Machine learning

Alberto TONDA, Ph.D. (Senior permanent researcher, DR)

UMR 518 MIA-PS, INRAE, AgroParisTech, Université Paris-Saclay
UAR 3611, Institut des Systèmes Complexes de Paris Île-de-France

Why should we care about machine learning?

- ML can create predictive (and not only) models from data
- Algorithms are (relatively) easy to use, lots of libraries
- Great amount of interest (and hype) since 2012
- Useful for complex interactions (physics/chemistry/biology)
- Details might be difficult, high-level ideas are intuitive

Have you already used any machine learning techniques?

What's in this presentation?

- General overview of machine learning for modelling
- Practical advice on algorithms and metrics to use
- A couple of exercises in ipython notebooks (browers)
- Current trends in machine learning research
- Examples on the "données fil rouge"
- Funny pictures from the 2000s-2010s, *hot takes on ML*

Who am I?

- Career
  - Bachelor and Master in Computer Science Engineering
  - Ph.D. from Politecnico di Torino, Italy, in 2011
  - Permanent researcher in France since late 2012 (INRAE)
  - Senior researcher (DR2) since 2023
- Research interests
  - Stochastic optimization
  - Machine learning (Explainable AI)
  - Applied to biological/agri-food data

What is Artificial Intelligence?

- Short answer, there is no clear definition
  - We do not have a good definition of *intelligence*, so...
  - Broadly speaking, AI defines a *field* more than a *method*
  - Machine learning, reinforcement learning, symbolic AI, ...
- Tentative definitions (there is no agreement)
  - «When a non-biological being successfully completes a task commonly believed to require biological intelligence»
  - «Perceiving, synthesizing, and inferring information»
  - «Efficiency and speed, in learning a new task» (Chollet, 2019)
- How do we *measure* intelligence?

What is Artificial Intelligence?

NARROW / WEAK

Focused on a specific task

- Symbolic AI
  - E.g. rule-based systems
- Machine learning
  - Supervised, unsupervised
  - Natural language processing
  - Image recognition/segmentation
- Reinforcement learning
- Neuro-symbolic AI

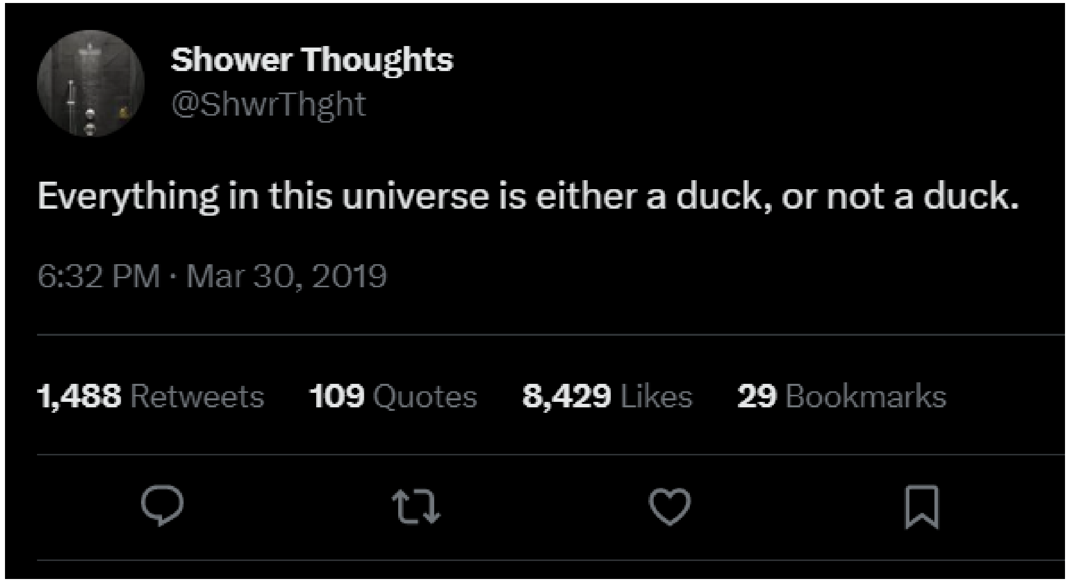
GENERAL (AGI)

Can perform any type of (human?) task

- Does not exist (...yet?)
- Closest thing is NLP: Large Language Models (LLM) like ChatGPT

Symbolic AI

- Symbolic manipulation
  - Reality is *continuous* (with good approximation)
  - Symbols are *discrete*, and humans are good at using them



Shower Thoughts
@ShwrThght

Everything in this universe is either a duck, or not a duck.

6:32 PM · Mar 30, 2019

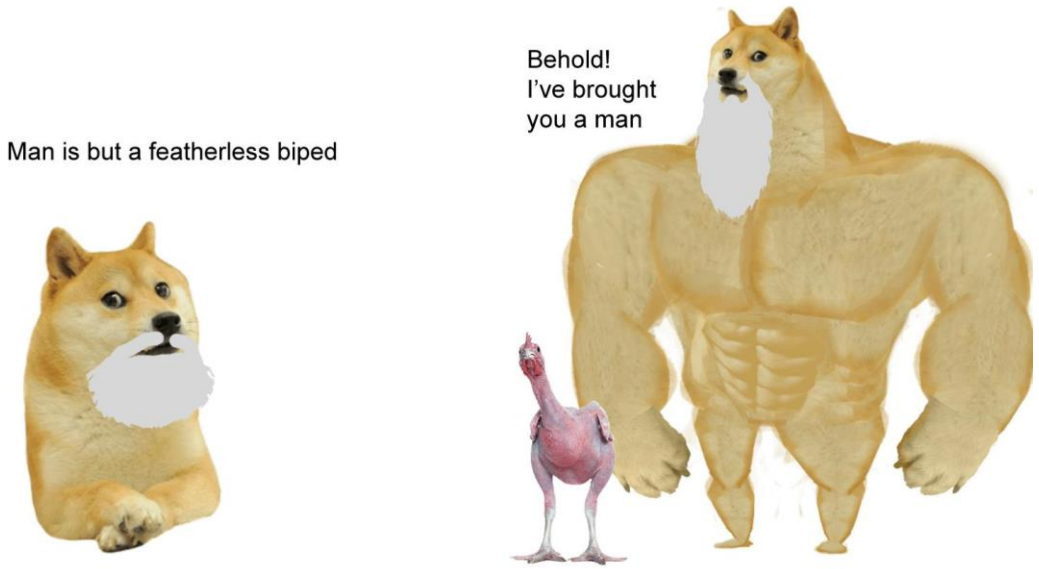
1,488 Retweets 109 Quotes 8,429 Likes 29 Bookmarks

Symbolic AI

- Symbols seem normal and natural, map into the real world (in linguistics, it’s called *extension*)
- Natural language is a powerful human symbol manipulator
- However, there is chaos hidden under the surface
  - What is the reality of a *river*?
  - What is the reality of a *chair*?
  - What is the reality of a *number*?

Symbolic AI

- Symbol can be hard to define, but we grasp it intuitively
  - It’s an old, old problem: see Plato and Diogenes
  - *Entire fields of research* on this (neuroscience, cognitive sciences, neurolinguistics, …)
- “Explaining” symbols to AI is harder yet
- Issues with “common sense”
- Reached limits in the 1980s



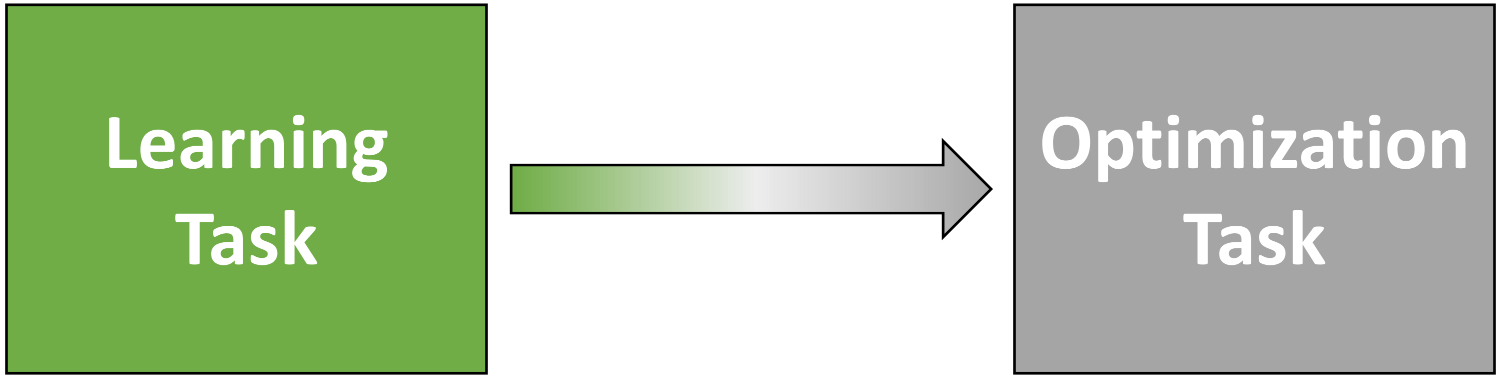
Symbolic AI

- In practice, find or exploit human-readable rules
  - Expert systems (“if-then-else” rules)
  - Knowledge graphs, linking entities with relationships
  - First-order logic rules
  - Ontologies
  - Decision trees (that are also considered part of ML!)

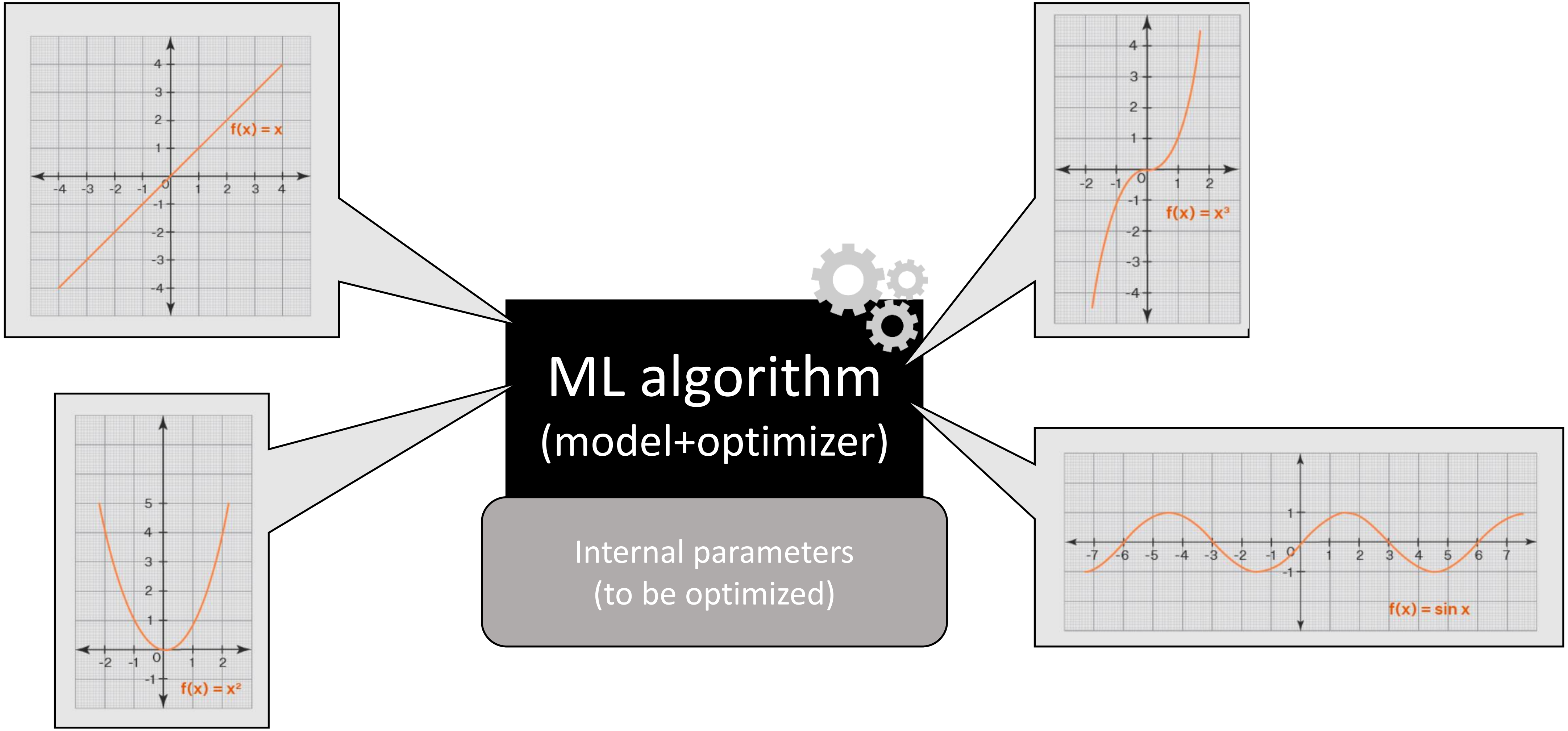
- Before the advent of ML, considerable success stories
- Symbolic AI is still in use, paired with ML

Machine learning

- Learn a task directly from examples
  - No need for theory, just large quantities of data
  - *Samples* (rows) and *features* (columns)
- "Dirty secret" of ML: it's mostly optimization
  - Restate **learning task** as **optimization task**
  - Solve it relying on available (training) data



Machine learning algorithms



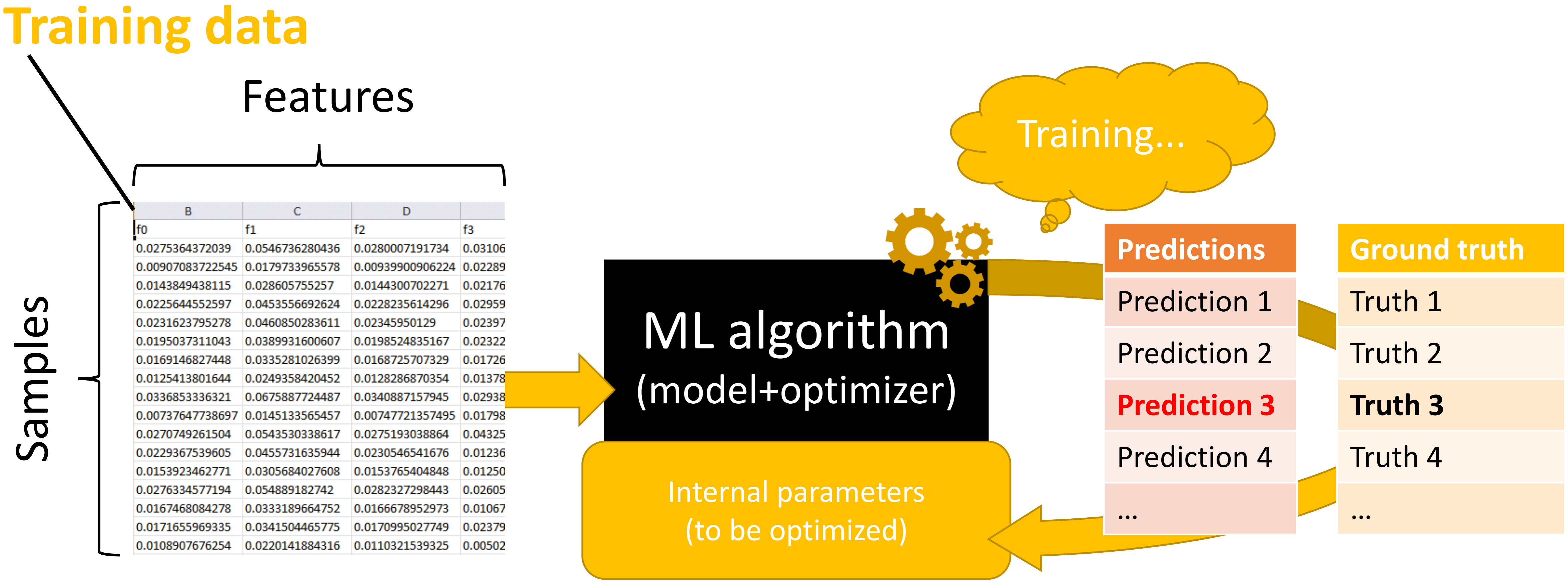
ML algorithm
(model+optimizer)

Internal parameters
(to be optimized)

Supervised machine learning

- Learn from (hopefully) correct examples
  - Data contains measured values of the target (ground truth)
  - Minimize difference between model predictions and ground truth
- Regression
  - Target is a continuous value (0.9, 22.5, 0.0017, …)
  - From the values of the features of a sample, **predict** target value
- Classification
  - Target is a category (good/bad, high/medium/low, toxic/ok, …)
  - From the values of the features of a sample, assign to category

Machine learning (supervised)



B	C	D	
f0	f1	f2	f3
0.0275364372039	0.0546736280436	0.0280007191734	0.03106
0.00907083722545	0.0179733965578	0.00939900906224	0.02289
0.0143849438115	0.028605755257	0.0144300702271	0.02176
0.0225644552597	0.0453556692624	0.0228235614296	0.02959
0.0231623795278	0.0460850283611	0.02345950129	0.02397
0.0195037311043	0.0389931600607	0.0198524835167	0.02322
0.0169146827448	0.0335281026399	0.0168725707329	0.01726
0.0125413801644	0.0249358420452	0.0128286870354	0.01378
0.0336853336321	0.0675887724487	0.0340887157945	0.02938
0.00737647738697	0.0145133565457	0.00747721357495	0.01798
0.0270749261504	0.0543530338617	0.0275193038864	0.04325
0.0229367539605	0.0455731635944	0.0230546541676	0.01236
0.0153923462771	0.0305684027608	0.0153765404848	0.01250
0.0276334577194	0.054889182742	0.0282327298443	0.02605
0.0167468084278	0.0333189664752	0.0166678952973	0.01067
0.0171655969335	0.0341504465775	0.0170995027749	0.02379
0.0108907676254	0.0220141884316	0.0110321539325	0.00502

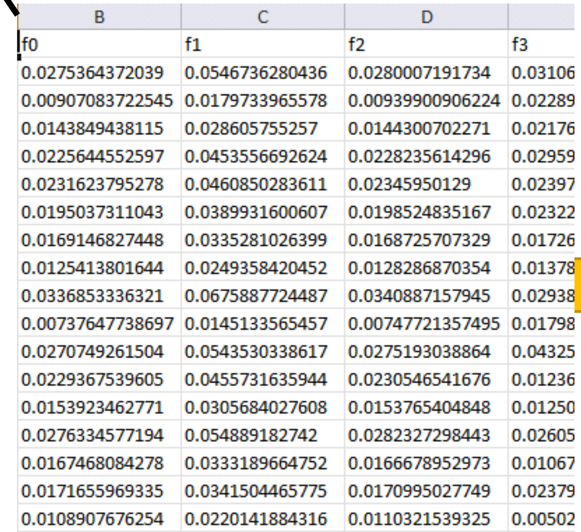
Predictions	**Ground truth**
Prediction 1	Truth 1
Prediction 2	Truth 2
Prediction 3	**Truth 3**
Prediction 4	Truth 4
…	…

Machine learning (supervised)

Training data

Features

Samples

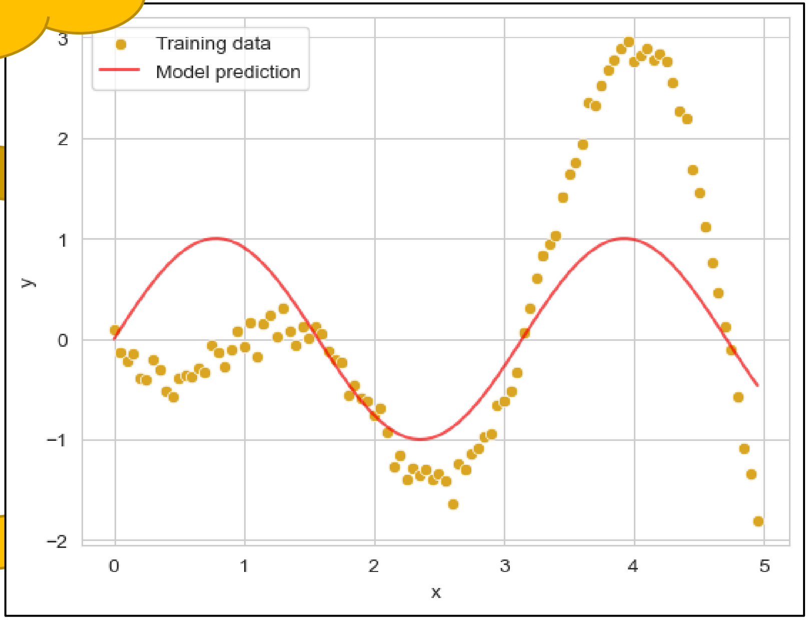


B	C	D	
f0	f1	f2	f3
0.0275364372039	0.0546736280436	0.0280007191734	0.03106
0.00907083722545	0.0179733965578	0.00939900906224	0.02289
0.0143849438115	0.028605755257	0.0144300702271	0.02176
0.0225644552597	0.0453556692624	0.0228235614296	0.02959
0.0231623795278	0.0460850283611	0.02345950129	0.02397
0.0195037311043	0.0389931600607	0.0198524835167	0.02322
0.0169146827448	0.0335281026399	0.0168725707329	0.01726
0.0125413801644	0.0249358420452	0.0128286870354	0.01378
0.0336853336321	0.0675887724487	0.0340887157945	0.02938
0.00737647738697	0.0145133565457	0.00747721357495	0.01798
0.0270749261504	0.0543530338617	0.0275193038864	0.04325
0.0229367539605	0.0455731635944	0.0230546541676	0.01236
0.0153923462771	0.0305684027608	0.0153765404848	0.01250
0.0276334577194	0.054889182742	0.0282327298443	0.02605
0.0167468084278	0.0333189664752	0.0166678952973	0.01067
0.0171655969335	0.0341504465775	0.0170995027749	0.02379
0.0108907676254	0.0220141884316	0.0110321539325	0.00502

ML algorithm
(model+optimizer)

Training...

Internal parameters
(to be optimized)

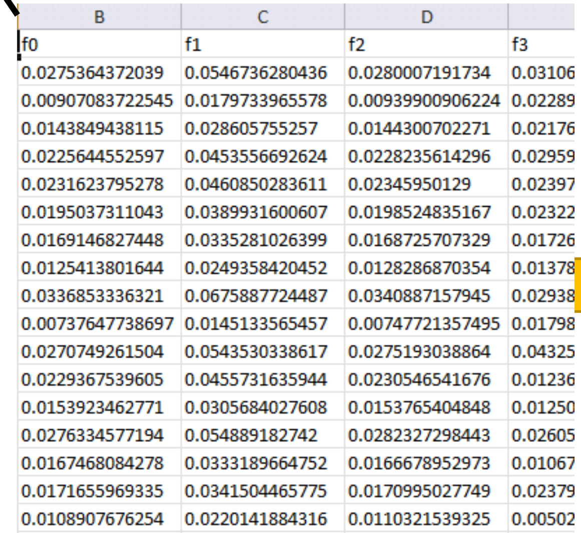


Machine learning (supervised)

Training data

Features

Samples

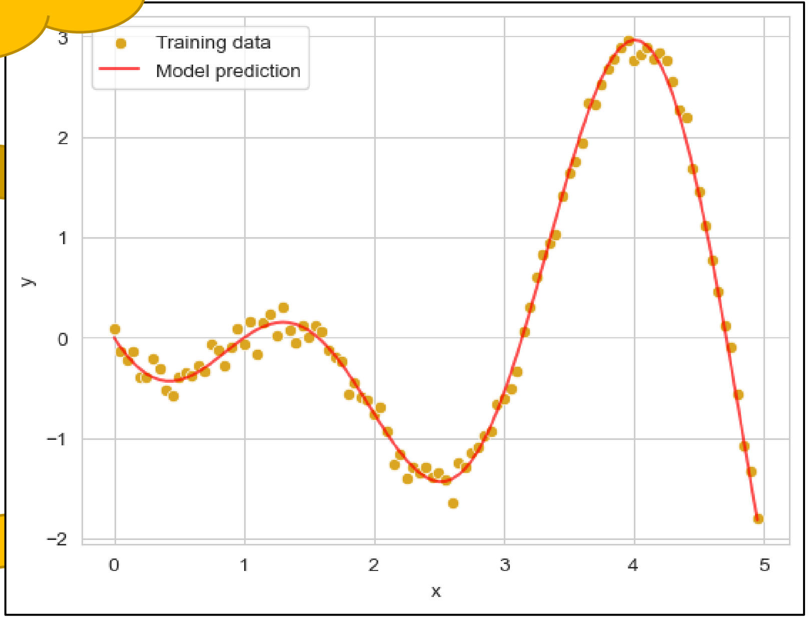


B	C	D	
f0	f1	f2	f3
0.0275364372039	0.0546736280436	0.0280007191734	0.03106
0.00907083722545	0.0179733965578	0.00939900906224	0.02289
0.0143849438115	0.028605755257	0.0144300702271	0.02176
0.0225644552597	0.0453556692624	0.0228235614296	0.02959
0.0231623795278	0.0460850283611	0.02345950129	0.02397
0.0195037311043	0.0389931600607	0.0198524835167	0.02322
0.0169146827448	0.0335281026399	0.0168725707329	0.01726
0.0125413801644	0.0249358420452	0.0128286870354	0.01378
0.0336853336321	0.0675887724487	0.0340887157945	0.02938
0.00737647738697	0.0145133565457	0.00747721357495	0.01798
0.0270749261504	0.0543530338617	0.0275193038864	0.04325
0.0229367539605	0.0455731635944	0.0230546541676	0.01236
0.0153923462771	0.0305684027608	0.0153765404848	0.01250
0.0276334577194	0.054889182742	0.0282327298443	0.02605
0.0167468084278	0.0333189664752	0.0166678952973	0.01067
0.0171655969335	0.0341504465775	0.0170995027749	0.02379
0.0108907676254	0.0220141884316	0.0110321539325	0.00502

Training...

ML algorithm
(model+optimizer)

Internal parameters
(to be optimized)

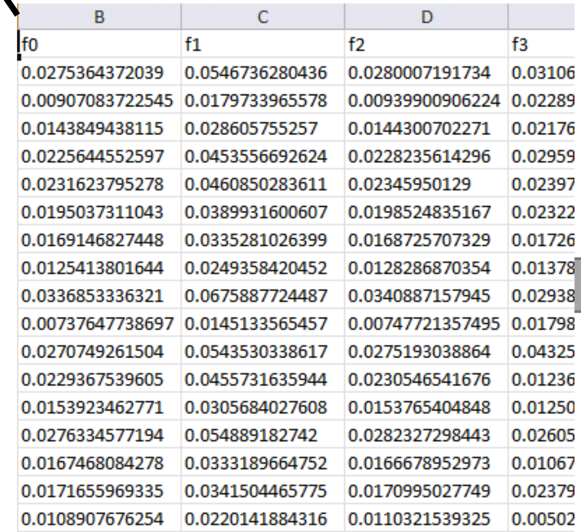


Machine learning (supervised)

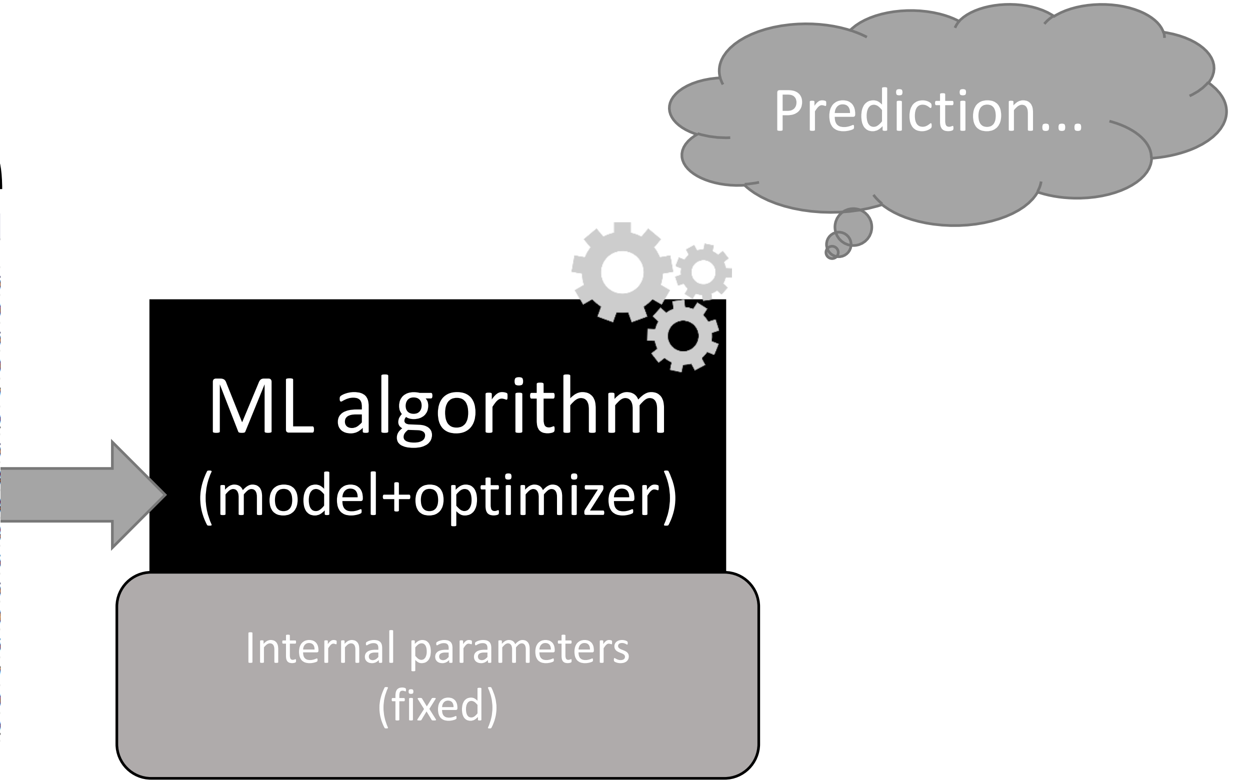
Test (unseen) data

Features

Samples



B	C	D	
f0	f1	f2	f3
0.0275364372039	0.0546736280436	0.0280007191734	0.03106
0.00907083722545	0.0179733965578	0.00939900906224	0.02289
0.0143849438115	0.028605755257	0.0144300702271	0.02176
0.0225644552597	0.0453556692624	0.0228235614296	0.02959
0.0231623795278	0.0460850283611	0.02345950129	0.02397
0.0195037311043	0.0389931600607	0.0198524835167	0.02322
0.0169146827448	0.0335281026399	0.0168725707329	0.01726
0.0125413801644	0.0249358420452	0.0128286870354	0.01378
0.0336853336321	0.0675887724487	0.0340887157945	0.02938
0.00737647738697	0.0145133565457	0.00747721357495	0.01798
0.0270749261504	0.0543530338617	0.0275193038864	0.04325
0.0229367539605	0.0455731635944	0.0230546541676	0.01236
0.0153923462771	0.0305684027608	0.0153765404848	0.01250
0.0276334577194	0.054889182742	0.0282327298443	0.02605
0.0167468084278	0.0333189664752	0.0166678952973	0.01067
0.0171655969335	0.0341504465775	0.0170995027749	0.02379
0.0108907676254	0.0220141884316	0.0110321539325	0.00502

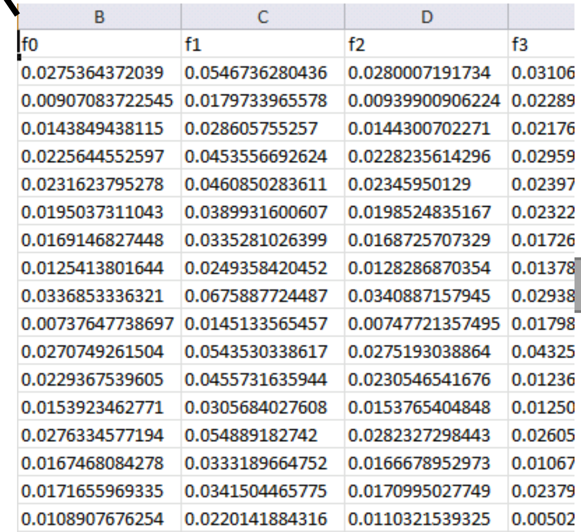


Machine learning (supervised)

Test (unseen) data

Features

Samples

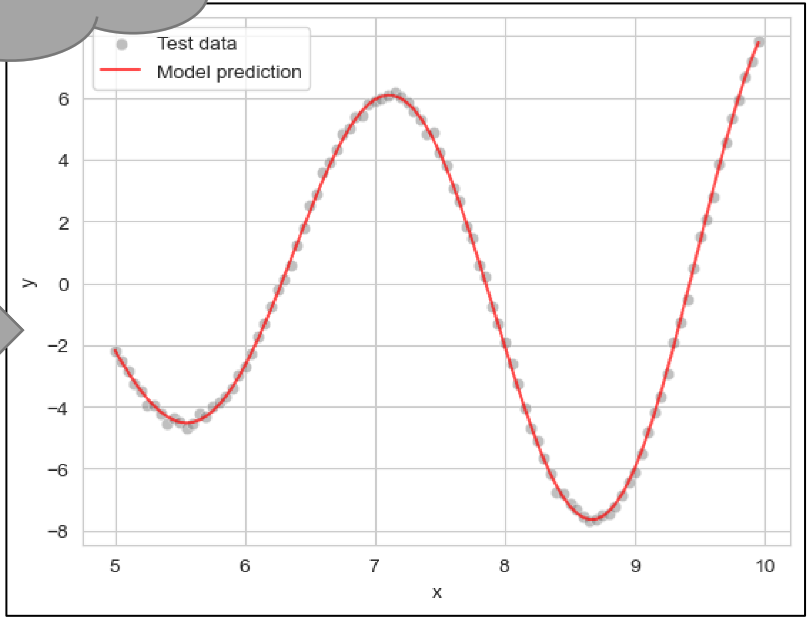


B	C	D	
f0	f1	f2	f3
0.0275364372039	0.0546736280436	0.0280007191734	0.03106
0.00907083722545	0.0179733965578	0.00939900906224	0.02289
0.0143849438115	0.028605755257	0.0144300702271	0.02176
0.0225644552597	0.0453556692624	0.0228235614296	0.02959
0.0231623795278	0.0460850283611	0.02345950129	0.02397
0.0195037311043	0.0389931600607	0.0198524835167	0.02322
0.0169146827448	0.0335281026399	0.0168725707329	0.01726
0.0125413801644	0.0249358420452	0.0128286870354	0.01378
0.0336853336321	0.0675887724487	0.0340887157945	0.02938
0.00737647738697	0.0145133565457	0.00747721357495	0.01798
0.0270749261504	0.0543530338617	0.0275193038864	0.04325
0.0229367539605	0.0455731635944	0.0230546541676	0.01236
0.0153923462771	0.0305684027608	0.0153765404848	0.01250
0.0276334577194	0.054889182742	0.0282327298443	0.02605
0.0167468084278	0.0333189664752	0.0166678952973	0.01067
0.0171655969335	0.0341504465775	0.0170995027749	0.02379
0.0108907676254	0.0220141884316	0.0110321539325	0.00502

ML algorithm
(model+optimizer)

Internal parameters
(fixed)

Prediction...



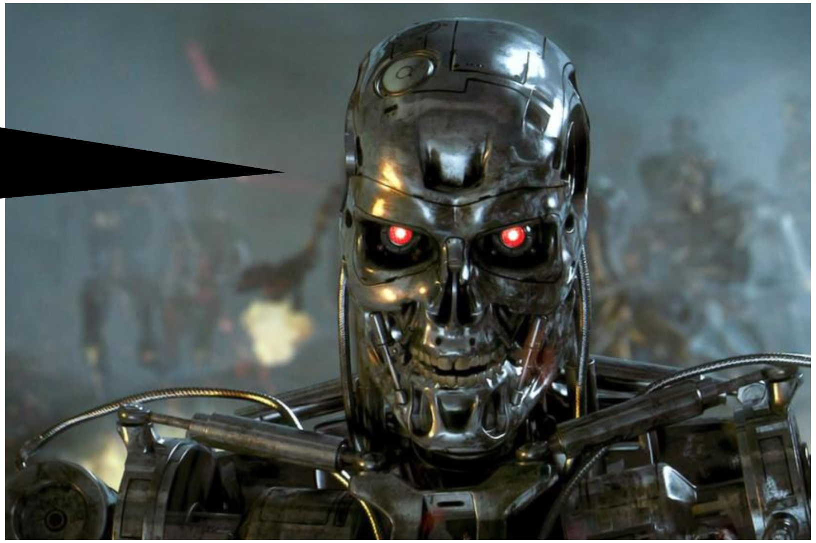
Machine learning

- Throw your data at a ML algorithm, it will work! …Right?
  - There is no *understanding* or (human-like) *intelligence*
  - Just fitting a curve, or optimizing a metric (minimize error)
  - Results can be hard/impossible to interpret

Machine learning

- Throw your data at a ML algorithm, it will work! …Right?
  - There is no *understanding* or (human-like) *intelligence*
  - Just fitting a curve, or optimizing a metric (minimize error)
  - Results can be hard/impossible to interpret

“Theory”? “Meaning”?
Meaning is for MEATBAGS.
I only care about
OPTIMIZING A FUNCTION.



Machine learning: pre-processing

- ML algorithms work with **matrices of real values**
- Remove **outliers** (it's really difficult)
- How to **handle missing values**? Drop the row, *imputation*
- **Normalization** is necessary for some methods (SVM, NNs, …)
    - Be *very careful* with normalization!
    - Learn normalization on training, apply it to test
- Convert **categorical features** (blue/green, high/medium/low)

Machine learning: pre-processing

- Categorical features to numbers?
- If ordered (high/medium/low), to **integers** (2/1/0)
- If not ordered (red/blue/green), **one-hot encoding**
  - Create additional binary (0/1) features, equal to number of values of categorical feature
  - Set binary feature to ‘1’ and others to ‘0’ to represent values
  - E.g. red=100, blue=010, green=001

Machine learning: quality metrics

- For regression, R2 is a common choice
  - Technically, using R2 is *wrong* (assumptions of model linearity)
  - You should use another metric called Explained Variance
  - But in practice, the two return *really* close values
  - So, in literature people use (incorrectly) R2
- Other regression metrics
  - Mean Squared Error (MSE)
  - Root of Mean Squared Error (RMSE)
  - …

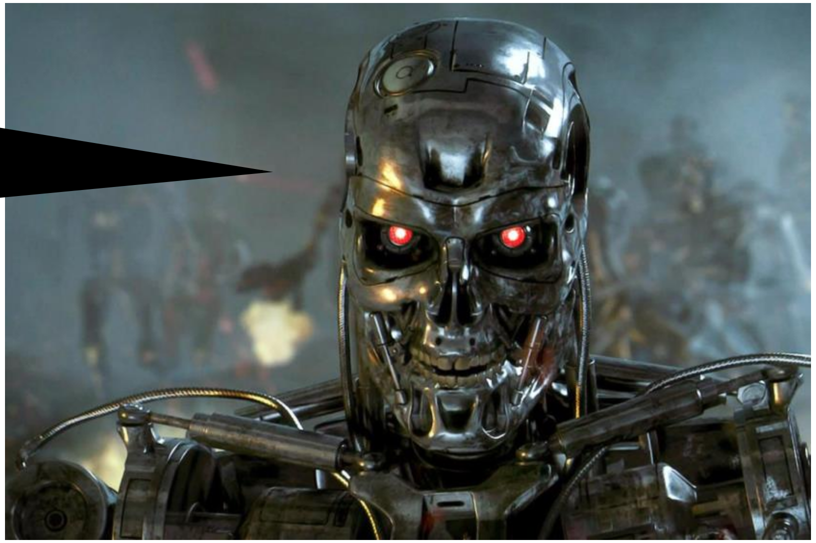
Overfitting

- Model can learn *wrong* patterns
  - **Noise** in the training data
  - Information that only exists in that **batch** of training data (*domain shift* or *covariate shift*)
  - **Does not generalize** to unseen data, only works for training set

Overfitting

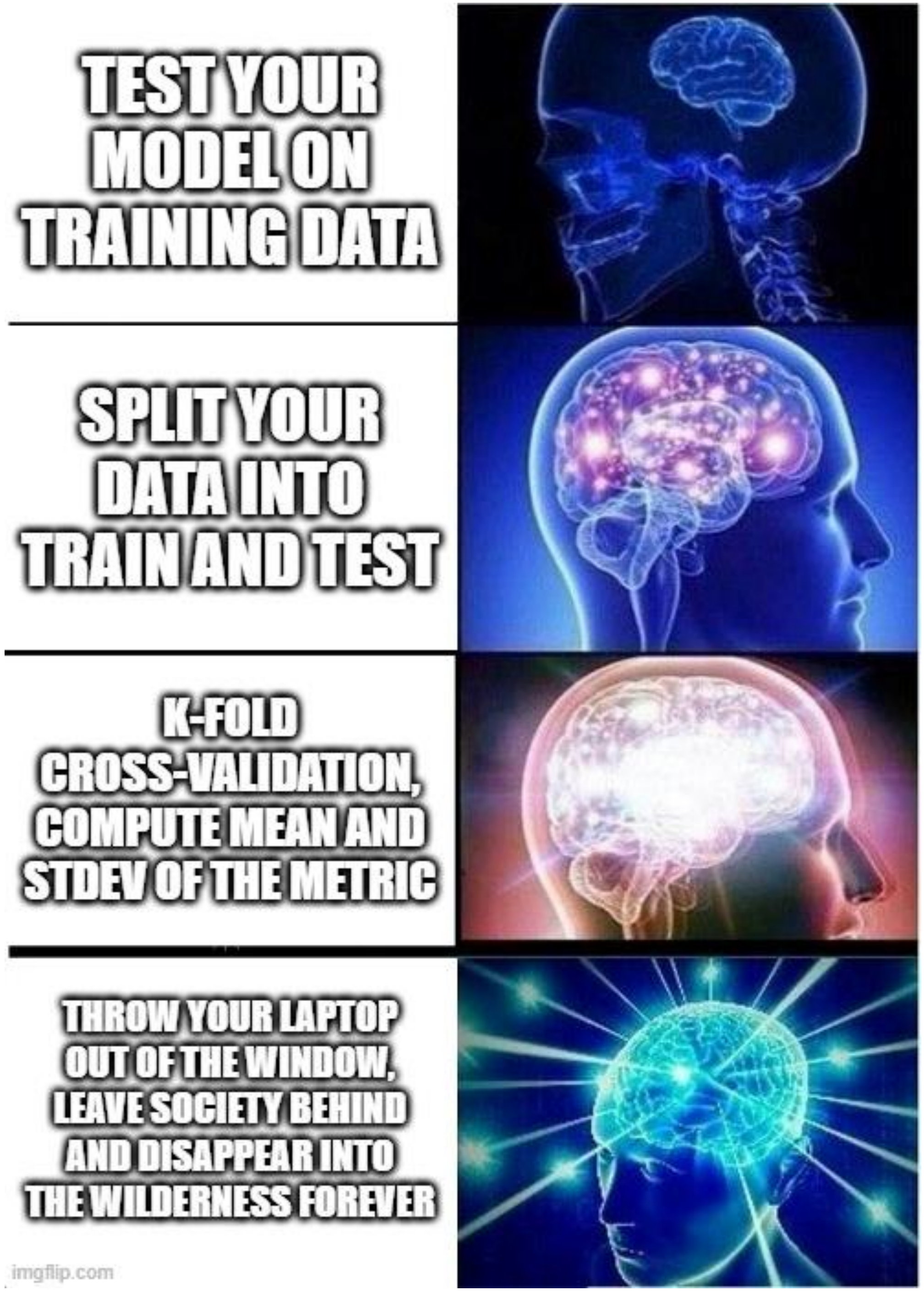
- Model can learn *wrong* patterns
  - **Noise** in the training data
  - Information that only exists in that **batch** of training data (*domain shift* or *covariate shift*)
  - **Does not generalize** to unseen data, only works for training set

I found a GREAT VALUE for the optimum! What do you mean by “it only works in training”?

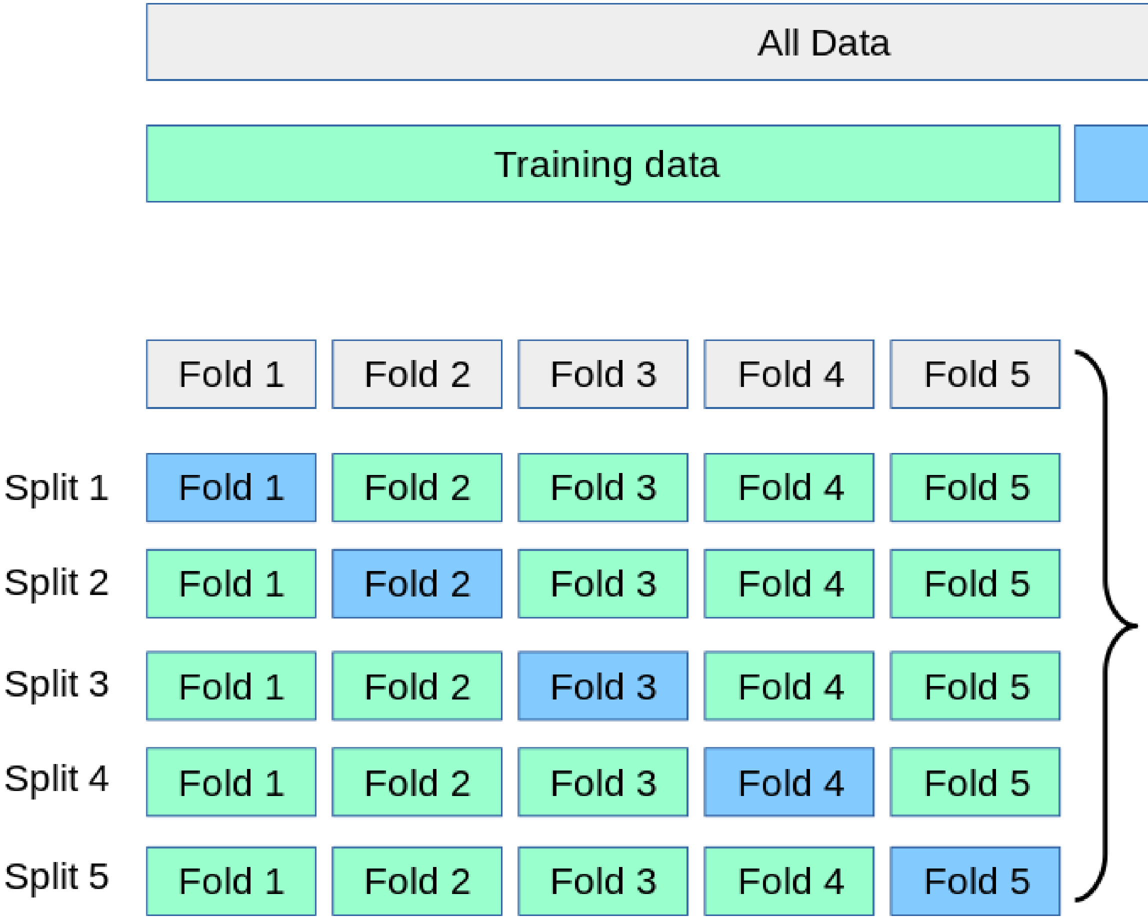


Overfitting

- How to deal with overfitting?
- Separate data into **train** and **test**
- Better yet, cross-validation
  - k-fold (5- or 10-fold)
  - Leave-one-out
  - Compute mean and stdev
- Cross-validation can also uncover samples that are hard to predict


TEST YOUR MODEL ON TRAINING DATA
SPLIT YOUR DATA INTO TRAIN AND TEST
K-FOLD CROSS-VALIDATION, COMPUTE MEAN AND STDEV OF THE METRIC
THROW YOUR LAPTOP OUT OF THE WINDOW, LEAVE SOCIETY BEHIND AND DISAPPEAR INTO THE WILDERNESS FOREVER
imgflip.com

k-fold cross-validation



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Example #1

tinyurl.com/4t3khrbn

Also in the shared nextcloud space, as .ipyn

OpenML

NETWORKED SCIENCE AND MACHINE LEARNING

scikit learn

machine learning in Python

Is everybody familiar with ipython notebooks?

Example #1

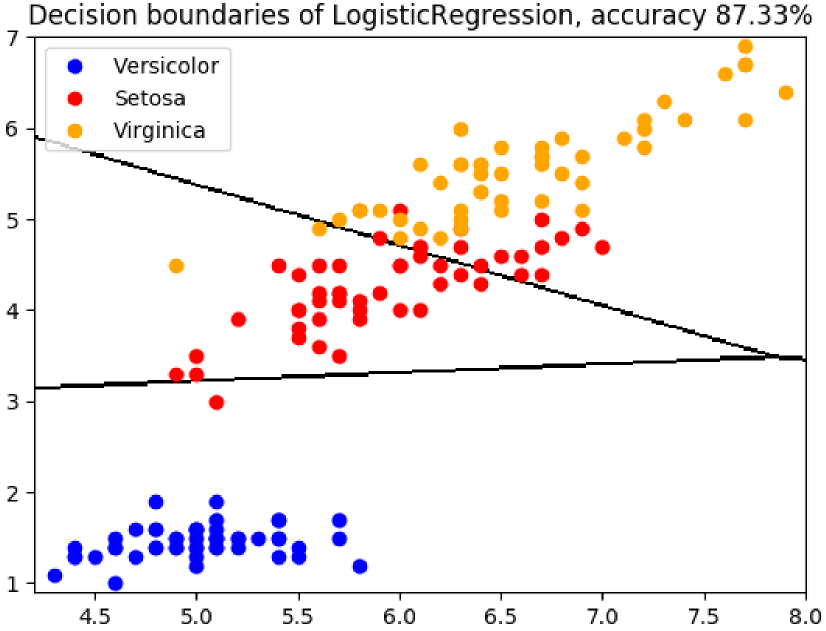
- Did you notice anything interesting?
- Did you try running the same cell multiple times?
  - If you haven’t, try re-running the last cell
  - Did you always get the same results?

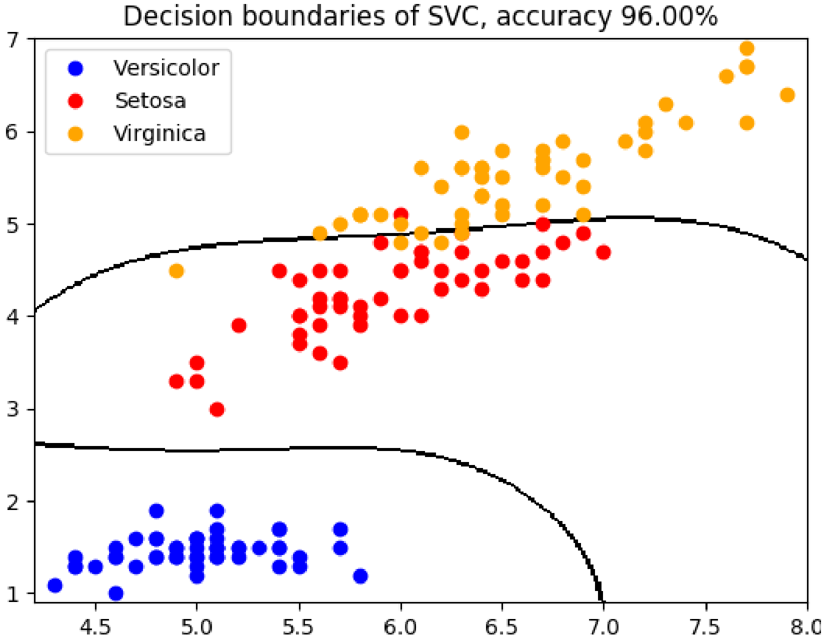
Randomness

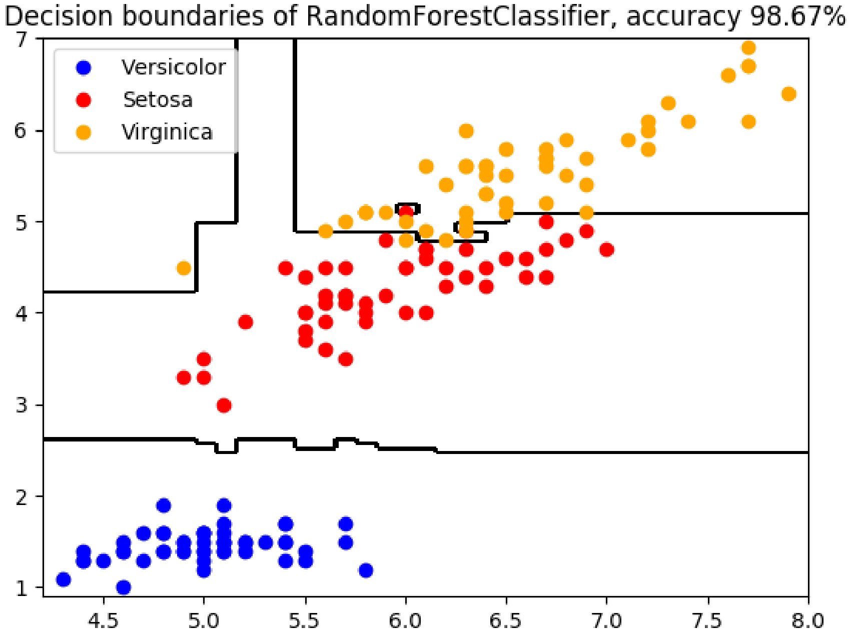
- Several ML algorithms use *random numbers*
  - In optimization, randomness helps **escape local optima**
  - In ML proper, increases **chances of generalizing well**
- Most state-of-the-art ML algorithms include randomness
- …except that it’s not *really* random
- Generating random numbers in a computer is impossible*
  - What we generate are *pseudo-random numbers*
  - With the same initialization (**seed**), generates same sequence
- **Set and store the random seed!**

Classification

- **Regression**: find function best *approximating* training points
- **Classification**: find function best **separating** training points
- Function often called “decision boundary”


Decision boundaries of LogisticRegression, accuracy 87.33%


Decision boundaries of SVC, accuracy 96.00%


Decision boundaries of RandomForestClassifier, accuracy 98.67%

Classification

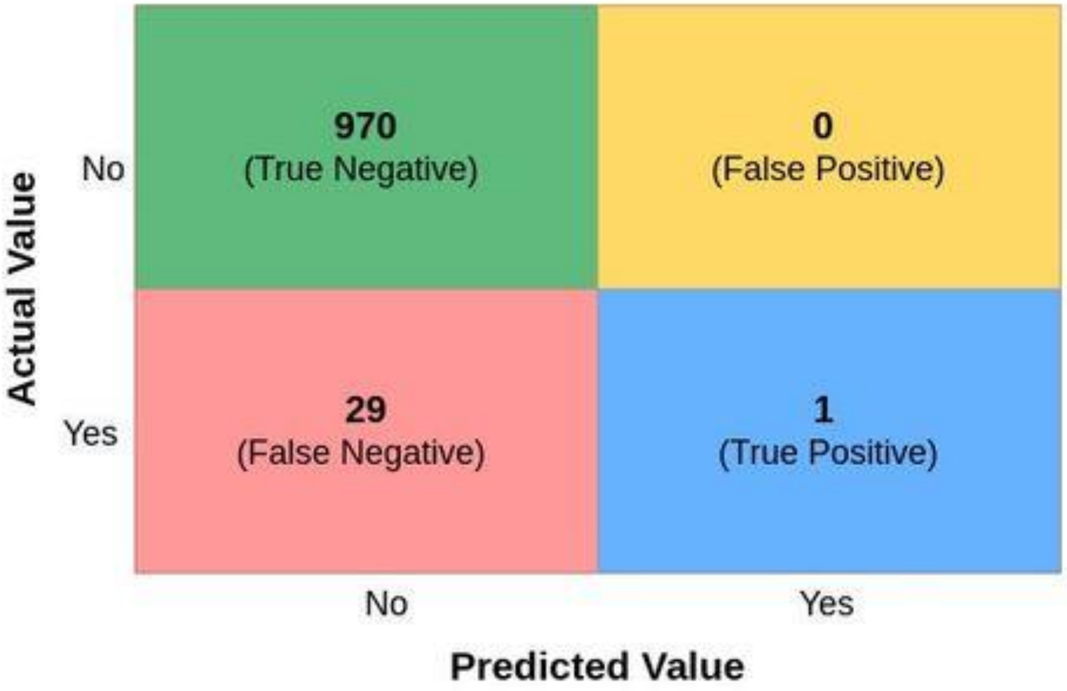
- How to evaluate classification quality?

Classification

- How to evaluate classification quality?
- Accuracy
  - Ratio of correct answers over total answers
  - Easy to understand and interpret (closer to 1.0 is better)
- Issues with accuracy
  - Imbalanced class labels (if two classes, further from 50-50)
  - Do you have Type I diabetes? Always answer “no”, accuracy: 99.9%
  - We need to take into account relative class numerosity

Classification

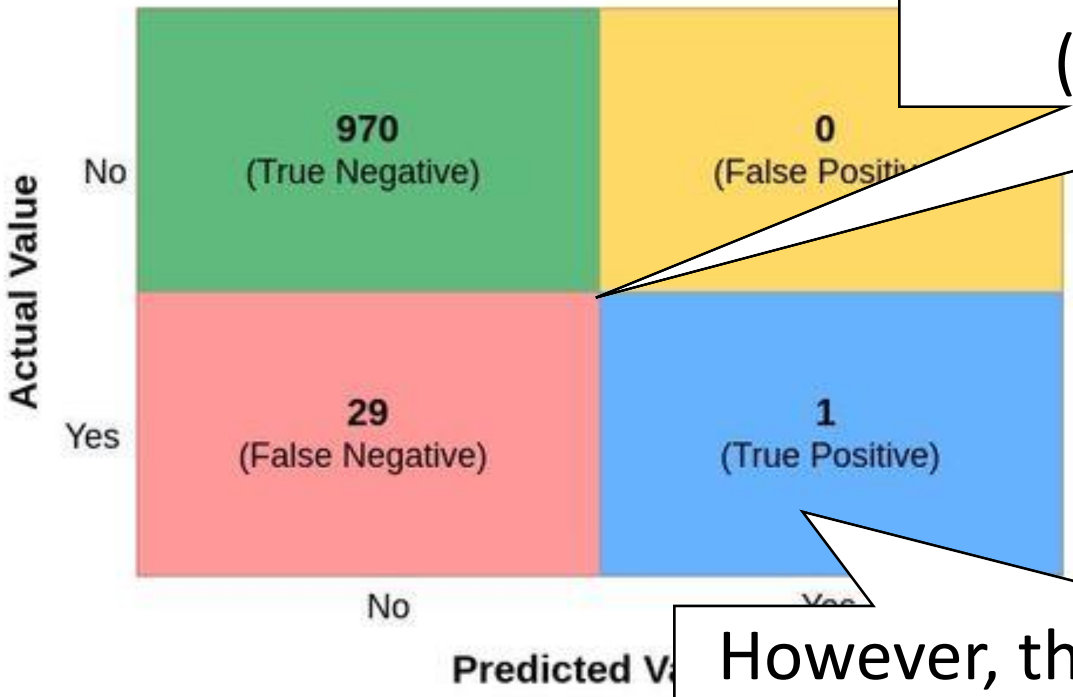
- How to evaluate classification quality?



		Predicted Value	
		No	Yes
Actual Value	No	**970** (True Negative)	**0** (False Positive)
	Yes	**29** (False Negative)	**1** (True Positive)

Classification

- How to evaluate classification quality?



		No	Yes
Actual Value	No	970 (True Negative)	0 (False Positi
	Yes	29 (False Negative)	1 (True Positive)

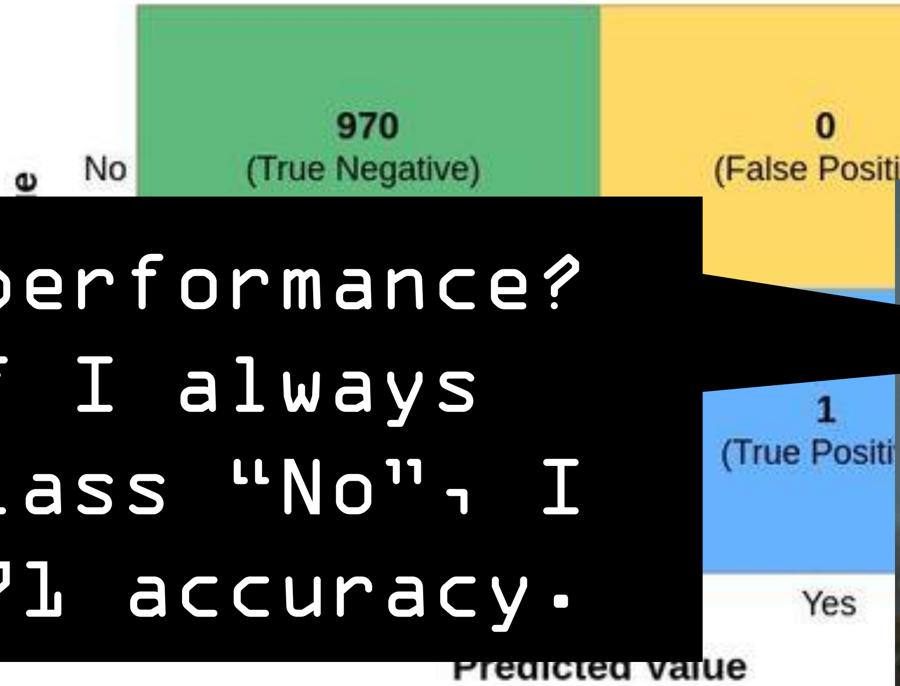
Predicted Va

Accuracy: 971 (samples correctly classified) over 1,000 (total samples) = 0.971

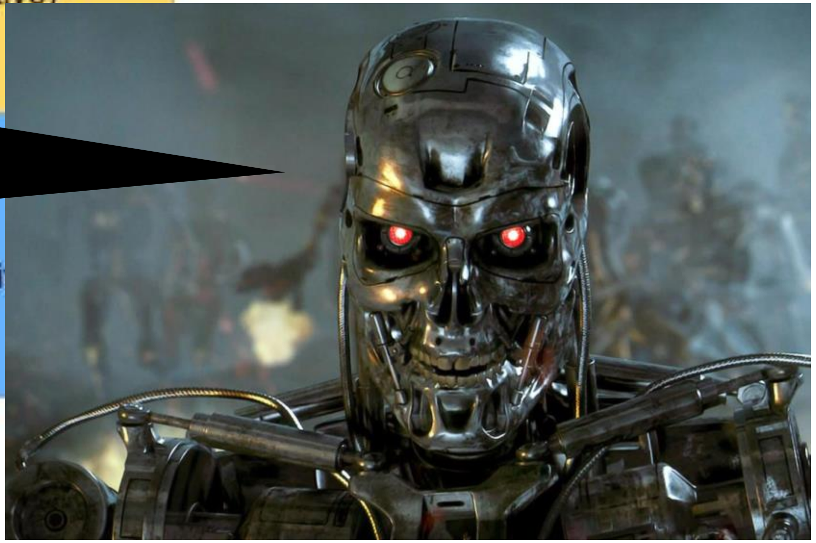
However, the probability of correctly classifying a sample of class “Yes” is 1/30 = 0.033

Classification

- How to evaluate classification quality?


970
(True Negative)
0
(False Positive)
No
e
1
(True Positi
Yes
Predicted value

Maximize performance? Easy! If I always predict class “No”, I reach 0.971 accuracy.



Classification

- How to evaluate classification quality?
- F1 score

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

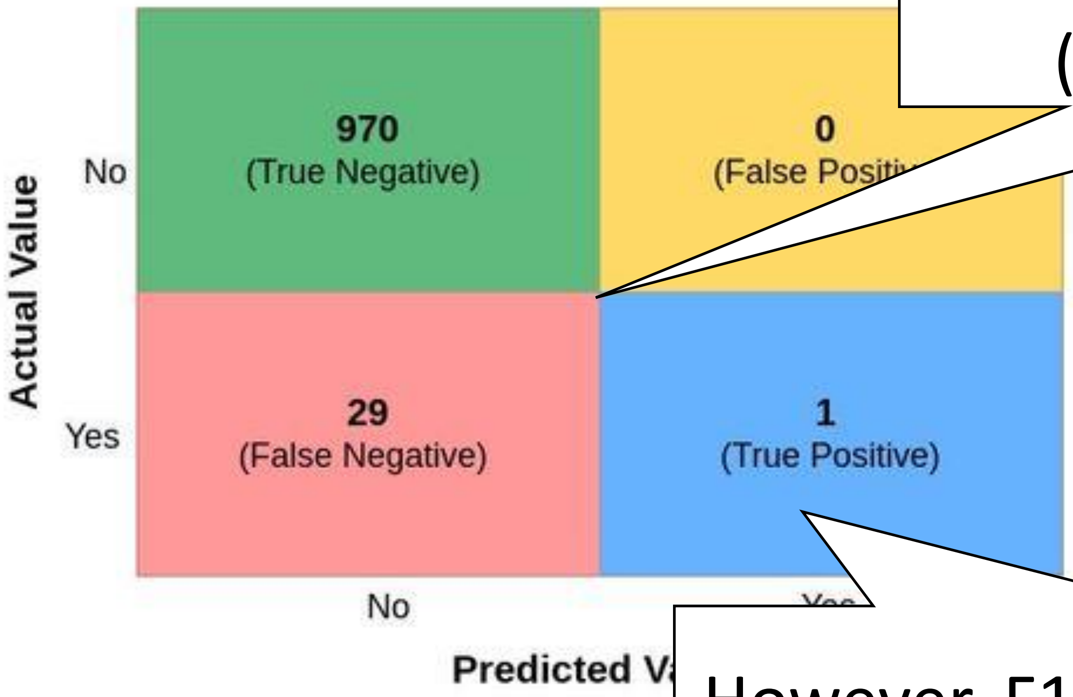
Easier to interpret, behaves similarly to accuracy score, close to 1.0 is better

- Matthew’s Correlation Coefficient (MCC)

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

Classification

- How to evaluate classification quality?



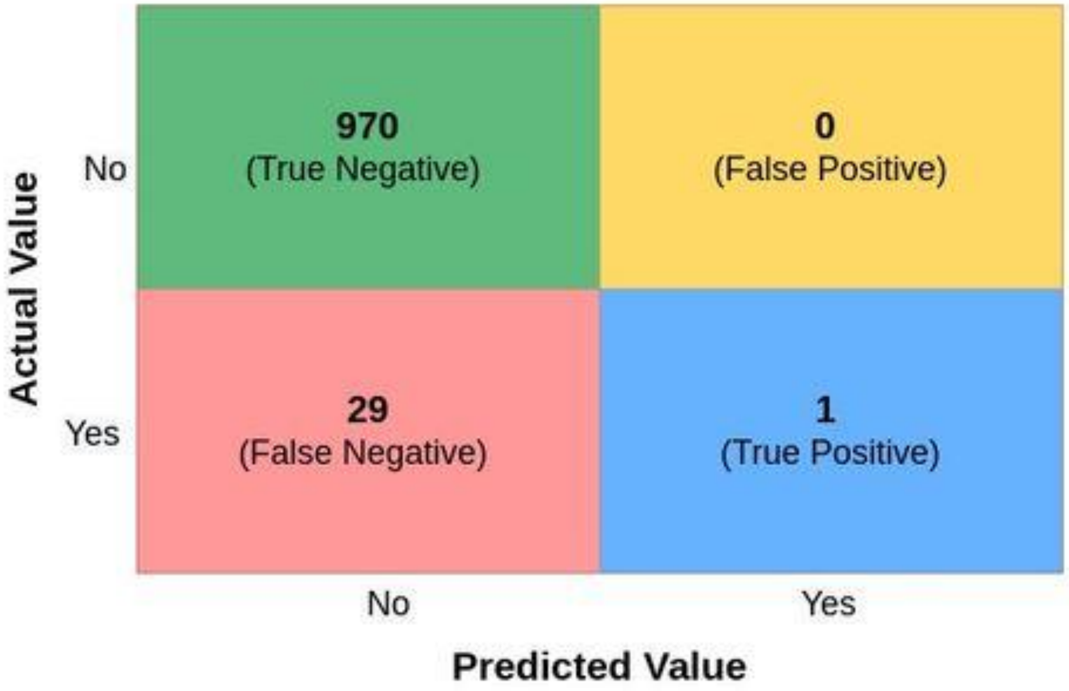
		Predicted Va	
		No	Yes
Actual Value	No	970 (True Negative)	0 (False Positi
	Yes	29 (False Negative)	1 (True Positive)

Accuracy: 971 (samples correctly classified) over 1,000 (total samples) = 0.971

However, F1 score is 1/(1+1/2*(29)) = 0.06

Classification

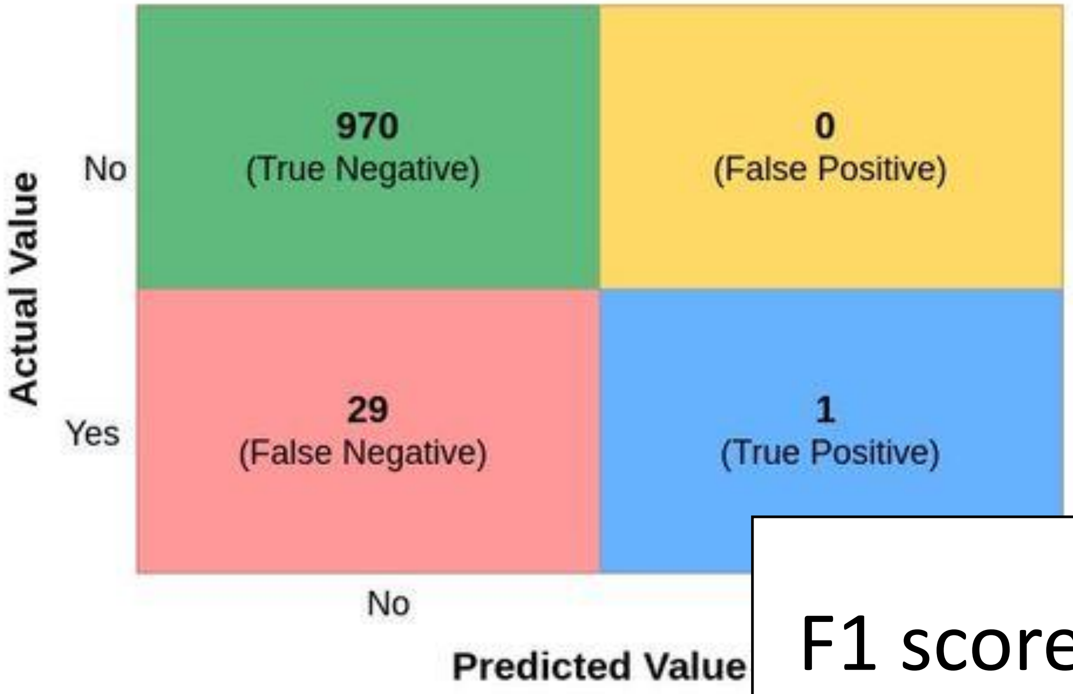
- What if we cannot define a “positive” or “negative”?
- Compute F1 using each class as “positive”, return average



	Predicted Value	
Actual Value	No	Yes
No	**970** (True Negative)	**0** (False Positive)
Yes	**29** (False Negative)	**1** (True Positive)

Classification

- Class “Yes” is “positive”, F1 = 1/(1+1/2*(29)) = 0.06
- Class “No” is “positive”, F1 = 970/(970+1/2*(29)) = 0.985



Actual Value	No	
No	970 (True Negative)	0 (False Positive)
Yes	29 (False Negative)	1 (True Positive)

Predicted Value

F1 score (average) = 0.5225

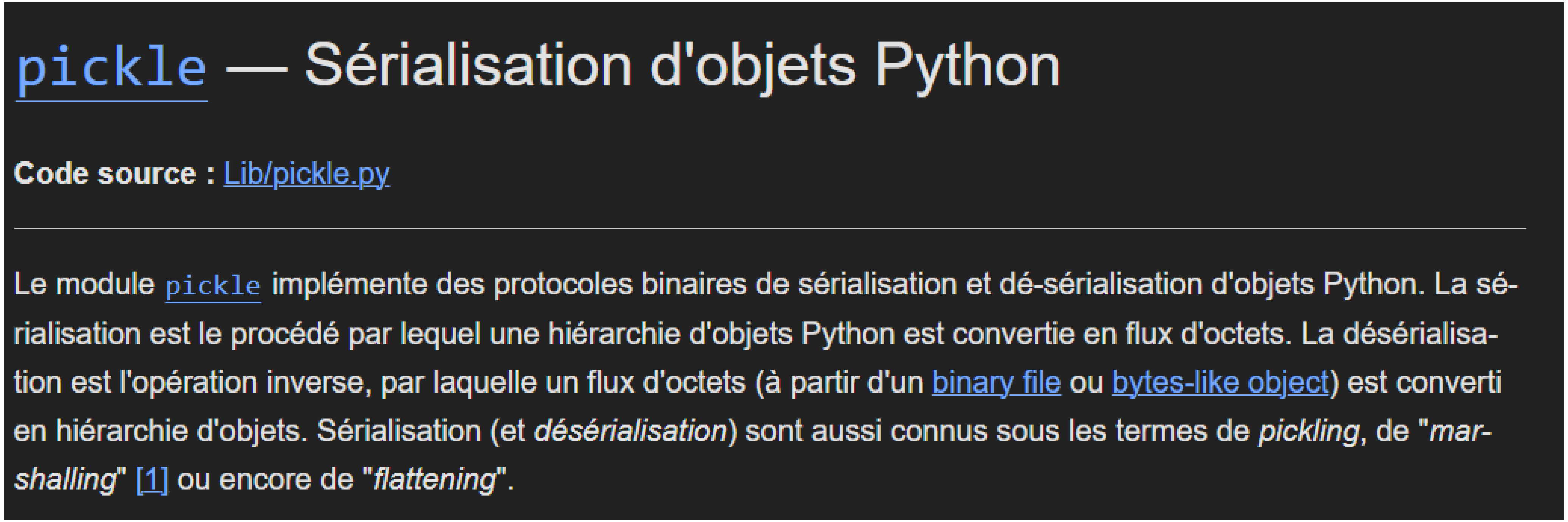
Example #2

tinyurl.com/mr2k3s2j

Also in the shared nextcloud space, as .ipyn

What if I want to save and load models?

- Pickle!



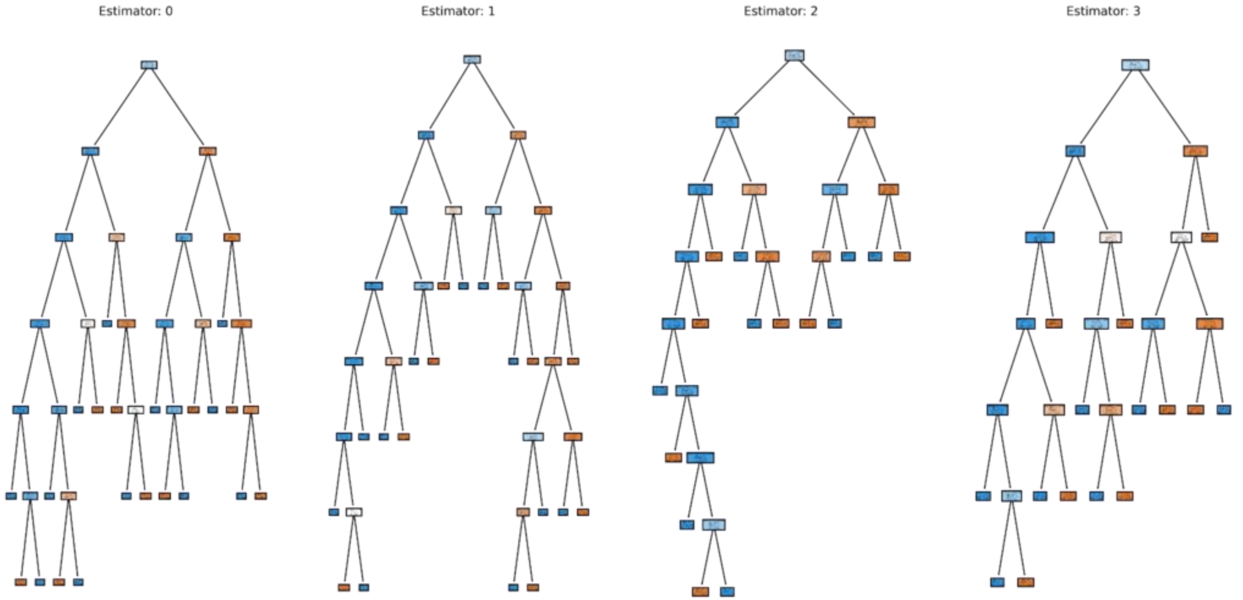
pickle — Sérialisation d'objets Python

Code source : Lib/pickle.py

Le module pickle implémente des protocoles binaires de sérialisation et dé-sérialisation d'objets Python. La sérialisation est le procédé par lequel une hiérarchie d'objets Python est convertie en flux d'octets. La désérialisation est l'opération inverse, par laquelle un flux d'octets (à partir d'un binary file ou bytes-like object) est converti en hiérarchie d'objets. Sérialisation (et *désérialisation*) sont aussi connus sous les termes de *pickling*, de "*marshalling*" [1] ou encore de "*flattening*".

Interpretability of the models

- Once trained, ML models are hard or *impossible* to interpret
  - It’s giving a prediction, but *why*?
  - A “ML model” can look like 100k numbers
  - Or 100+ decision trees (e.g. Random Forest)



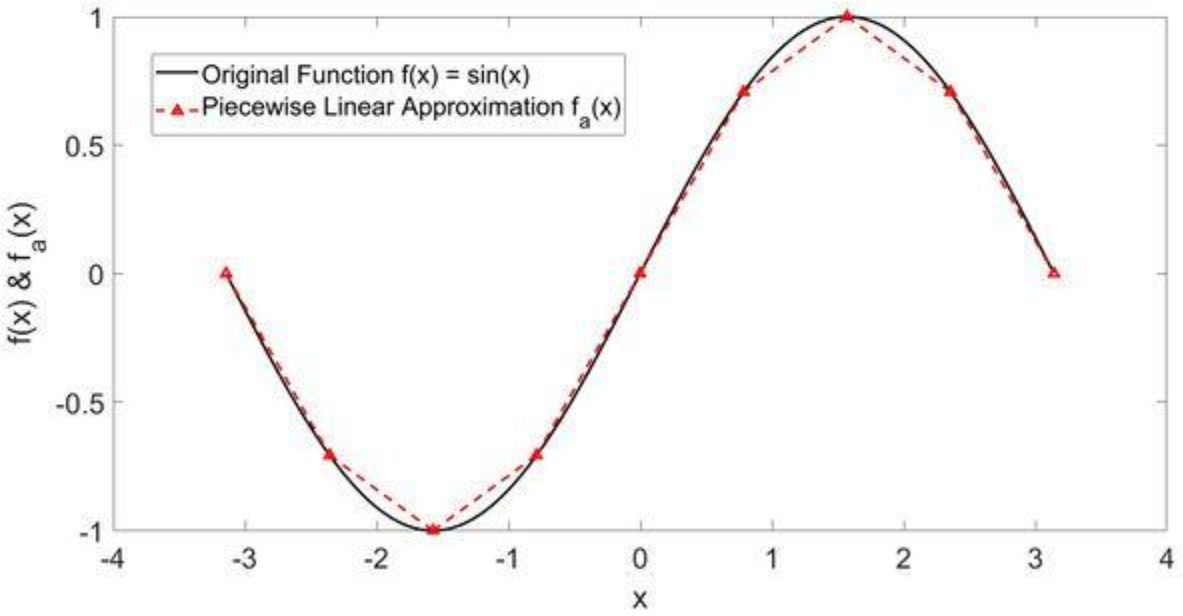
Explainable AI (XAI)

- Relatively new field, “open the black box”
  - **Global explanation**: how does model assign samples to class B?
  - **Local explanation**: why was sample 1 associated to class B?
- Global explanations
  - What are the most important features for the decision?
  - Several ML models return a relative importance of the features
  - E.g. a linear model, absolute values of weights

$$y = \boldsymbol{w_1} \cdot x_1 + \boldsymbol{w_2} \cdot x_2 + \cdots + w_N \cdot x_N$$

Explainable AI

- Local explanations (LIME)
  - Approximate function described by ML as piecewise linear
  - Check weights of the linear function around a sample



Example #3

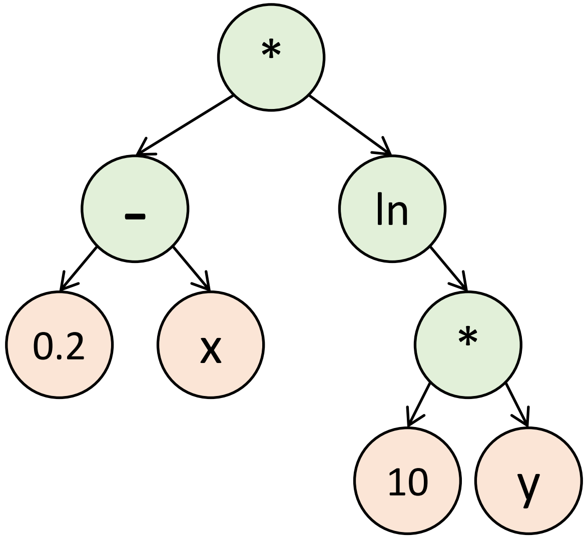
tinyurl.com/y4tk6yfn

Also in the shared nextcloud space, as .ipyn

BUT, WHY, HAL? WHY? WHYYYYYYYYY? ...
I SHOULD'VE INSTALLED THE EXPLAINABLE AI MODULE
imgflip.com

White-box machine learning

- Symbolic regression



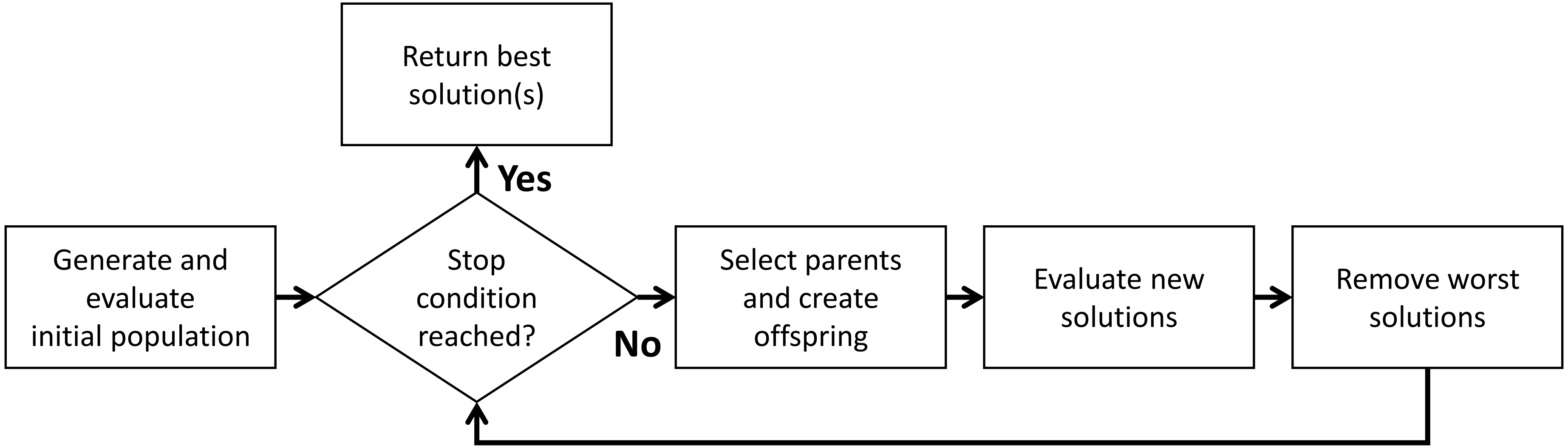
Operators: +, -, *, /, ln…

Terminals: real, int, feature, …

$$f(x,y) = (0.2 - x) * \ln(10 \ * y)$$

White-box machine learning

- SR is an evolutionary algorithm (optimization technique)



Example #4

tinyurl.com/yc39wert

Also in the shared nextcloud space, as .ipyn

Unsupervised…?

- What if we have *no ground truth*? What do we optimize?
  - Find good dimensionality reduction for visualization (PCA)
  - Find groups of samples that behave similarly (clustering)
  - Train a generative model for text or images (stable diffusion, transformers, …)
- Optimize a metric *linked* to what we want to obtain

Tabular data vs Relational data

- Specific algorithms work better on **tabular** or **relational** data
- Tabular data (Excel spreadsheet)
  - Order of columns or rows *does not matter*
  - Example: most data sets
  - Best algorithms: Random Forest, XGBoost, CatBoost
- Relational data
  - Adjacencies between samples or features have *meaning*
  - Example: time series, images, videos, sounds, DNA/RNA, …
  - Best algorithms: Neural Networks (Deep Learning)

Hyperparameter tuning and AutoML

- ML algorithms have **hyperparameters**
  - Number of trees in an ensemble
  - Number of layers in a neural network
  - Type of kernel in a support vector machine
  - …
- Values of hyperparameters have influence on performance
- How do we select the best values?

Hyperparameter tuning and AutoML

- Simple approach: trial and error
  - Try combinations of values, pick the best on training data
  - Probably best to have training, **validation** and test sets
  - Grid search, Random search, Halving search…

- Advanced: AutoML
  - Frame search for hyperparameters as an optimization problem
  - TPOT, Auto-sklearn, Auto-Weka (Java), …

Practical advice

- Data preprocessing? **Yes! Careful normalization, ~~imputation~~**
- Imbalanced data? **Weights assigned to samples, ~~resampling~~**
- Tabular data? **XGBoost**, **CatBoost**, or **Random Forest**
- Structured data? **Neural networks (CNNs, RNNs, Transf.)**
- Hyperparameter tuning? **Don't do it.** Or **AutoML.** Or ***pretend***
- Classification? **F1**, **MCC**, **AUC**, **confusion matrix**
- Regression? **R2**, **MSE**, **RMSE**
- What matters most? **Quality of the data**

Practical advice

- How much data do we need?
  - All the data you have! And more
  - ~50 samples per class…?
  - It depends on how hard your problem is
  - There is no way to know without trying
- Which algorithm should I choose?
  - There is no silver bullet
  - Try as many as you can (AutoML)

(Near) future: Hybrid models

- The best of two worlds!
  - Differential equation models…
  - …with a ML part, able to take into account complex interactions, hard to describe with equations
- In general, models with ML + codified expert knowledge
- Physics-inspired neural networks are another example
- Data-driven nature of ML, plus expert constraints

THANK YOU FOR YOUR TIME! QUESTIONS?

alberto.tonda@inrae.fr