Outline of this lecture

1. General introduction

2. Short selection of [simple] applications of supervised learning to chemistry

3. Tutorials / Live demonstrations ← Jupyter notebooks

github repository

Python in the [Physical] Chemistry Lab

[PytChem]

https://github.com/rpoteau/PytChem

General context

Context

Artificial Intelligence (AI)

intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals

Goals

reasoning & (basic) problem solving

knowledge representation

planning: making choices and hierarchy of events

learning (*i.e.* machine learning)

natural language processing

perception of the world from sensors

ability to move and manipulate objects

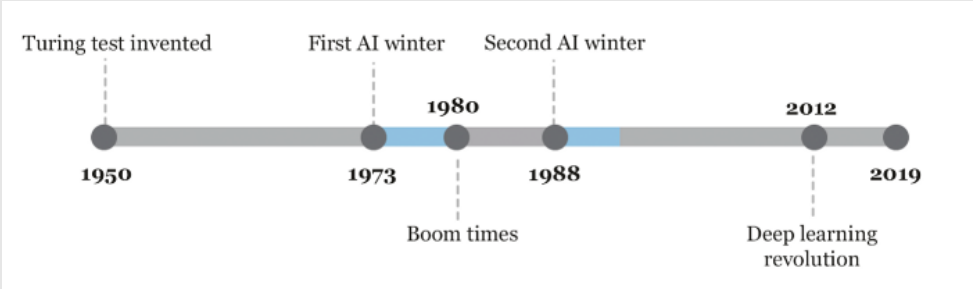
simulating human affects

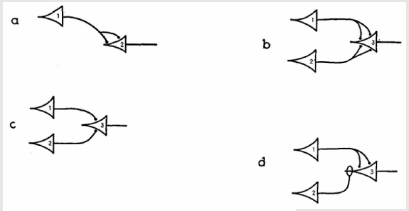
long-term goal: ability to solve an arbitrary problem

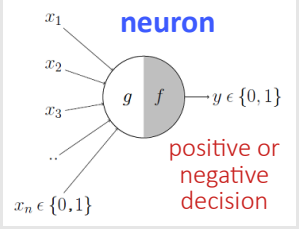
N.B. in some fields "artificial intelligence" means "machine learning with neural networks"

Artificial Intelligence & Deep learning timelines

Since the first McC&P mathematical model for a neuron & the pioneering work of Turing, AI has a long history, with two "winters"







were the first to suggest that neural activity is computational

1940 Dark Era Until 1940

???

1943 Neural Nets McCulloch & Pitt

1950 Computing Machinery and Intelligence Alan Turing

1958 Perceptron Rosenblatt

1960 ADALINE Widrow & Hoff

1969 XOR problem Minsky & Papert

1974 Backpropagation Werbos (and more)

1980 Neocogitron Fukushima

1980 Self Organizing Map Kohonen

1982 Hopfield Network John Hopfield

1985 Boltzmann Machine Hinton & Sejnowski

1986 Multilayer Perceptron Rumelhart, Hinton & Williams

1986 Restricted Boltzmann Machine Smolensky

1986 RNNs Jordan

1990 LeNet Lecun

1997 Bidirectional RNN Schuster & Paliwal

1997 LSTMs Hochreiter & Schmidhuber

2006 Deep Belief Networks - pretraining Hinton

2006 Deep Boltzmann Machines Salakhutdinov & Hinton

2012 Dropout Hinton

2014 GANs Goodfellow

2017 Capsule Networks Sabour, Frosst, Hinton

Two neural networks contest with each other

W. S. McCulloch & W. Pitts (**1943**)
A logical calculus of the ideas immanent in nervous activity
Bull. Math. Biophys. **5**, 115-133

Context

Artificial Intelligence and Scientific Research

FIDLE

https://fidle.cnrs.fr

1st paradigm

Experimental science

2nd paradigm

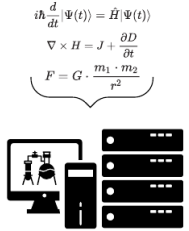
$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle$$

$$\nabla \times H = J + \frac{\partial D}{\partial t}$$

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2}$$

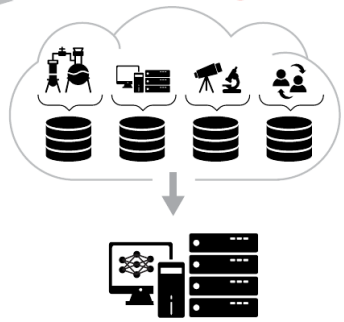
Theoretical science

3rd paradigm



Computational science

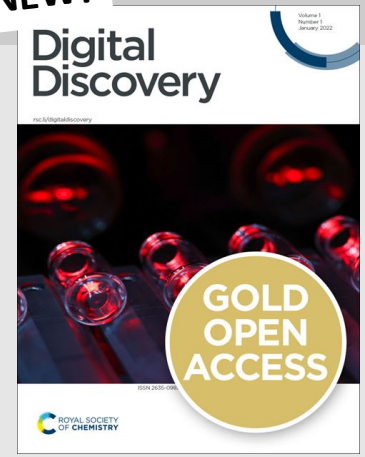
4th paradigm[1]



Data-driven science

1600 1950 2000

NEW!



“a new forum for data-driven approaches to scientific discoveries”

experimental and computational work

all topics related to the acceleration of discovery (screening, robotics, databases and advanced data analytics)

broadly defined, but anchored in chemistry

(1) Hey, T.; Tansley, S.; Tolle, K. The Fourth Paradigm: Data-Intensive Scientific Discovery; The Fourth Paradigm: Data-Intensive Scientific Discovery; Microsoft Research, 2009

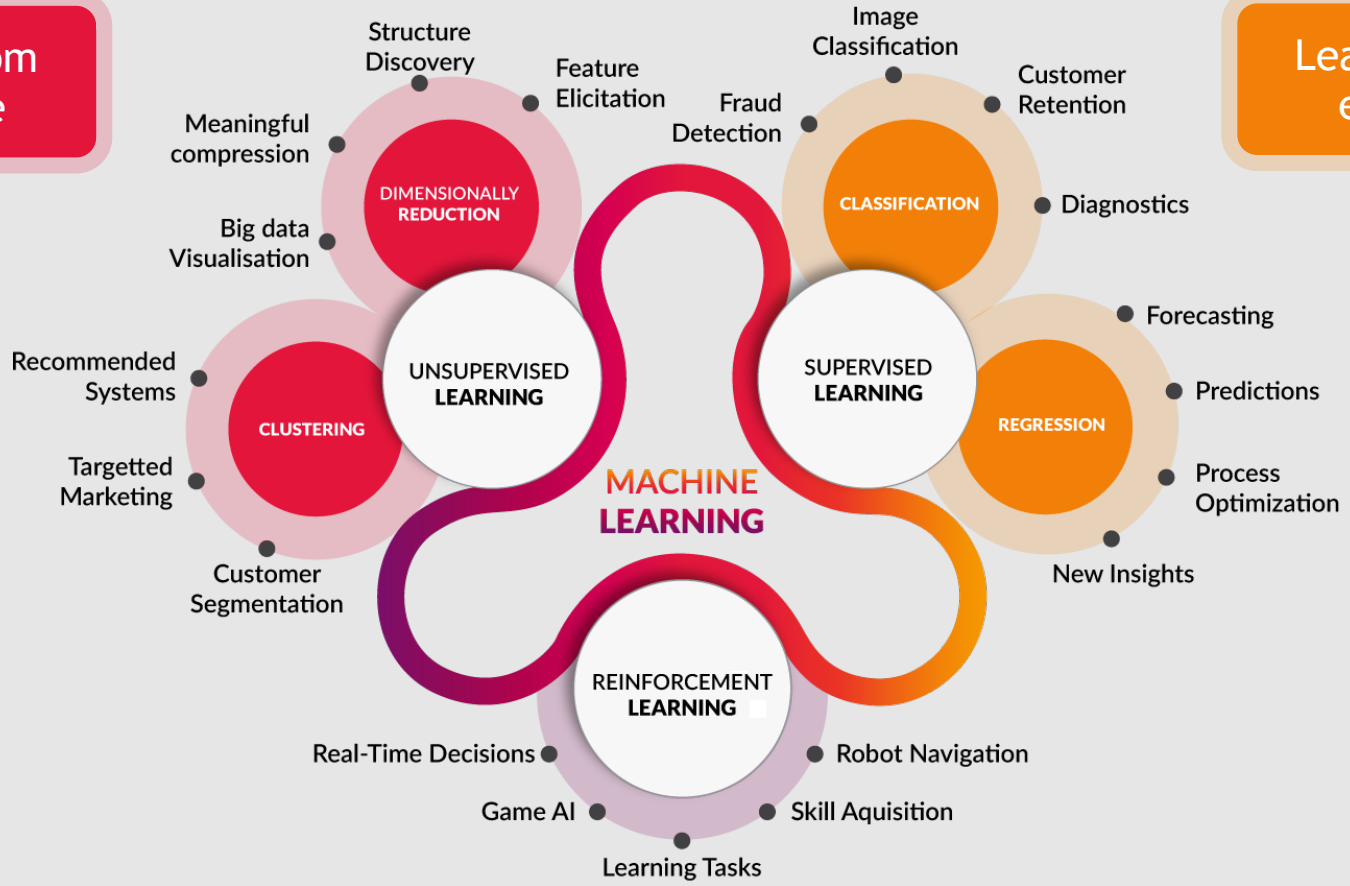
LPCNO

Machine Learning

Machine learning

Learning from data alone

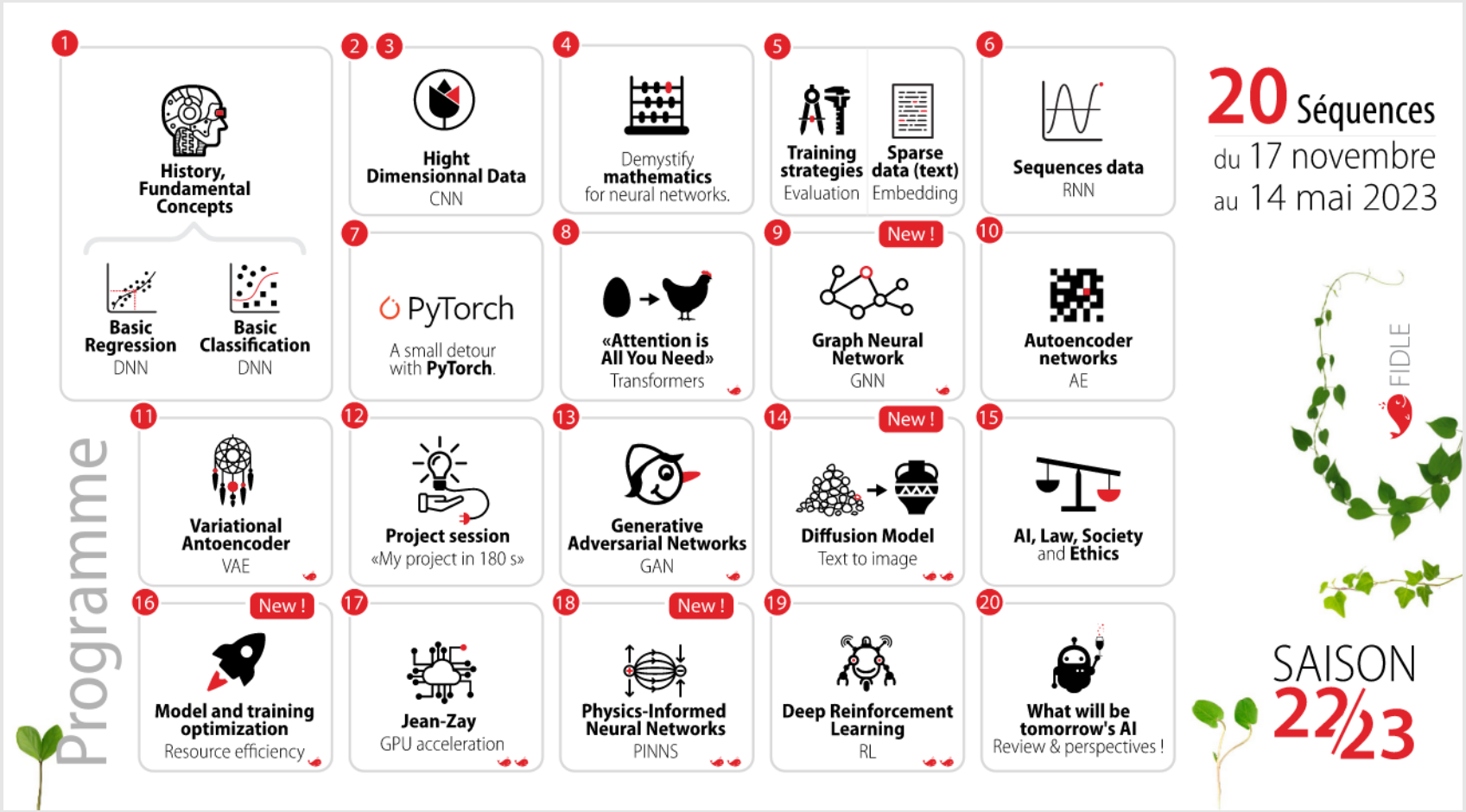
Learning from examples



https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d

Introduction to deep learning

https://fidle.cnrs.fr



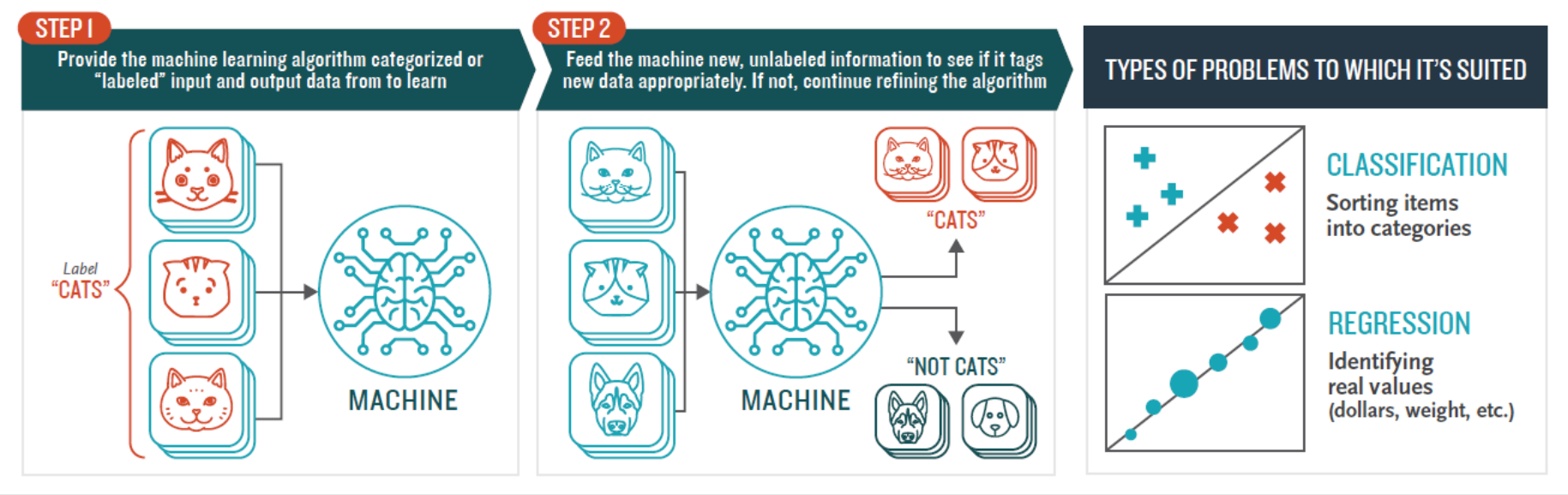
Supervised learning

Data are provided along with the desired output (*i.e.* labelled data)

Example of cats detection:

- **collect thousands of images of cats**
- **draw a bounding box around each cat**
- **feed the entire dataset to the machine so it can learn all by itself**

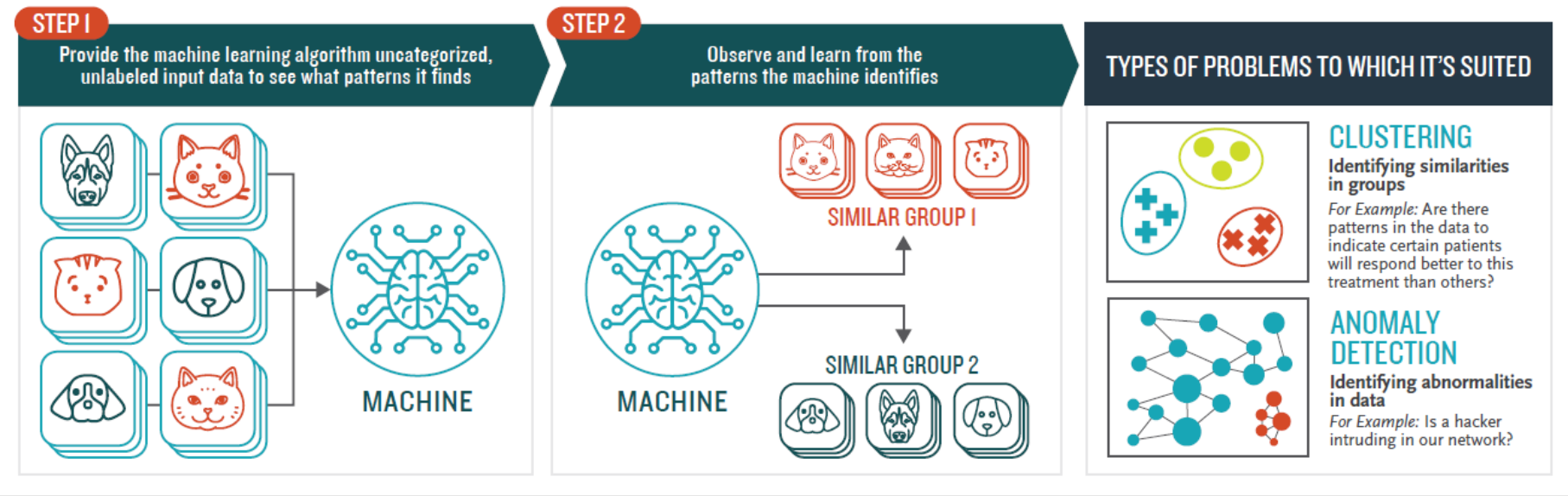
Learning from examples



Unsupervised learning

Learning from data alone

- Just provide data
- Let the machine find out (or cluster) the patterns in the dataset



How to develop home-made ML tools?

mathematica

Python is a high-level, interpreted, object-oriented, general-purpose programming language

Core philosophy:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

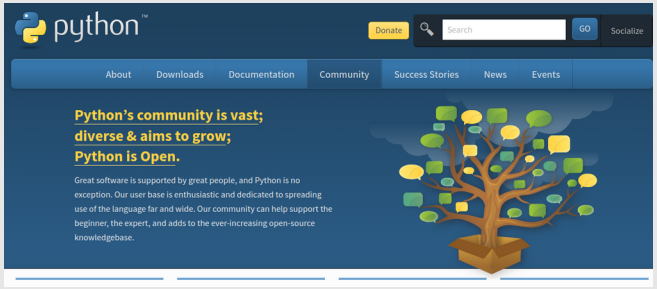
~ 250 additional libraries are available
- data science
- machine learning
- modern scientific computation
- visualization

Using Python as an everyday tool for scientific calculation and computation

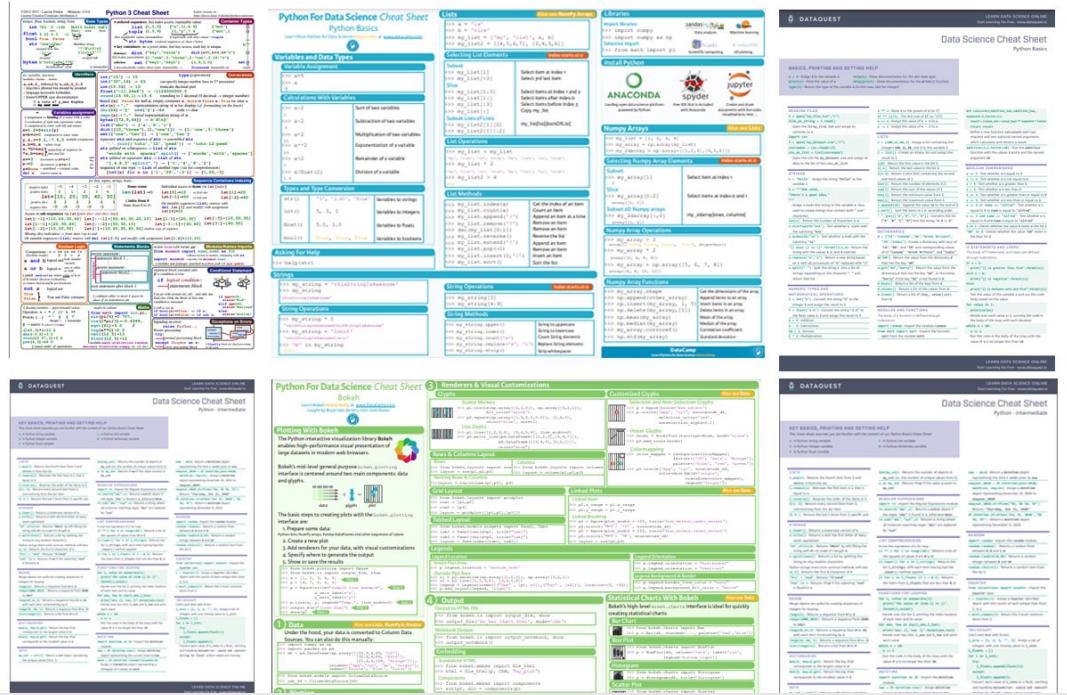
(it can even replace excel… by far)

→ basic knowledge of programming languages (variables, arrays, loops, conditional tests...)

→ enthusiastic and vast community



→ cheatsheets & cut/paste



Uneasy?
- yes and no
- important initial investment
- worth the effort

ML using the Python programming language

https://scikit-learn.org/

Machine learning in Python with scikit-learn

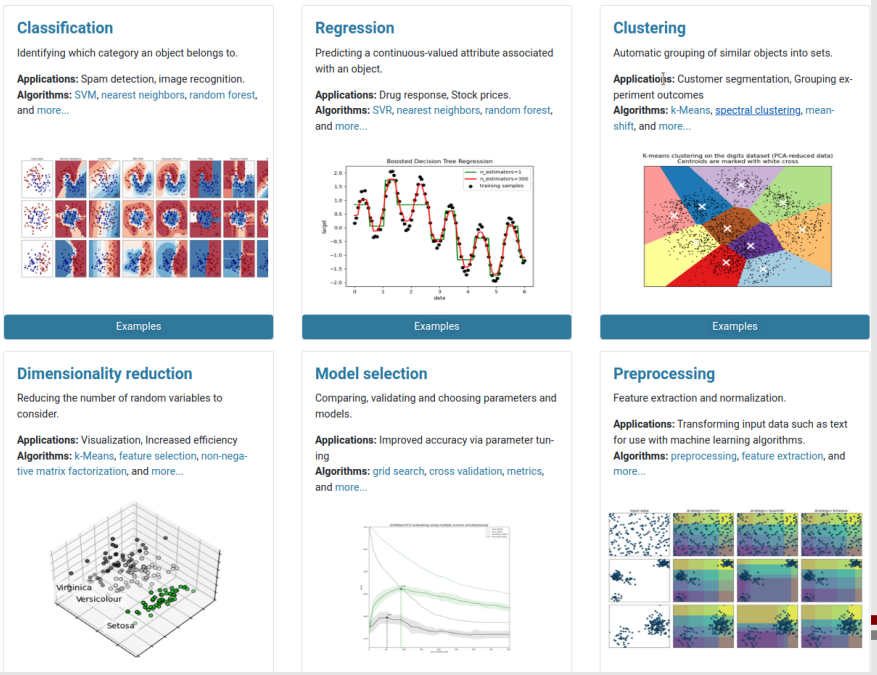
Simple and efficient tools for predictive data analysis

Accessible to everybody, and reusable in various contexts

Built on NumPy, SciPy, and matplotlib

INRIA took leadership of the project and made the first public release on February 2010

3-clause BSD License (permissive free software license, compatible with the GNU GPL)

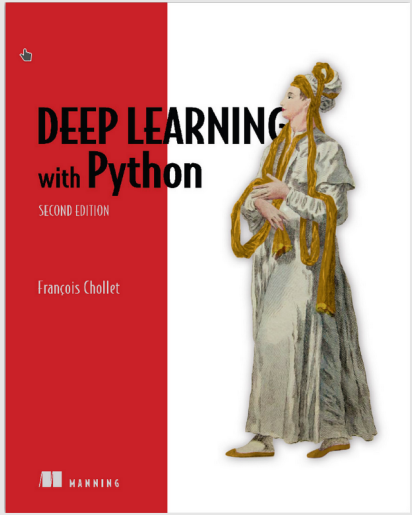


FUN MOOC

Machine learning in Python with scikit-learn

Keras

https://keras.io/



High Level Deep Learning Application Programming Interface (API)

By François Cholet (Google)

Part on TensorFlow since 2017

MIT license (permissive free software license)

how to start?

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
```

TensorFlow

https://www.tensorflow.org/

Google Brain's second-generation system

Supported by Google

Low level API

Apache license (yet another permissive free software license)

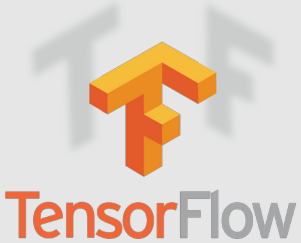
https://pytorch.org/

From Torch library
Supported by Facebook
BSD licence
(permissive free software license)

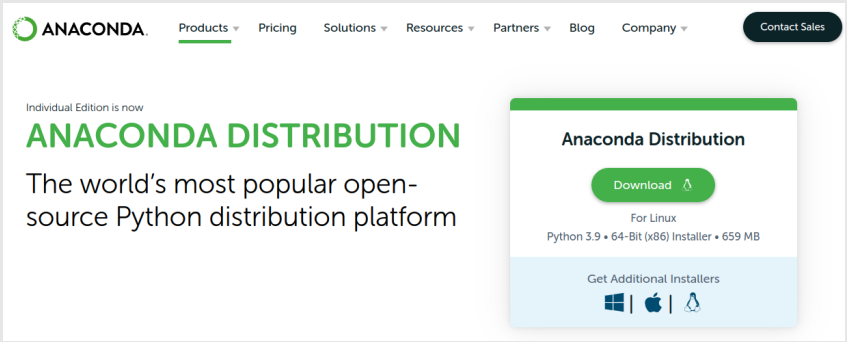
An open source machine learning framework that accelerates the path from research prototyping to production deployment

Widely used in the field of AI research

The ML galaxy



Python programming language



Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment

JupyterLab is a web-based interactive development environment for notebooks, code, and data

Users can arrange workflows in **data science**, **scientific computing**, and **machine learning**

you can save your everyday data manipulation / visualization as you do in your chemistry laboratory notebooks