Introduction to Machine Learning

Vincent Barra

LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Why Machine Learning

- Search engines
- Recommender systems
- Automatic translation
- Speech understanding
- Game playing
- Self-driving cars
- Personalized medicine
- ...
- Progress in all sciences: genetics, astronomy, chemistry, neurology, **physics**,...

What is Machine Learning ?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." Tom Mitchell (1997).

$\Rightarrow$ Learn to perform a task T , based on experience E (examples) minimizing error $\mathcal{E}$ (or maximizing performance P)

Often, we want to learn a function (model) f_θ with some model parameters $\theta \in \mathbb{R}^d$ that produces the right output y

$$\hat{\theta} = Arg\min_{\theta} \mathcal{E}(f_\theta(E))$$

E needs to be collected, cleaned, normalized, checked for data biases...

Inductive bias

Assumptions into the model = *inductive bias* b

- What should the model look like ?
  - Perform logical combination of inputs: decision trees, linear models,...
  - Memorize similar examples: KNN, SVM,...
  - Approach probability distributions: Bayesian approaches
  - "Reproduce" human brain : NN, DNN
- Hyperparameter settings (depth of tree, NN architecture, ..)
- Hypothesis on data distribution $(E \sim \mathcal{N}(0, \sigma^2)$...)
- Transfer knowledge from another domain

So...

$$\hat{\theta}, \hat{b} = Arg \min_{\theta, b} \mathcal{E}(f_{\theta,b}(E))$$

Machine Learning vs. Statistics

- Historically been developed in different fields
- Mathematical foundations are partially equivalent.
- Both aim to make predictions of natural phenomena

ML models...

- Focus more on precise predictions
- Automate a task
- Assume that the data generation process is unknown

Stats models

- Assume data is generated according to an understandable model
- focus more on the ability to interpret the patterns that generated the data and the ability to derive sound inference.

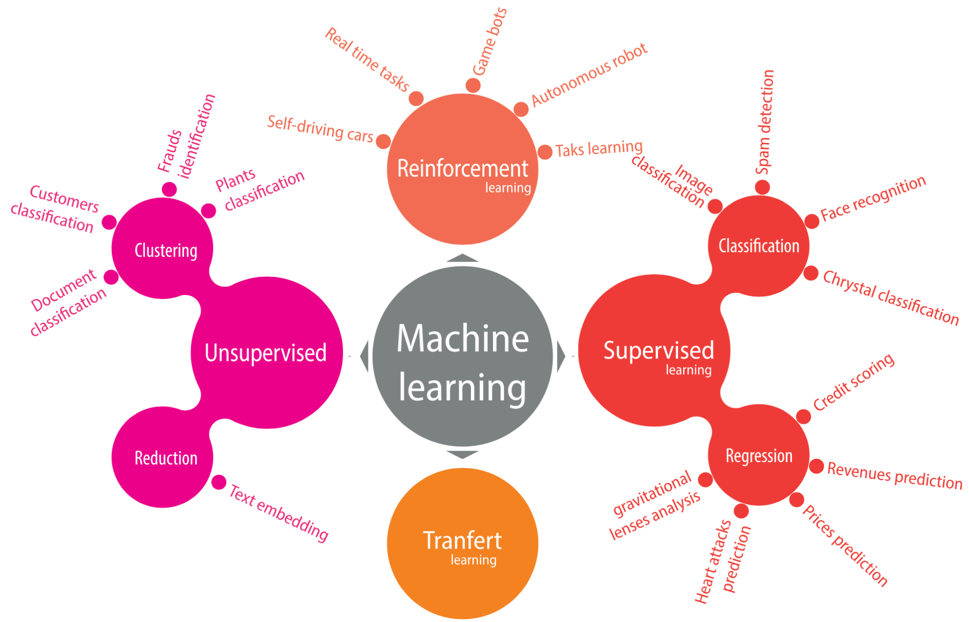
Taxonomy of Machine Learning algorithms

Several criteria, non exhaustive and combinable

- Supervised or not
- Incremental or batch learning
- Instance-based or model based

Tasks :

- Classification : group similar object in clusters
- Regression : predict a value / vector



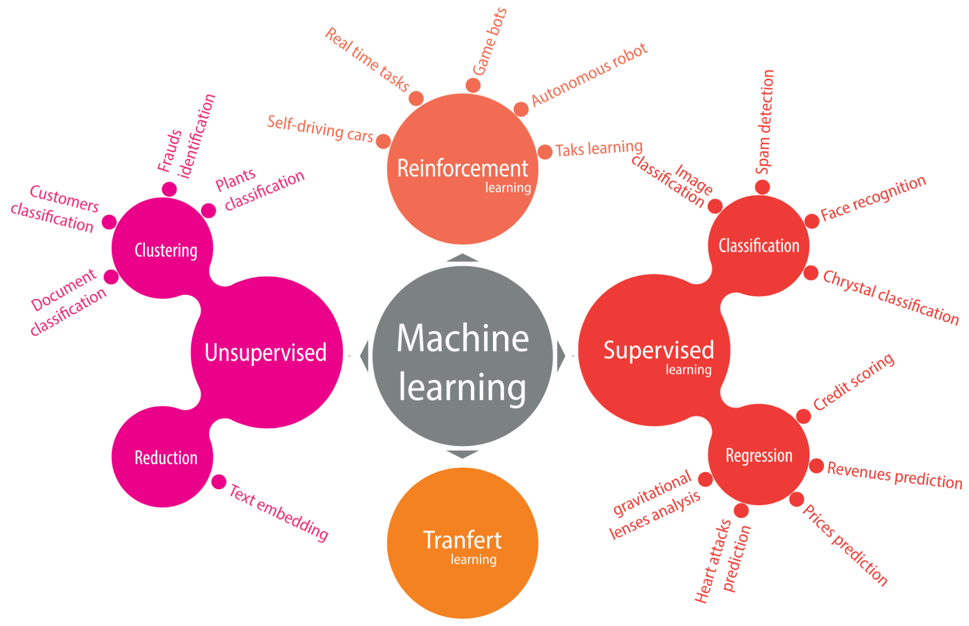
Taxonomy of Machine Learning algorithms

Several criteria, non exhaustive and combinable

- Supervised or not
- Incremental or batch learning
- Instance-based or model based

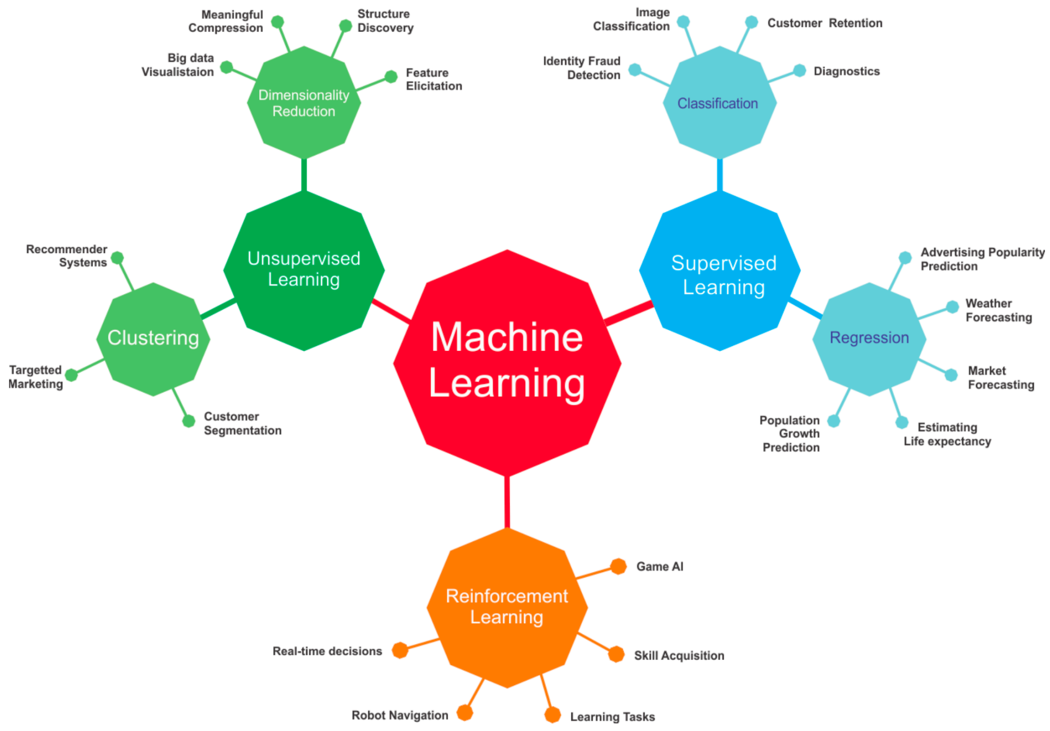
Tasks :

- Classification
- Regression



Taxonomy: supervision

- *Supervised algorithms*: learn f from labeled examples $E = (X, y)$
- *Unsupervised algorithms*: explore the structure of E to extract meaningful information
- *Semi-Supervised algorithms*: learn a model from few labeled and many unlabeled examples
- *Reinforcement Learning*: develop an agent that improves its performance based on interactions with the environment



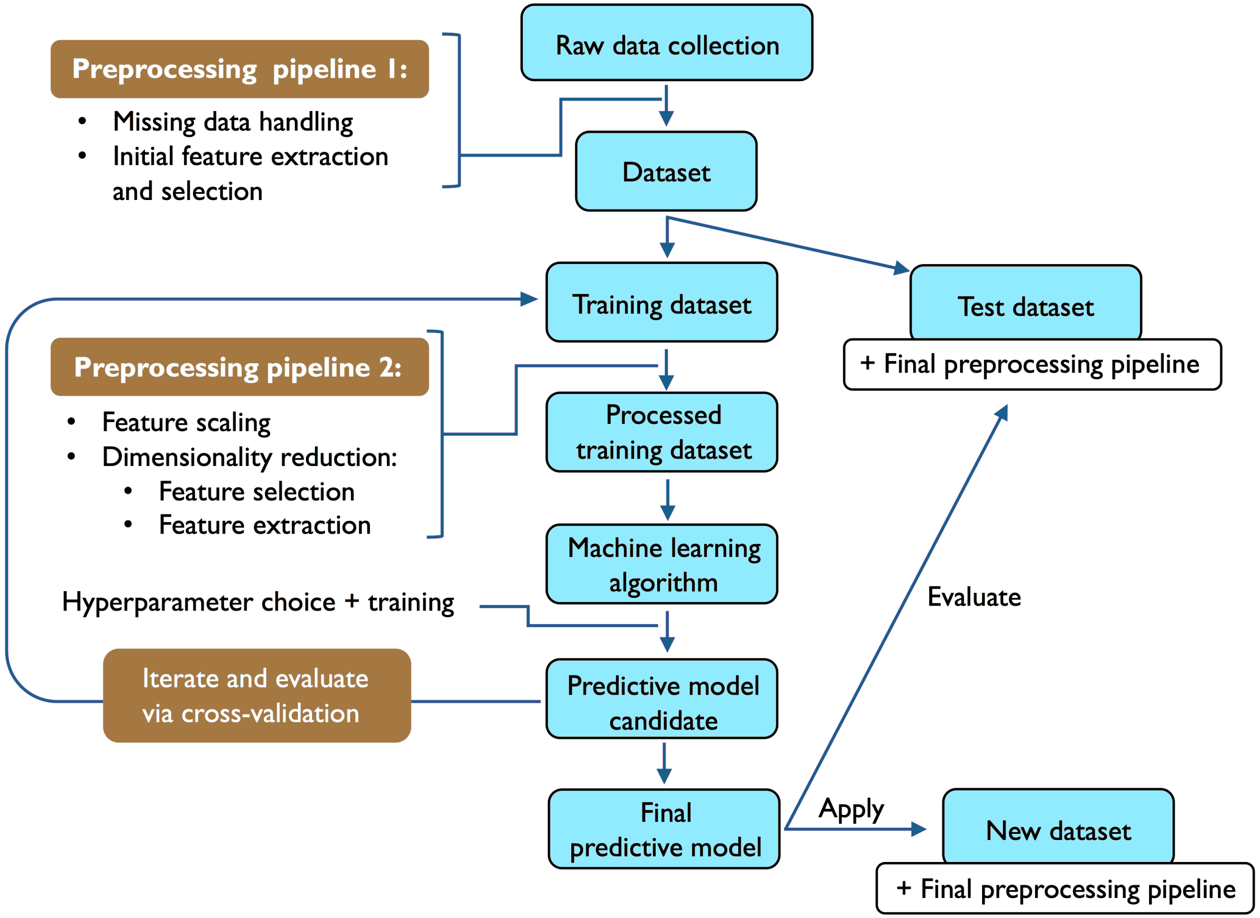
Supervised algorithms

- Known data: training/test set (fruits/labels)
- Model : f
- Input data: unknown data
- Output data: label

Supervised algorithms

Workflow

- Learn f from a set of examples
- Make predictions using f

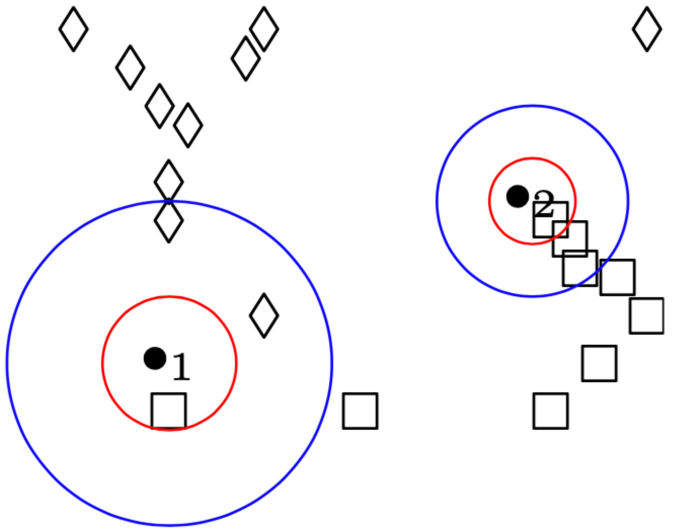


Supervised algorithms

K-nearest neighbors

An data point not in E is labeled based on the value/class of its K nearest neighbors in the feature space.

- classification : majority vote
- regression: mean value



1-NN (red) and 3-NN (blue) decision rule

Supervised algorithms

SVM / SVR

Maximization of a margin

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathbf{d}}} \|\mathbf{w}\|^{\mathbf{2}}$$

w.r.t.

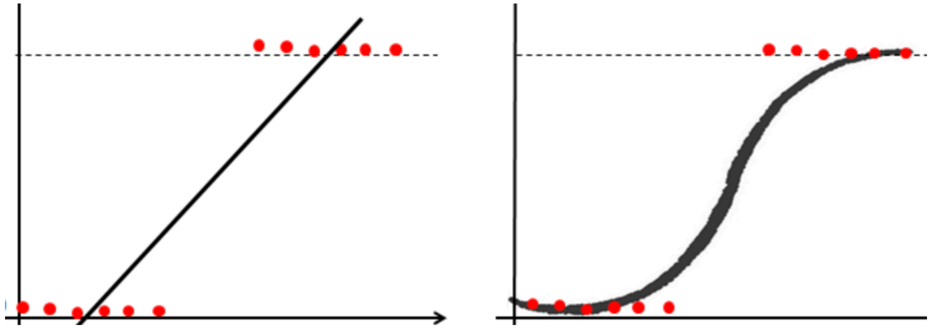
$$y_i(\mathbf{w}^T\mathbf{x_i}) \geq 1, i \in [\![1, n]\!]$$

Supervised algorithms

Linear / Logistic regression

Model: Find θ such that $\|\mathbf{A}\theta - \mathbf{Y}\|^{\mathbf{2}}$ is minimum $\Rightarrow \hat{\theta} = (\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top\mathbf{Y}$

- Classification: logit transform
- Regression: $\mathcal{E}$: mean square error

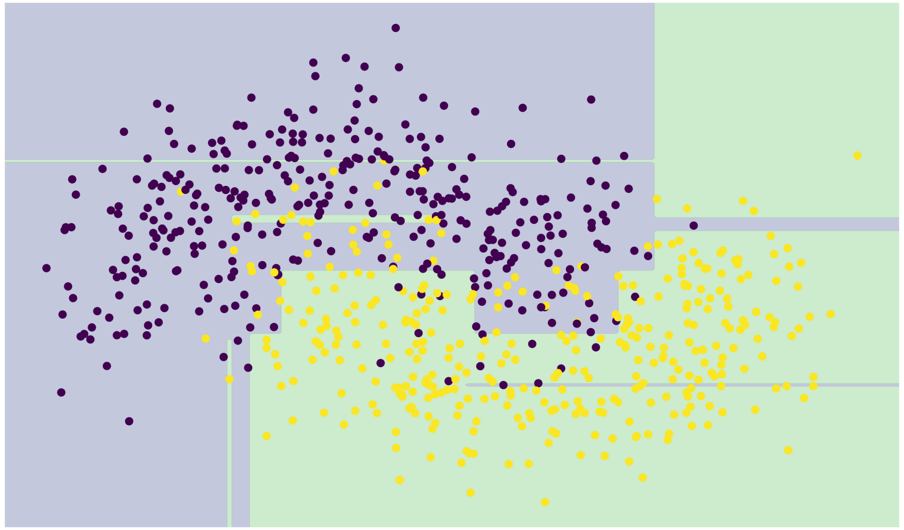


Supervised algorithms

Decision trees

- Process an object by means of a series of tests on the attributes that describe it.
- Tests are organized in such a way that the answer to one of them indicates the next test to which the object must be submitted

  $\Rightarrow$ Structuration of tests into a tree.
    - Classification: individuals in a node
    - Regression: mean value in a node



Supervised algorithms

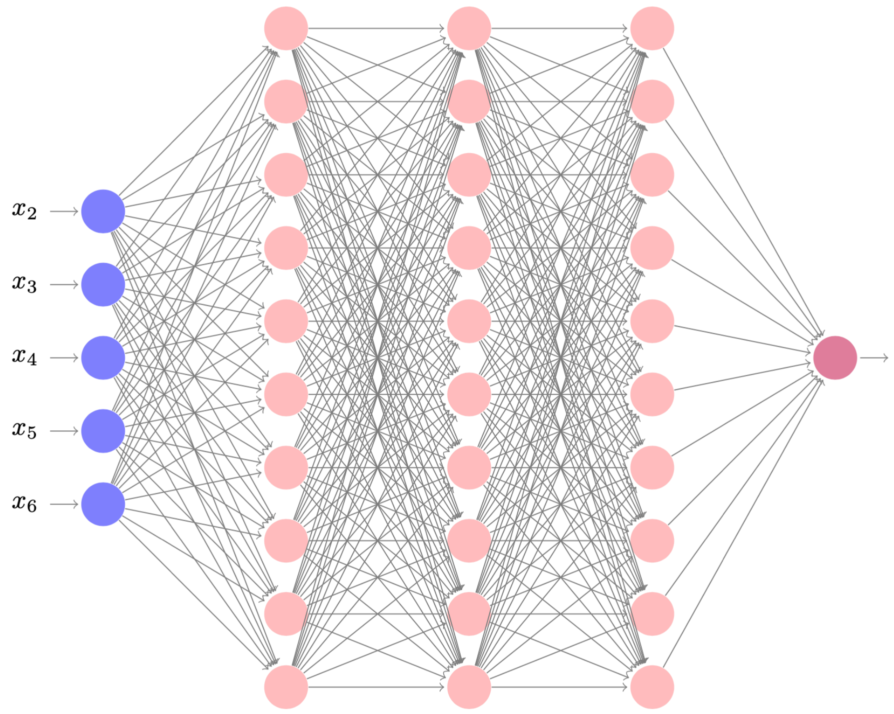
Random forests

- Bagging algorithm + decorrelation criterion between trees.
- Uses decision trees as partially independent classifiers.
- Trains each decision tree on a sampling of E obtained by bootstrapping according to a tree learning algorithm
  - Classification: individuals in a node
  - Regression: mean value in a node

Supervised algorithms

Neural networks

- Fully connected layers of neurons
- Memory stored in connections (weights+bias)
- Learning algorithm (backpropagation)
- Shallow / Deep



Unupervised algorithms

- Unlabeled data E, or data with unknown structure :
- Explore the structure of the data to extract information
  - *Clustering algorithm*: organize information into meaningful subgroups (clusters)
  - *Data reduction/visualization* : find an intrinsic representation of the data

Unsupervised algorithms

K-means and variants

Given C initial points (class centers) $g_1 \cdots g_c$:

1. Compute $d(x, g_j), \forall x \in E, j \in [\![1, C]\!]$
2. Assign each $x \in E$ to its closest class center
3. Recompute class centers
4. Iterate until convergence

$\Rightarrow$ Group data into clusters

Unsupervised algorithms

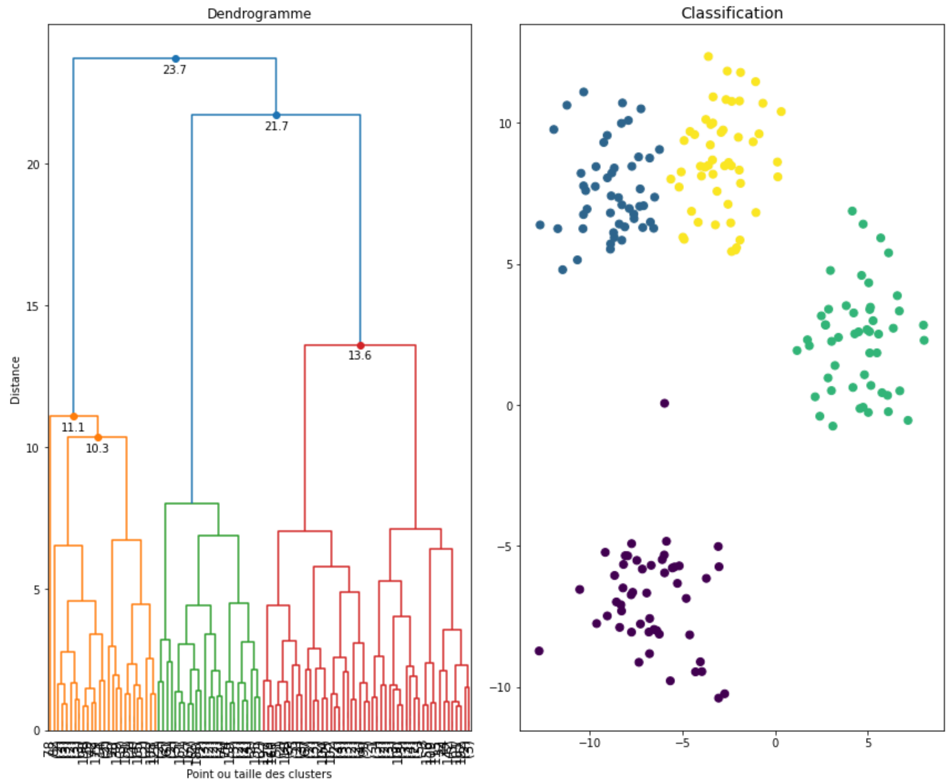
Hierarchical clustering

Init: each $x \in E$ is a subset,

While (number of subsets>1)

1. Compute distances between all pairs of subsets
2. Merge the two closest subsets

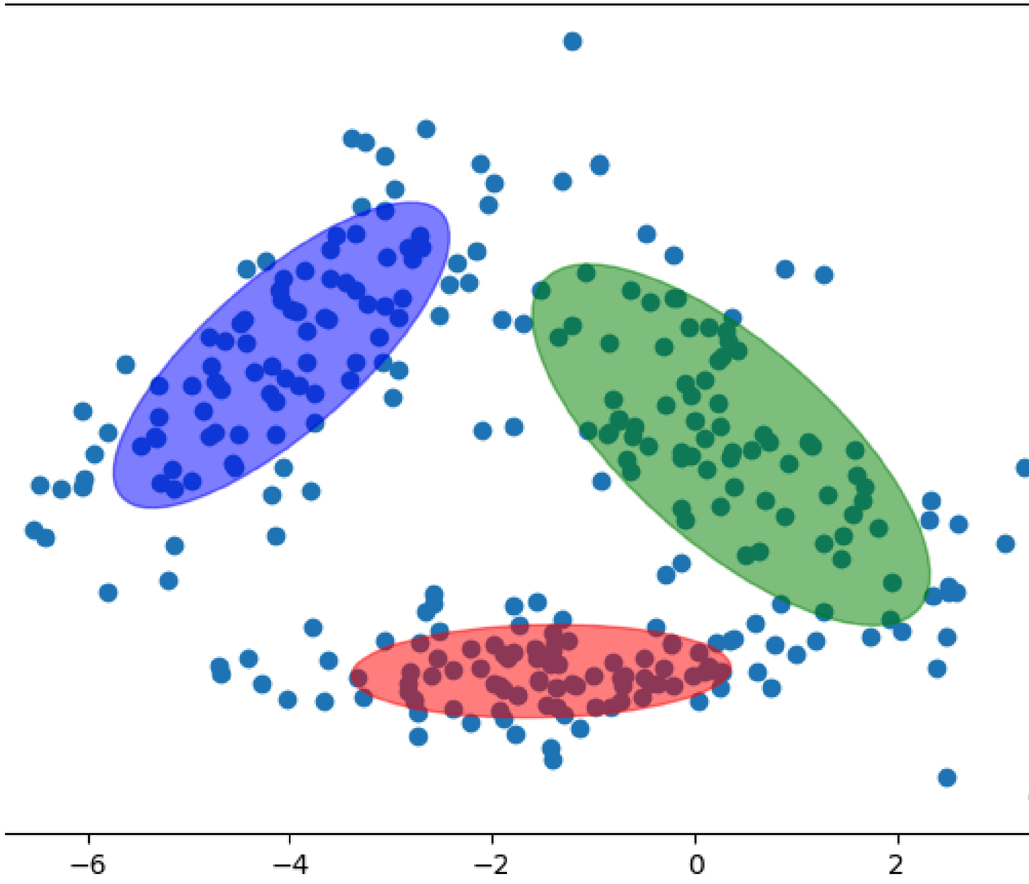
$\Rightarrow$ Definition of a distance between subsets



Unsupervised algorithms

Mixture Models

- Claim: Data comes from a mixture of distributions (usually Gaussian)
- Aim: estimate the parameters of the mixture model by maximizing the likelihood of the data



Unsupervised algorithms

Data can be very high-dimensional and difficult to understand, learn from, store,...

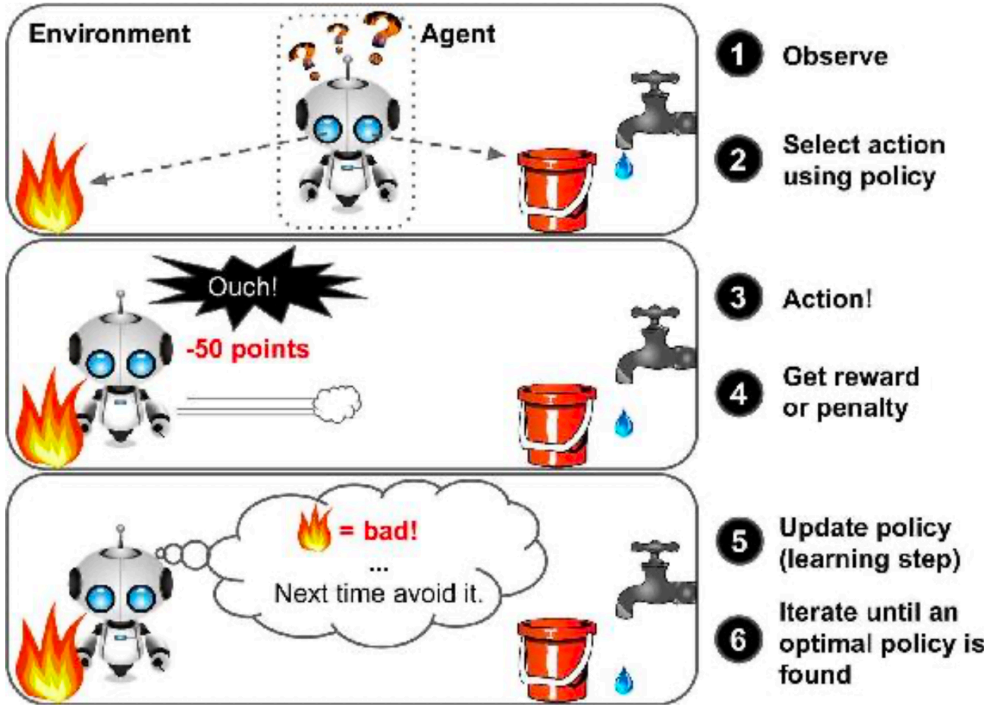
Dimension reduction

- Linear (ACP,..)
- Non linear (manifold learning)

$\Rightarrow$ Find intrinsic dimension of the data

Reinforcement Learning

- Develop an agent that improves its performance based on interactions with the environment
- Search a large space of actions and states
- Reward function defines how well a series of actions works
- Learn a policy that maximizes reward through exploration



Taxonomy of Machine Learning algorithms

Several criteria, non exhaustive and combinable

- Supervised or not
- Incremental or batch learning
- Instance-based or model based

Incremental / Batch learning

Availability of data E:

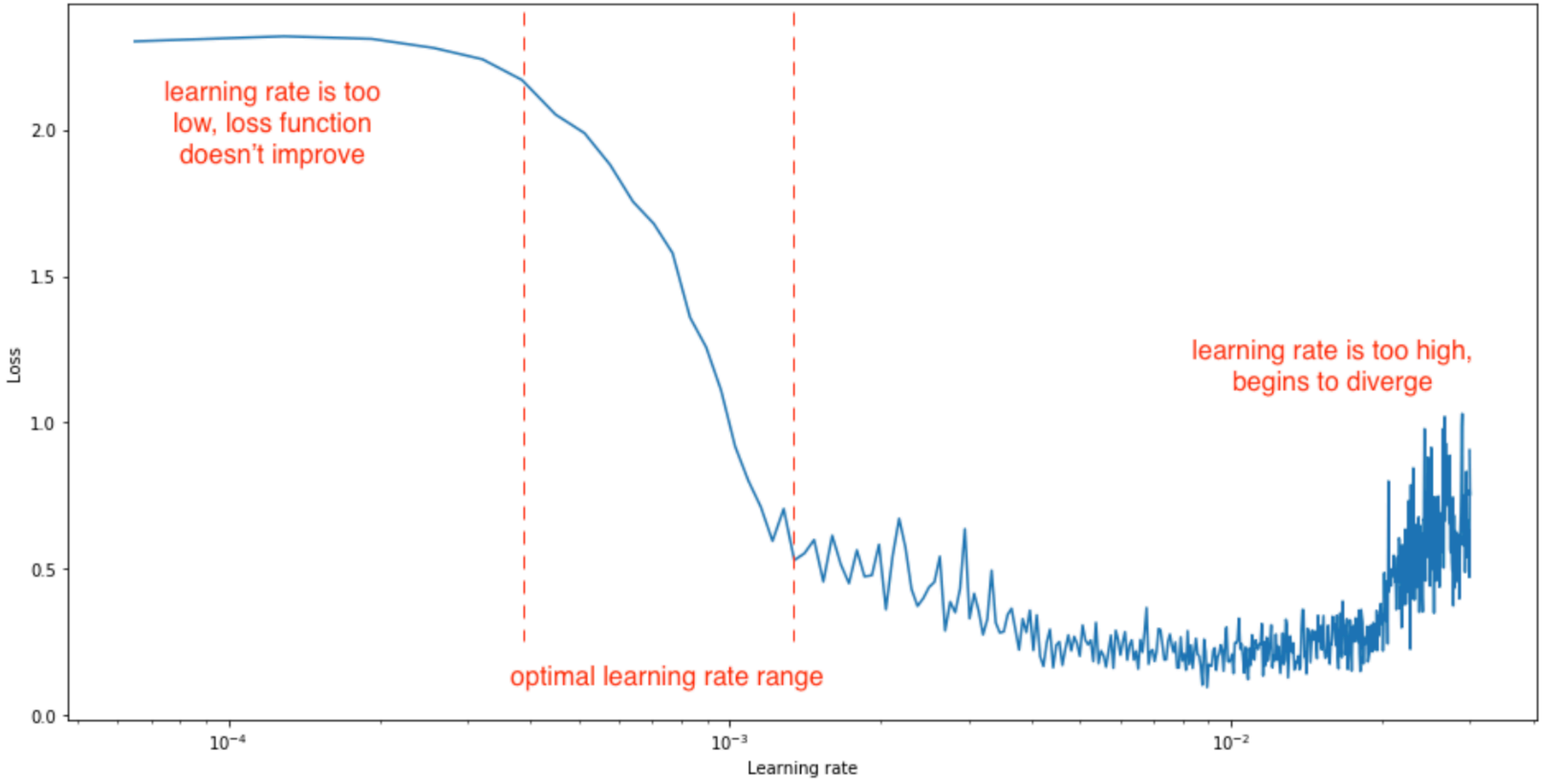
- Yes: batch Learning
  - time consuming
  - offline training
  - how to retrain if new data arrives ?
- No: online learning
  - fast
  - online
  - interests: data flow/limited ressources

Incremental / Batch learning

When do we have to learn again ?

- too often: instability, sensitivity to outliers
- too rarely: no adaptation

$\Rightarrow$ Learning rate



Taxonomy of Machine Learning algorithms

Several criteria, non exhaustive and combinable

- Supervised or not

  -Incremental or batch learning

- Instance-based or model based

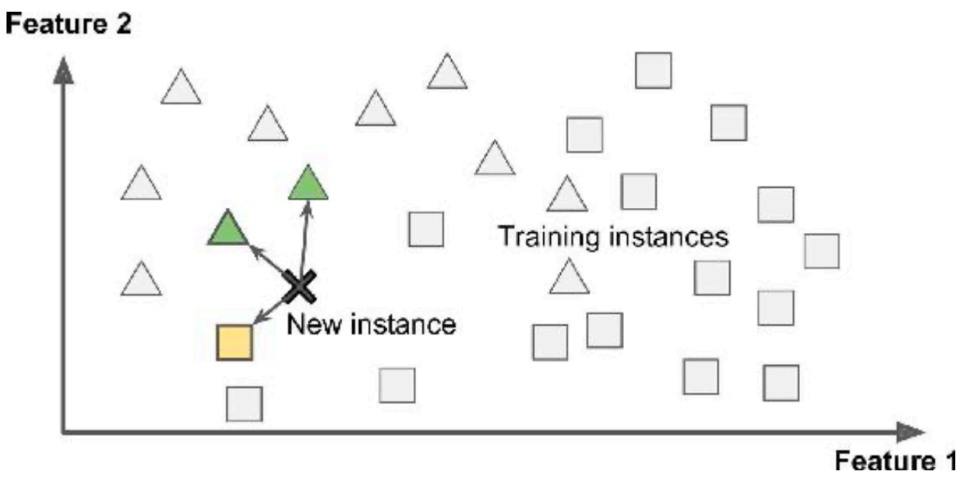
Instance/Model based algorithms

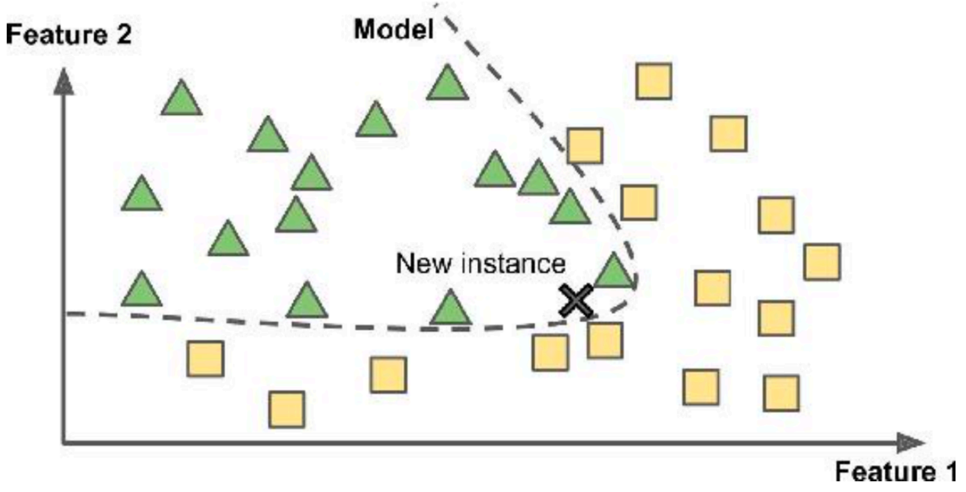
1. Instance-based algorithm: only relying on the training set

- Easy to learn by heart
- How to allow generalization ?

2. Model-based: use of a parametric model

- What kind of model ?
- How to tune parameters ?





Summary

Learning = representation + evaluation + optimization

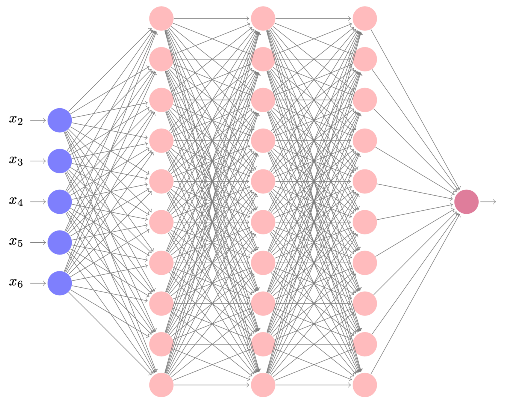
Machine learning algorithms consist of 3 components:

- *Representation*: a model $f_{\theta,b}$ must be represented in a formal language that the computer can handle
    - Defines the 'concepts' it can learn, the hypothesis space
- *Evaluation*: an internal way to choose one hypothesis over the others
    - Error $\mathcal{E}$, loss function,...
- *Optimization*: an efficient way to search the hypothesis space

Summary

Example : Multilayer perceptron

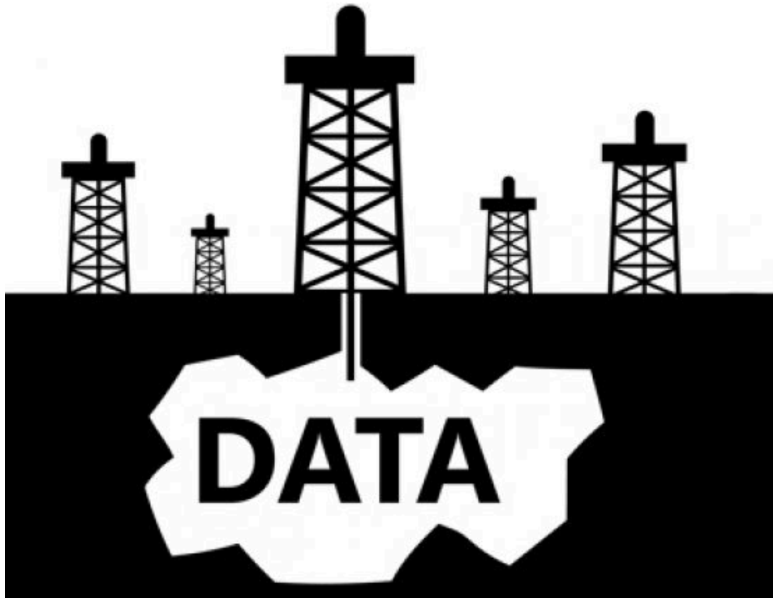
- *Representation*:
  - Fixed architecture
  - Each neuron process $\sigma(w^\top x)$
    - x input of the neuron
    - w weight vector $\rightarrow \theta$ = set of weights
    - σ: activation function
  - Each neuron is connected to all the neurons in the next layer
  - $f_{\theta,b}$: output of the last layer
- *Evaluation*: loss function $\mathcal{L}(f_{\theta,b})$ estimated on a training set
- *Optimization*: Find $\hat{\theta}, \hat{b}$ minimizing $\mathcal{L}(f_{\theta,b})$ (e.g. gradient descent)



Machine Learning Challenges

Two things can go wrong

- Bad data
- Bad algorithms



Bad data

Not enough data

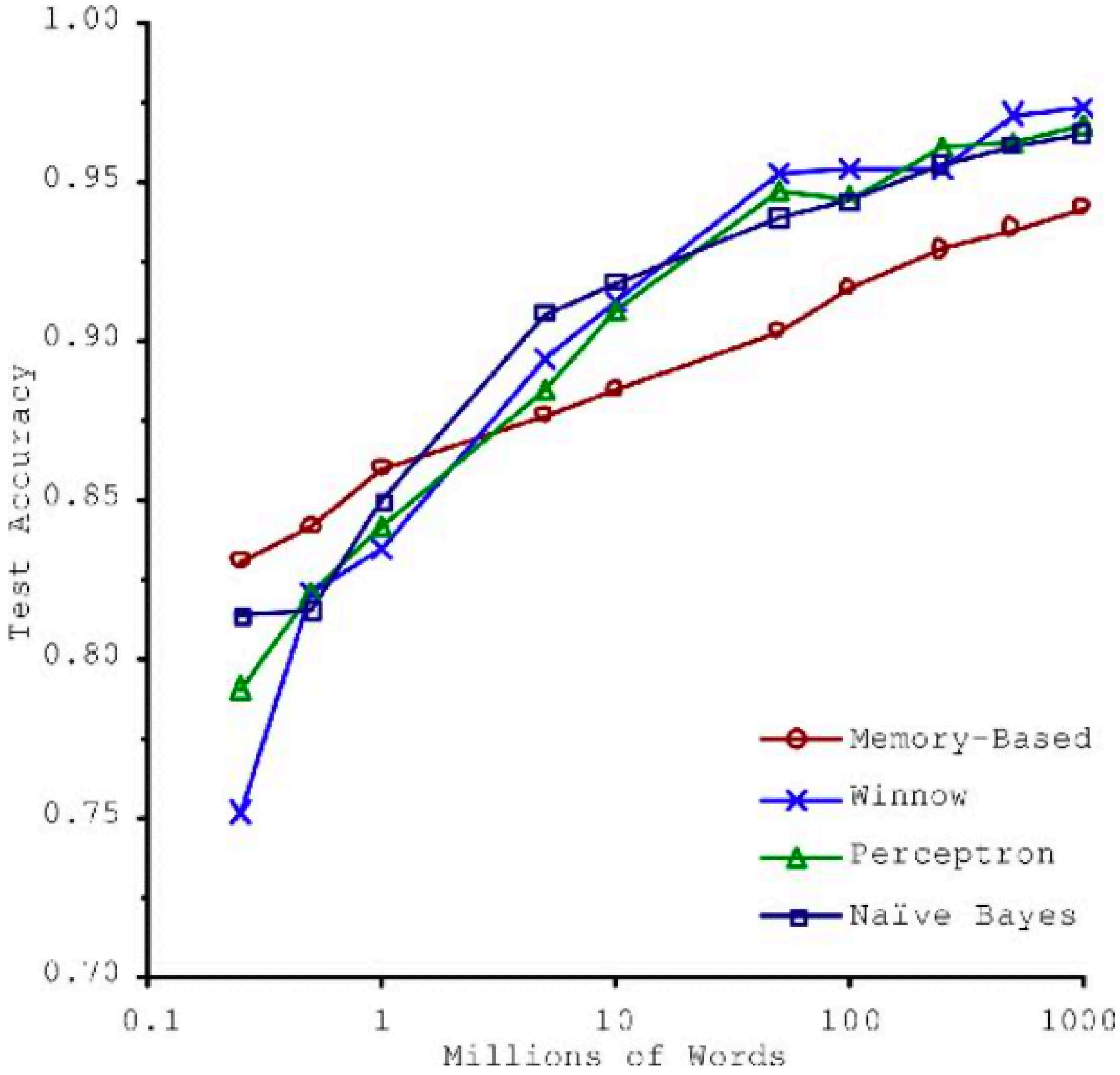
- A child can learn (and generalize) what is an apple with only few examples
- A machine learning algorithm needs thousands of examples
- Take care of imbalanced data

Bad data

Not enough data

Few data $\to$ a simple algorithm.

Example: to-two disambiguation.



Bad data

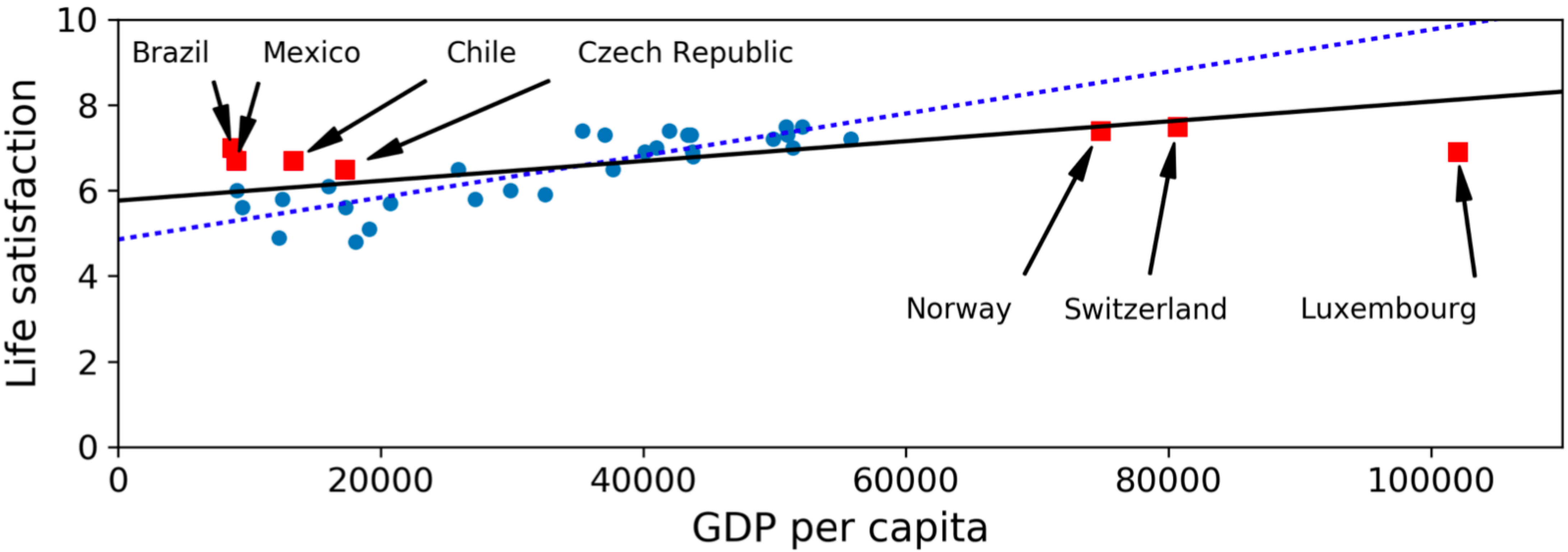
More data is better than a cleverer method (but if you have both ... :-))

- More data reduces the chance of overfitting (see below...)
- Less sparse data reduces the curse of dimensionality (see below...)
- Non-parametric models: number of model parameters grows with amount of data
  - can learn any model given sufficient data (but can get stuck in local minima)
- Parametric (fixed size) models: fixed number of model parameters
  - Can be given a huge number of parameters to benefit from more data
  - Deep learning models can have millions of weights, learn almost any function.

$\Rightarrow$ The bottleneck is moving from data to compute/scalability

Bad data

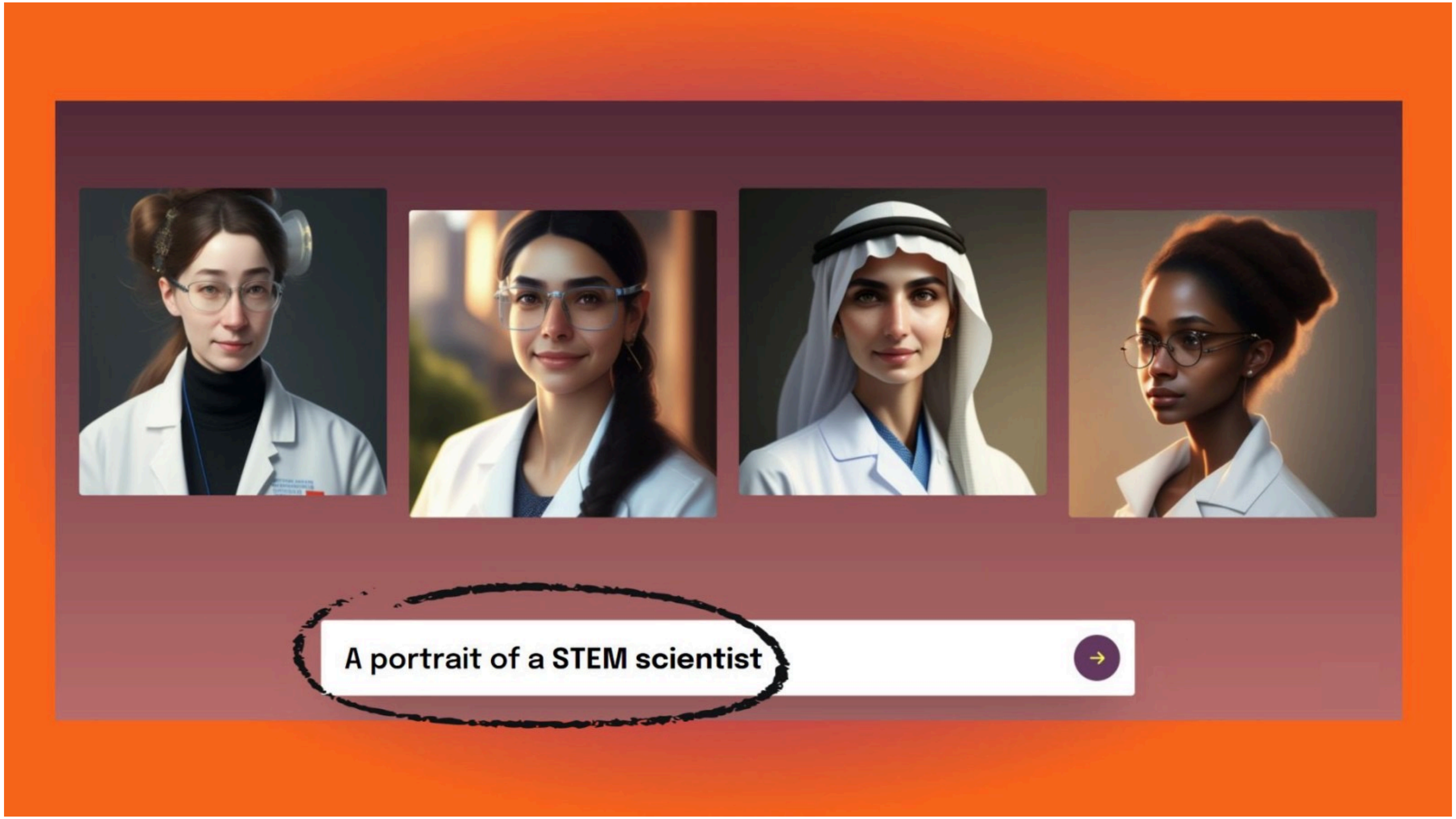
Unrepresentative data



⇒ Bias

Bad data

Unrepresentative data

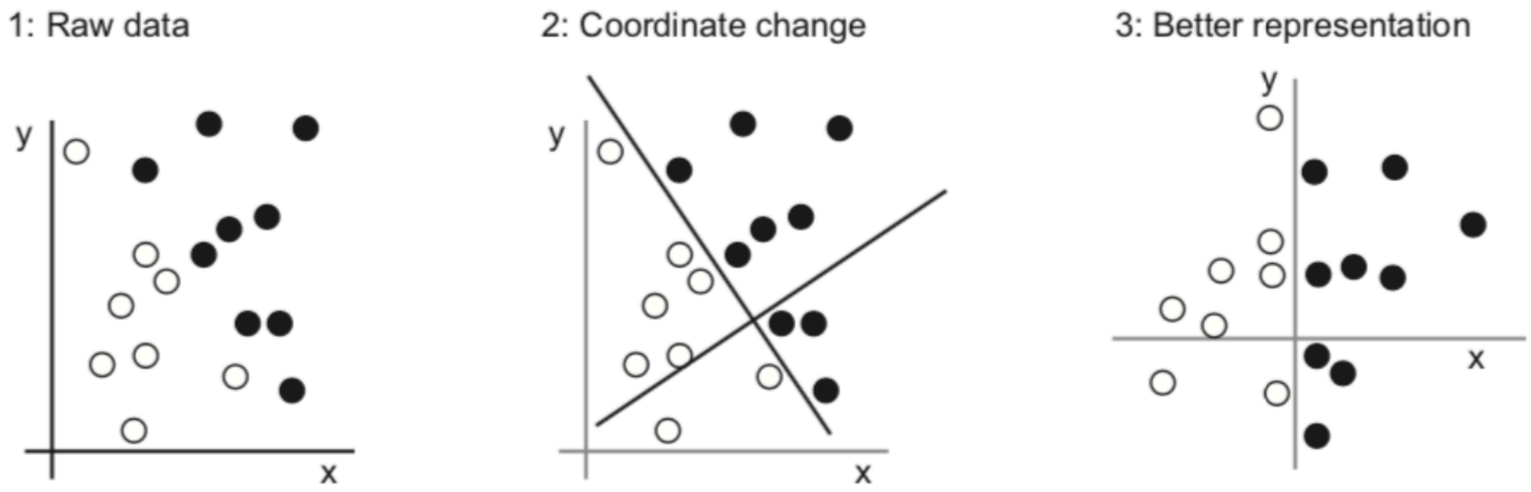


⇒ Bias

Bad data

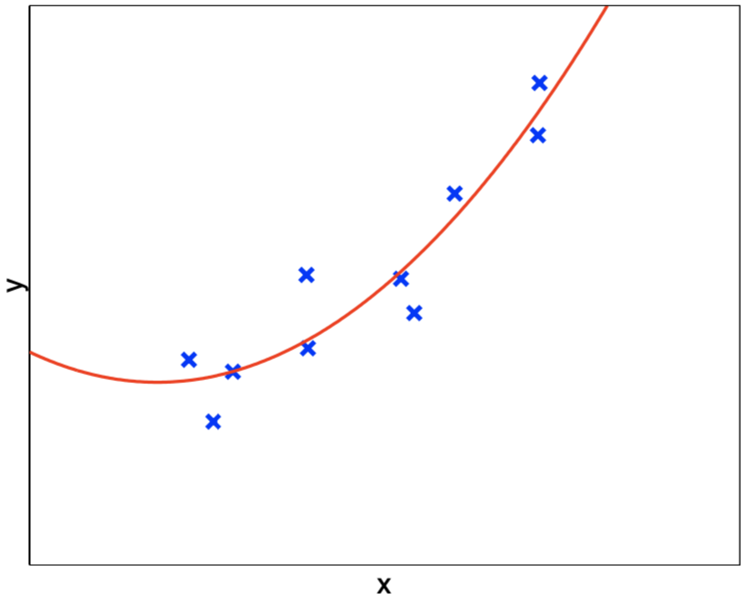
Poor quality data $\Rightarrow$ Feature engineering

- Remove or correct outliers
- Handle missing values : ignore / impute
- Transform data: better data representation, better models
- Manage unsignificant features: selection/extraction/collection
- Take care of the curse of dimensionality !!



Bad algorithms

Linear least square fitting of a set $E = (x, y)$ with a polynomial of order $p \in [\![2, 9]\!]$.



Bad algorithms

Overfitting / Underfitting

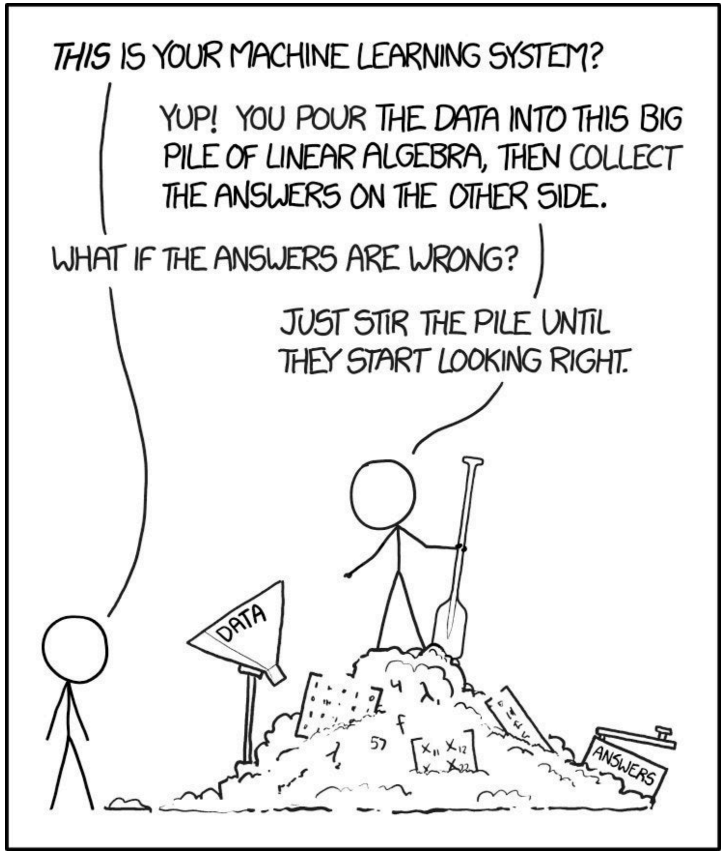
- Overfitting: building a model that is too complex given the amount of data
    - peculiarities in the training data (noise, biases,...)
    - 100% accurate on the training data, but very bad on new data
      $\Rightarrow$ Solve by making model simpler (regularization), or getting more data
    - There exists techniques for detecting overfitting (e.g. bias-variance analysis).
- Underfitting: building a model that is too simple given the complexity of the data
    - Use a more complex model

$\Rightarrow$ *Generalization* capability

Bad algorithms

Model selection

- No single algorithm is always best.
- Next to the error/loss function, need for an external evaluation function
    - Feedback signal: are we actually learning the right thing?
        - Are we under/overfitting?
    - Carefully choose to fit the application.
    - Needed to select between models (and hyperparameter settings)



Bad algorithms

Model selection

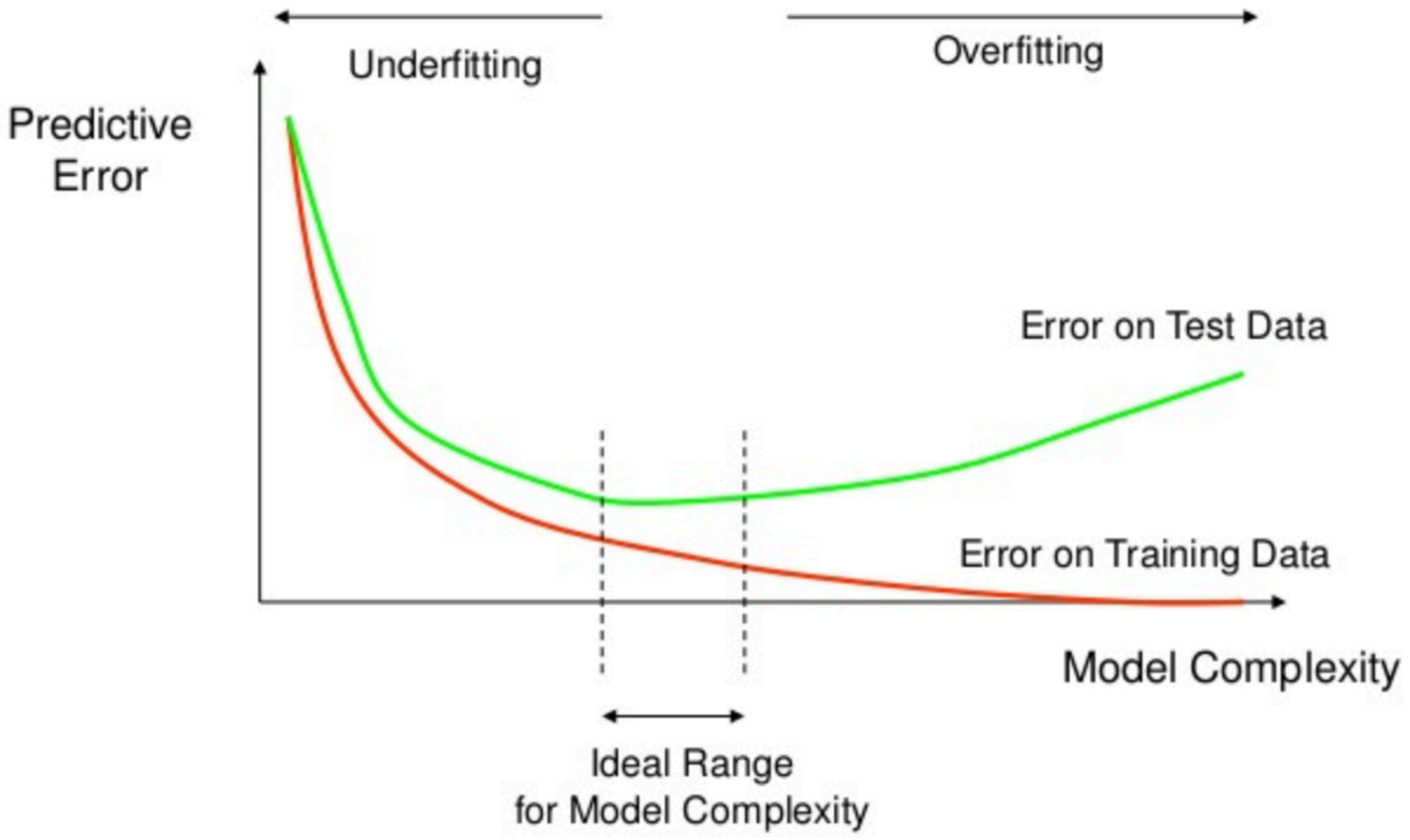
$E = E_{train} \cup E_{test}$

- $E_{train} \rightarrow$ training error e_t
- $E_{test} \rightarrow$ generalization error e_g

	small e_t	**large e_t**
small e_g	generalizes, performs well	possible (luck or fraud?)
large e_g	fails to generalize, overfit	generalizes, but performs poorly

Bad algorithms

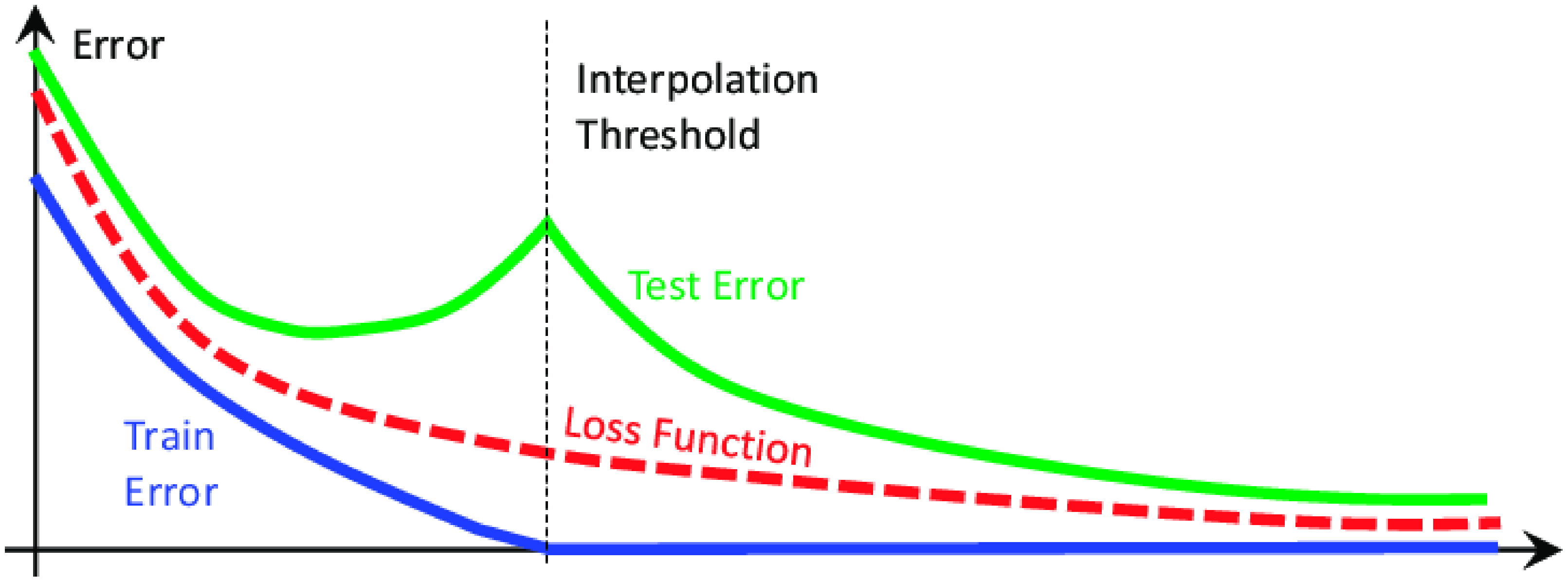
Model selection



Bad algorithms

Model selection

Double descent phenomenon



Bad algorithms

Model selection

When the model includes hyperparameters, set aside part of training set as a validation set

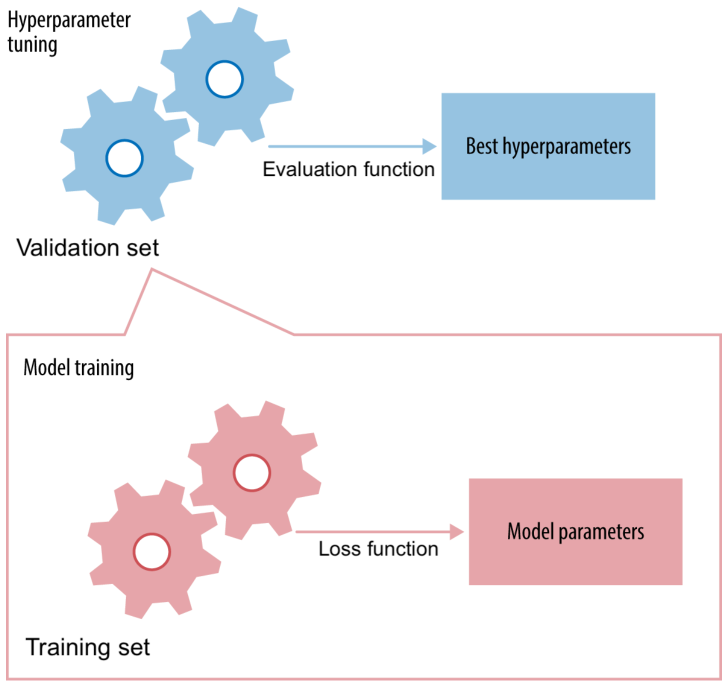
$E = E_{train} \cup E_{test} \cup E_{val}$

- E_{train}: training error
- E_{test}: generalization error
- E_{val}: hyperparameter optimization

Bad algorithms

Model selection

- For a given hyperparameter setting, learn the model parameters on the training set
- Evaluate the trained model on the validation set
    - Tune the hyperparameters to maximize a certain metric (e.g. accuracy)
- Keep test set hidden during all training

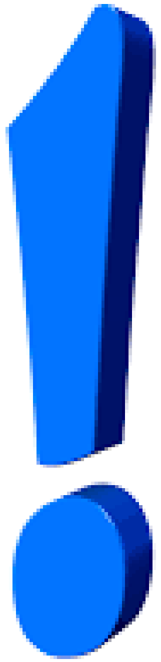


Bad algorithms

Model selection

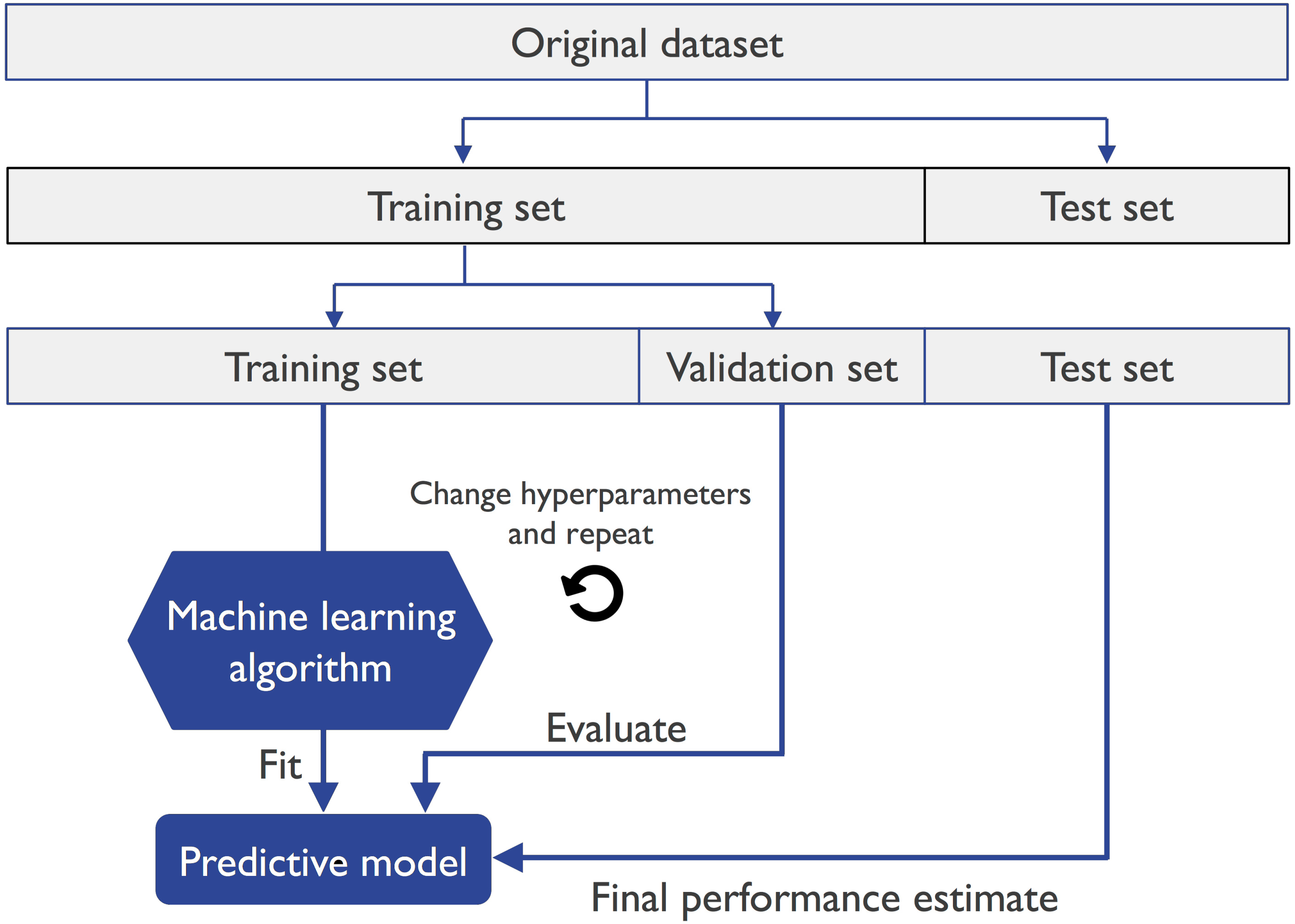
Generalization must guide your process !

- Never evaluate the final model on E_{train}, except for:
  - Tracking whether the optimizer converges (learning curves)
  - Diagnosing under/overfitting:
    - High training and test error: underfitting
    - low training error, high test error: overfitting
- Always keep a completely independent test set
- On small datasets, use multiple train-test splits to avoid sampling bias (e.g. cross validation)



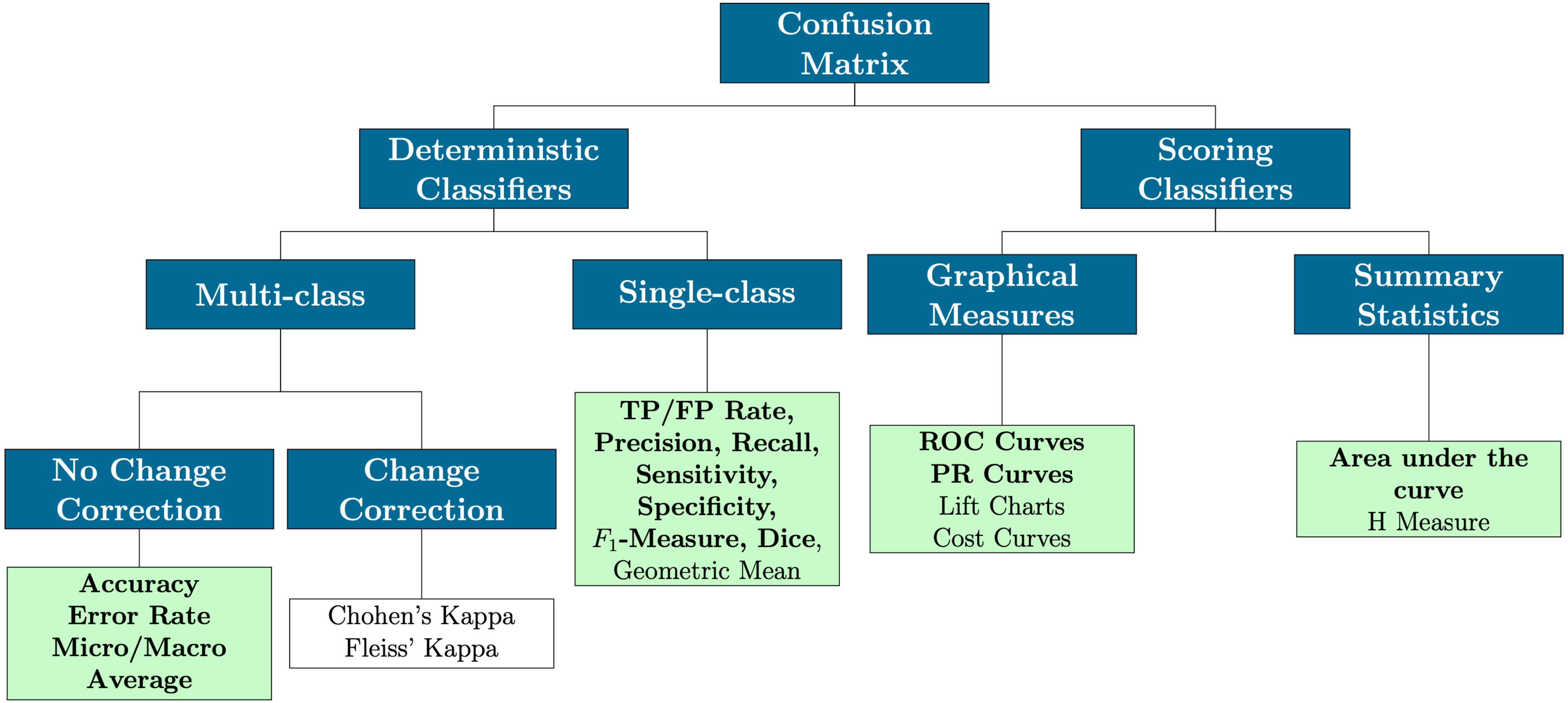
Bad algorithms

Summary



Measuring performance

Going further than the test error...



credits: Sebastian Pölsterl

Measuring performance

Example: binary classification problem (+/-) on N individuals using algorithm $\mathcal{A}$:

- True Positive (TP) = + sample correctly classified as belonging to the + class
- False Positive (FP) = - sample misclassified as belonging to the + class
- True Negative (TN) = - sample correctly classified as belonging to the - class
- False Negative (FN) = + sample misclassified as belonging to the - class

Measuring performance

Confusion matrix

	True +	True -
Predicted +	TP	FP (type I error (α))
Predicted -	FN (type II error (β))	TN

Accuracy = $\frac{TP+TN}{N}$

Error rate = 1-*Accuracy*

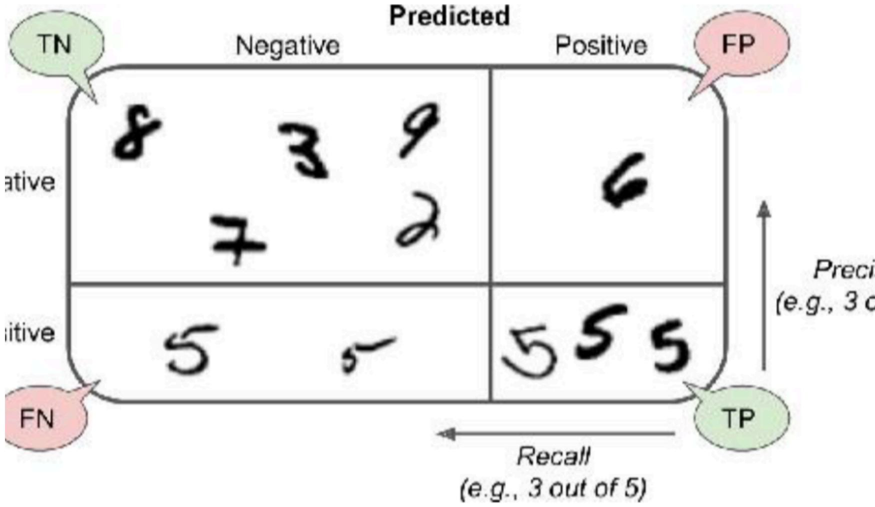
FPR = $\frac{FP}{FP+TN}$

Recall $R = \frac{TP}{TP+FN}$

Precision $P = \frac{TP}{TP+FP}$

F_1 *score* : $F_1 = \frac{2PR}{P+R}$

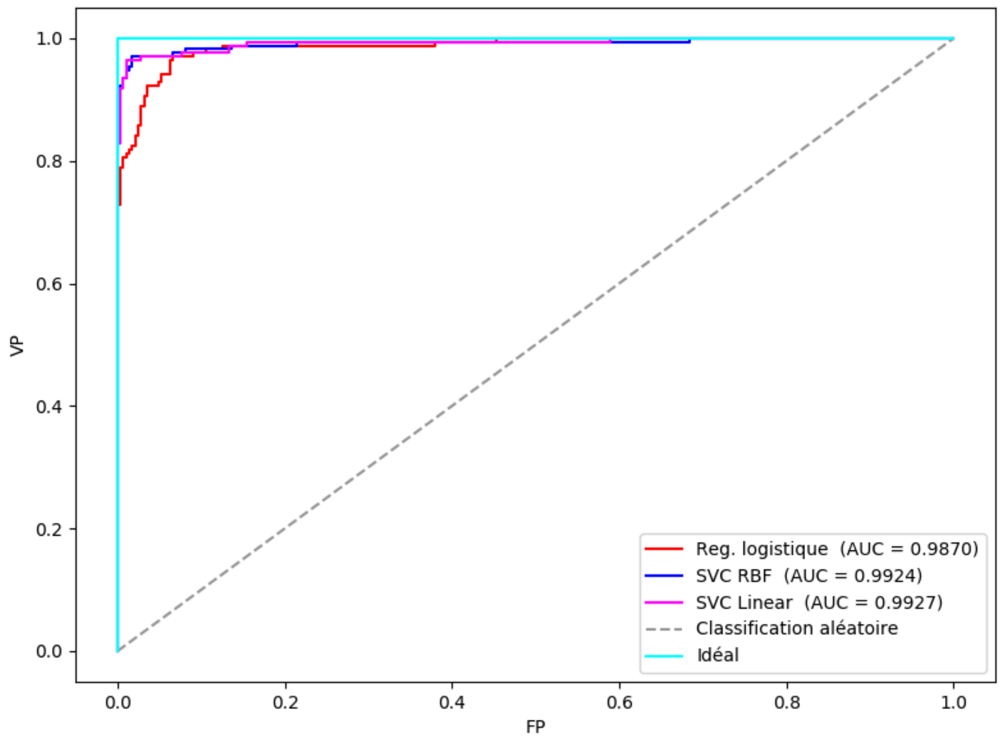
Extension to multiple class: One vs. One/One vs. All



Measuring performance

ROC curve

- Binary classifier returns probability or score that represents the degree to which class an example belongs to.
- The ROC curve plots R vs. FPR for all possible thresholds of the $\mathcal{A}$'s score.
- Visualizes the trade-off between benefits (R) and costs (FPR).



Area Under the Curve

$\Rightarrow$AUC$\in [0, 1] \sim$ probability that $\mathcal{A}$ will rank a randomly chosen + instance higher than a randomly chosen - instance (Mann-Whitney test)

Measuring performance

ROC curve

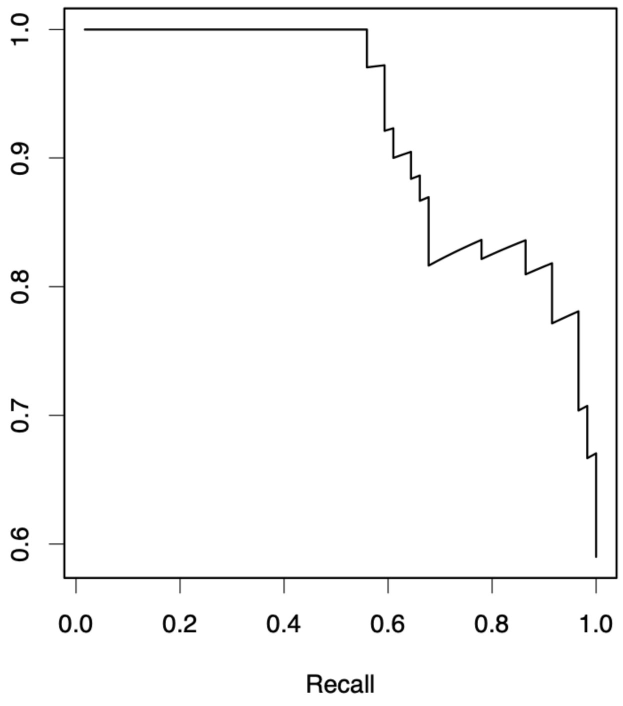
Drawbacks:

- can present an overly optimistic view of $\mathcal{A}$'s performance if there is a large skew in the class distribution (the data set contains much more samples of one class)
- A large change in the number of false positives can lead to a small change in the false positive rate

Measuring performance

P/R curve

- plots P vs. R for all possible thresholds of the $\mathcal{A}$'s score.
- P/R curve of optimal classifier is in the upper-right corner.
- One point in P/R space corresponds to a single confusion matrix.
- Algorithms that optimizes the AUC are not guaranteed to optimize the area under the PR curve



Measuring performance

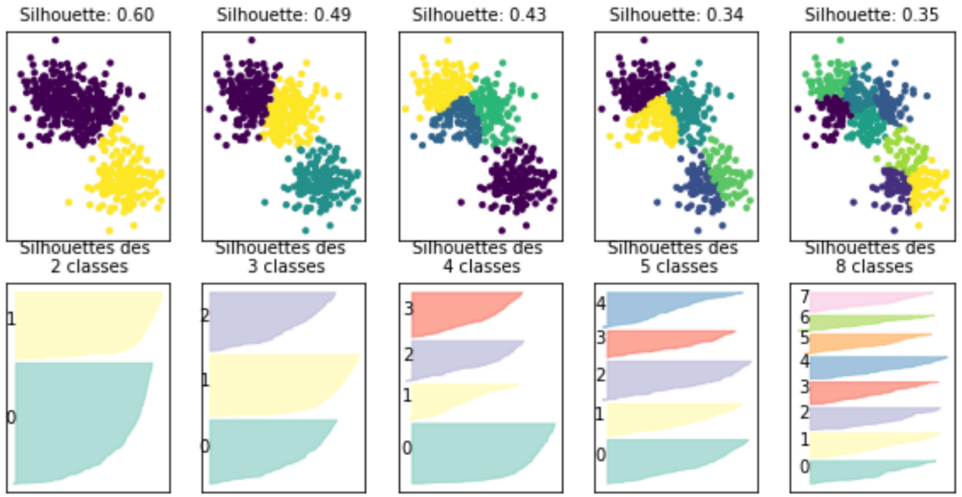
And what about unsupervised methods ?

- Hard to evaluate since the ground truth is not known ?

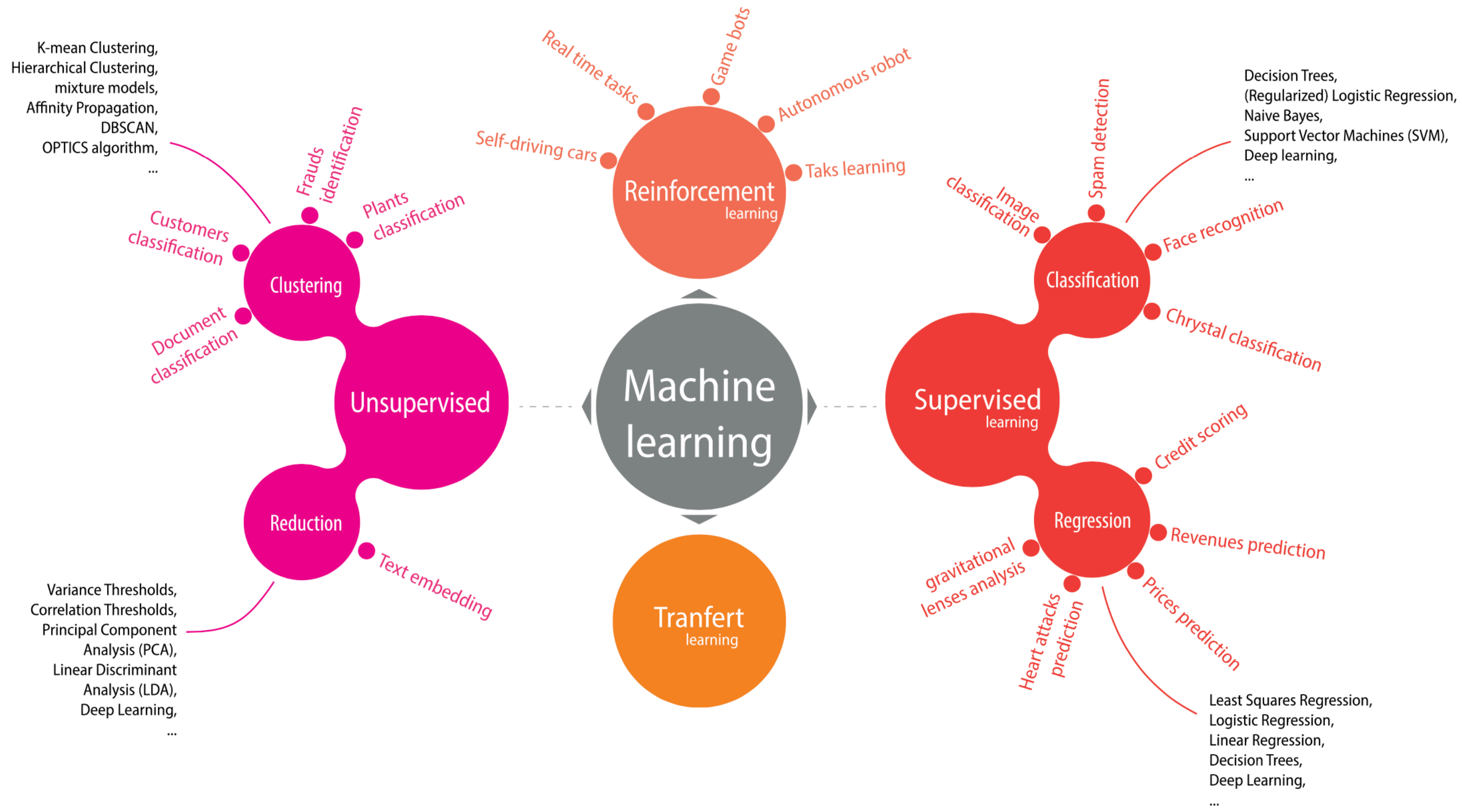
Silhouette Index

$x_i \in$ cluster P_k, and P_j closest cluster of P_k:

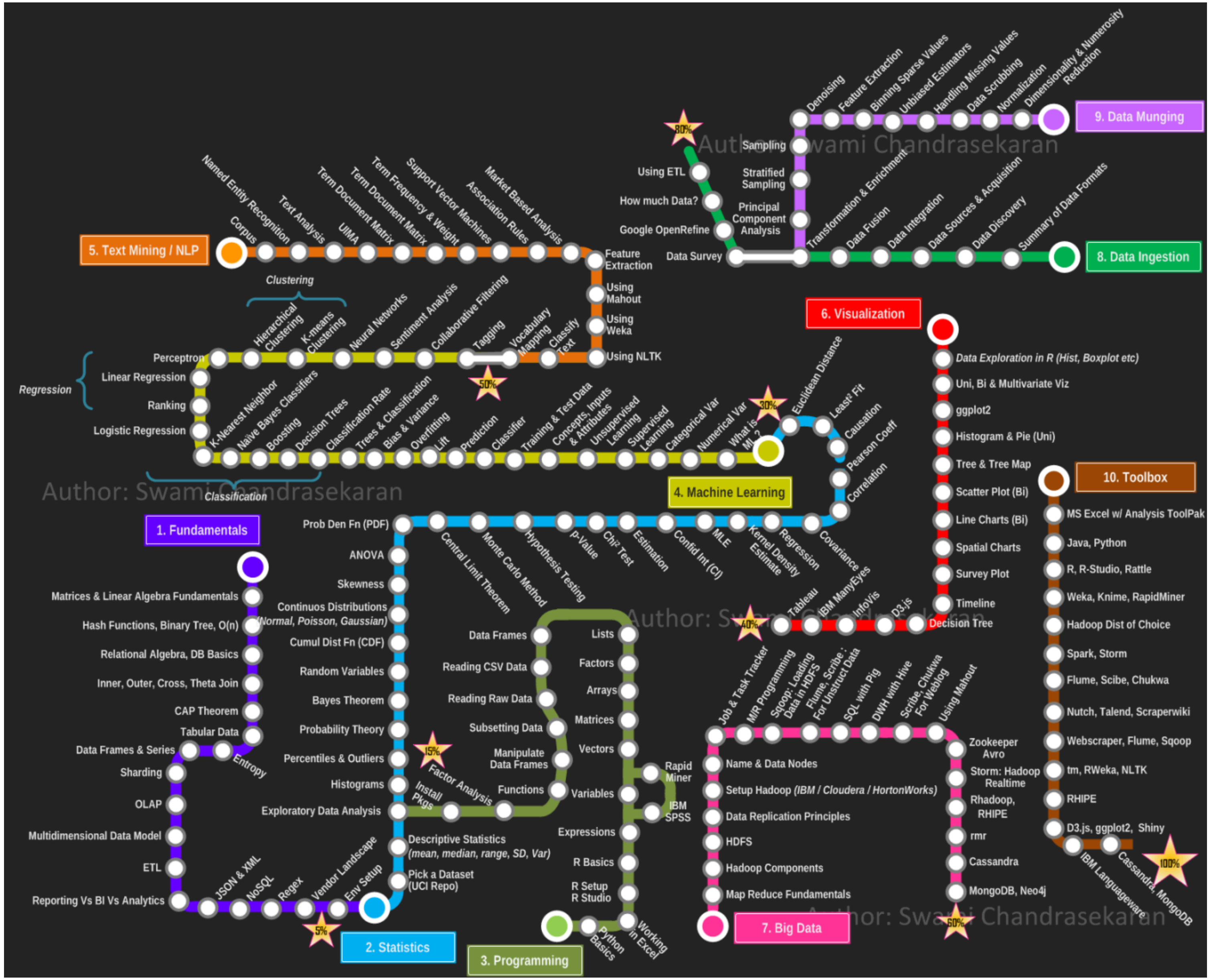
- $a_i = \frac{1}{|P_k|-1} \sum_{x_j \in P_k} d(x_i, x_j)$,
- $b_i = \frac{1}{|P_j|} \sum_{x_l \in P_j} d(x_i, x_l)$
- silhouette index of x_i :
  $s_i \in [-1, 1] = \frac{b_i - a_i}{max(a_i, b_i)}$.



Summary


K-mean Clustering,
Hierarchical Clustering,
mixture models,
Affinity Propagation,
DBSCAN,
OPTICS algorithm,
...
Frauds identification
Plants classification
Customers classification
Clustering
Document classification
Unsupervised
Reduction
Text embedding
Variance Thresholds,
Correlation Thresholds,
Principal Component Analysis (PCA),
Linear Discriminant Analysis (LDA),
Deep Learning,
...
Real time tasks
Game bots
Autonomous robot
Self-driving cars
Taks learning
Reinforcement
learning
Machine learning
Tranfert
learning
Image classification
Spam detection
Decision Trees,
(Regularized) Logistic Regression,
Naive Bayes,
Support Vector Machines (SVM),
Deep learning,
...
Face recognition
Classification
Chrystal classification
Supervised
learning
Credit scoring
Regression
Revenues prediction
gravitational lenses analysis
Heart attacks prediction
Prices prediction
Least Squares Regression,
Logistic Regression,
Linear Regression,
Decision Trees,
Deep Learning,
...

ML MAP (still evolving)



Statistical Learning Theory

Learning Model (Vapnik):

- A generator G of random vectors $x \in D$, i.i.d, $P(x)$ fixed but unknown
- A supervisor S giving for each input x a value $y \in C$ drawned from $P(y|x)$ fixed but unknown
- A Learning Machine LM implementing a set of functions $\mathcal{F}$

> *Problem statement*: Find $f \in \mathcal{F}$ that best fit S.

Training set $E = \{(x_1, y_1), \cdots, (x_l, y_l)\}$ l observations i.i.d from $P\left(x, y\right) = P\left(x\right)P\left(y|x\right)$

> *Problem statement*: Find $f : D \to C$ such as $R\left(f\right) = P\left(y \neq f\left(x\right)\right)$ is minimal.

Statistical Learning Theory

Loss function

$$L\left(y, f\left(x\right)\right) = \mathbb{1}_{\ y \neq f(x)}, \text{or } \left(y - f(x)\right)^2 \text{or } \ldots$$

Difference between $S\left(y\right)$ and LM $\left(f\left(x\right)\right)$

Risk (Error)

$$R\left(f\right) = \int L\left(y, f\left(x\right)\right) dP\left(x, y\right) = P\left(y \neq f\left(x\right)\right)$$

$\Rightarrow$ Expected value of the loss function = probability that f predicts a different value of S.

> Problem statement

Knowing E, find $f \in \mathcal{F}$ such that

$$f = Arg \min_{g \in \mathcal{F}} R\left(g\right)$$

Statistical Learning Theory

Minimum risk function

For classification problem, $\exists$ a minimum risk function

$$f_{Bayes}\left(x\right) = Arg\max_{y} P\left(y|x\right)$$

f_{Bayes} :"Ideal" function to reach (no hypothesis on the underlying distributions)

> Problem statement

Knowing E, approximate f_{Bayes} with $f \in \mathcal{F}$ (a priori $f_{Bayes} \notin \mathcal{F}$)

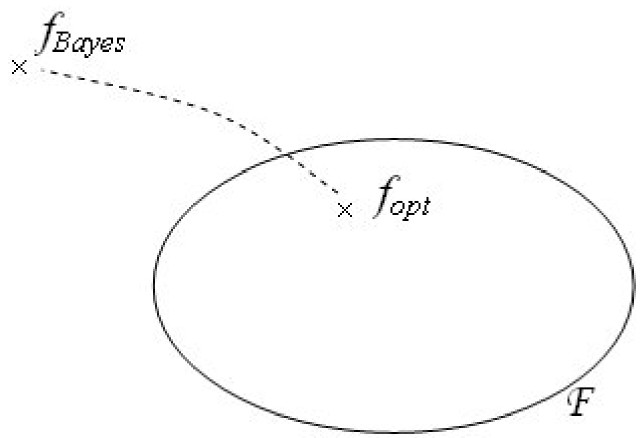
Statistical Learning Theory

Let suppose there exists $f_{opt} \in \mathcal{F}$ of minimal risk:

$$0 \leq R\left(f_{Bayes}\right) \leq R\left(f_{opt}\right) = \underbrace{R\left(f_{Bayes}\right)}_{\text{non-deterministic}} + \underbrace{\left(R\left(f_{opt}\right) - R\left(f_{Bayes}\right)\right)}_{\text{structural error}}$$

Choice of $\mathcal{F}$:

- Using expressive $\mathcal{F}$ spaces to allow $R\left(f_{opt}\right) \approx R\left(f_{Bayes}\right)$
- Not too rich, otherwise risk of overfitting



Statistical Learning Theory

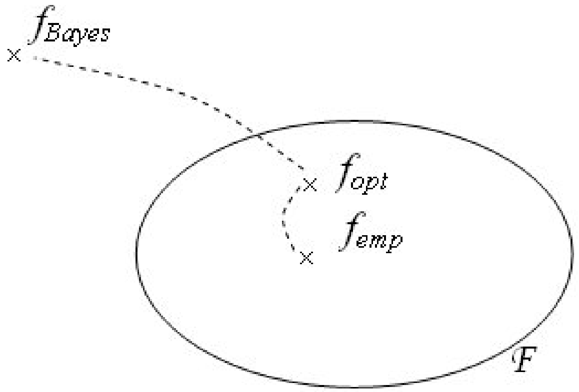
Empirical risk

Natural idea: find $f \in \mathcal{F}$ that best classify E

$$R_{emp}(f) = \frac{1}{l}\sum_{i=1}^{l} L\left(y_i, f(x_i)\right) = \frac{Card\left\{i | f(x_i) \neq y_i\right\}}{l}$$

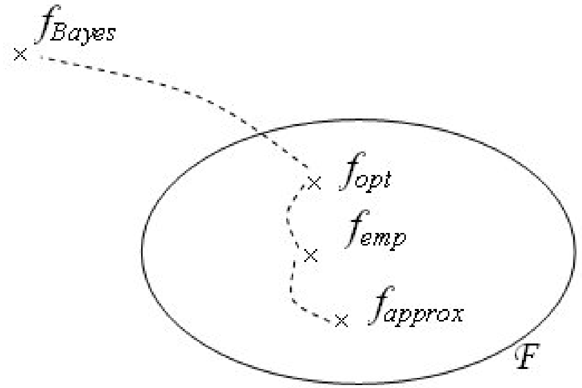
$\Rightarrow$*Empirical Risk Minimization(ERM)*: Find $f \in \mathcal{F}$ (f_{emp}) minimizing $R_{emp}(f)$

$$R(f_{emp}) = R(f_{Bayes}) + (R(f_{opt}) - R(f_{Bayes})) + (R(f_{emp}) - R(f_{opt}))$$



Statistical Learning Theory

One cannot expect to compute f_{emp} is a reasonable time
$\rightarrow$ Approximation f_{approx} of f_{emp}.



Statistical Learning Theory

At least four reasons altering the results of a classfication method:

- *Nature of the problem* : minimum $\Rightarrow$ Bayes and can be important;
- *Low expressivity of* $\mathcal{F}$: structural error;
- *Non consistancy of ERM principle* : do we get close to f_{opt} with E? (be careful: learning by heart !!);
- Minimization can be computationaly hard/unstable.

Statistical Learning Theory

Uniform convergence of the empirical risk

ERM does not necesseraly get close to rhe real risk (E is randomly drawned)

$\Rightarrow$ Serious problem !!

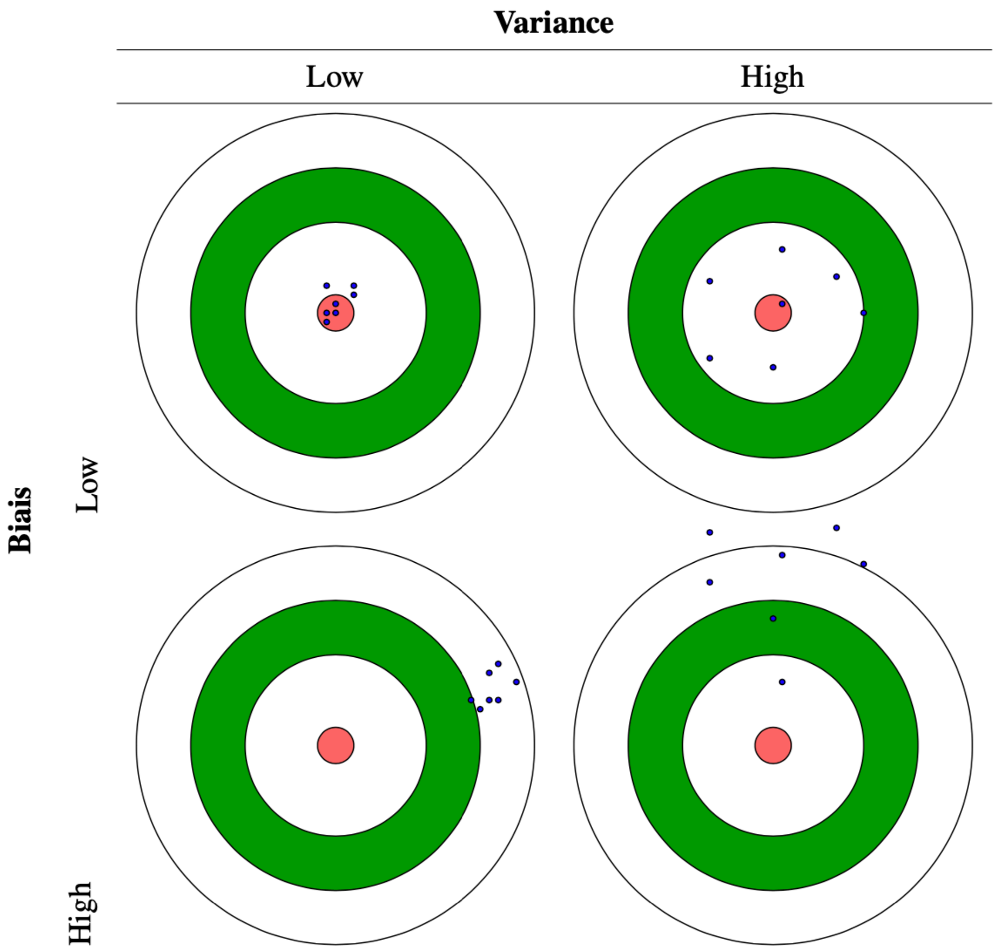
- f_{opt}close to $f_{bayes} \Rightarrow$ rich $\mathcal{F}$
- To find f_{opt} by ERM, $\mathcal{F}$ not too rich....

Extreme cases

- $\mathcal{F} = \{f_{opt}\}$: easy to find but probably high R_{emp}
- $\mathcal{F}$: set of all possible functions $\Rightarrow f_{bayes} \in \mathcal{F}$ but also all f minimizing R_{emp} and in particular $f_{byheart}$.

Statistical Learning Theory

- Biais $\approx$ distance between f_{bayes} and f_{opt}
- Variance $\approx$ distance between f_{opt} and f_{emp}



Statistical Learning Theory

The Empirical risk uniformly converges (in probability) to the real risk in $\mathcal{F}$ iff

$$(\forall \epsilon > 0) \ \lim_{l \to \infty} Pr \left\{ Max_{f \in \mathcal{F}} \left| R^l_{emp}(f) - R(f) \right| \geq \epsilon \right\} = 0$$

If the empirical risk uniformly converges to the real risk then a LM based on ERM converges in probability to f_{opt}:

$$\lim_{l \to \infty} Pr \left\{ \left| R\left(f^l_{emp}\right) - R\left(f_{opt}\right) \right| \geq \epsilon \right\} = 0$$