Earth Data Science

Master-level lecture at the institut de physique du globe de Paris with contents from the scikit-learn documentation and the deep learning book of Ian Goodfellow.

`Léonard Seydoux` `Antoine Lucas` `Éléonore Stutzmann`
`Alexandre Fournier` `Geneviève Moguilny`

leonard-seydoux/earth-data-science-public

Goals of the class

- **Identify** data-related problems
- **Define** the problem properly
- **Build** machine-learning solutions
- **Criticize** the scientific literature

Bergen et al. (2019)

REVIEW SUMMARY

GEOPHYSICS

Machine learning for data-driven discovery in solid Earth geoscience

Karianne J. Bergen, Paul A. Johnson, Maarten V. de Hoop, Gregory C. Beroza*

BACKGROUND: The solid Earth, oceans, and atmosphere together form a complex interacting geosystem. Processes relevant to understanding Earth's geosystem behavior range in spatial scale from the atomic to the planetary, and in temporal scale from milliseconds to billions of years. Physical, chemical, and biological processes interact and have substantial influence on this complex geosystem, and humans interact with it in ways that are increasingly consequential to the future of both the natural world and civilization as the finiteness of Earth becomes increasingly apparent and limits on available energy, mineral resources, and fresh water increasingly affect the human condition. Earth is subject to a variety of geohazards that are poorly understood, yet increasingly impactful as our exposure grows through increasing urbanization, particularly in hazard-prone areas. We have a fundamental need to develop the best possible predictive understanding of how the geosystem works, and that understanding must be informed by both the present and the deep past. This understanding will come through the analysis of increasingly large geo-datasets and from computationally intensive simulations, often connected through inverse problems. Geoscientists are faced with the challenge of extracting as much useful information as possible and gaining new insights from these data, simulations, and the interplay between the two. Techniques from the rapidly evolving field of machine learning (ML) will play a key role in this effort.

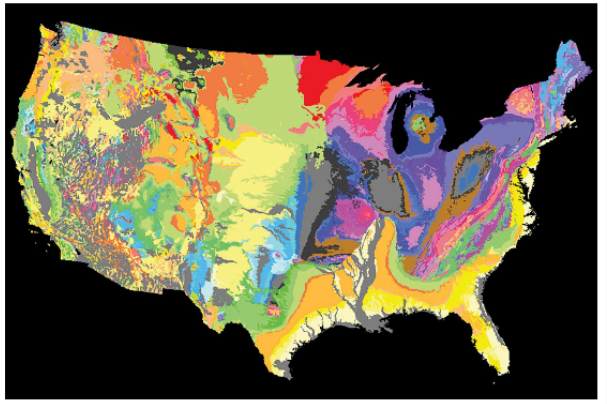
ADVANCES: The confluence of ultrafast computers with large memory, rapid progress in ML algorithms, and the ready availability of large datasets place geoscience at the threshold of dramatic progress. We anticipate that this progress will come from the application of ML across three categories of research effort: (i) automation to perform a complex prediction task that cannot easily be described by a set of explicit commands; (ii) modeling and inverse problems to create a representation that approximates numerical simulations or captures relationships; and (iii) discovery to reveal new and often unanticipated patterns, structures, or relationships. Examples of automation include geologic mapping using remote-sensing data, characterizing the topology of fracture systems to model subsurface transport, and classifying volcanic ash particles to infer eruptive mechanism. Examples of modeling include approximating the viscoelastic response for complex rheology, determining wave speed models directly from tomographic data, and classifying diverse seismic events. Examples of discovery include predicting laboratory slip events using observations of acoustic emissions, detecting weak earthquake signals using similarity search, and determining the connectivity of subsurface reservoirs using groundwater tracer observations.

ON OUR WEBSITE

Read the full article at http://dx.doi.org/10.1126/science.aau0323

OUTLOOK: The use of ML in solid Earth geosciences is growing rapidly, but is still in its early stages and making uneven progress. Much remains to be done with existing datasets from long-standing data sources, which in many cases are largely unexplored. Newer, unconventional data sources such as light detection and ranging (LiDAR), fiber-optic sensing, and crowd-sourced measurements may demand new approaches through both the volume and the character of information that they present.

Practical steps could accelerate and broaden the use of ML in the geosciences. Wider adoption of open-science principles such as open source code, open data, and open access will better position the solid Earth community to take advantage of rapid developments in ML and artificial intelligence. Benchmark datasets and challenge problems have played an important role in driving progress in artificial intelligence research by enabling rigorous performance comparison and could play a similar role in the geosciences. Testing on high-quality datasets produces better models, and benchmark datasets make these data widely available to the research community. They also help recruit expertise from allied disciplines. Close collaboration between geoscientists and ML researchers will aid in making quick progress in ML geoscience applications. Extracting maximum value from geoscientific data will require new approaches for combining data-driven methods, physical modeling, and algorithms capable of learning with limited, weak, or biased labels. Funding opportunities that target the intersection of these disciplines, as well as a greater component of data science and ML education in the geosciences, could help bring this effort to fruition. ■

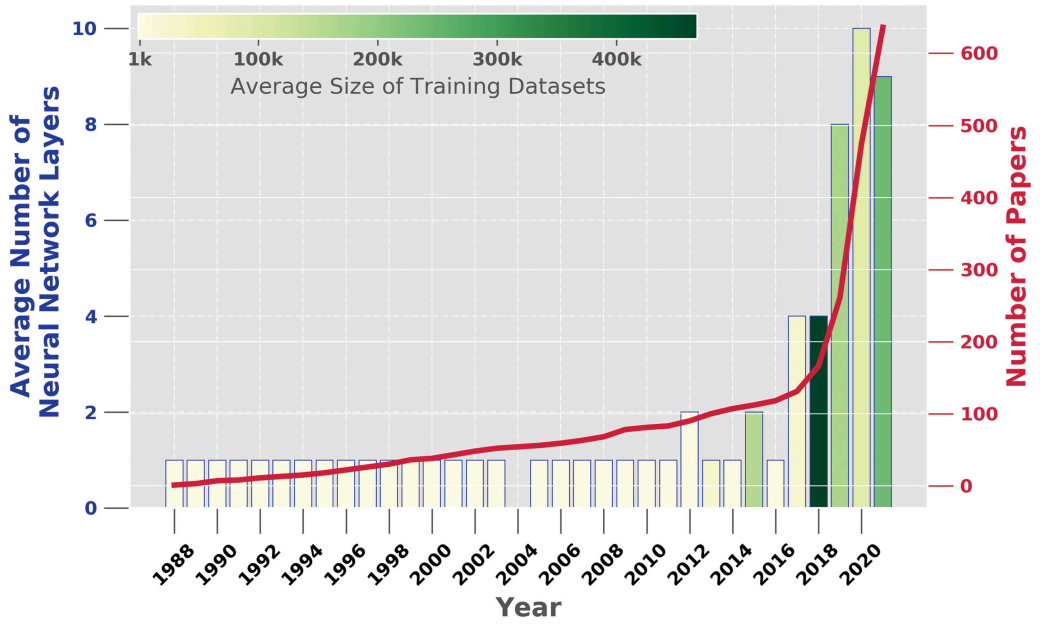


Digital geology. Digital representation of the geology of the conterminous United States. [Geology of the Conterminous United States at 1:2,500,000 scale; a digital representation of the 1974 P. B. King and H. M. Beikman map by P. G. Schruben, R. E. Arndt, W. J. Bawiec]

The list of author affiliations is available in the full article online.
*Corresponding author. Email: beroza@stanford.edu
Cite this article as K. J. Bergen et al., Science 363, eaau0323 (2019). DOI: 10.1126/science.aau0323

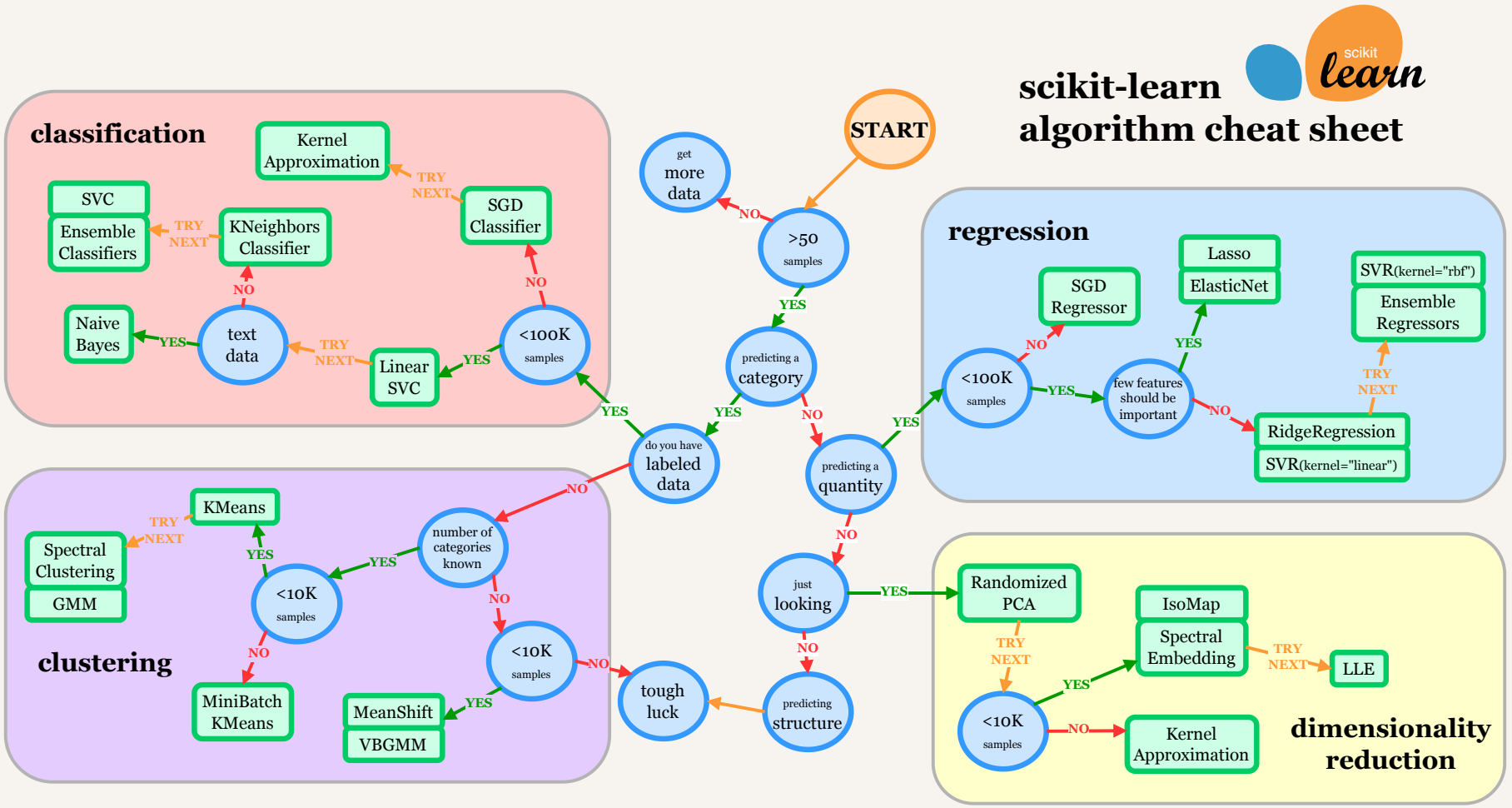
Goals of the class

Keep up with the ongoing pace!



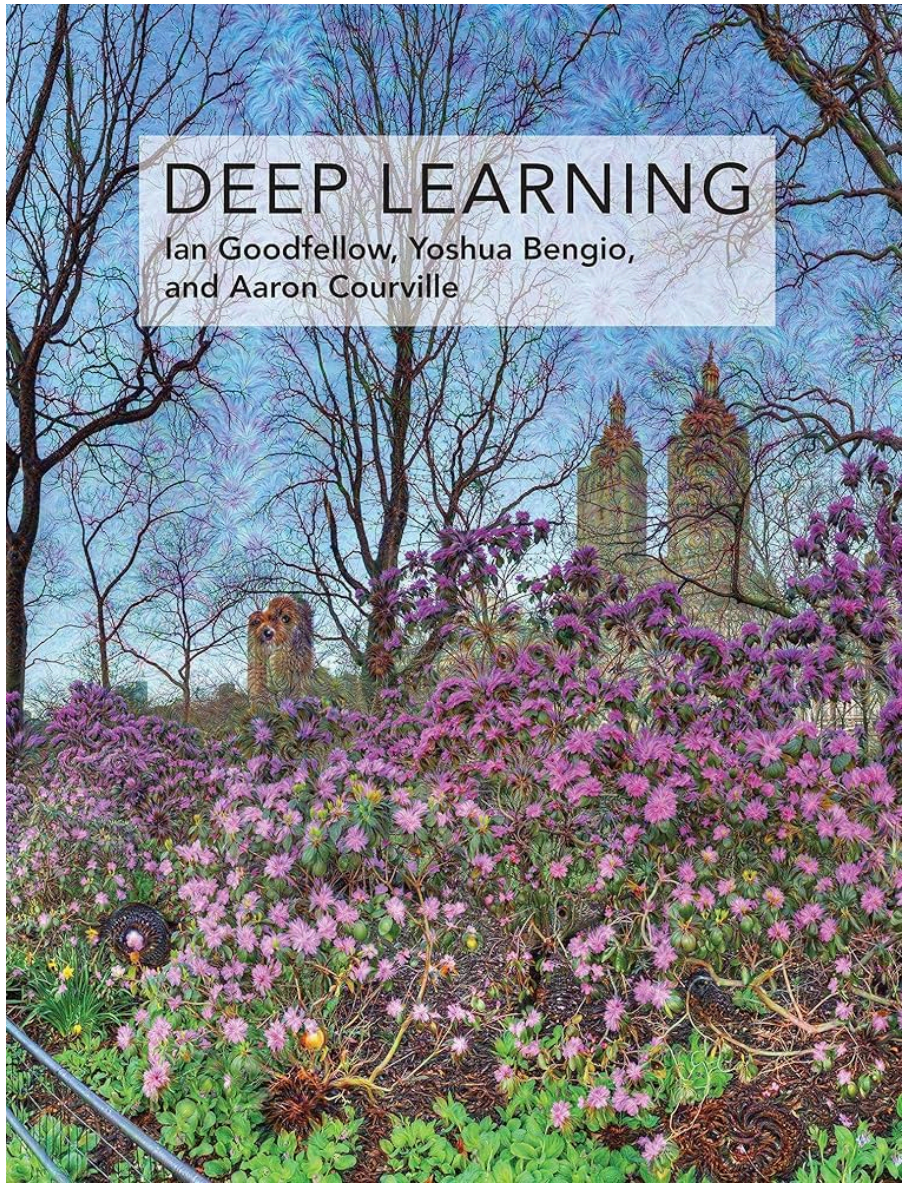
Mousavi et al. (2022)

The scikit-learn library



The deep learning book

- **Historical** aspects
- **From scratch** to deep learning
- **Examples** and exercises
- **Freely** accessible online



1. Introduction

Machine learning for earth science: why, what, and how? How to read papers that use machine learning?

How fast can you describe the following images?

How accurate are those descriptions?

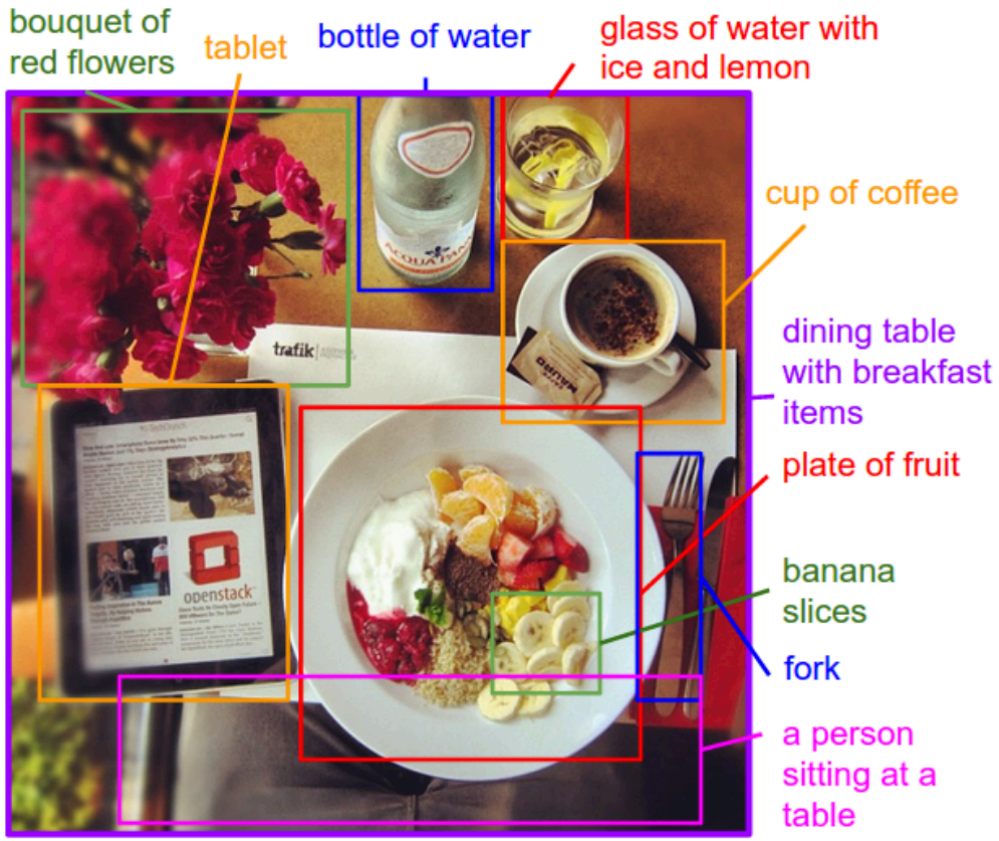
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

From data to knowledge

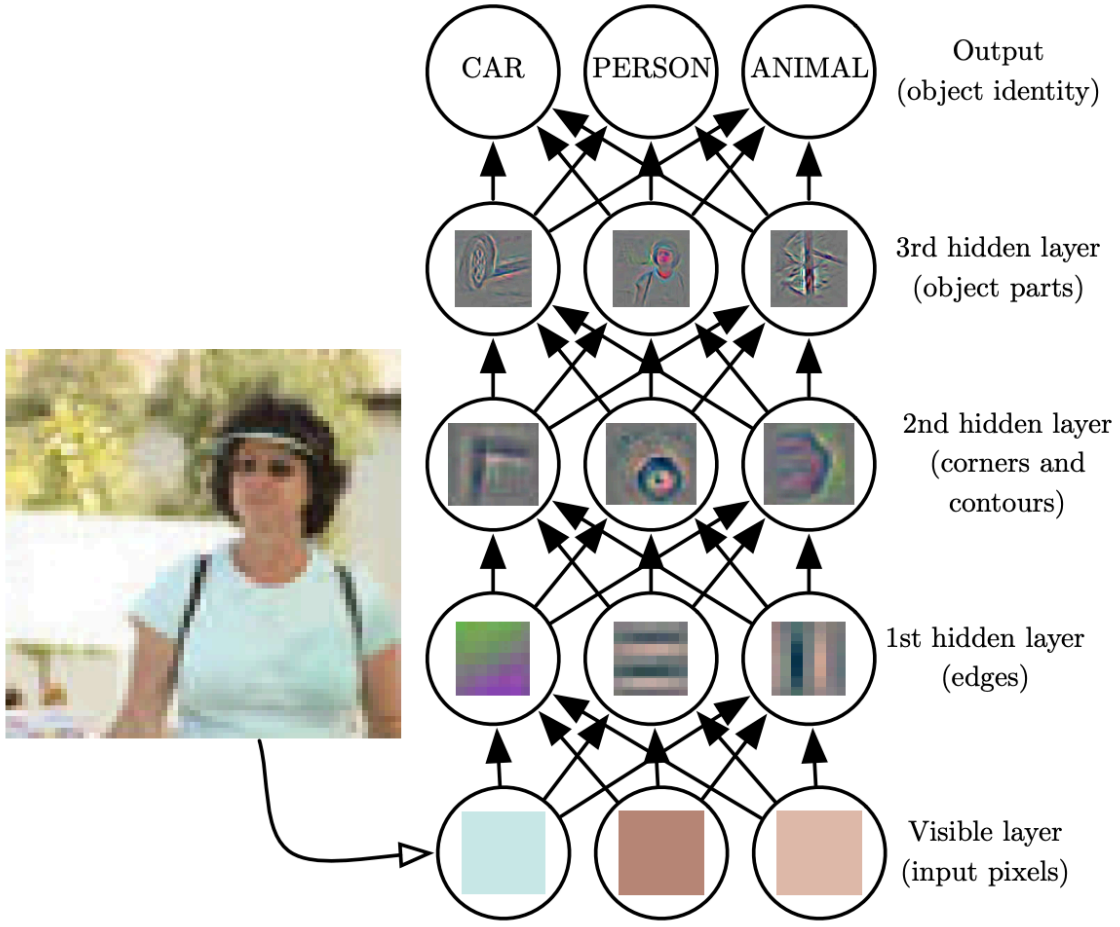
- Identify objects from pixels
- Recognize objects category
- Relate objects
- Sort links by priority
- Generate text out of it



Karpathy & Fei-Fei (2015)

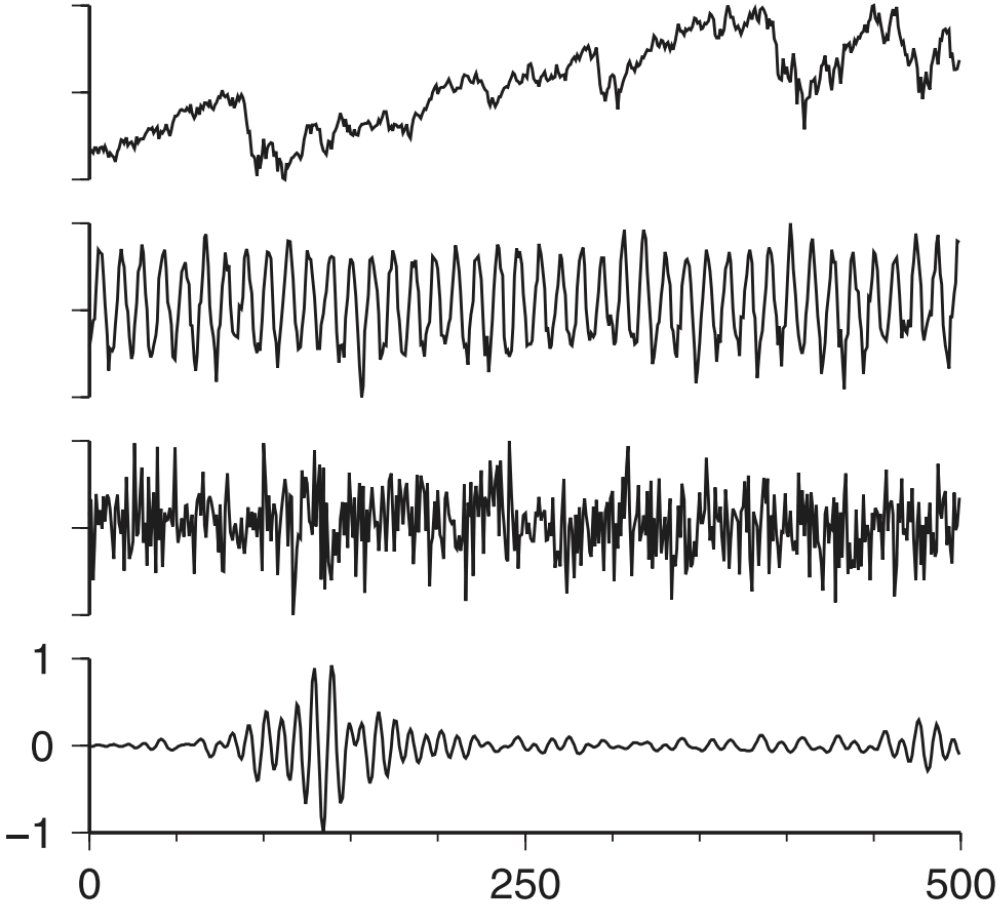
From data to knowledge

- Identify objects from pixels
- Recognize objects category
- Relate objects
- Sort links by priority
- Generate text out of it



Goodfellow et al. (2016)

Can you spot the seismogram?

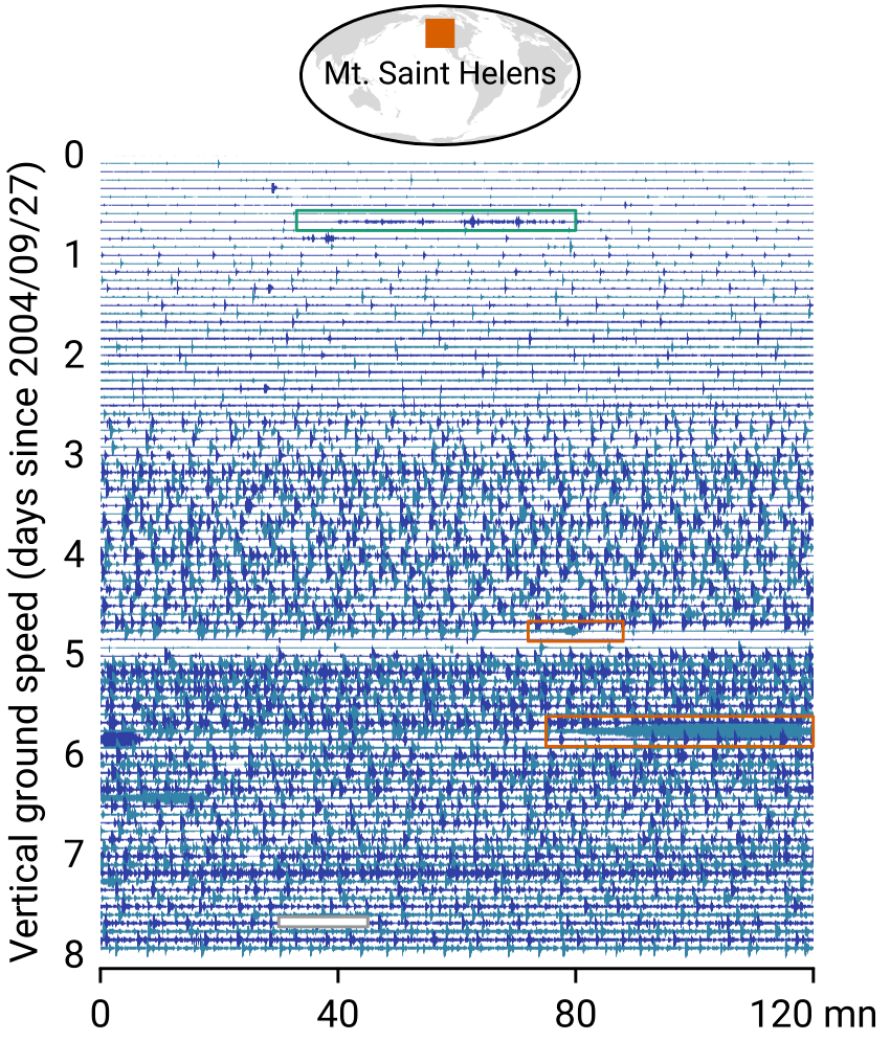


Valentine & Trampert (2012)

Top to bottom: UK stock exchange; Temperature in Central England; Gaussian noise; Long-period seismogram.

Task: seismic events detection and classification

Most humans can pinpoint events.

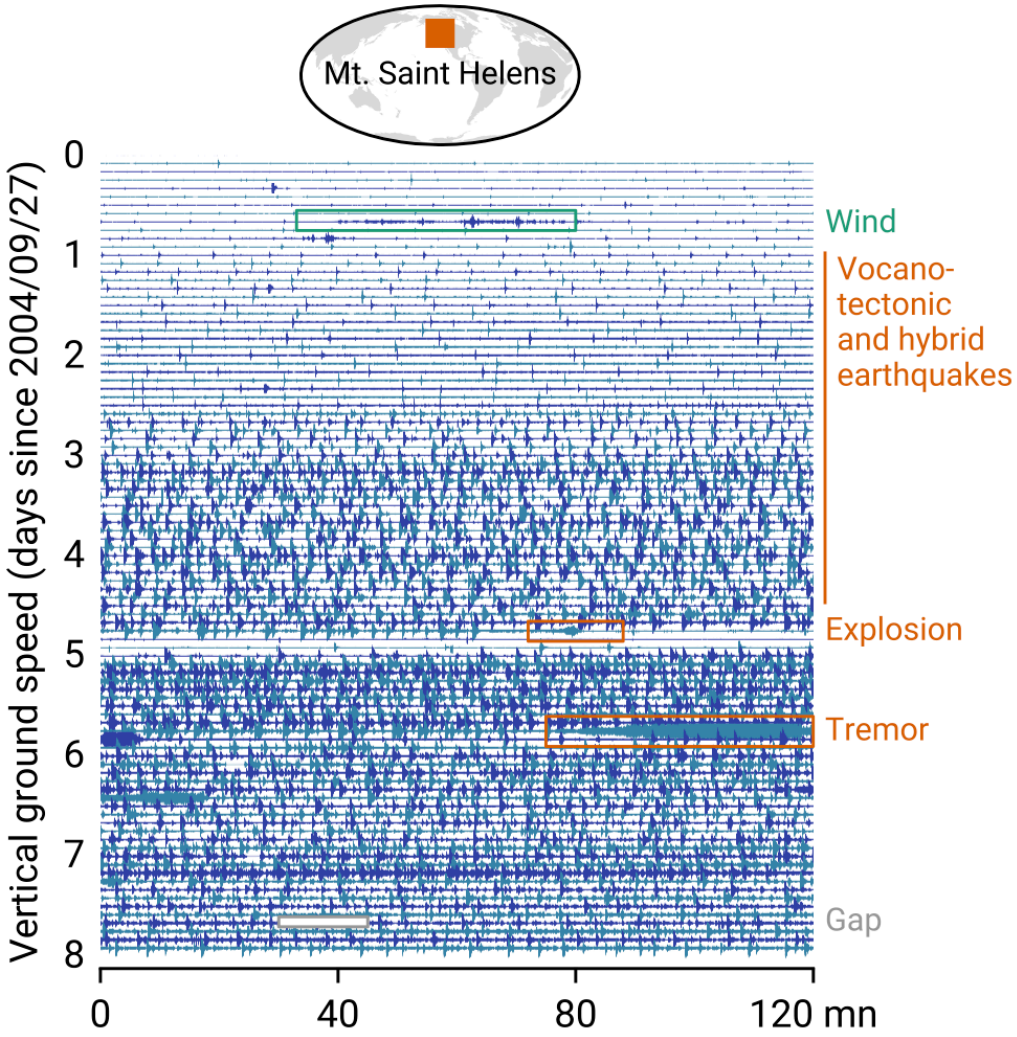


from Moran et al. (2008)

Task: seismic events detection and classification

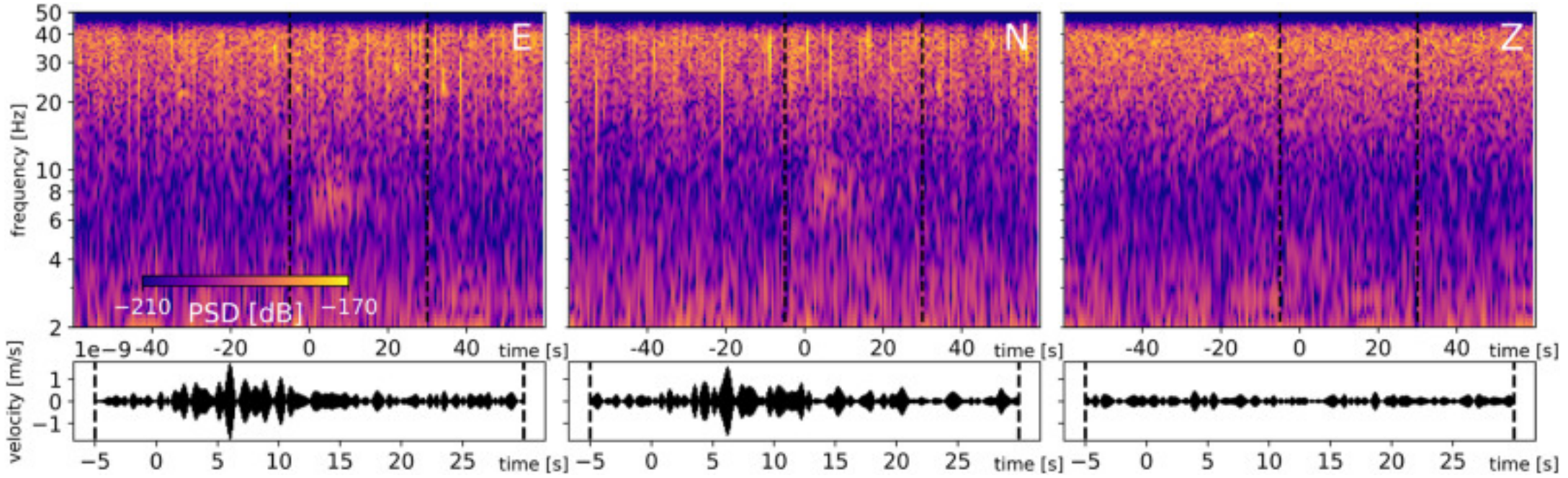
Most humans can pinpoint events.

Experts can **classify** them.



from Moran et al. (2008)

Task: diving into previously unseed data

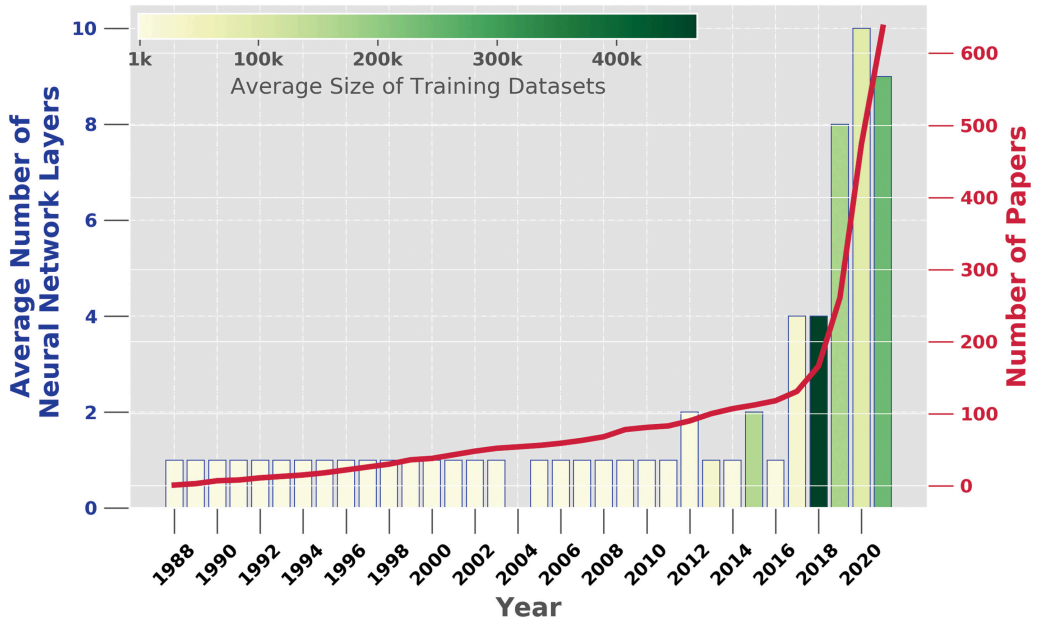


Expert-detected marsquake within continuous insight data.

Clinton et al. (2021)

Machine learning tasks

- Time-consuming tasks
- Hard-to-describe tasks
- Exploration of new data

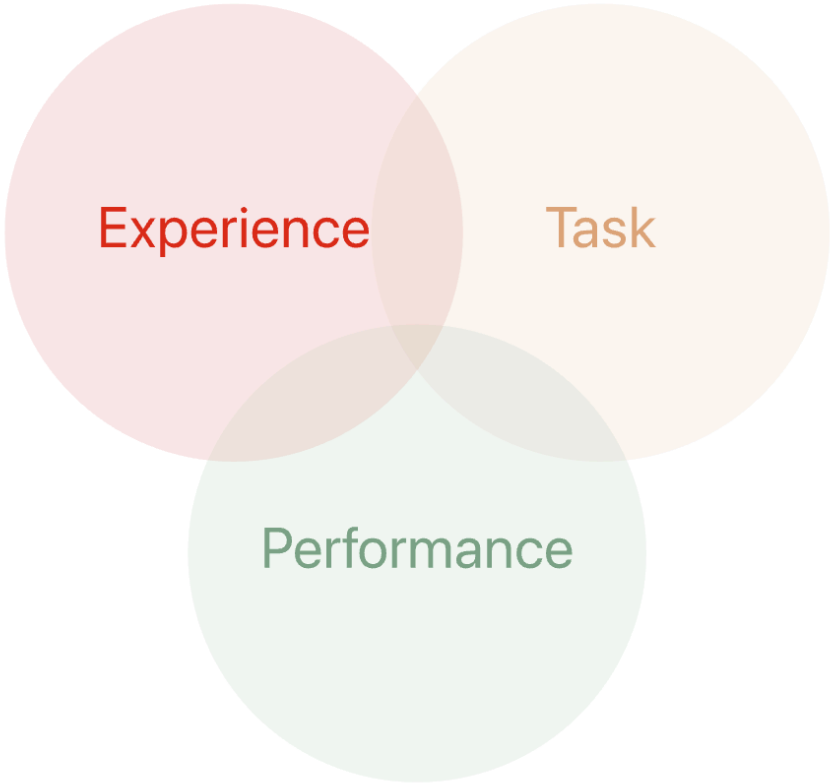


2. Definitions

Machine learning is a field of study in artificial intelligence of statistical algorithms that can effectively generalize and thus perform tasks without explicit instructions.

General definition

An algorithm learns from **experience** with respect to a **task** and **performance**, if its performance at solving the task improves with experience.



The data, the model, and the loss

the data

A set of samples $\mathbf{x}_i$ and labels $\mathbf{y}_i$ to learn from:

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$$

the model

A parametric function f_θ that maps data $\mathbf{x}$ to $\hat{\mathbf{y}}$

$$f_\theta : \mathbf{x} \mapsto \hat{\mathbf{y}}$$

the loss

A measurement of the model performance

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$$

Learning is equivalent to find the optimal parameters θ^* that minimize the loss function $\mathcal{L}$, as in

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}\Big(f_\theta(\mathbf{x}), \mathbf{y}\Big)$$

Vocabulary and symbols

An image is a sample $\mathbf{x}$ with

$$\mathbf{x} \in \mathbb{X} = \mathbb{R}^{H \times W \times C}$$

H is the height, W the width, and C the channels. The labels are a category y with

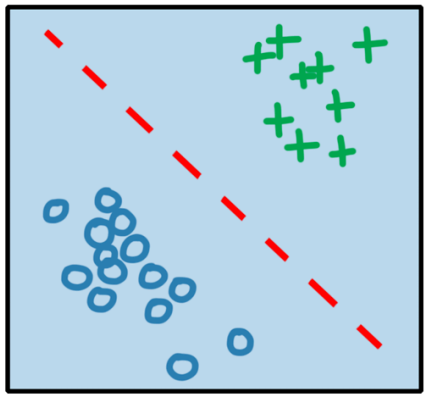
$$y \in \mathbb{Y} = \{0, 1, \ldots, K\}$$

with K the number of categories.
Note that y is scalar in this case.

Symbol	Name
$\{\mathbf{x}_i \in \mathbb{X}\}_{i=1\ldots N}$	Collection of **samples**
$\{\mathbf{y}_i \in \mathbb{Y}\}_{i=1\ldots N}$	Collection of **labels**
$\mathbf{x} = (x_1, \ldots, x_F)$	Set of sample **features**
$\mathbf{y} = (y_1, \ldots, y_T)$	Set of label **targets**
N	Dataset size
F	Feature space dimensions
T	Target space dimension
$\mathbb{X}$	Data space
$\mathbb{Y}$	Label space

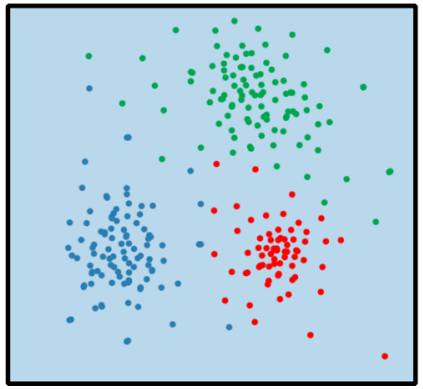
Main types of machine learning strategies

supervised learning



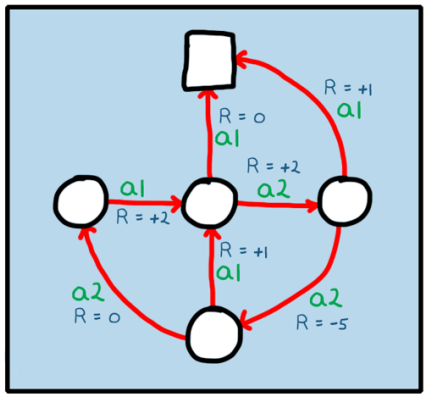
Predict $\mathbf{y}$ from $\mathbf{x}$ (regression, classification).

unsupervised learning



Learn some distribution $p(\mathbf{x})$ (clustering, reduction).

reinforcement learning



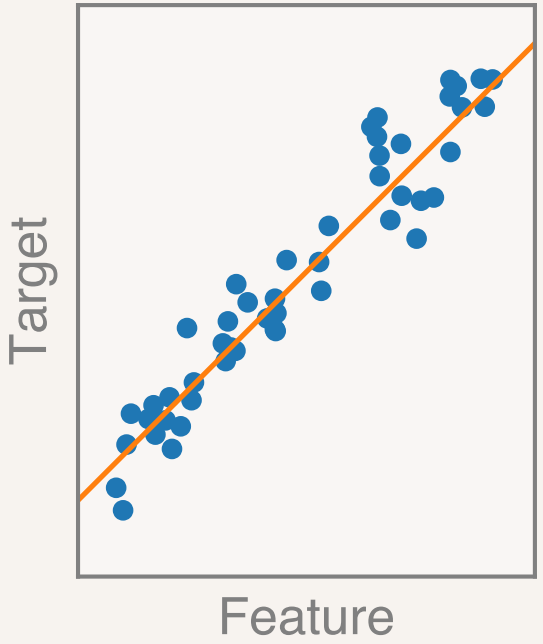
Learn a policy to maximize a reward (gaming, robotics).

from mathworks.com

The two main tasks of supervised learning

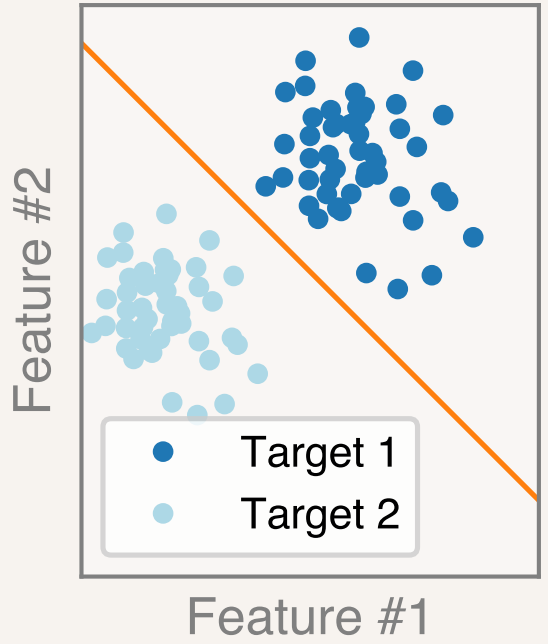
Regression

x and y are continuous



Classification

x is continuous and y is discrete



3. Supervised learning: regression

How to solve a regression or classification task with machine learning?

Regression

Dataset

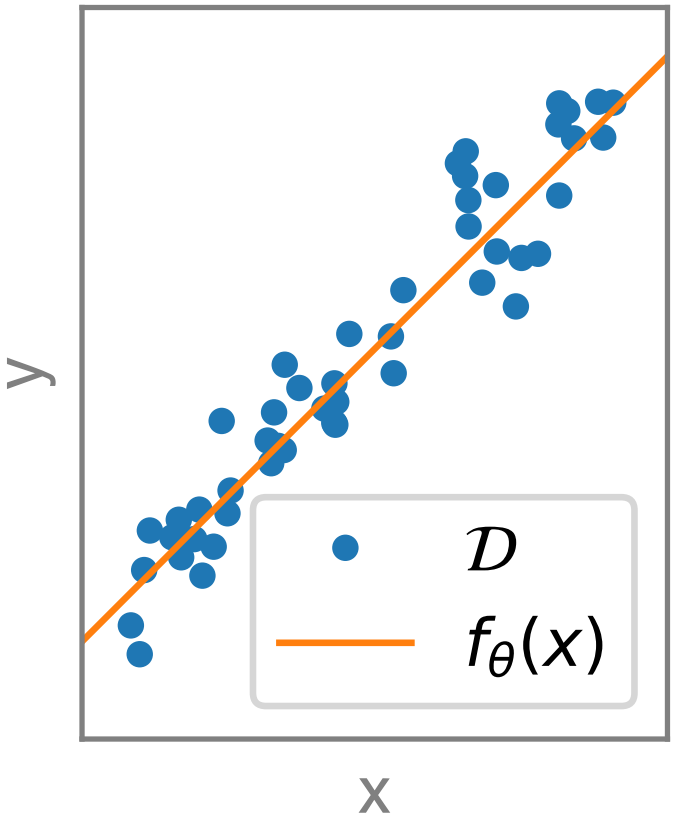
The set of N samples x_i and corresponding labels y_i such as

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

Formulation

Optimize the parameters θ of a function f_θ to predict y from x. Find the optimal parameters θ^* that minimize $\mathcal{L}$, such as

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}\Big(f_\theta(x), y\Big).$$



Linear regression

Linear model

coefficients $\theta = (a, b) \in \mathbb{R}^2$ that map x to y with

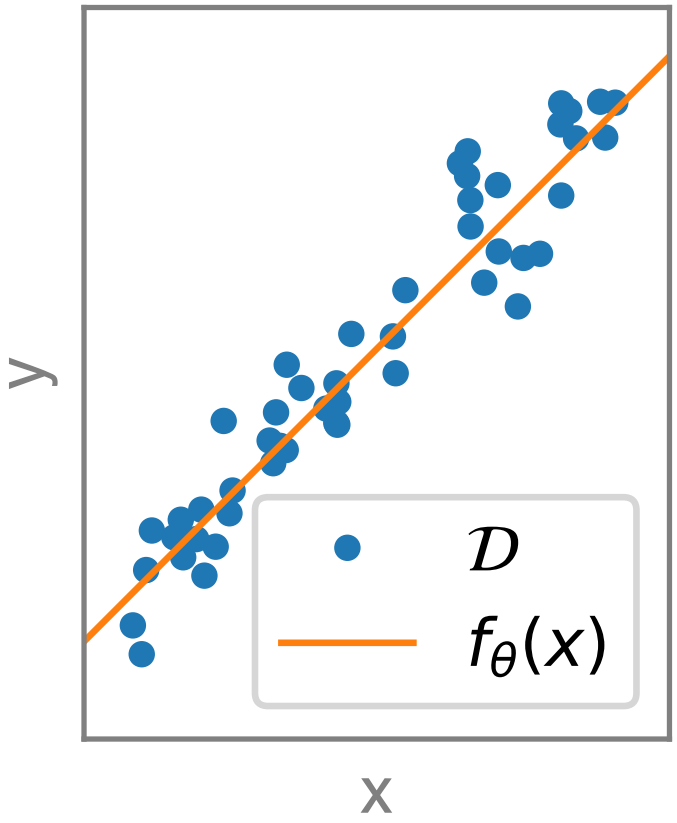
$$f_\theta : x \mapsto y = ax + b.$$

Loss function

For instance mean squared error:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(f_\theta(x_i) - y_i \right)^2 .$$

How do we minimize the loss?



Naive attempt: grid search

Implementation

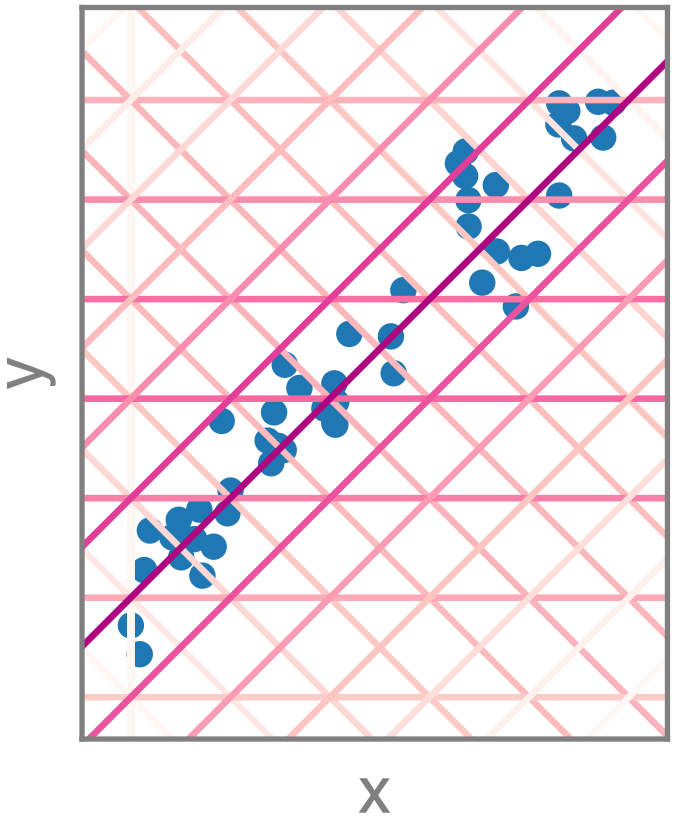
Find θ^* among regularly spaced tested values of θ.

Pros

Easy to implement, exhaustive search, uncertainty estimation.

Cons

Unscalable: if 0.1s / evaluation, then 2 parameters with 100 values each takes 1/4 hour. **For 5 parameters it takes more than 30 years!**



Random search

Implementation

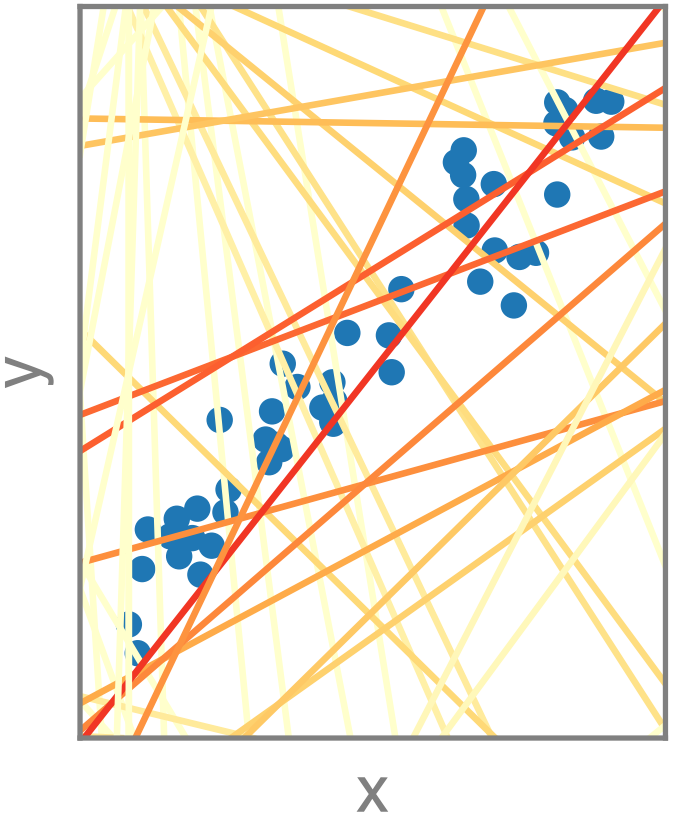
Sample random values of θ, keep the best one.

Pros

Easy to implement, scalable, uncertainty estimation, can include prior knowledge.

Cons

Not exhaustive, can be slow to converge.



Gradient descent

Implementation

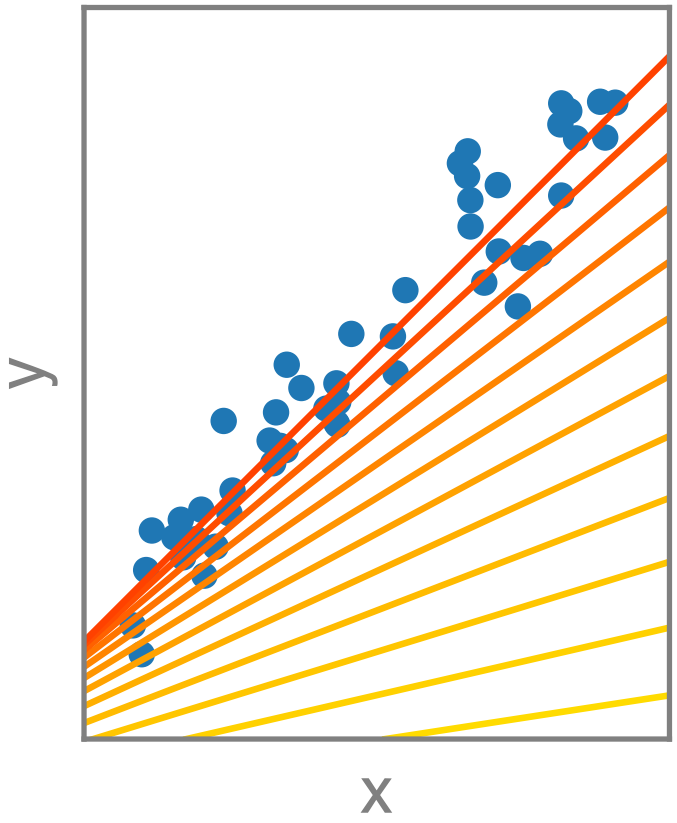
Estimate the gradient of $\mathcal{L}$ w.r.t. the parameters θ, update the parameters towards gradient descent.

Pros

Converges faster than random search.

Cons

Gets stuck in local minima, slow to converge, needs differentiability.



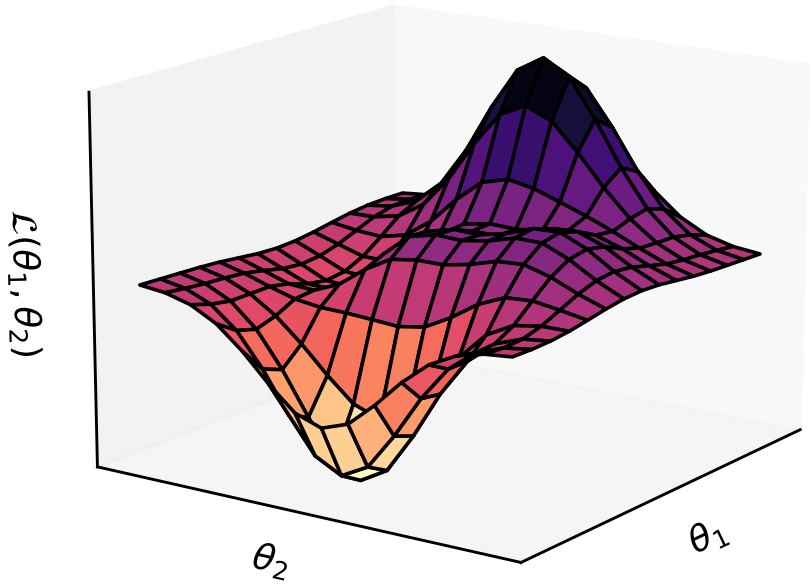
Gradient descent

Implementation

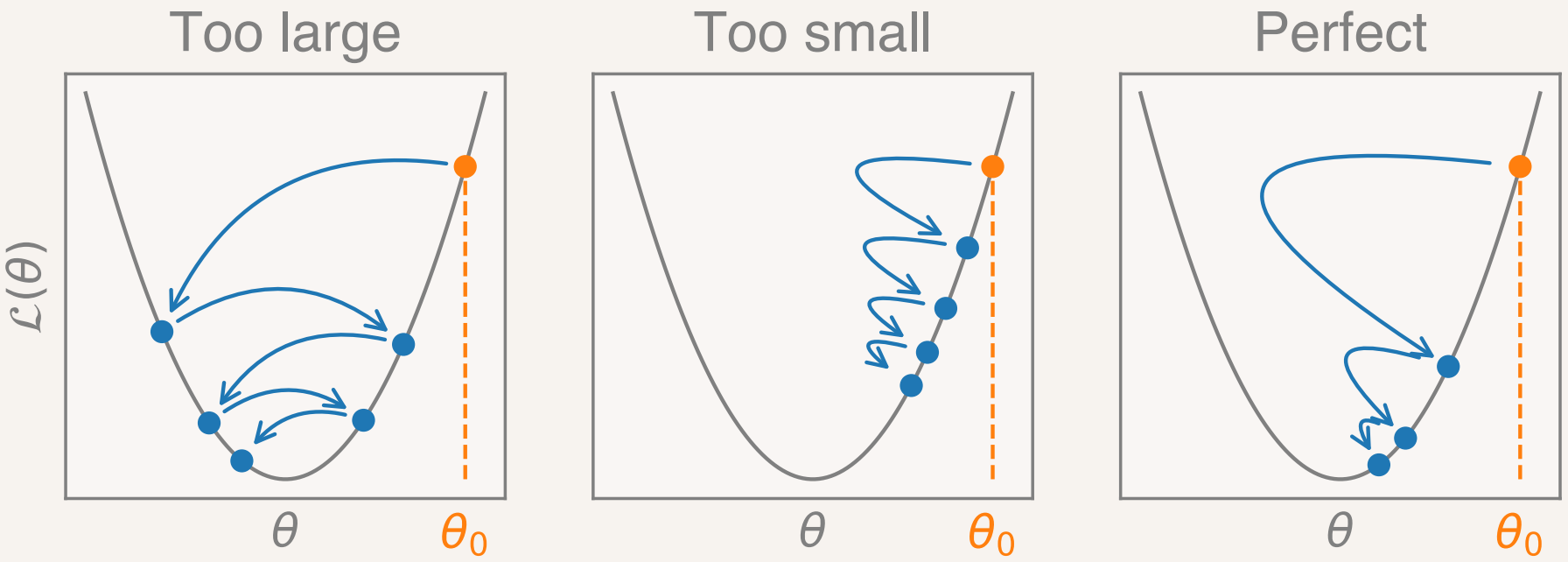
1. Initiate model $\theta = (a_0, b_0)$
2. Compute gradient $\nabla \mathcal{L}(\theta)$
3. Update $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta)$
4. Repeat until convergence

Hyperparameters

The **learning rate** η defines the update step.



How to deal with learning rate?



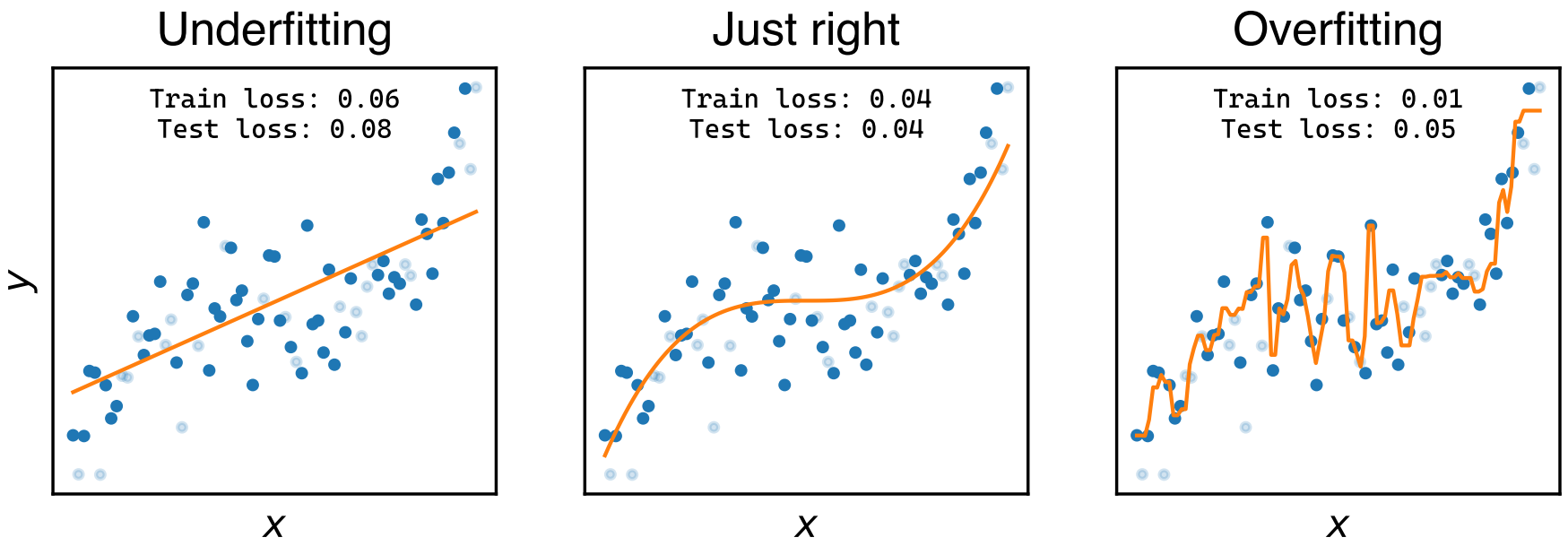
That's part of the **hyperparameters tuning**.

More about that in the deep learning lectures.

The problem of overfitting

Having a loss close to 0 does not mean that the model **generalizes** well.

Key concepts to prevent overfitting: split the dataset



By splitting the dataset into a **training** and a **testing** set,
we evaluate the performance on unseen (but **similar**) data.

Key concepts to prevent overfitting: regularization

Add a penalty term $\mathcal{R}$ to the loss $\mathcal{L}_{\mathcal{R}} = \mathcal{L} + \lambda\mathcal{R}$, with λ the regularization strength

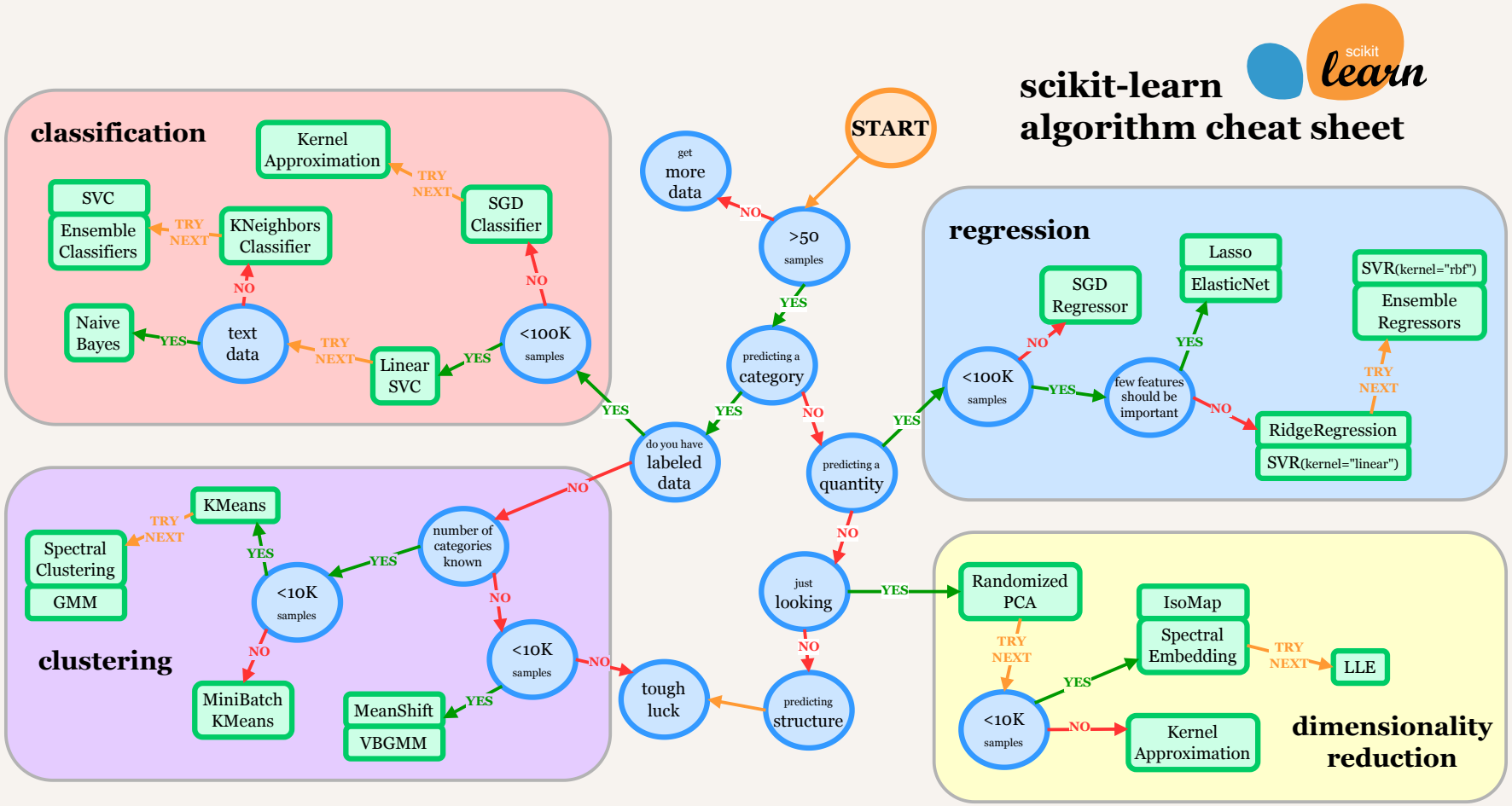
The regularization penalizes the model's complexity.

Why so many regression algorithms?

Because of combination of models, losses, and regularizations. The scikit-learn.org website provides a unified interface in a `greybox style`.

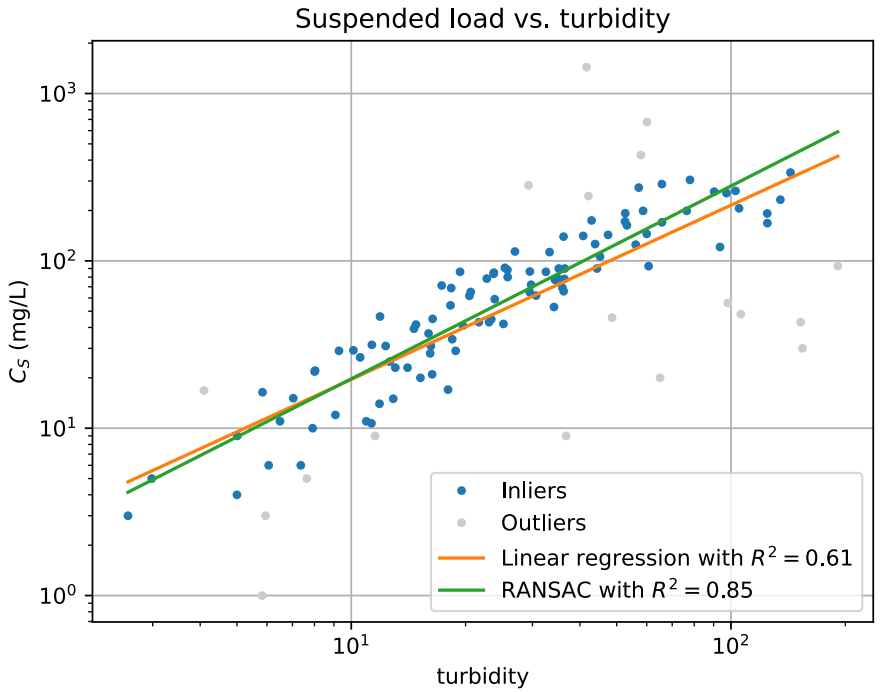
The model selection is made by experience or **trial and error**.

Guidelines for exploring relevant models



Notebook 1

High-quality data from cheap sensors

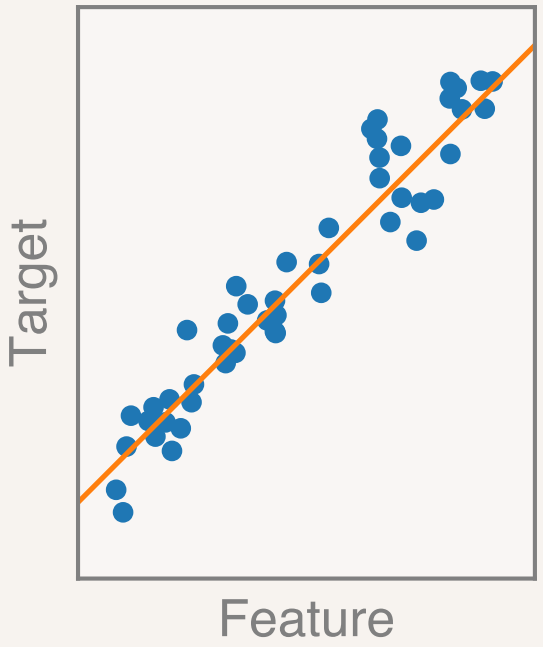


4. Supervised learning: classification

The two main tasks of supervised learning

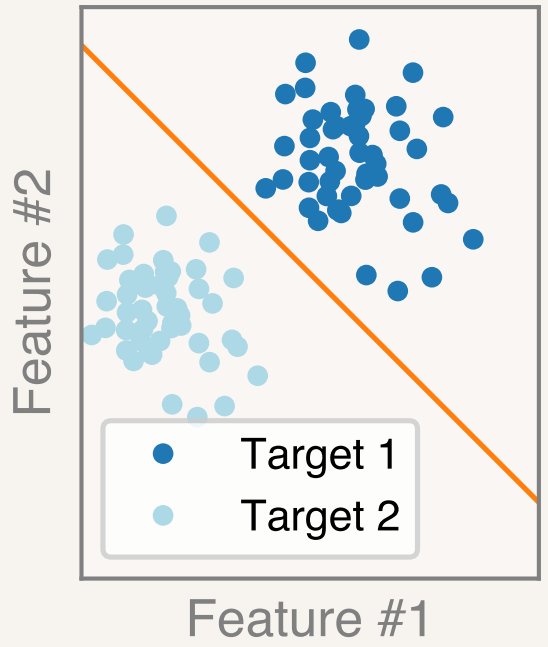
Regression

x and y are continuous

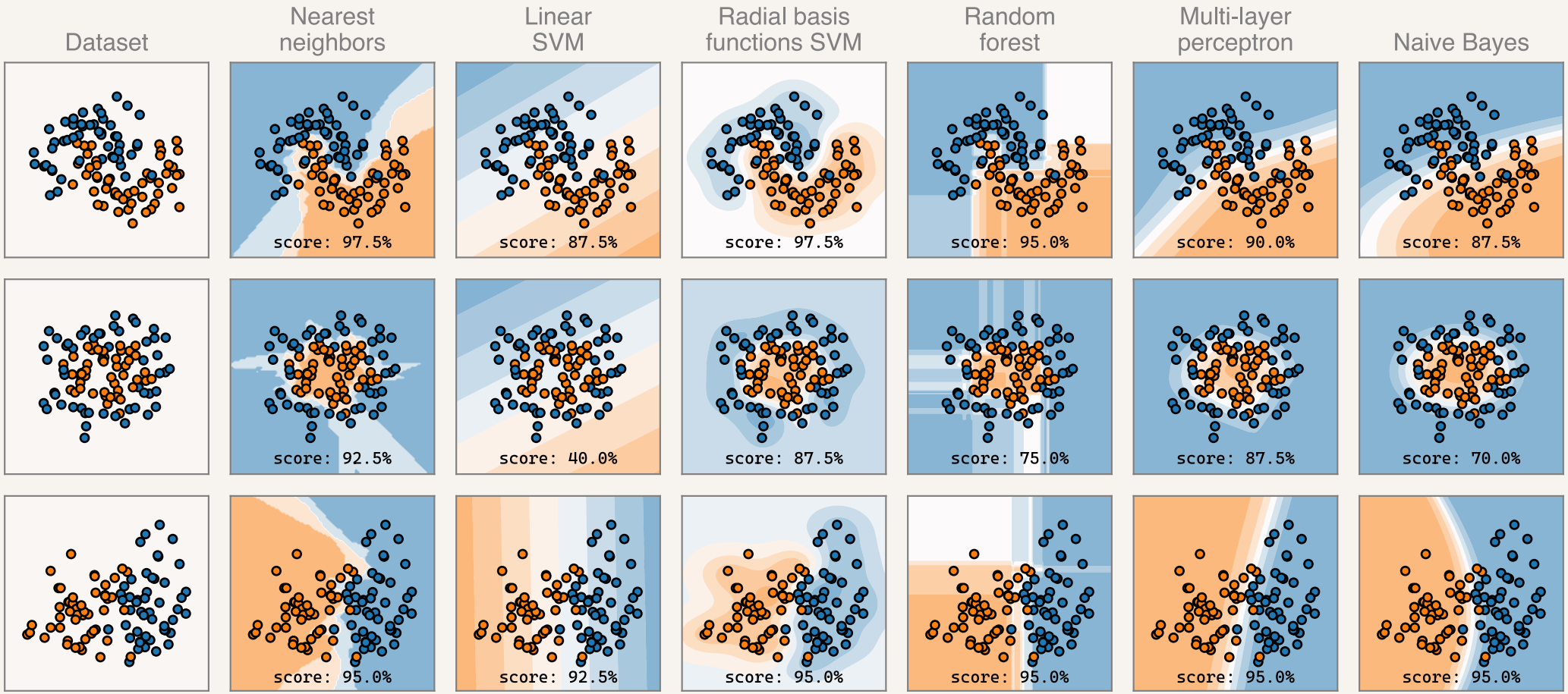


Classification

x is continuous and y is discrete



The classification task



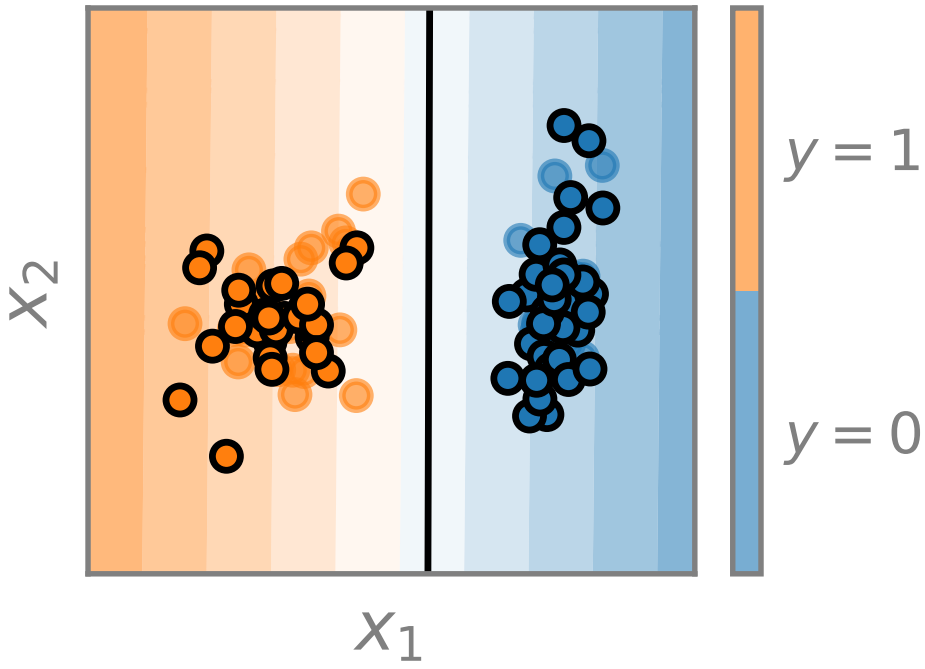
Here again, we have many possibilities.

The classification task

Experience: labels $y \in \{0, 1\}$ for two features $\mathbf{x} \in \mathbb{R}^2$.

Task: predict $\hat{y}$ of each sample $\mathbf{x}$.

Performance: how should we measure the performance of a classifier?

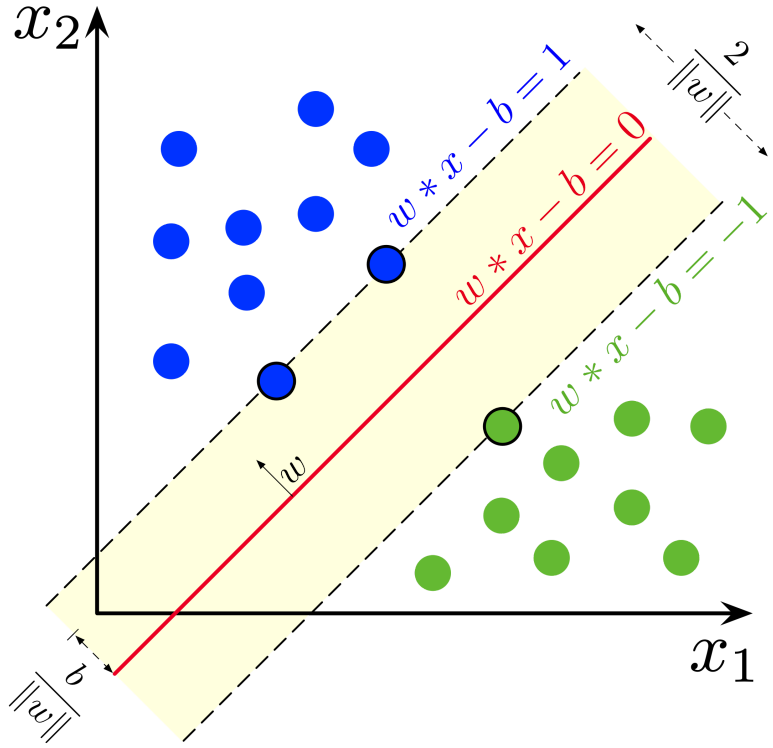


Support vector machines

Support vector machines search the hyperplane of normal vector $\mathbf{w}$ and bias b that split the classes.

> Note: in 2D, a hyperplane is a line.

The support vectors are the samples that are closest to the other class.



https://en.wikipedia.org/wiki/Support_vector_machine

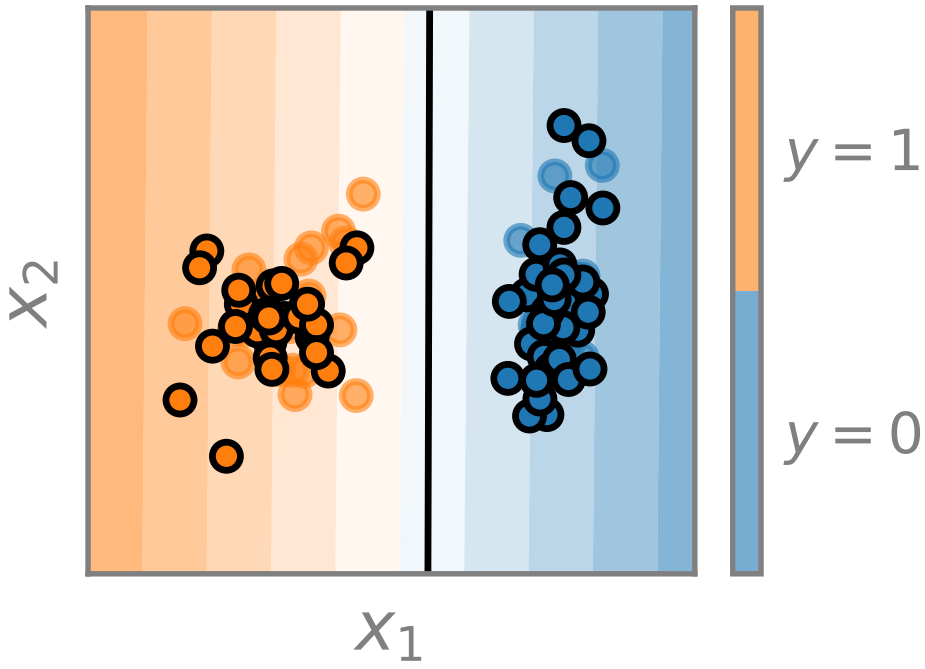
Support vector machines

The decision function $f(\mathbf{x})$ dependson the sign of the linear combination of the normal vector and the sample:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

The quantity to minimize is the **Hinge loss**:

$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right)\right)$$



Support vector machines

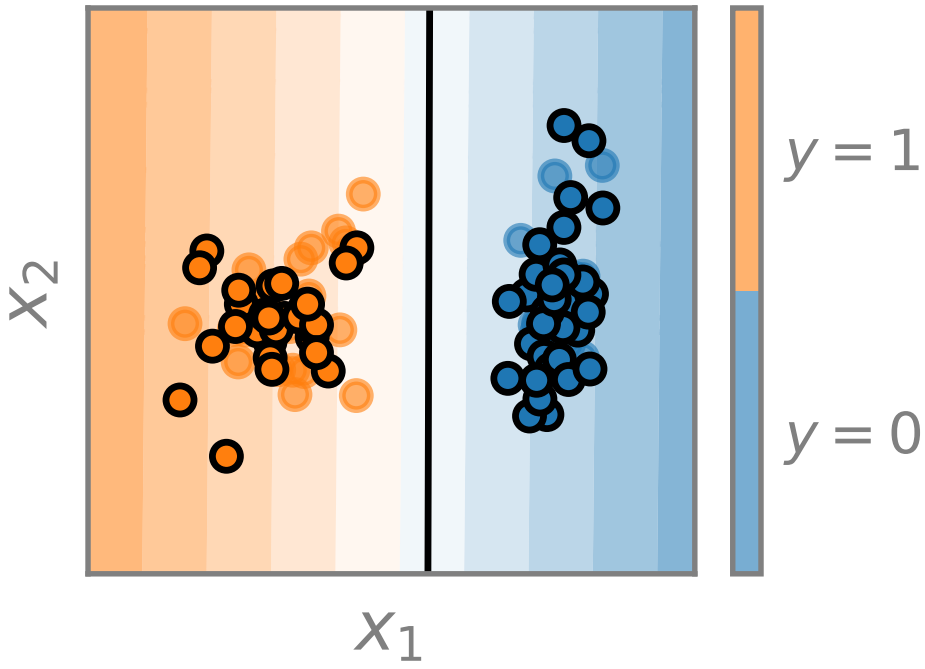
The decision function $f(\mathbf{x})$ depends on the sign of the linear combination of the normal vector and the sample:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

The quantity to minimize is the **hinge loss**:

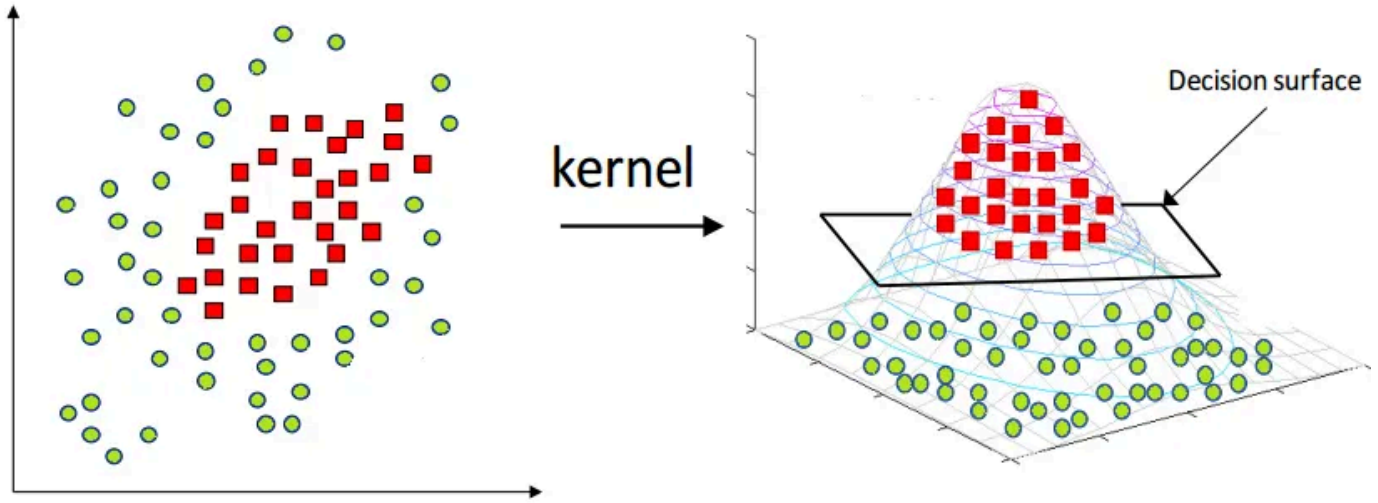
$$\mathcal{L}(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right)\right)$$

What about non linear problems?



The kernel trick for non linear classification problems

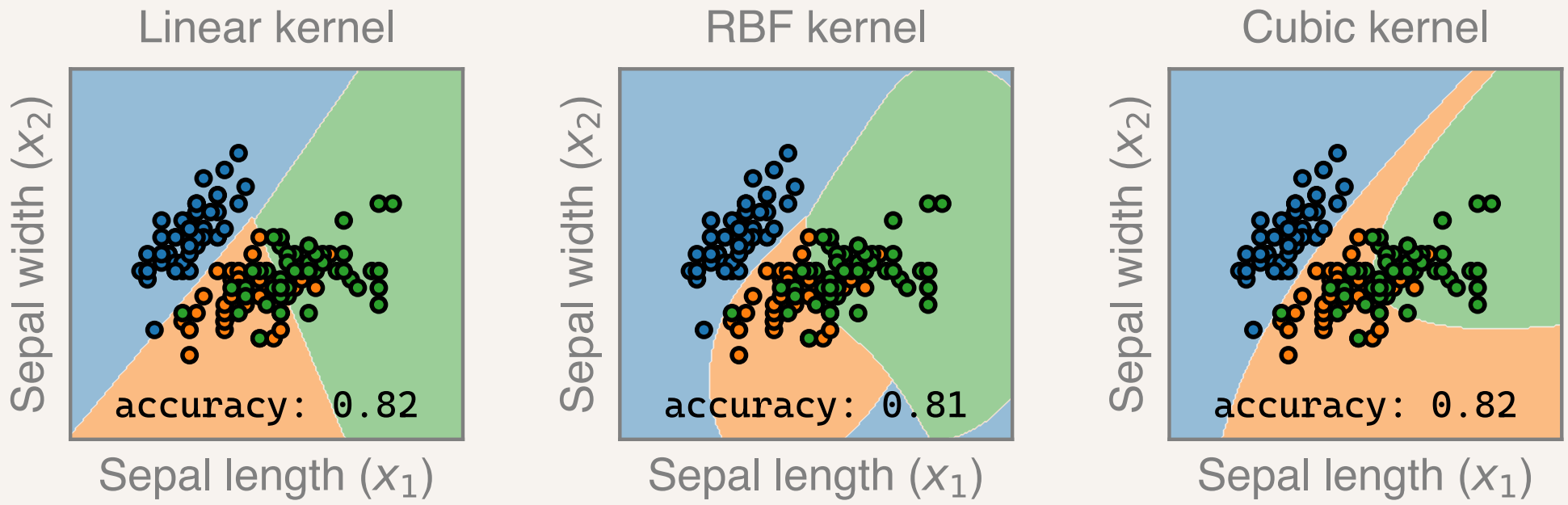
The kernel trick allows to map the data to a **higher dimensional** space made from the input features where the problem is **linearly separable**.



The **Radial Basis Functions** (RBF) is an infinite kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$

Support vector classifier with multiple classes

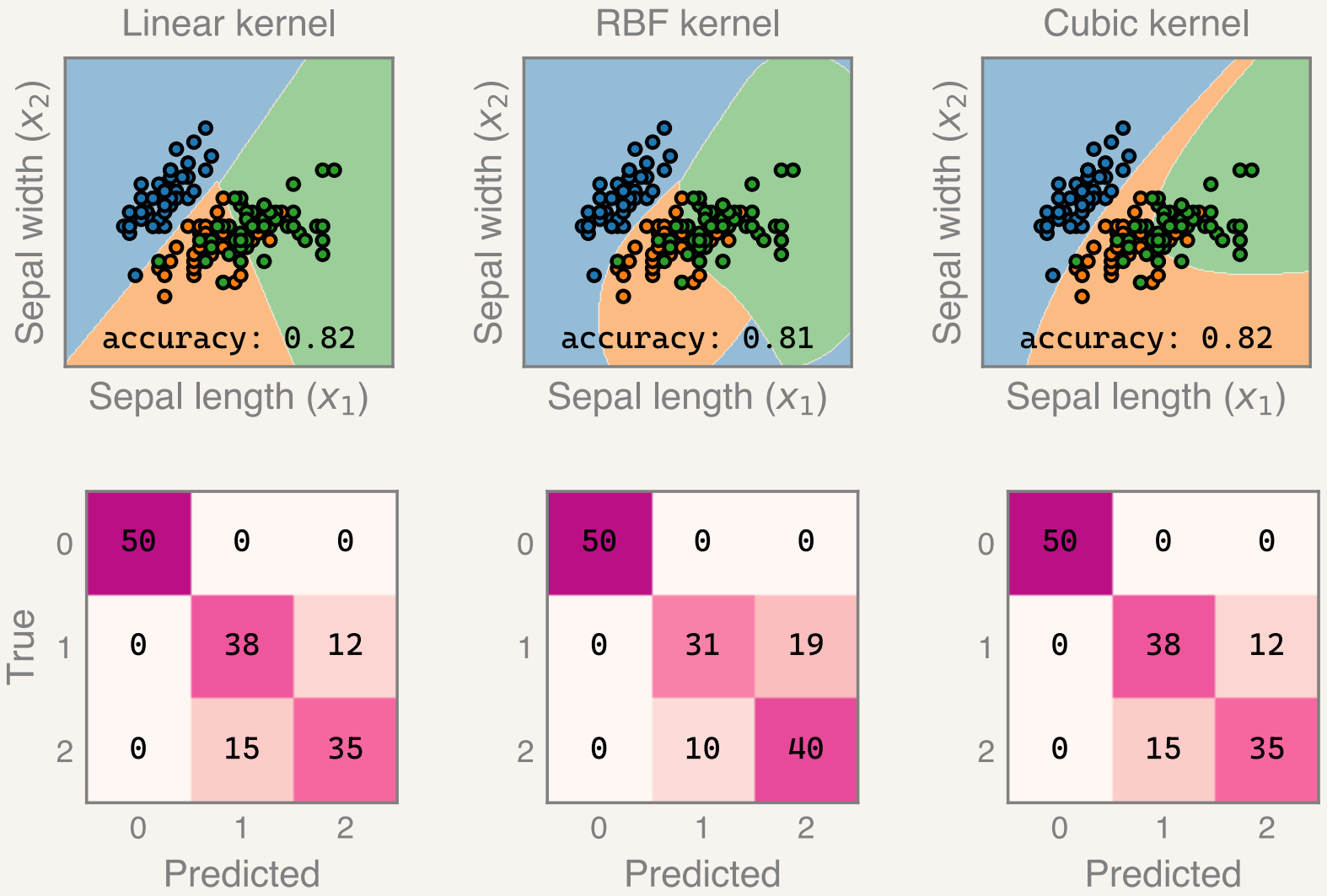
The SVC is a generalization of the SVM that digests more than two classes.



The decision function is linear in the kernel space only.

We can project it back to the data space to inspect it.

Confusion matrix



Accuracy, precision, and recall

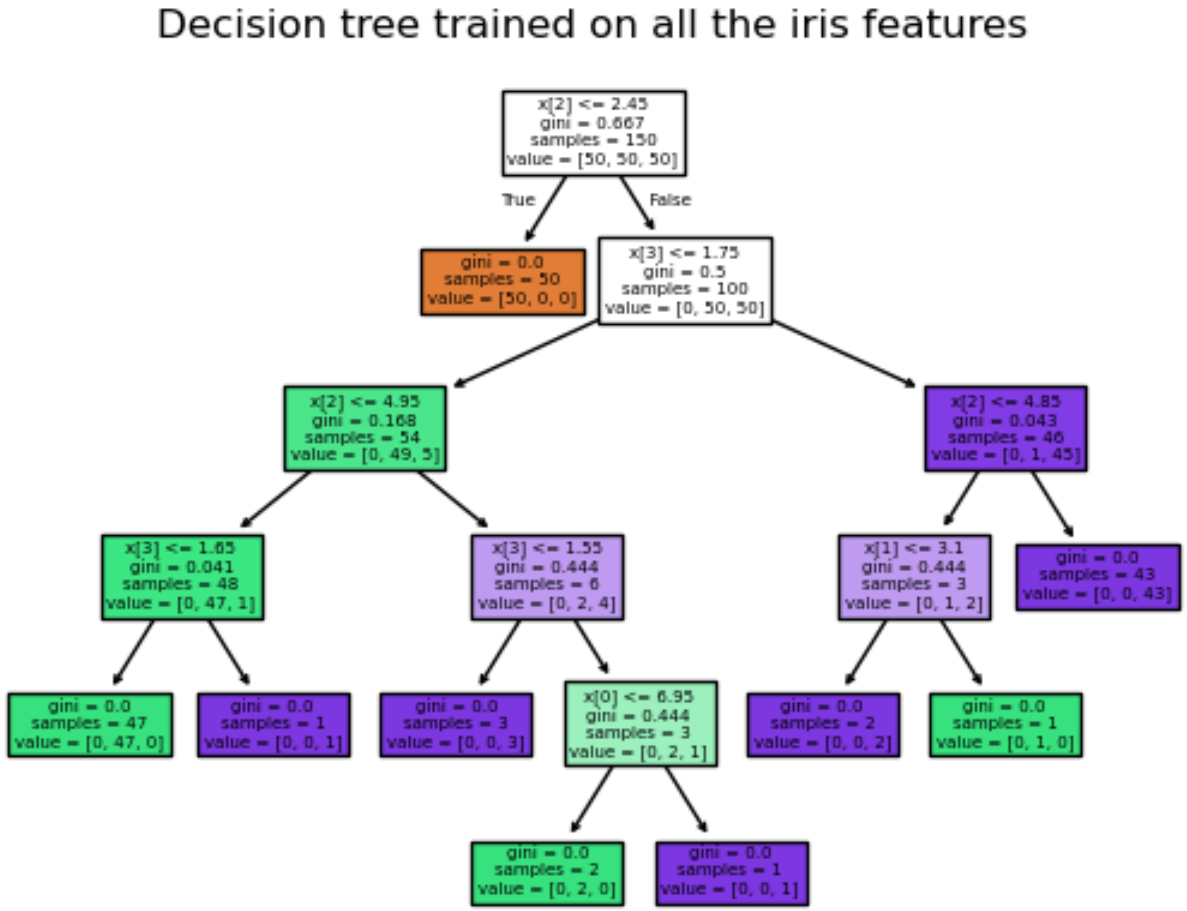
Ma et al. (2019)

Decision trees and random forests

Decision trees learn to predict y with feature splitting.

Random forests are ensembles of decision trees that vote for y.

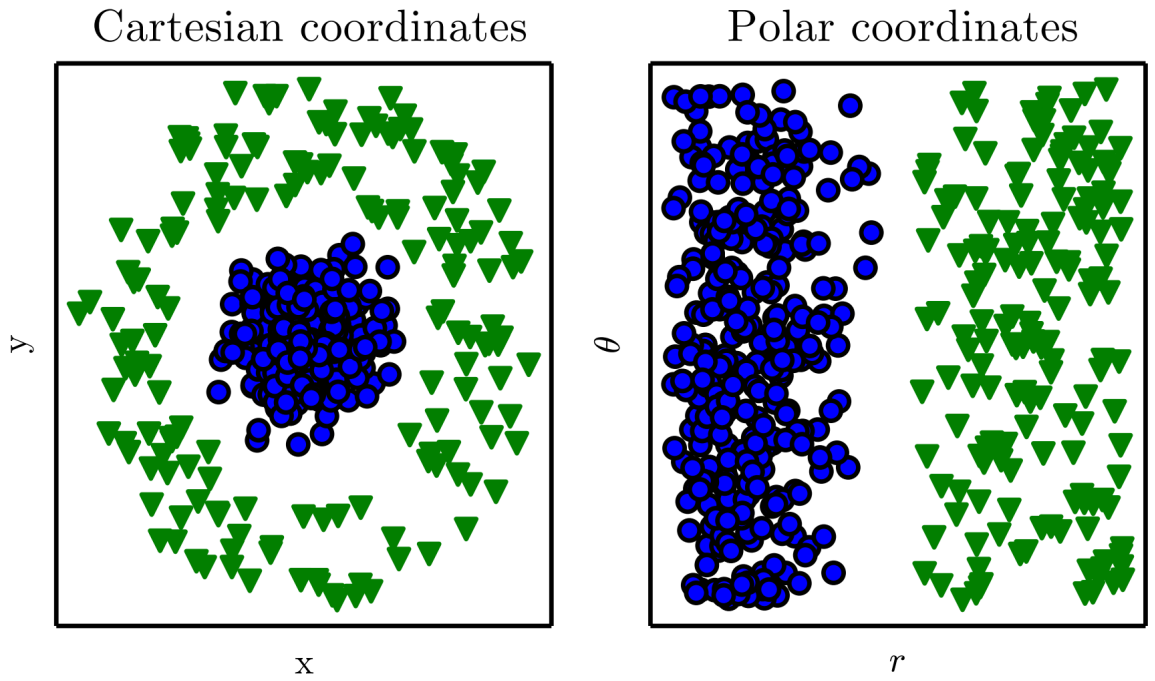
These algorithms are extremely powerful.



from scikit-learn.org

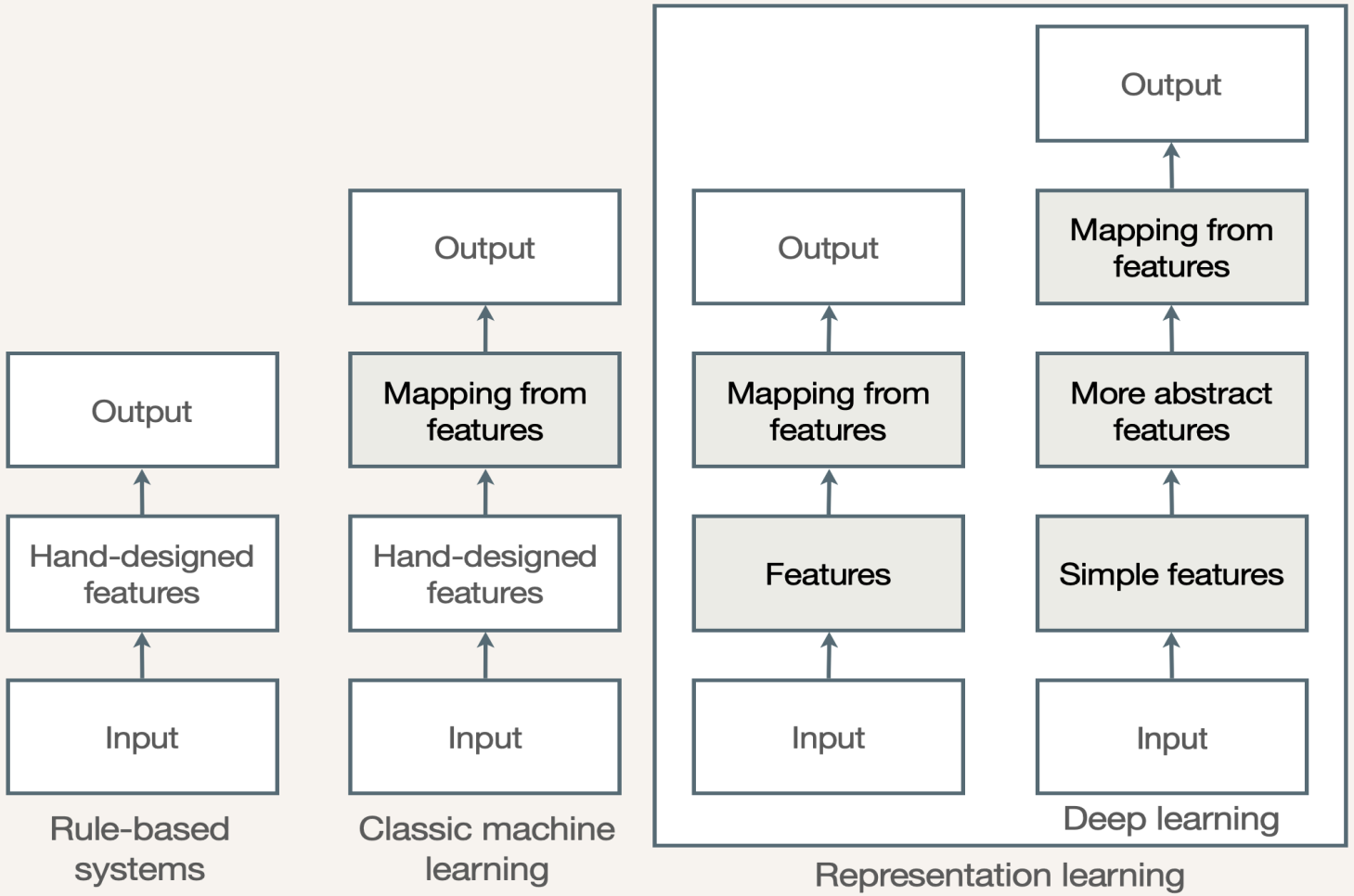
Representation matters

There is no need for a complex model if you have a good **representation** of the data.



Representation depending on the task complexity

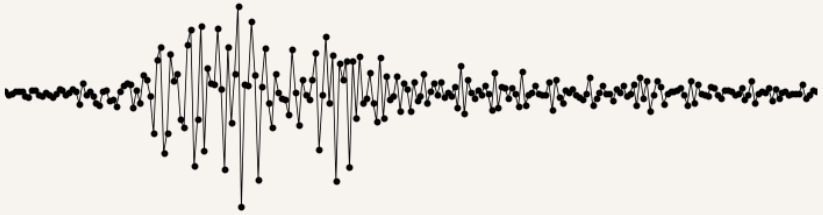
Hand-designed or learned features?



Goodfellow et al. (2016)

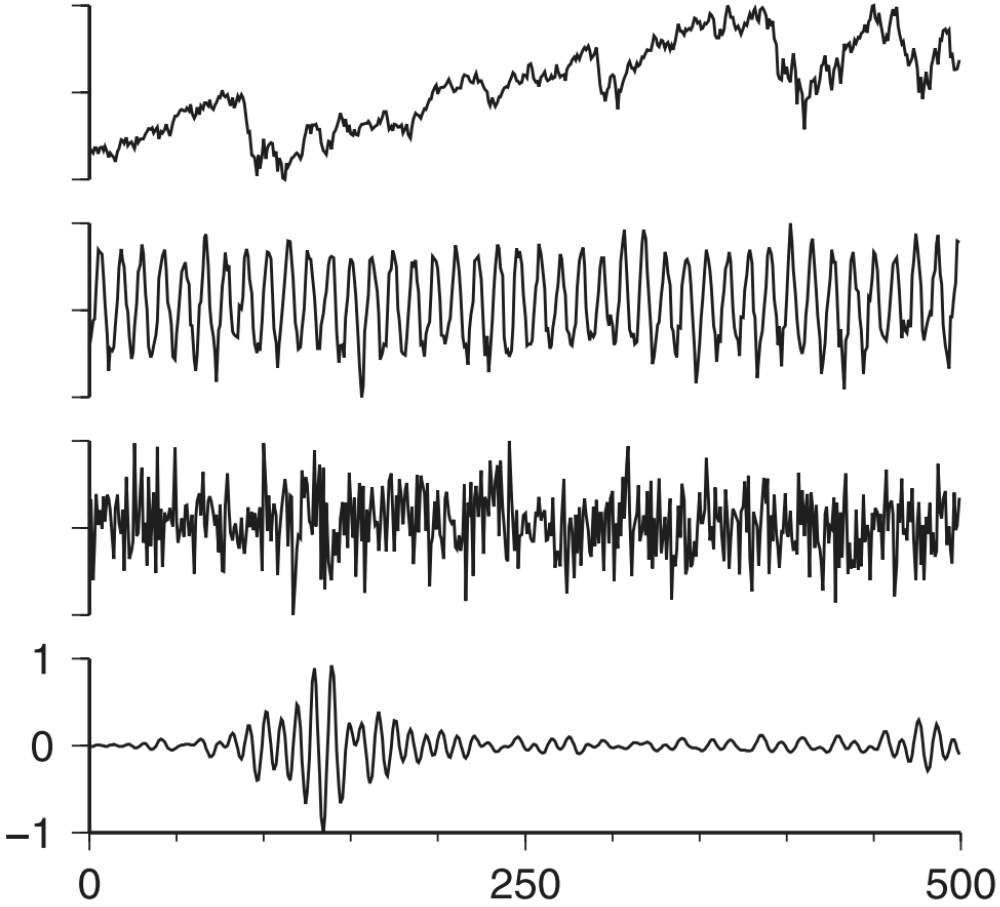
Representation matters

We can see waveforms $\mathbf{x} \in \mathbb{R}^N$ as points of a N-dimensional space



Yet, seismic waveform do not occupy this space fully, likely very sparse.

Dimension > Information

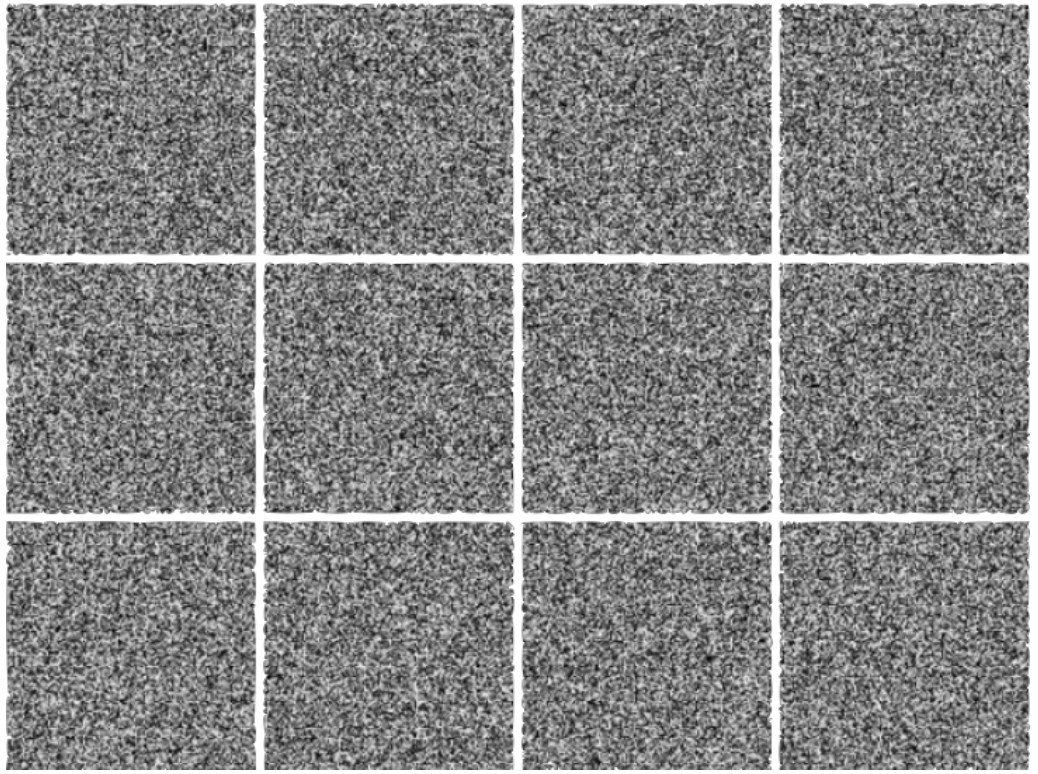


still from Valentine & Trampert (2012)

Representation matters

Random sampling of the pixels of a face. What is the likelihood that the reshuffled image *is* a face?

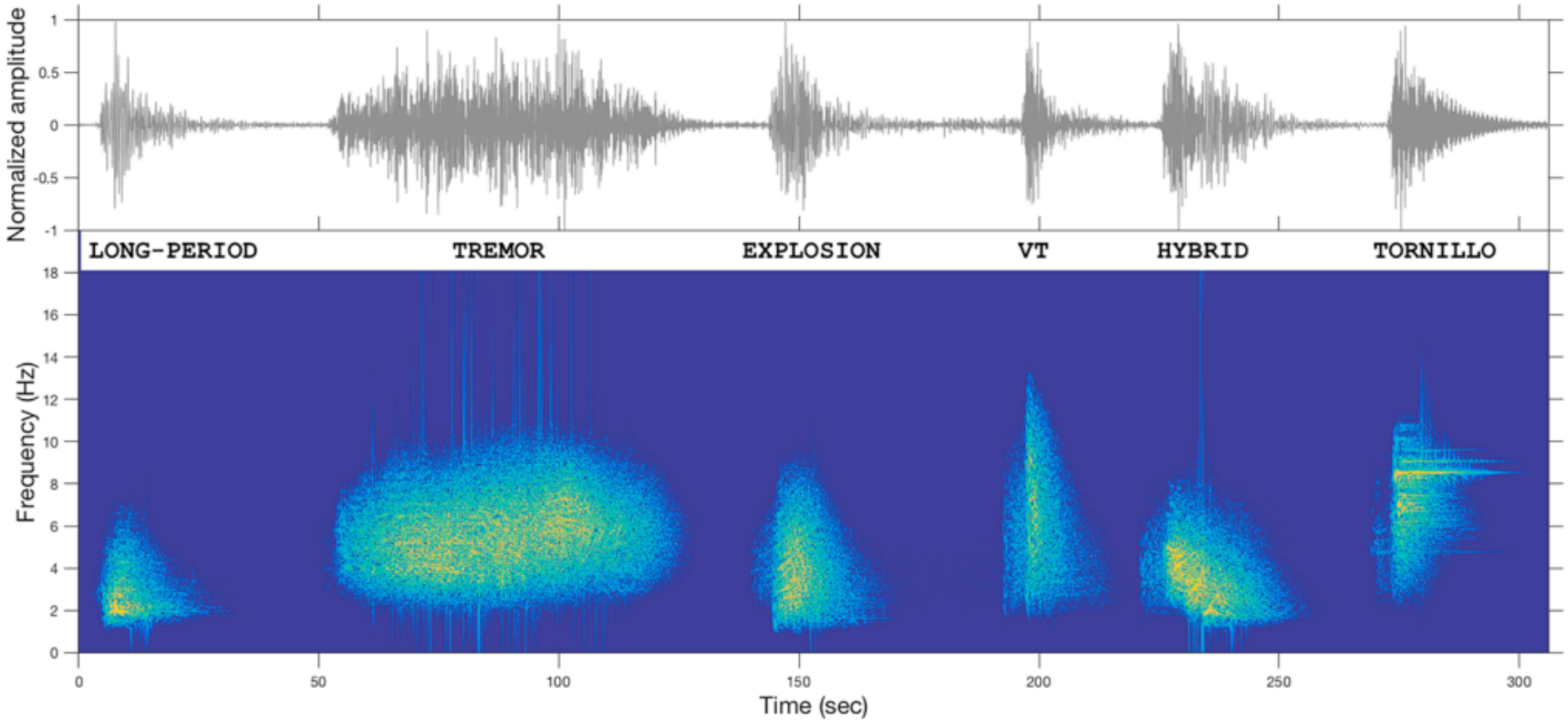
Like waveforms, **images are living on a manifold.**



modified from Goodfellow et al. (2016)

Seismo-volcanic signal classification

Supervised learning experiences a set of examples containing features $\mathbf{x}_i \in \mathbb{X}$ associated with labels $\mathbf{y} \in \mathbb{Y}$ to be predicted from the features (here, classification).

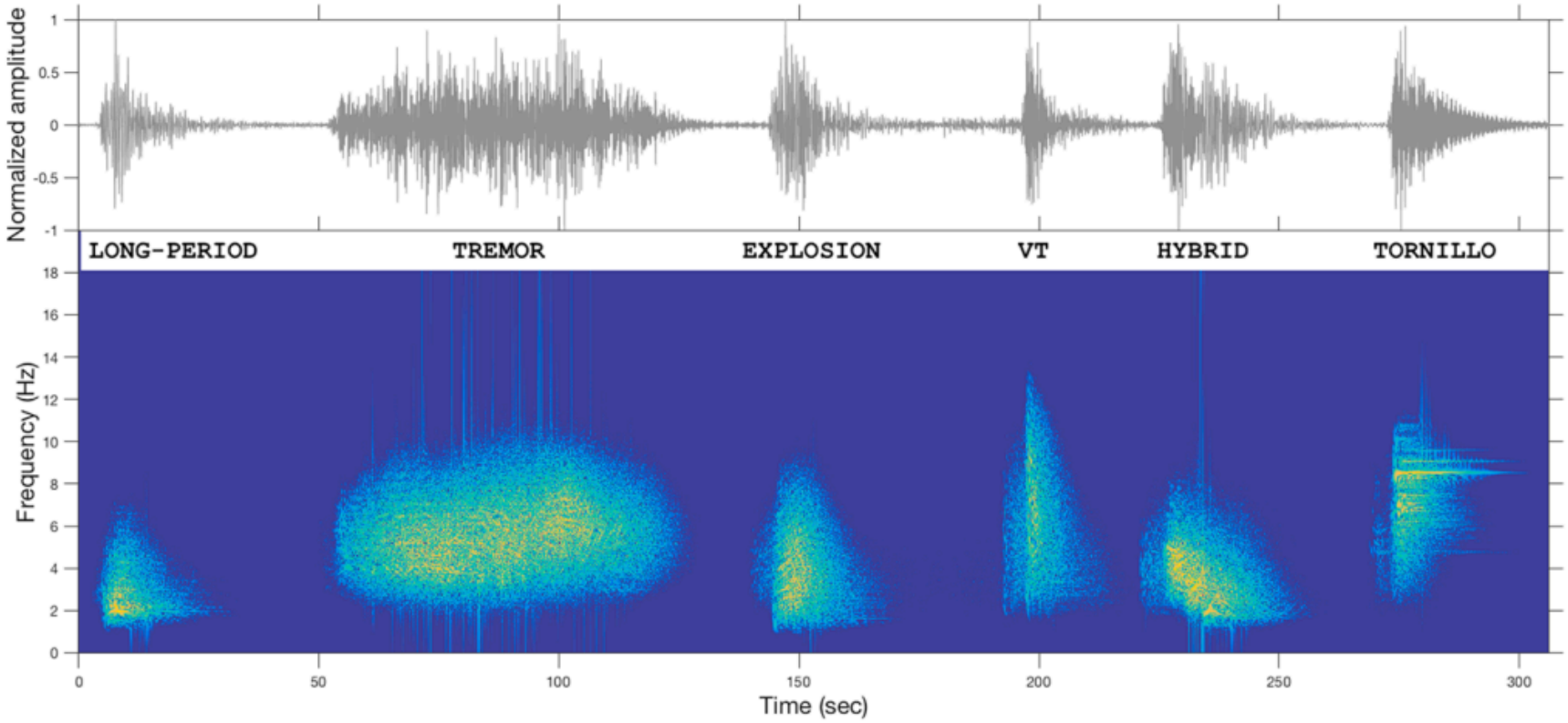


Malfante et al. (2018)

Seismo-volcanic signal classification

In this case, $\mathbf{x}$ lies in $\mathbb{R}^{3\times N}$, and $\mathbf{y}$ in $[0, \ldots, 5]$.

Which **representation** of $\mathbf{x}$ works best?



Malfante et al. (2018)

Handcrafted features

Table 1
List of Features

Statistic features

Feature	Definition	Used in	Ref.
Length	$n = \text{length}(s)$	Tucker and Brown (2005)	1
Mean	$\mu_s = \frac{1}{n}\sum_i s[i]$	Tucker and Brown (2005)	2
Standard deviation	$\sigma_s = \sqrt{\frac{1}{(n-1)}\sum_i (s[i]-\mu_s)^2}$		3
Skewness	$\frac{1}{n}\sum_i \left(\frac{s[i]-\mu_s}{\sigma_s}\right)^3$	Langet (2014) and Hibert et al. (2014)	4
Kurtosis	$\frac{1}{n}\sum_i \left(\frac{s[i]-\mu_s}{\sigma_s}\right)^4$	Langet (2014) and Hibert et al. (2014)	5
i of central energy	$\bar{i} = \frac{1}{E}\cdot\sum_i E_i \cdot i$	(Tucker & Brown, 2005)	6
RMS bandwidth	$B_i = \sqrt{\frac{1}{E}\sum_i i^2 \cdot E_i - \bar{i}^2}$	Tucker and Brown (2005)	7
Mean skewness	$\sqrt{\frac{\sum_i (i-\bar{i})^3 E_i}{E\cdot B_i^3}}$	Tucker and Brown (2005)	8
Mean kurtosis	$\sqrt{\frac{\sum_i (i-\bar{i})^4 E_i}{E\cdot B_i^4}}$	Tucker and Brown (2005)	9

Entropy features (with $p(s_j)$ the probability of amplitude level s_j)

Feature	Definition	Ref.	
Shannon entropy[a]	$-\sum_j p(s_j)\log_2\left(p(s_j)\right)$	Esmaili et al. (2004) and Han et al. (2011)	10 to 12
Rényi *entropy*[b]	$\frac{1}{1-\alpha}\cdot\log_2\left(\sum_j p(s_j)^\alpha\right)$	Han et al. (2011)	13 to 18

Shape descriptor features

Feature	Definition	Ref.	
Rate of attack	$\max_i\left(\frac{s[i]-s[i-1]}{n}\right)$	Tucker and Brown (2005)	19
Rate of decay	$\min_i\left(\frac{s[i]-s[i+1]}{n}\right)$	Tucker and Brown (2005)	20
Ratios	min/mean and max/mean	Langet (2014) and Hibert et al. (2014)	21 to 22
Energy descriptors	Signal energy, maximum, average, standard deviation, skewness, and kurtosis	Tucker and Brown (2005)	23 to 28
Specific values	min, max, i of min, i of max, threshold crossing rate, and silence ratio	Tucker and Brown (2005)	29 to 34

Note. Features computed for a signal $s[i]_{i=1}^n$ (in which i might correspond to a temporal, frequency, or cepstral sample). $E = \sum_{i=1}^n s[i]^2$ and $E_i = s[i]^2$ describe the signal energy and the energy at sample i, respectively. Some features have a dimension greater than others; e.g., entropy measurements are made on three different estimations of the amplitude probability (i.e., different histogram bin numbers).
[a]Bin numbers for probability estimation: 5, 30, and 500. [b]Bin numbers for probability estimation: 5, 30, 500, $\alpha = 2$, and inf.

Jasperson et al. (2022)

Performance

Accuracy of the predictions measures the model's performance (= confusion matrix)

Table 3
Confusion Matrix

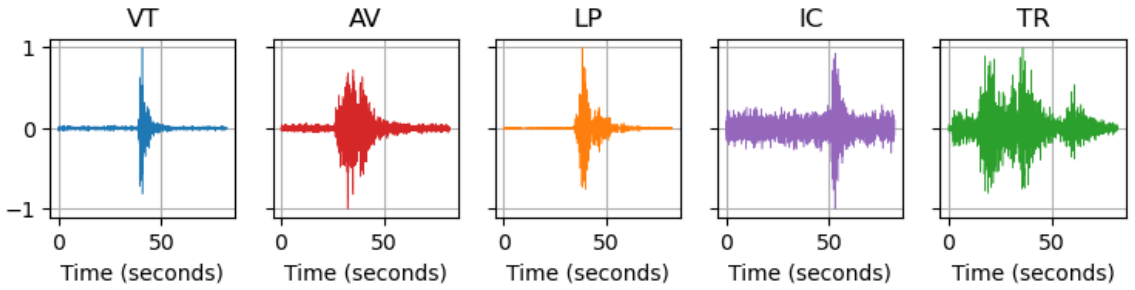
		True Class (ground truth)						
		LP	TR	VT	EXP	HYB	TOR	Precision
	LP	**58,363**	627	8	0	5	1	98.9%
	TR	3,000	**4,584**	0	1	2	0	60.4%
Predicted	VT	478	11	**475**	5	11	3	48.3%
Class	EXP	15	16	2	**29**	0	0	47.8%
	HYB	131	3	28	13	**125**	0	41.7%
	TOR	43	4	3	0	0	**28**	35.9%
Accuracy		**94.1%**	**87.4%**	**92.2%**	**59.8%**	**87.1%**	**84.6%**	

What is the guarantee that the features we choose are the best ones?

Malafante et al. (2018)

Notebook 2

Volcano seismology classification from Chilean volcanoes



Notebook 3

Identification of objects in a lidar cloud from labeled subset