

# 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

RESEARCH

## 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

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,

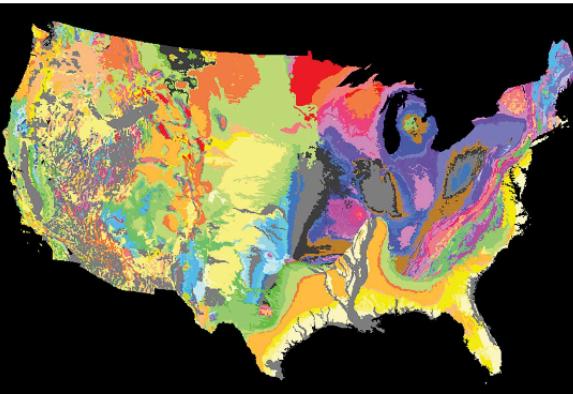
and classifying diverse seismic events. Examples of discovery include predicting laboratory slip events using observations of acoustic emissions, detecting weak earthquake signals using similarity search, and determining the connectivity of subsurface reservoirs using ground-water tracer observations.

**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. ■

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

Downloaded from https://www.science.org on December 09, 2023



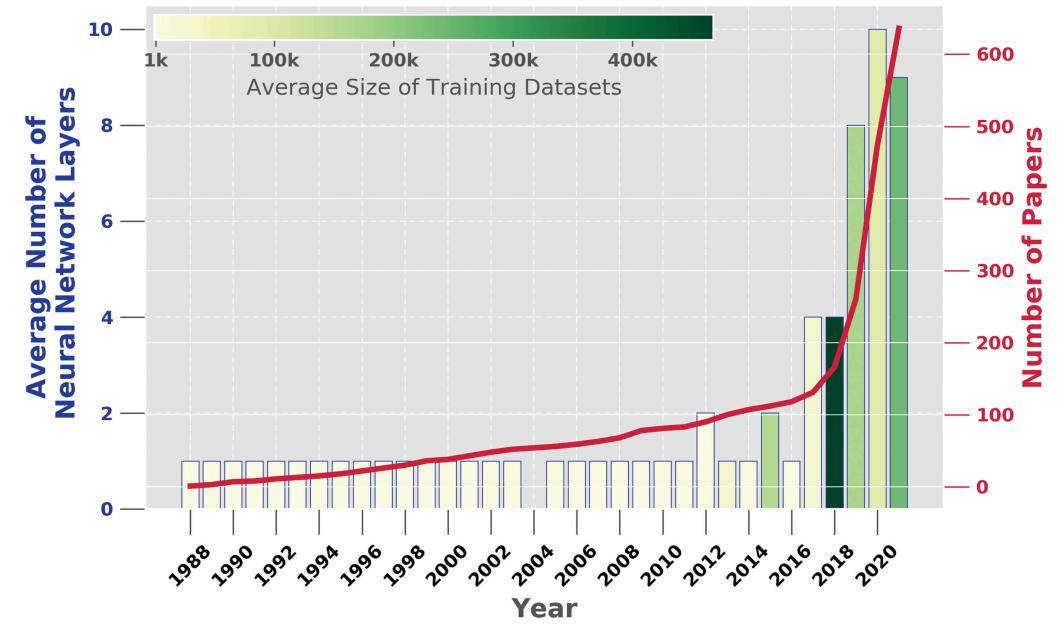
**Digital geology.** Digital representation of the geology of the conterminous United States. [Geology of the Conterminous United States at 1:2,500,000 scale; a digital representation of the 1974 P. B. King and H. M. Beikman map by P. G. Schruben, R. E. Arndt, W. J. Bawiec]

Bergen et al., Science 363, 1299 (2019) 22 March 2019

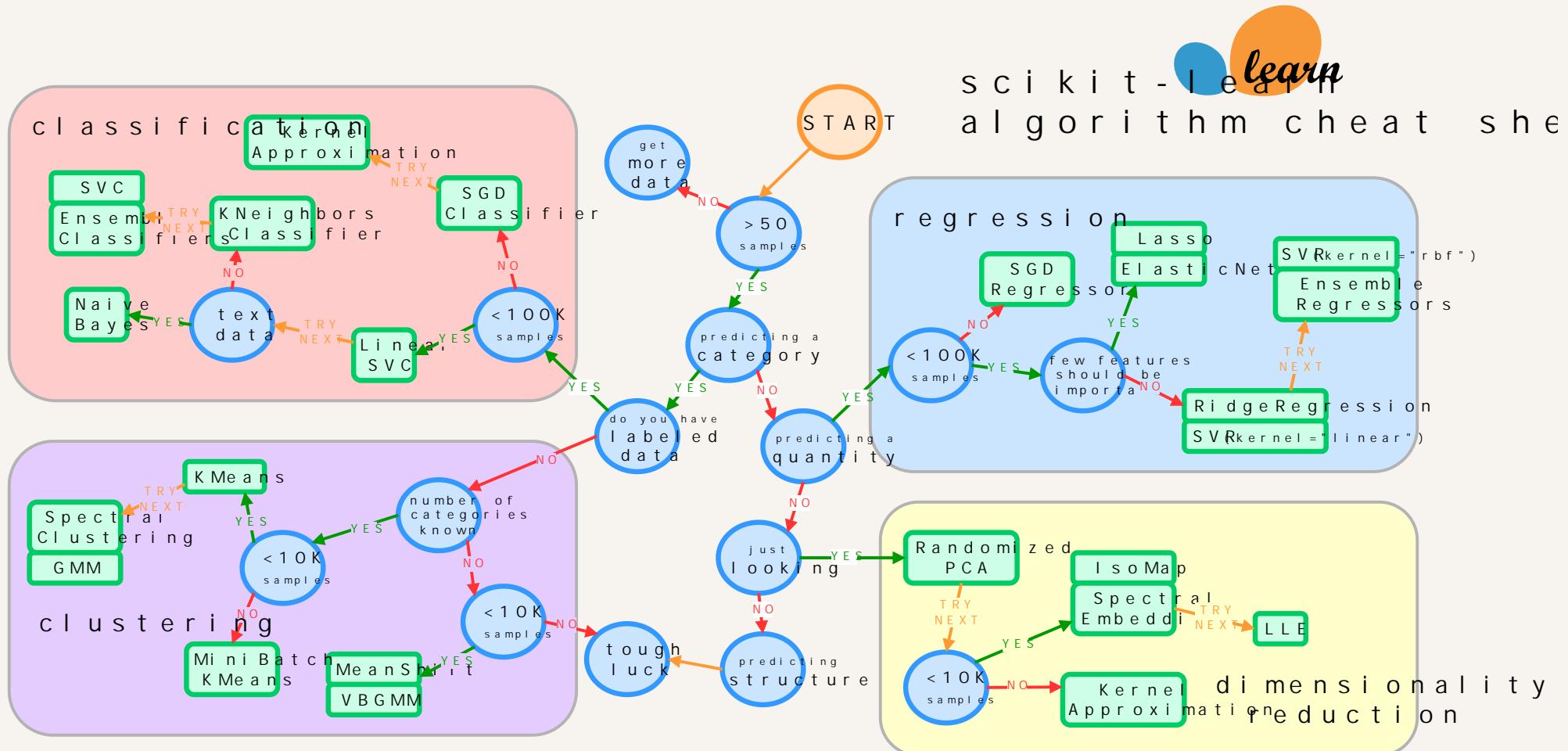
1 of 1

# Goals of the class

**Keep up** with the ongoing pace!

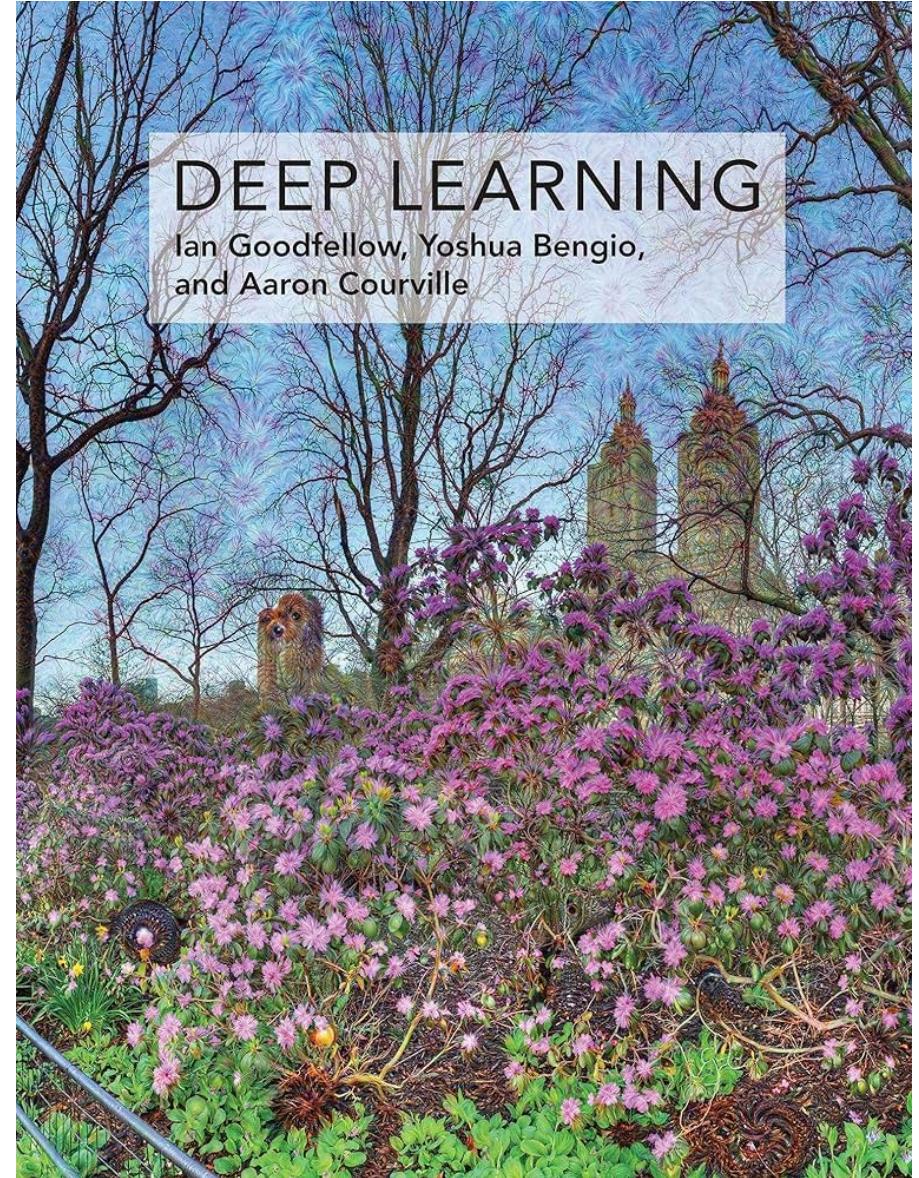


# The scikit-learn library



# The deep learning book

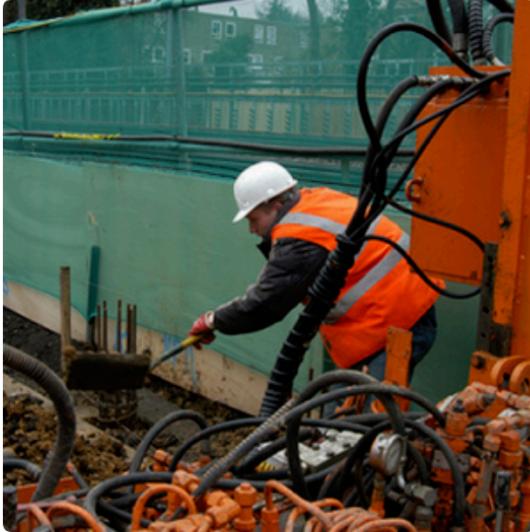
- **Historical** aspects
- **From scratch** to deep learning
- **Examples** and exercises
- **Freely** accessible online



# 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."



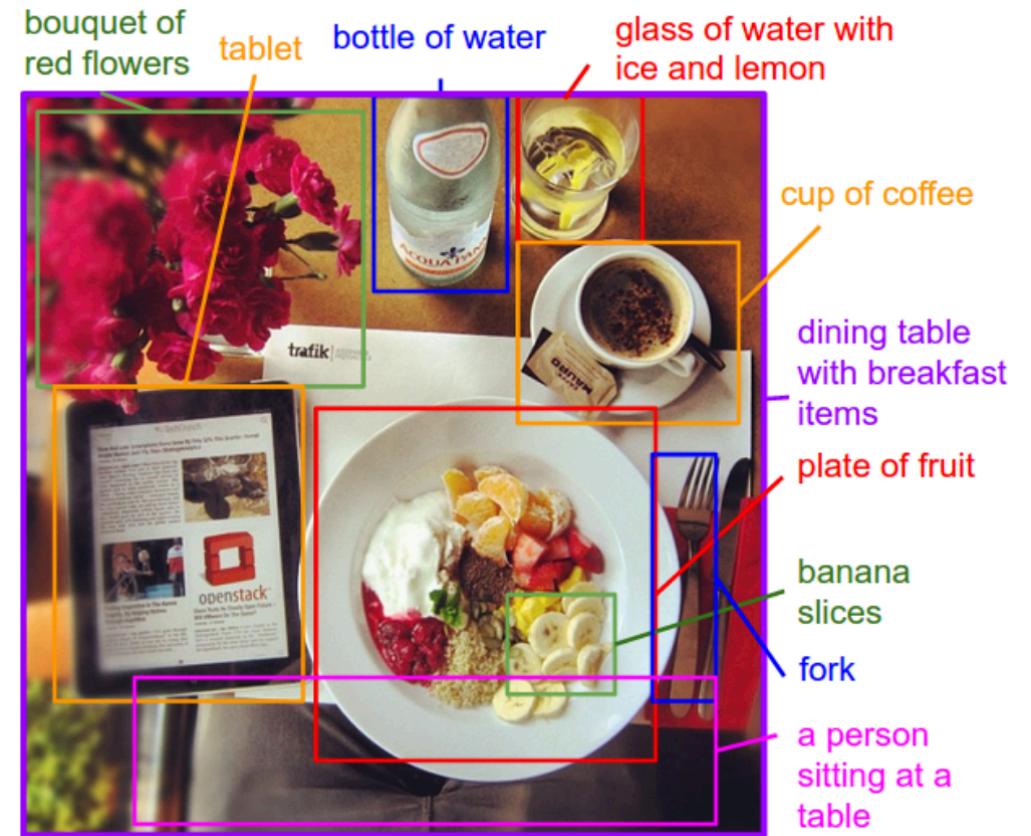
"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."

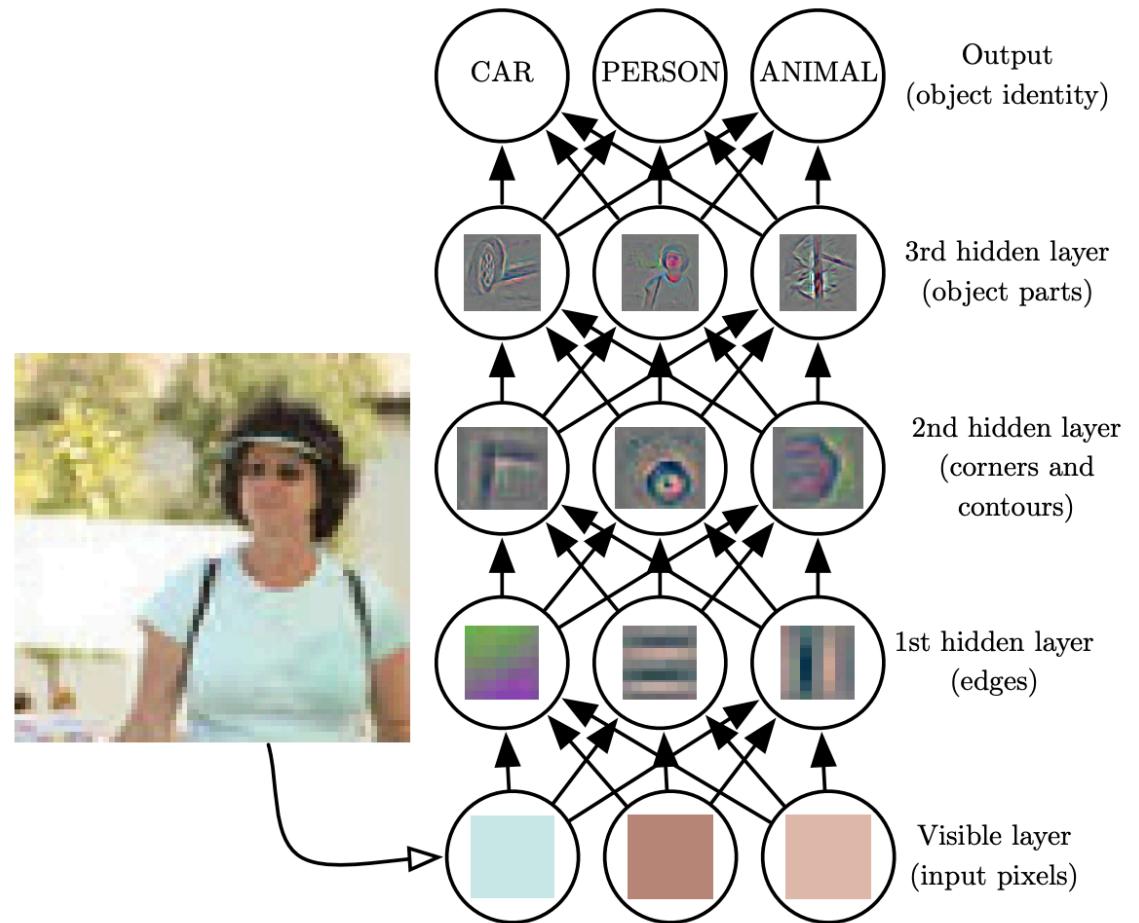
# From data to knowledge

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

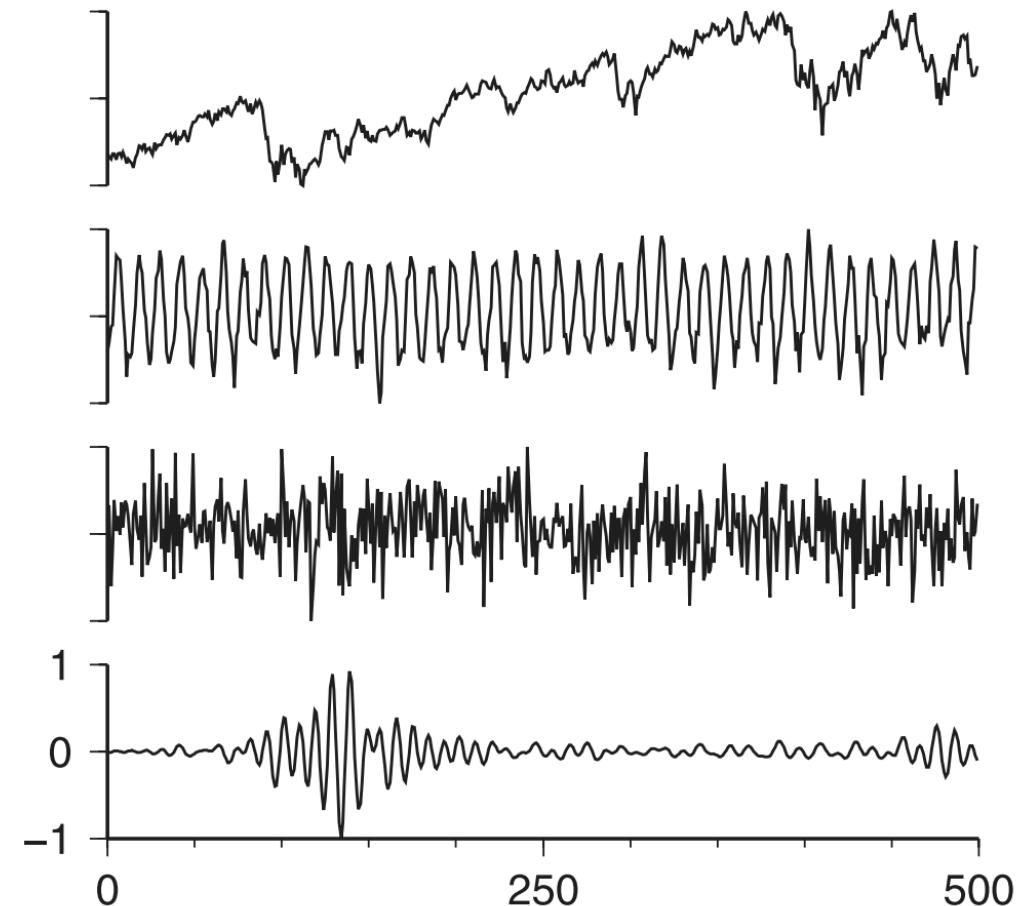


# From data to knowledge

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



**Can you spot  
the seismogram?**

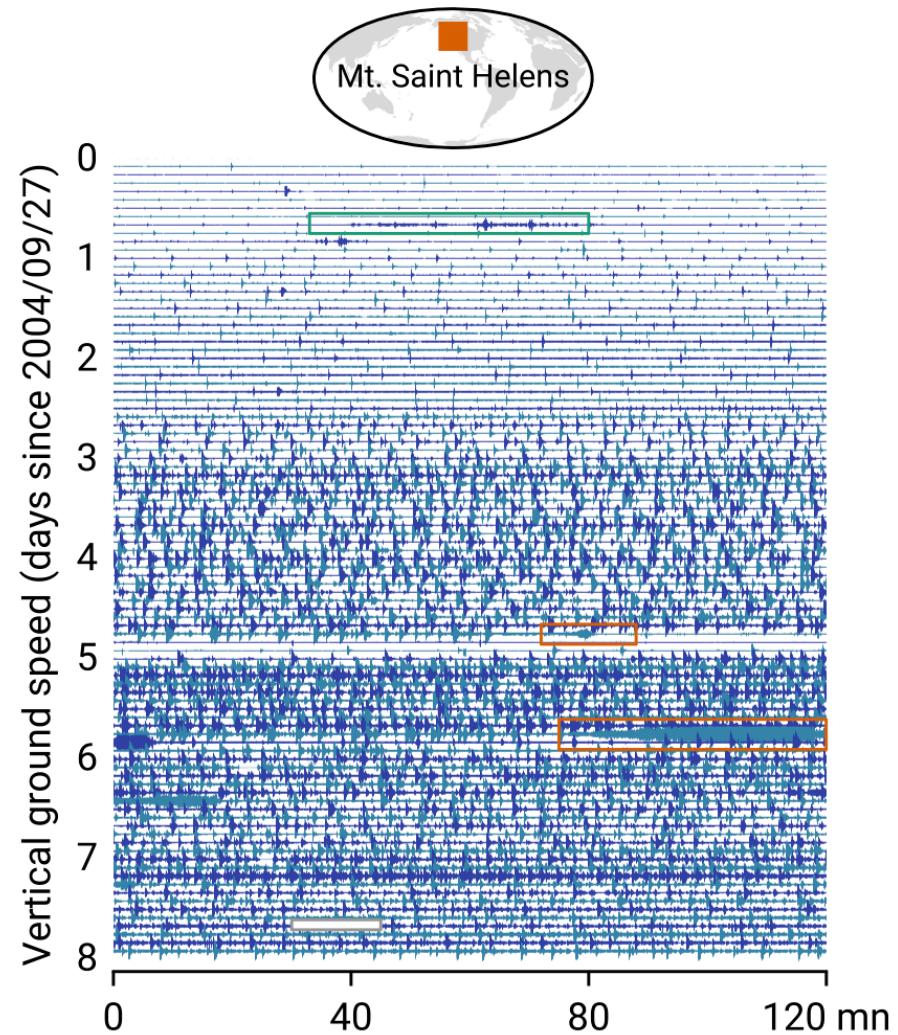


Valentine & Trampert (2012)

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

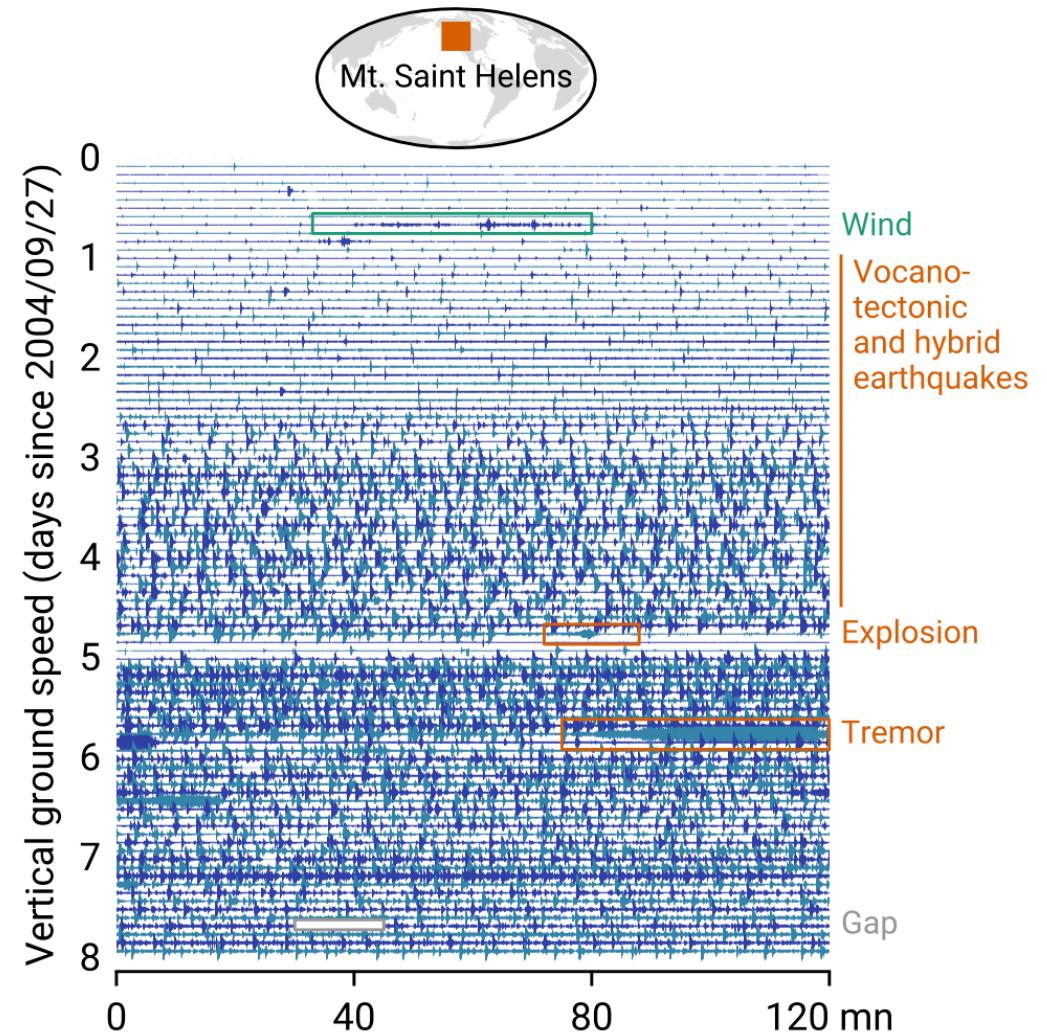
# Task: seismic events detection and classification

Most humans can pinpoint events.

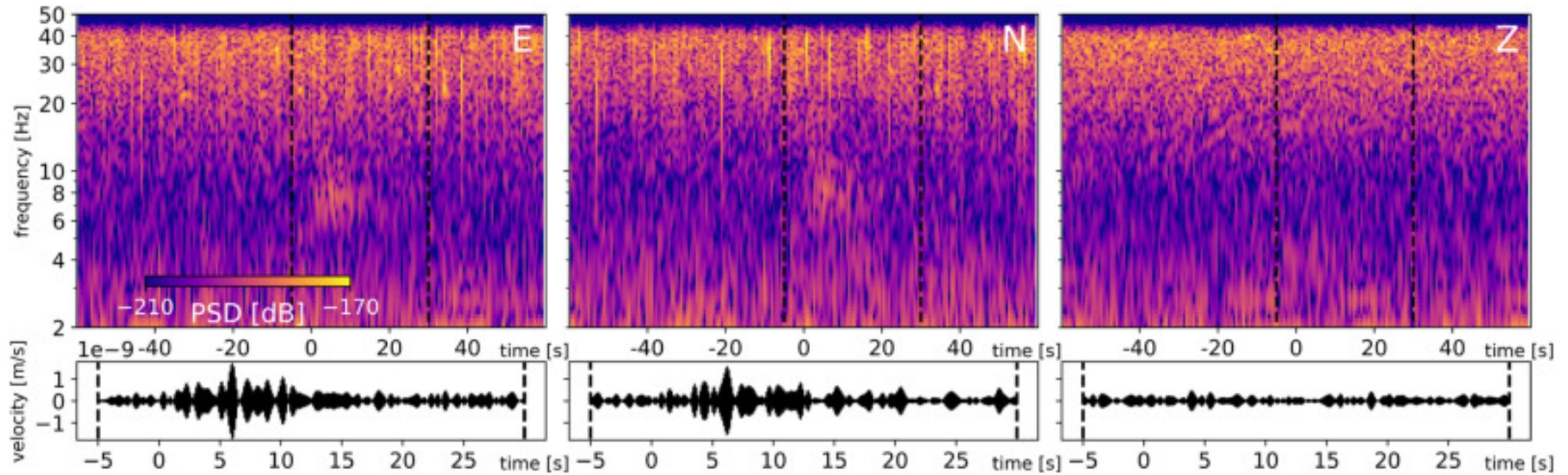


# Task: seismic events detection and classification

Most humans can pinpoint events.  
Experts can **classify** them.



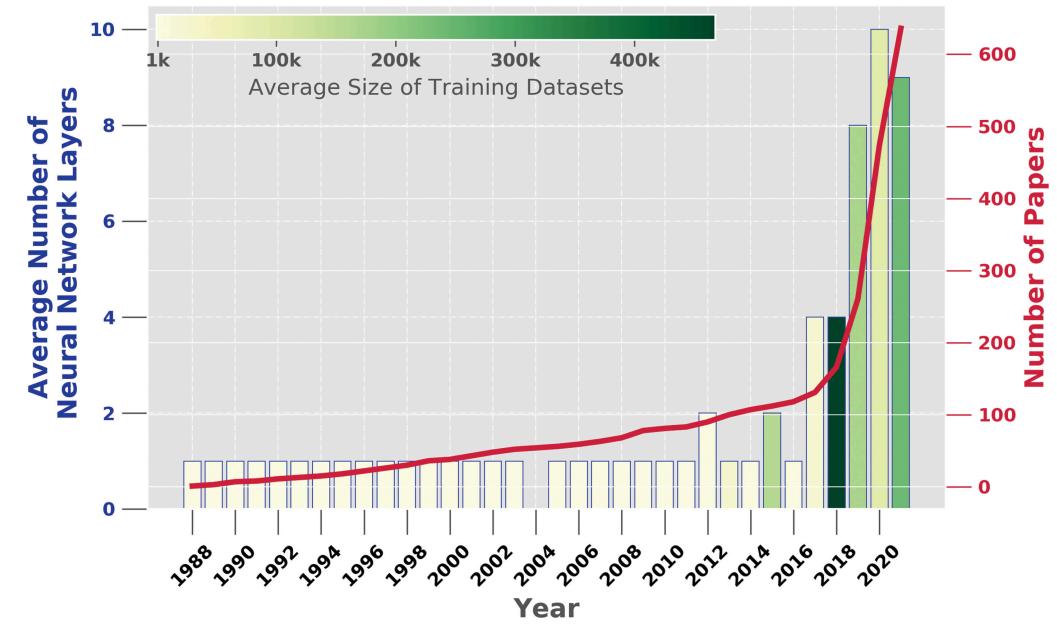
# Task: diving into previously unseed data



Expert-detected marsquake within continuous insight data.

# Machine learning tasks

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



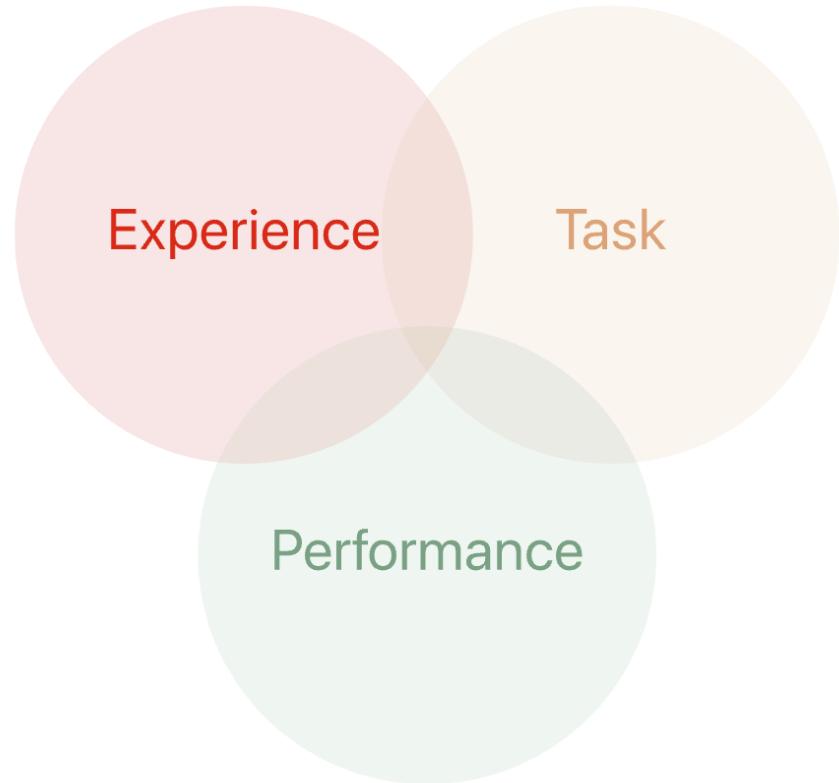
## 2. Definitions



**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  $x_i$  and labels  $y_i$  to learn from:

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



## the model

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

$$f : x \rightarrow \hat{y}$$



## the loss

A measurement of the model performance

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

**Learning** is equivalent to find the optimal parameters  $\theta$  that minimize the loss function  $L$ , as in

$$\theta^* = \operatorname{arg\,min}_{\theta} L(f_{\theta}(x), y)$$

# Vocabulary and symbols

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

$$\mathbf{x} = \mathbb{H} \times \mathbb{W} \times \mathbb{C}$$

$\mathbb{H}$  is the height,  $\mathbb{W}$  the width, and  $\mathbb{C}$  the channels. The labels are a category  $\mathbf{y}$  with

$$\mathbf{y} = \{0, 1, \dots, \mathbb{K}\}$$

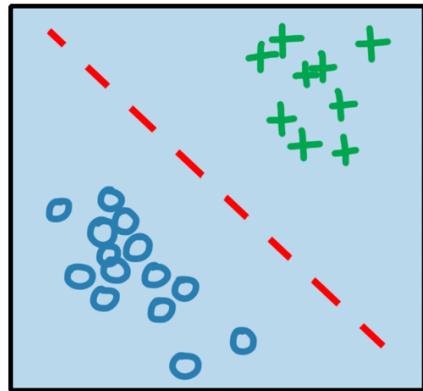
with  $\mathbb{K}$  the number of categories.

Note that  $\mathbf{y}$  is scalar in this case.

Symbol	Name
$\{\mathbf{x}_i\}_{i=1\dots N}$	Collection of <b>samples</b>
$\{\mathbf{y}_i\}_{i=1\dots N}$	Collection of <b>labels</b>
$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_F)$	Set of sample <b>features</b>
$\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$	Set of label <b>targets</b>
$N$	Dataset size
$F$	Feature space dimensions
$T$	Target space dimension
	Data space
	Label space

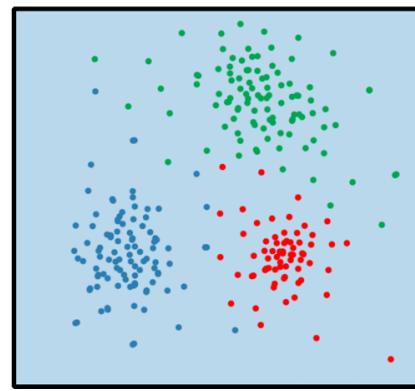
# Main types of machine learning strategies

supervised  
learning



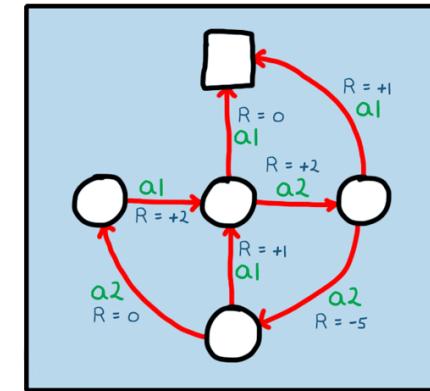
Predict  $y$  from  $X$  (regression,  
classification).

unsupervised  
learning



Learn some distribution  $p(x)$   
(clustering, reduction).

reinforcement  
learning

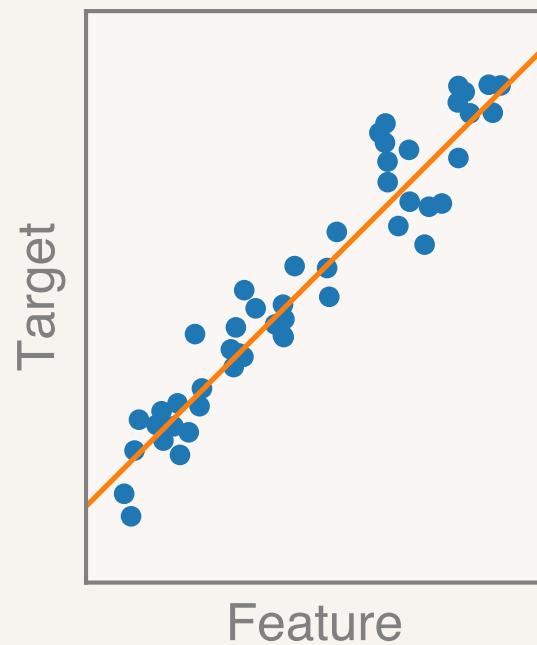


Learn a policy to maximize a  
reward (gaming, robotics).

# 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

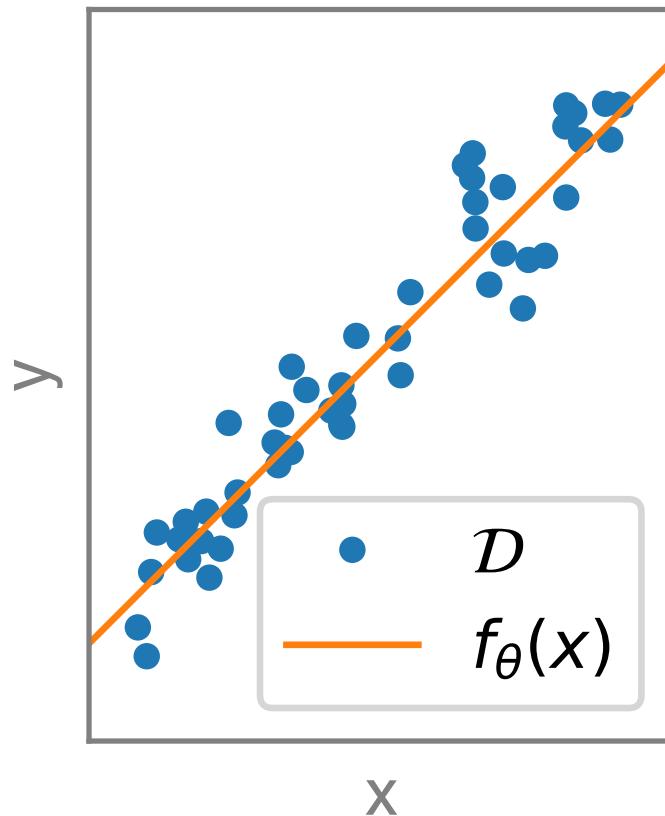
The set of  $N$  samples  $x_i$  and corresponding labels  $y_i$  such as

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

## Formulation

Optimize the parameters  $\theta$  of a function  $f$  to predict  $y$  from  $x$ . Find the optimal parameters  $\theta$  that minimize  $L$ , such as

$$\theta = \text{argmin}_\theta L(f_\theta(x), y).$$



# Linear regression

## Linear model

coefficients =  $(a, b)$  that map  $x$  to  $y$  with

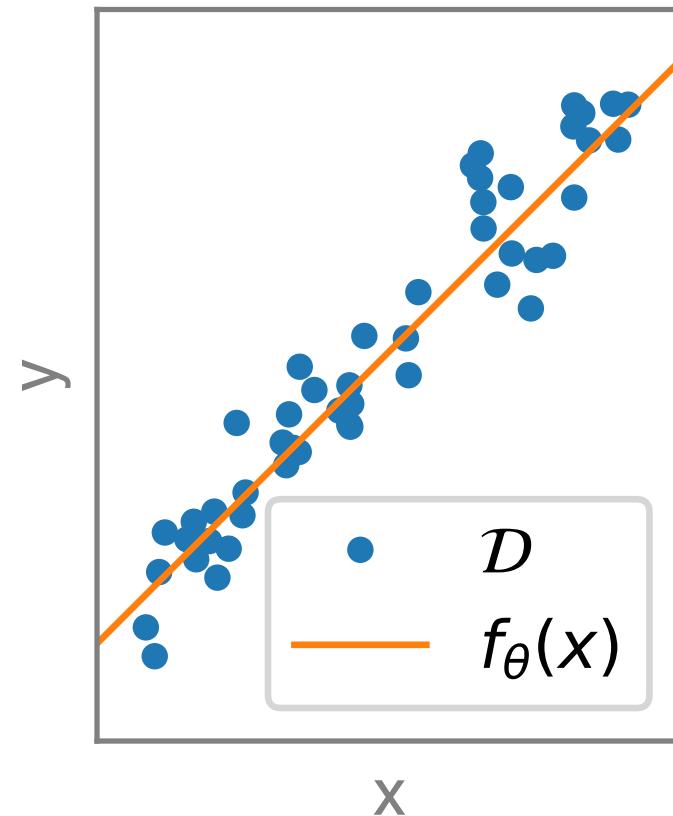
$$f : x \rightarrow y = ax + b.$$

## Loss function

For instance mean squared error:

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

How do we minimize the loss?



# Naive attempt: grid search

## Implementation

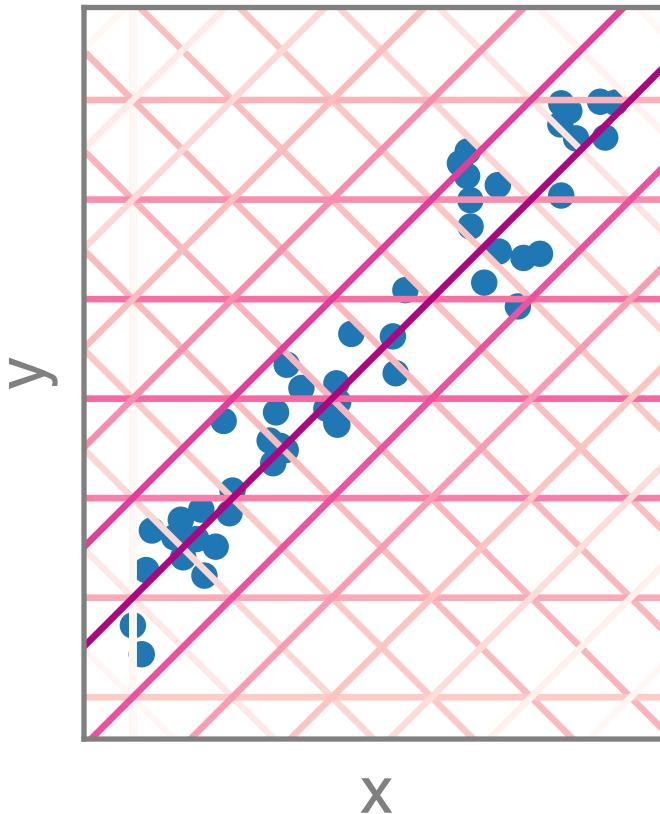
Find  $\theta$  among regularly spaced tested values of  $\theta$ .

## 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

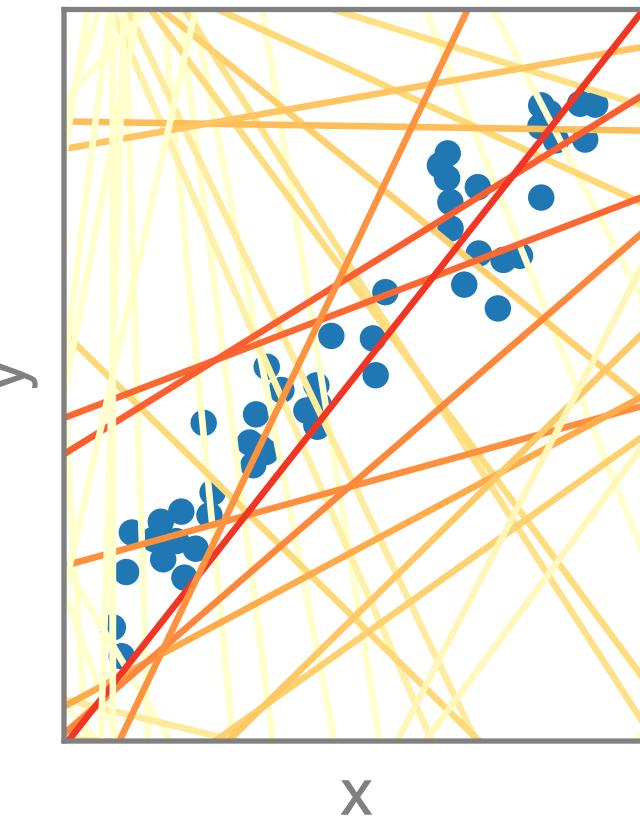
Sample random values of  $x$ , keep the best one.

## Pros

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

## Cons

Not exhaustive, can be slow to converge.



# Gradient descent

## Implementation

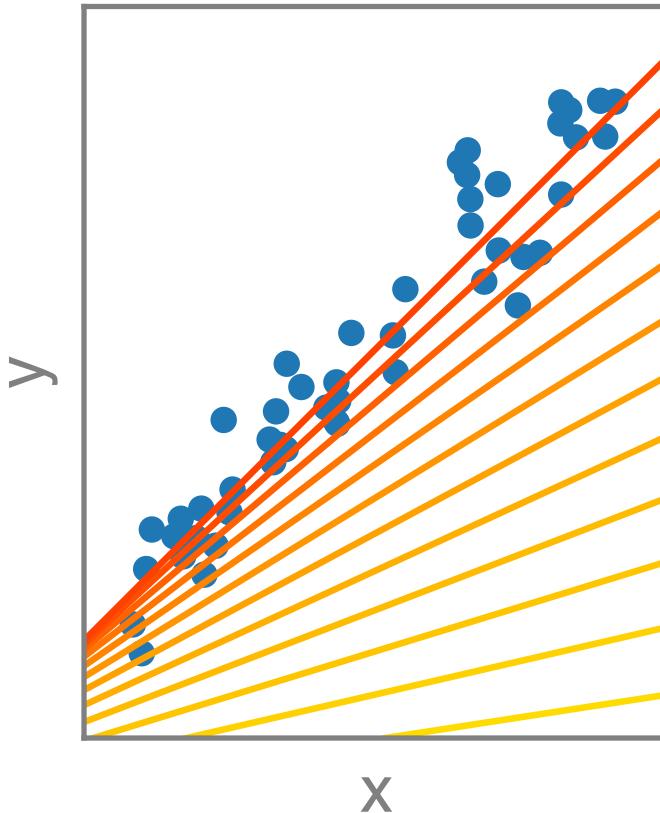
Estimate the gradient of  $L$  w.r.t. the parameters , update the parameters towards gradient descent.

## Pros

Converges faster than random search.

## Cons

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



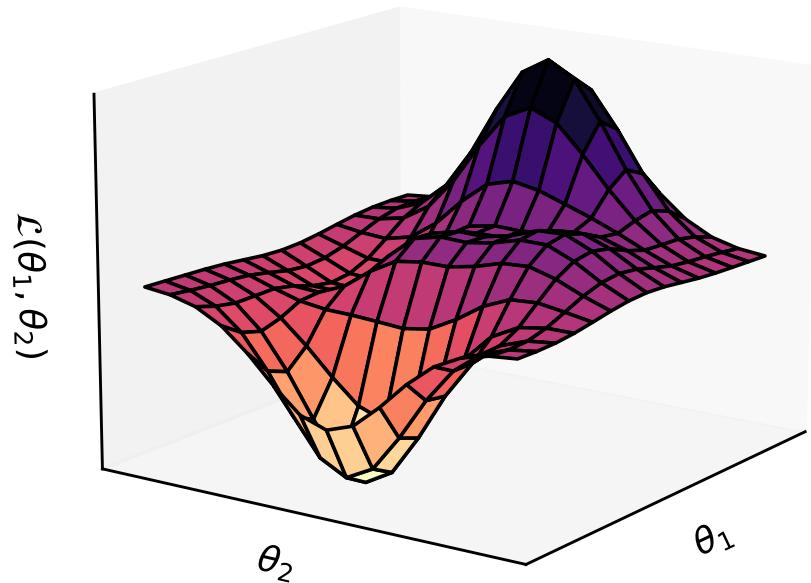
# Gradient descent

## Implementation

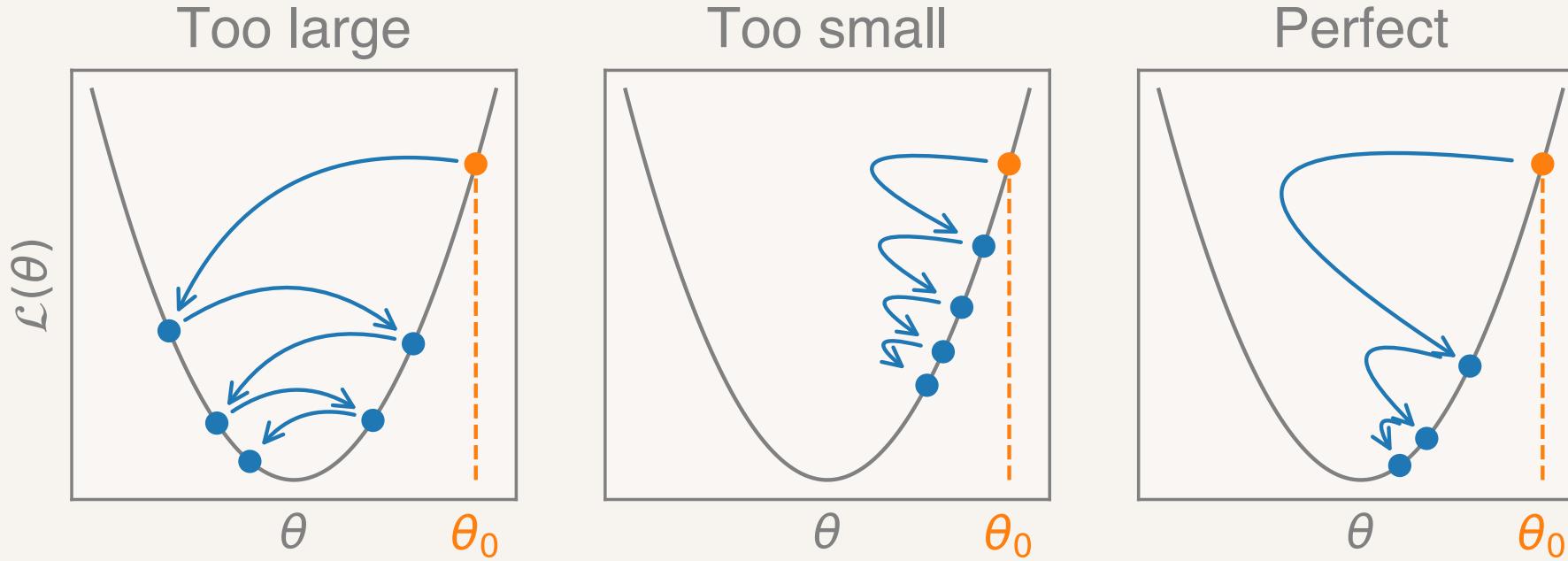
1. Initiate model  $= (a_0, b_0)$
2. Compute gradient  $L( )$
3. Update  $- L( )$
4. Repeat until convergence

## Hyperparameters

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



# How to deal with learning rate?



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

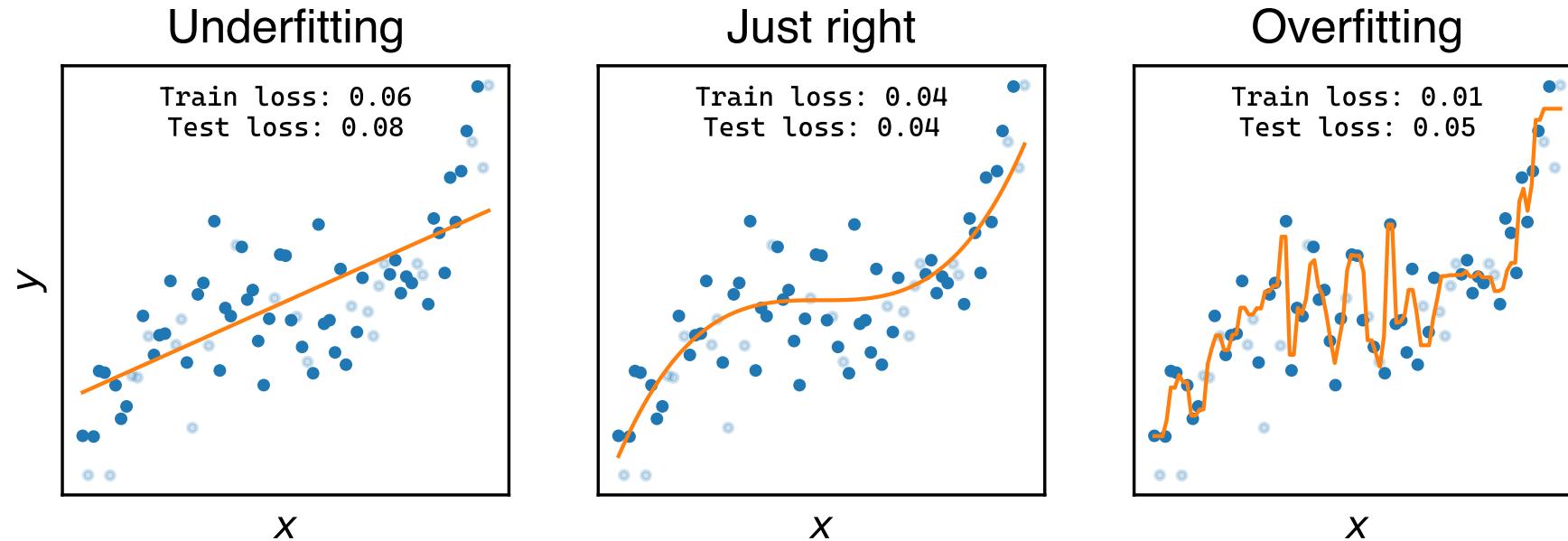
More about that in the deep learning lectures.

# The problem of overfitting



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

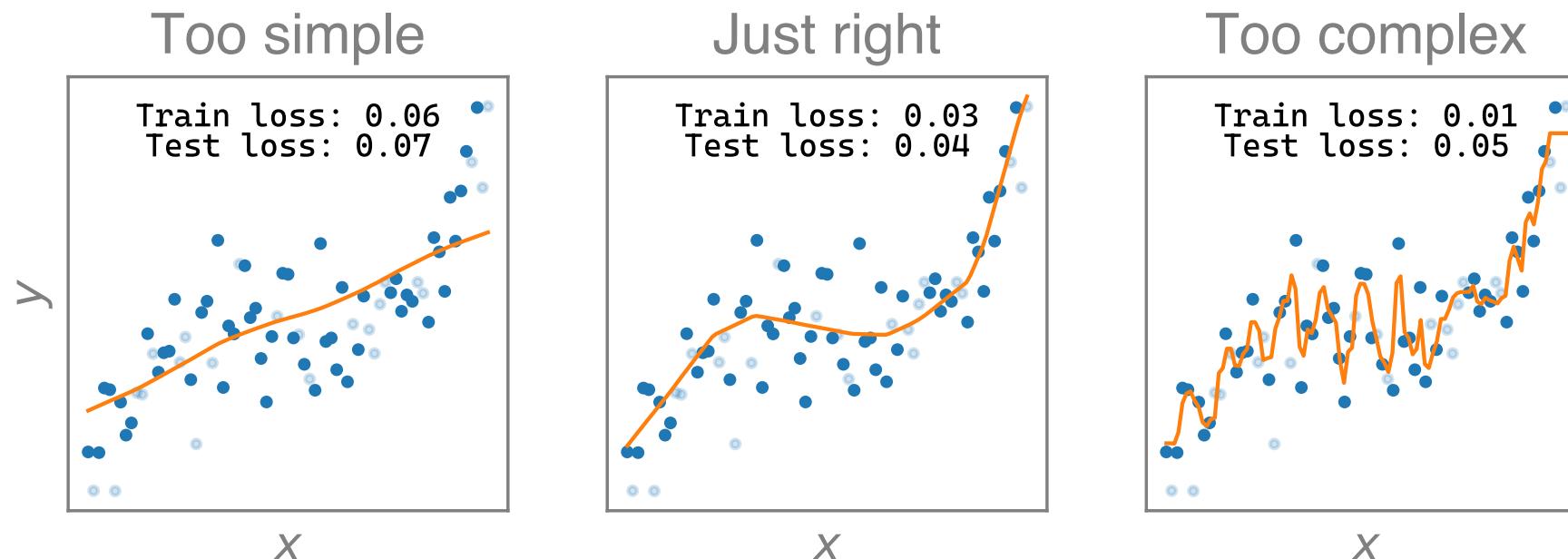
# Key concepts to prevent overfitting: split the dataset



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  $R$  to the loss  $L_R = L + \gamma R$ , with  $\gamma$  the regularization strength



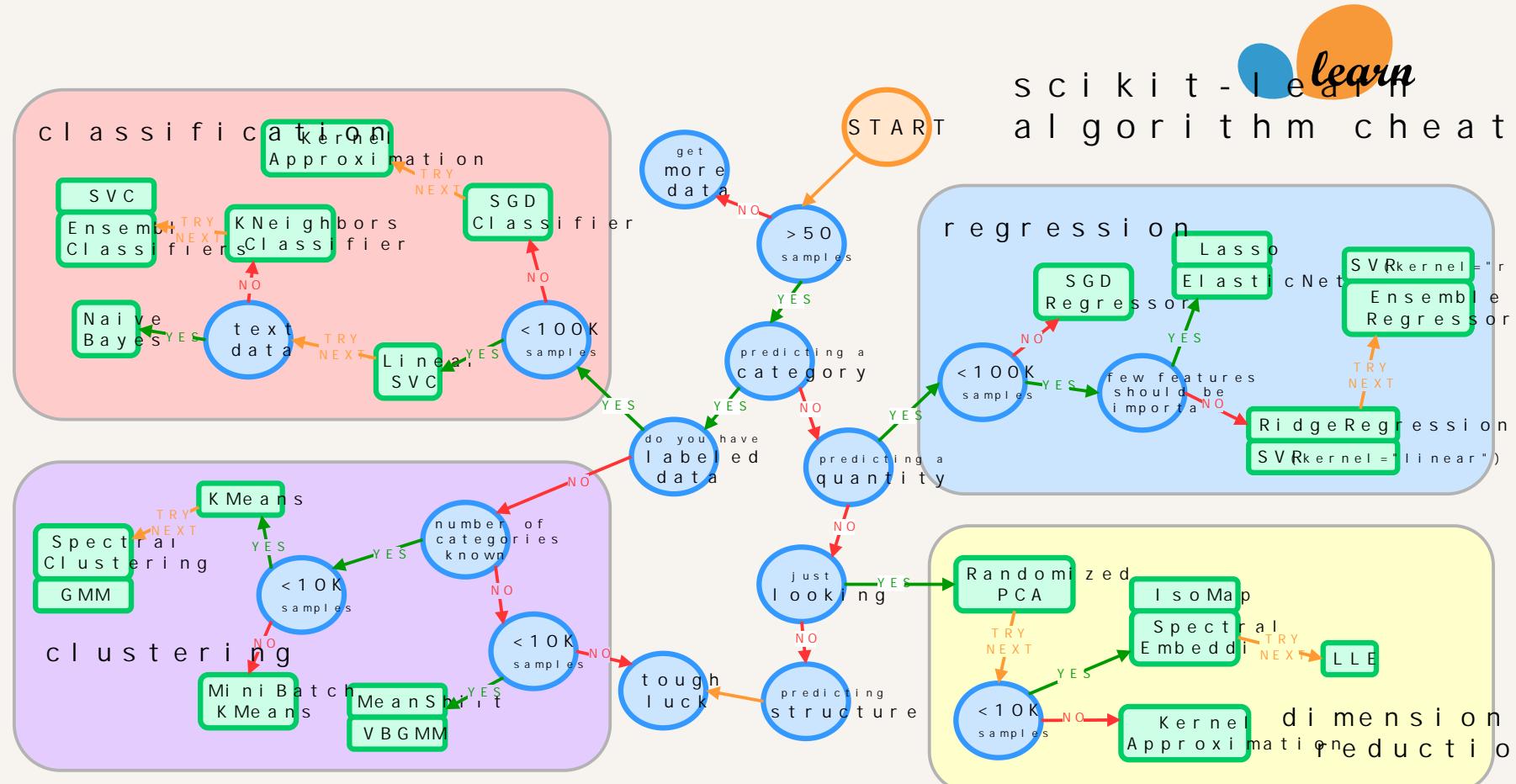
The regularization penalizes the model's complexity.

# Why so many regression algorithms?

Because of combination of models, losses, and regularizations. The [scikit-learn.org](http://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

