$\text{MSE} = \frac{1}{n}\sum(Y_i - \hat{Y}_i)^2$$

Objective: Find the minimum of the loss function (see optimisation problem)

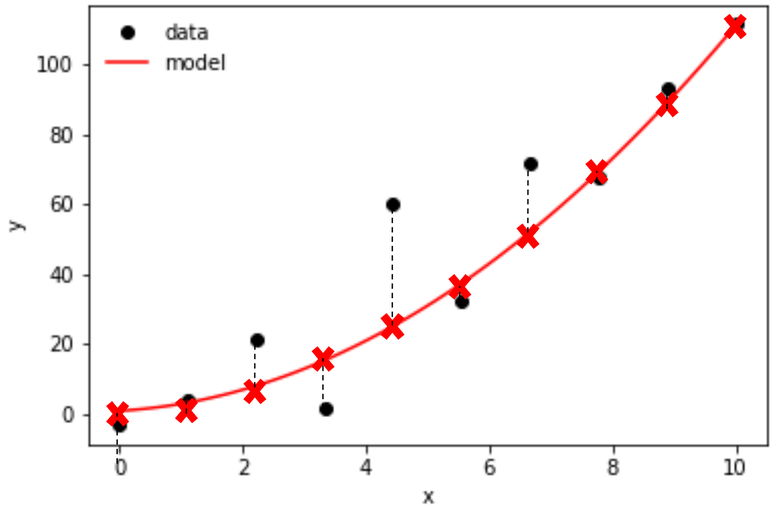


In machine learning, the step size of the gradient descent is called the **learning rate**

There is an *optimal learning* rate for every regression problem

Choosing the **best** optimiser is an optimisation problem in itself!

Linear Least Squares: A Summary



Model predicts values for each datapoint

$$\hat{y}_i = f(x_i; a, b, \ldots)$$

Residuals quantify prediction error

$$r_i = y_i - f(x_i; a, b, \ldots)$$

The best model minimizes the sum of squared residuals ("loss function")

$$E(a, b, \ldots) = \sum_{i=1}^{N} r_i^2$$

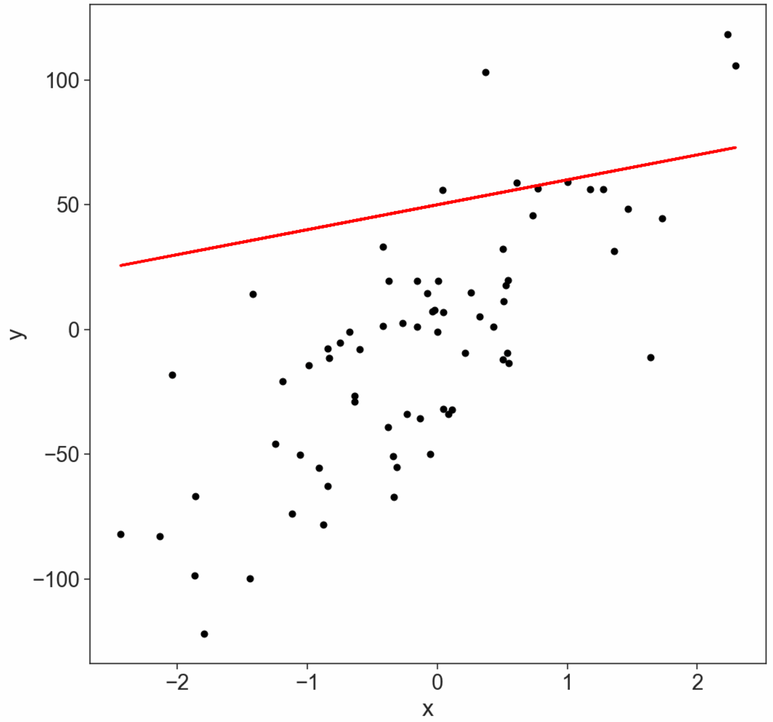
Solve: $\nabla E = 0$

-> find the minimum of the loss function
Note: This can be in many dimensions!

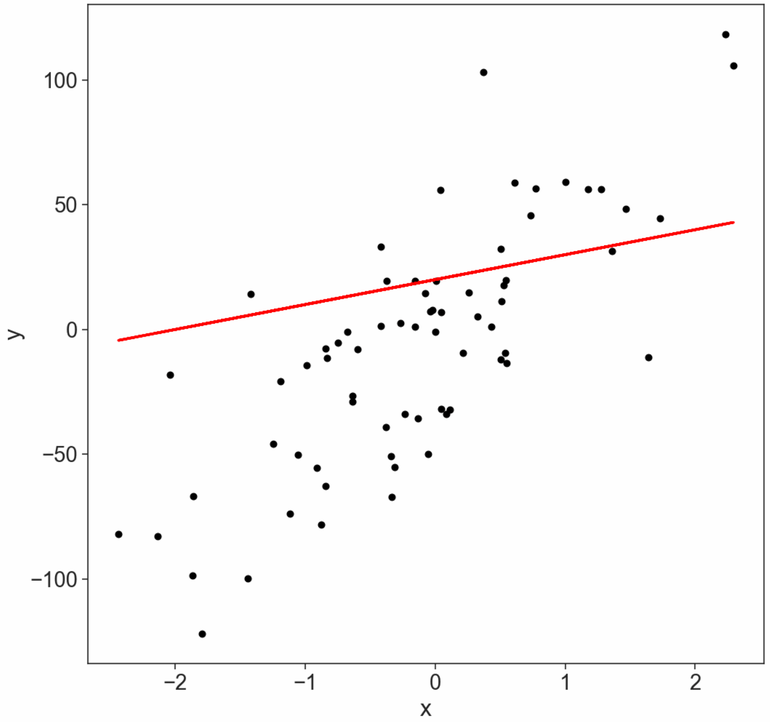
Use least squares (i.e. analytical solution) if:

- there are more data points than parameters (overdetermined)
- Model's parameters combine linearly (linear least squares)

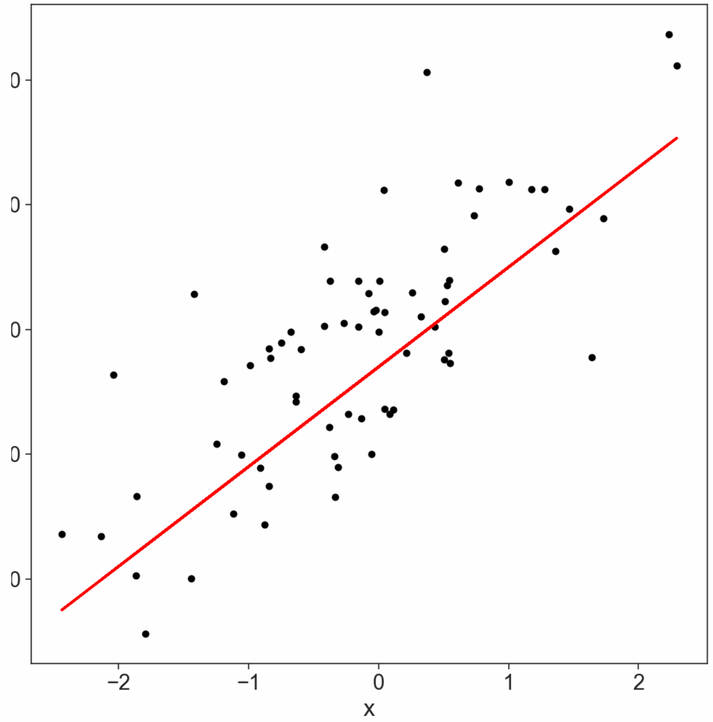
Gradient Descent: learning rate and initial conditions



Low learning rate
bad starting point
424 steps

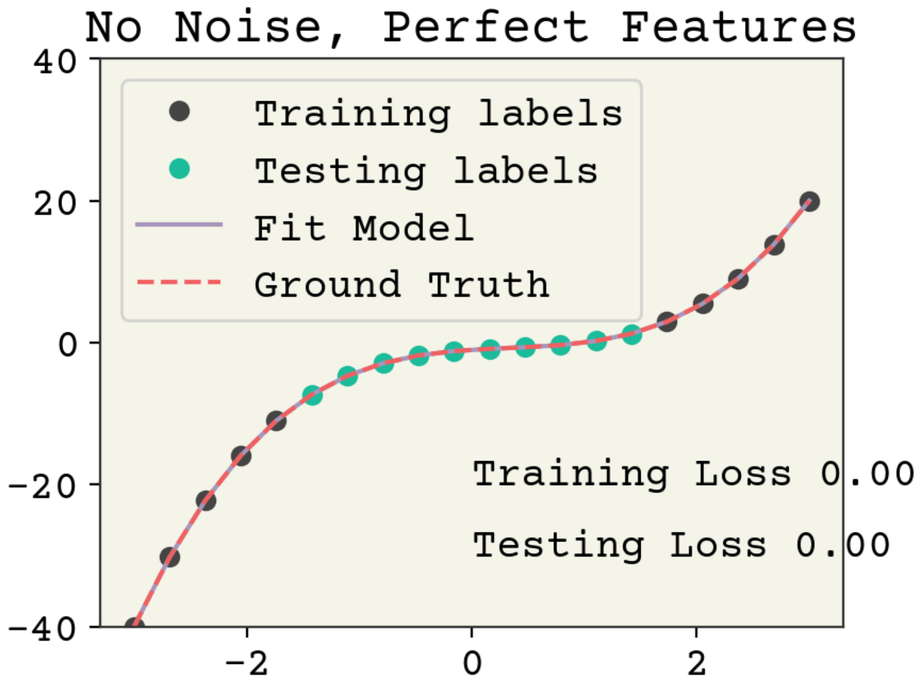


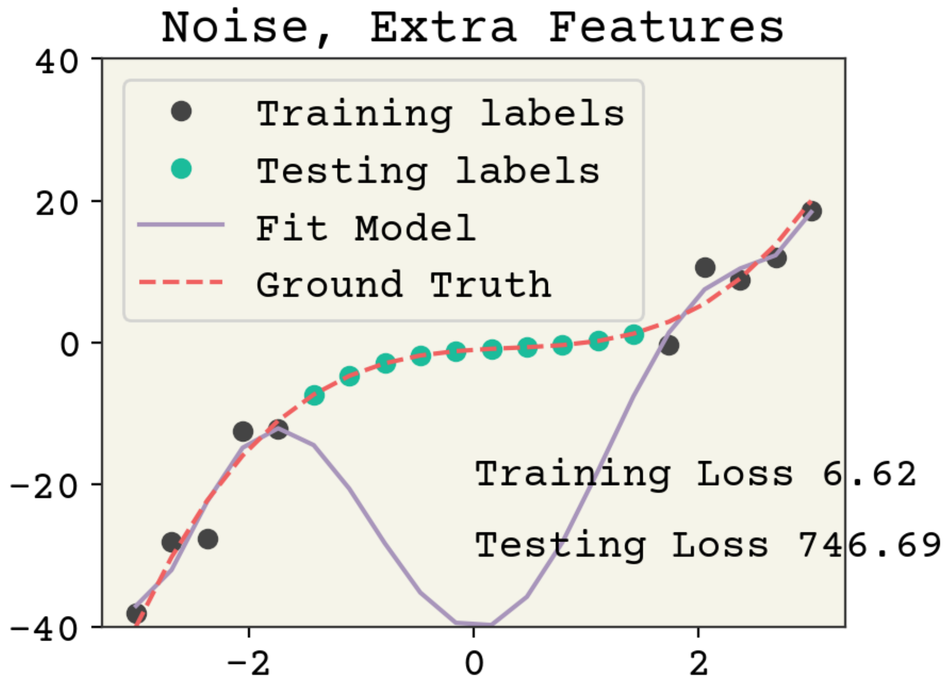
High learning rate
bad starting point
52 steps

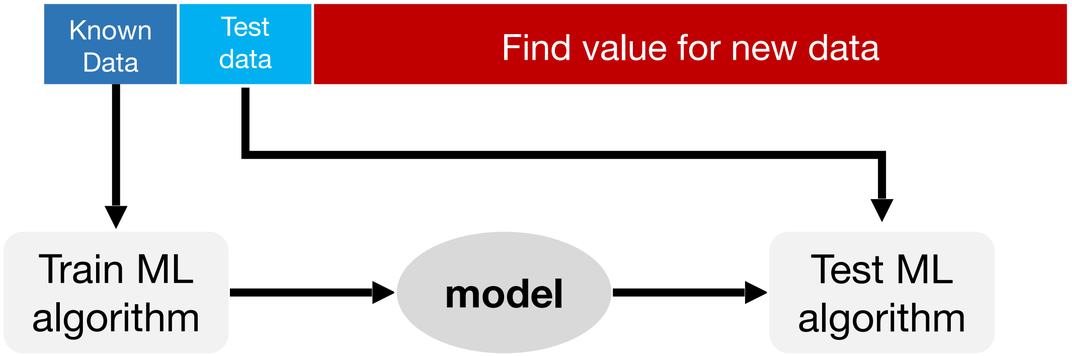


Low learning rate
Good starting point
379 steps

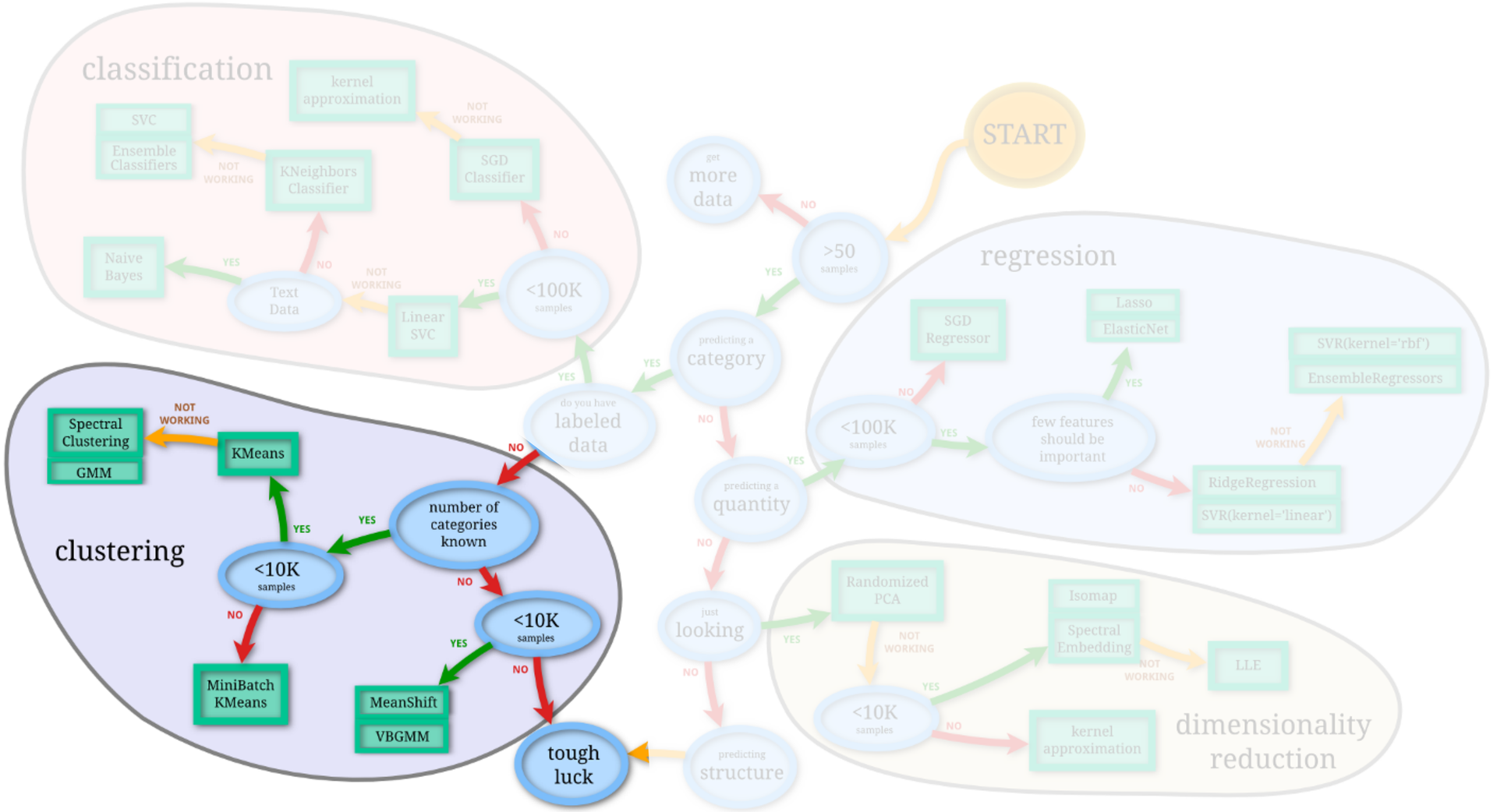
Models can only be as good as your data







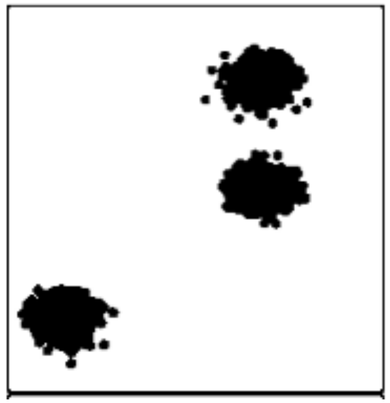
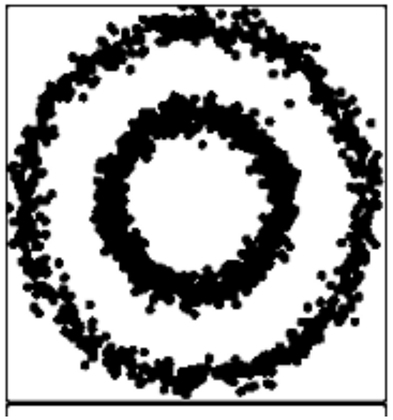
The Data Mining World

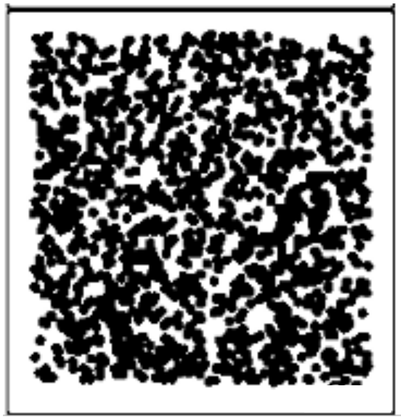


From **scikit-learn.org**

Clustering is an unsupervised learning technique

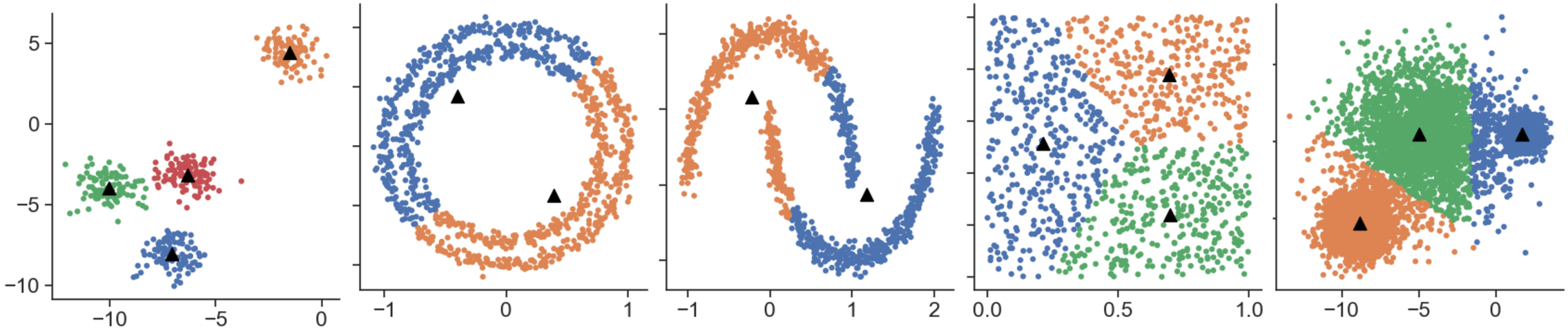
How many clusters are there?




Can the computer do better than the human?

This is how the k-means algorithm performs



How does k-means work?

Input: K, set of points $x_1 \ldots x_n$ this can be in N-dimensions

Place centroids, $c_1 \ldots, c_n$ at random locations

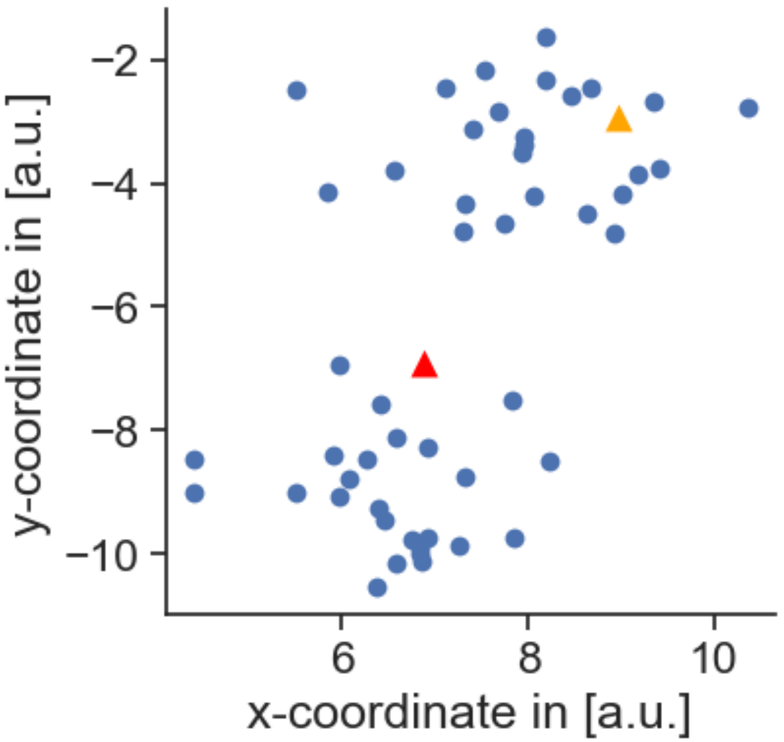
Repeat until convergence:

- For each point x_i:
  - Find nearest centroid c_j
  - Assign the point x_i to cluster j

$$\arg\min_j D(x_i, c_j)$$

- Fore each cluster j = 1…K:
  - Compute the centroid mean for all points in one cluster and update the centroid

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \to c_j} x_i(a)$$



How does k-means work?

Input: K, set of points $x_1 \ldots x_n$ this can be in N-dimensions

Place centroids, $c_1 \ldots, c_n$ at random locations

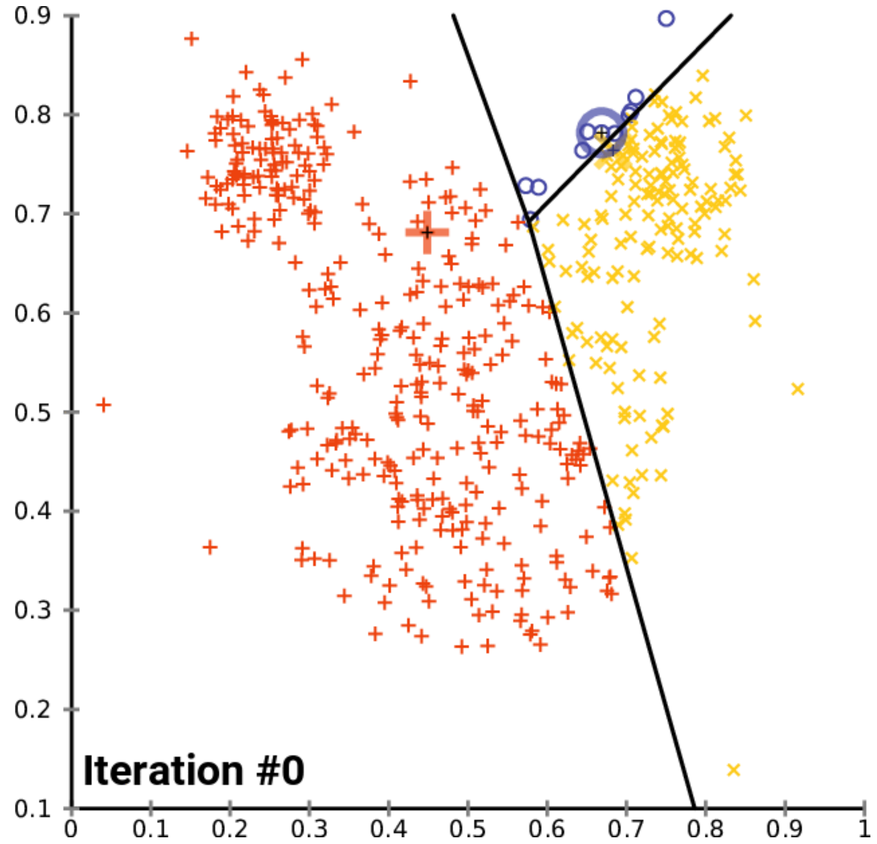
Repeat until convergence:

- For each point x_i:
  - Find nearest centroid c_j
  - Assign the point x_i to cluster j

$$\arg\min_j D(x_i, c_j)$$

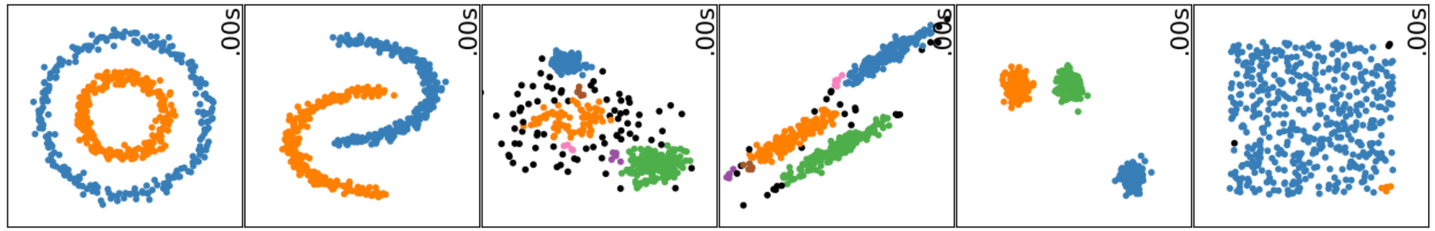
- Fore each cluster j = 1…K:
  - Compute the centroid mean for all points in one cluster and update the centroid

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \to c_j} x_i(a)$$

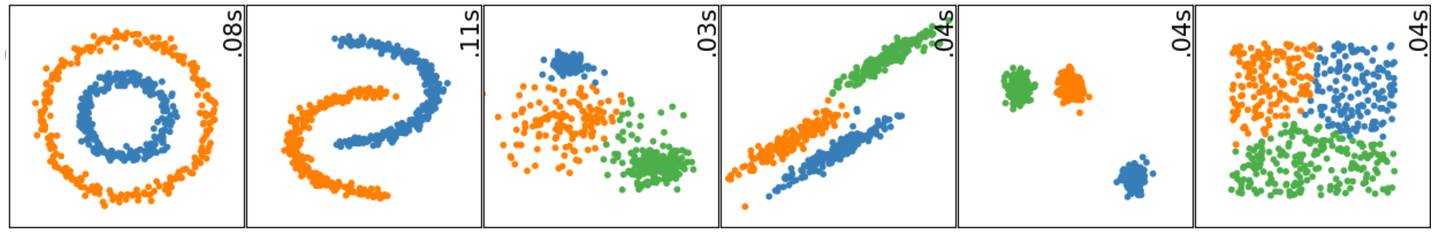


Not all clustering methods perform the same

DBSCAN

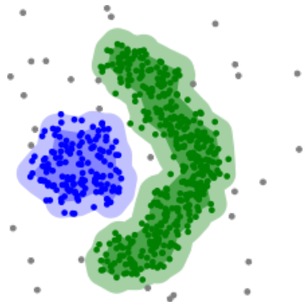


Spectral clustering



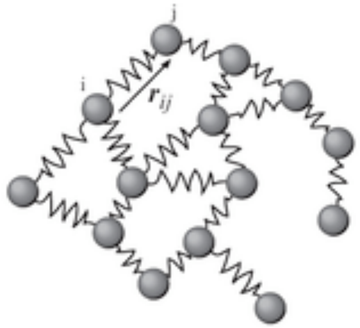
Density based clustering

DBSCAN



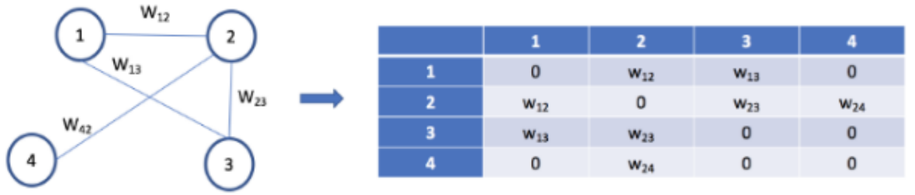
1. Find the points in the ε (eps) neighbourhood of every point, and identify the core points with more than *minPts* neighbours.
2. Find the connected components of core points on the neighbour graph, ignoring all non-core points.
3. Assign each non-core point to a nearby cluster if the cluster is an ε (eps) neighbour, otherwise assign it to noise.

Spectral clustering



$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

1. Calculate the Laplacian
2. Calculate the first k eigenvectors
3. Consider the matrix formed by the first k-eigenvectors
4. Cluster the graph nodes based on these features (e.g. k-means)



	1	2	3	4
1	0	w_{12}	w_{13}	0
2	w_{12}	0	w_{23}	w_{24}
3	w_{13}	w_{23}	0	0
4	0	w_{24}	0	0

L: Graph, R: n x n symmetric adjacency matrix

$$A_{i,j} = \exp(-\alpha ||x_i - x_j||^2)$$

α is a tuneable parameter now

D is the degree matrix for each node, which means how many connections it has.

There are many different clustering algorithms