

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**  python™
[PytChem]

<https://github.com/rpoteau/PytChem>

General 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

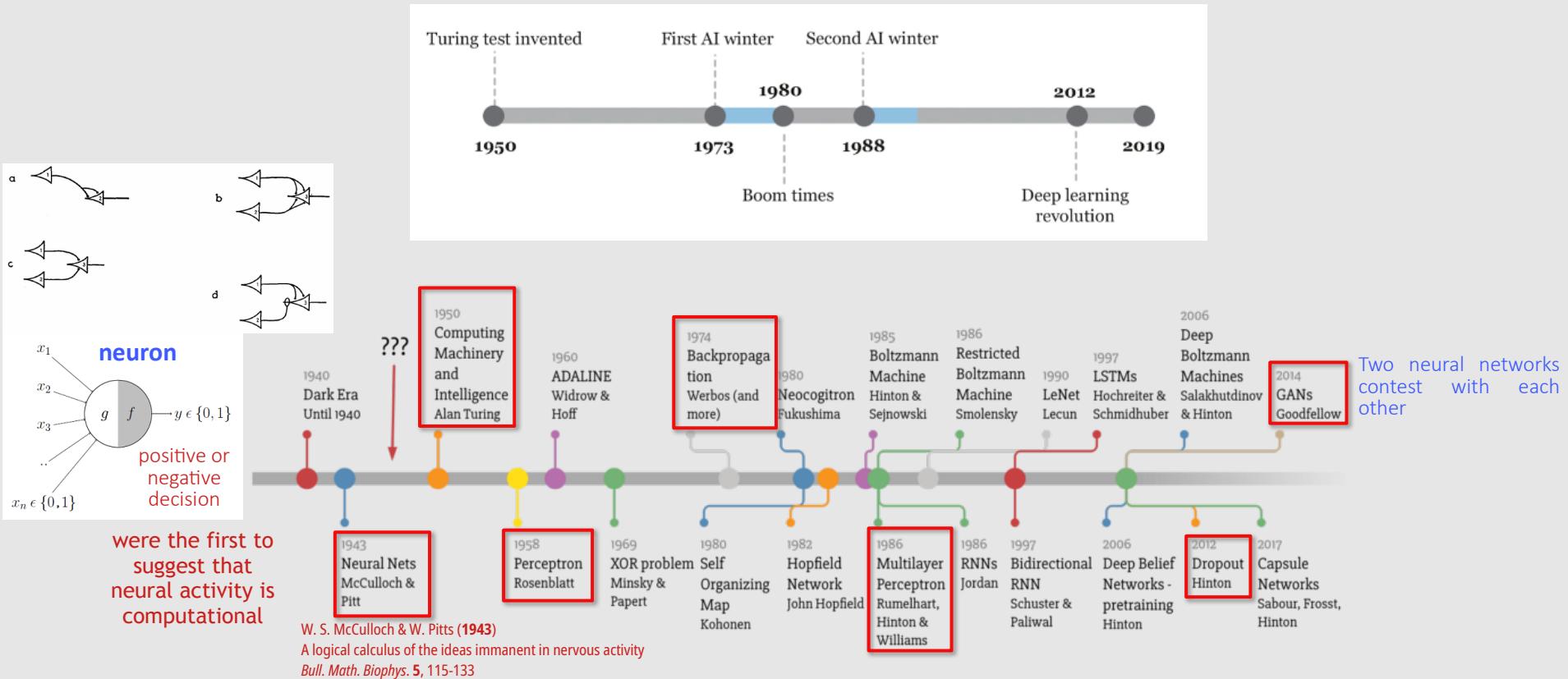
simulating human affects

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

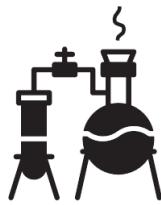
Since the first McC&P mathematical model for a neuron & the pioneering work of Turing, AI has a long history, with two “winters”





<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

Artificial Intelligence and Scientific Research

3rd paradigm

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

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

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



Computational science

4th paradigm¹



Data-driven science



1600



1950



2000



NEW!

Digital Discovery



GOLD
OPEN
ACCESS

“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)

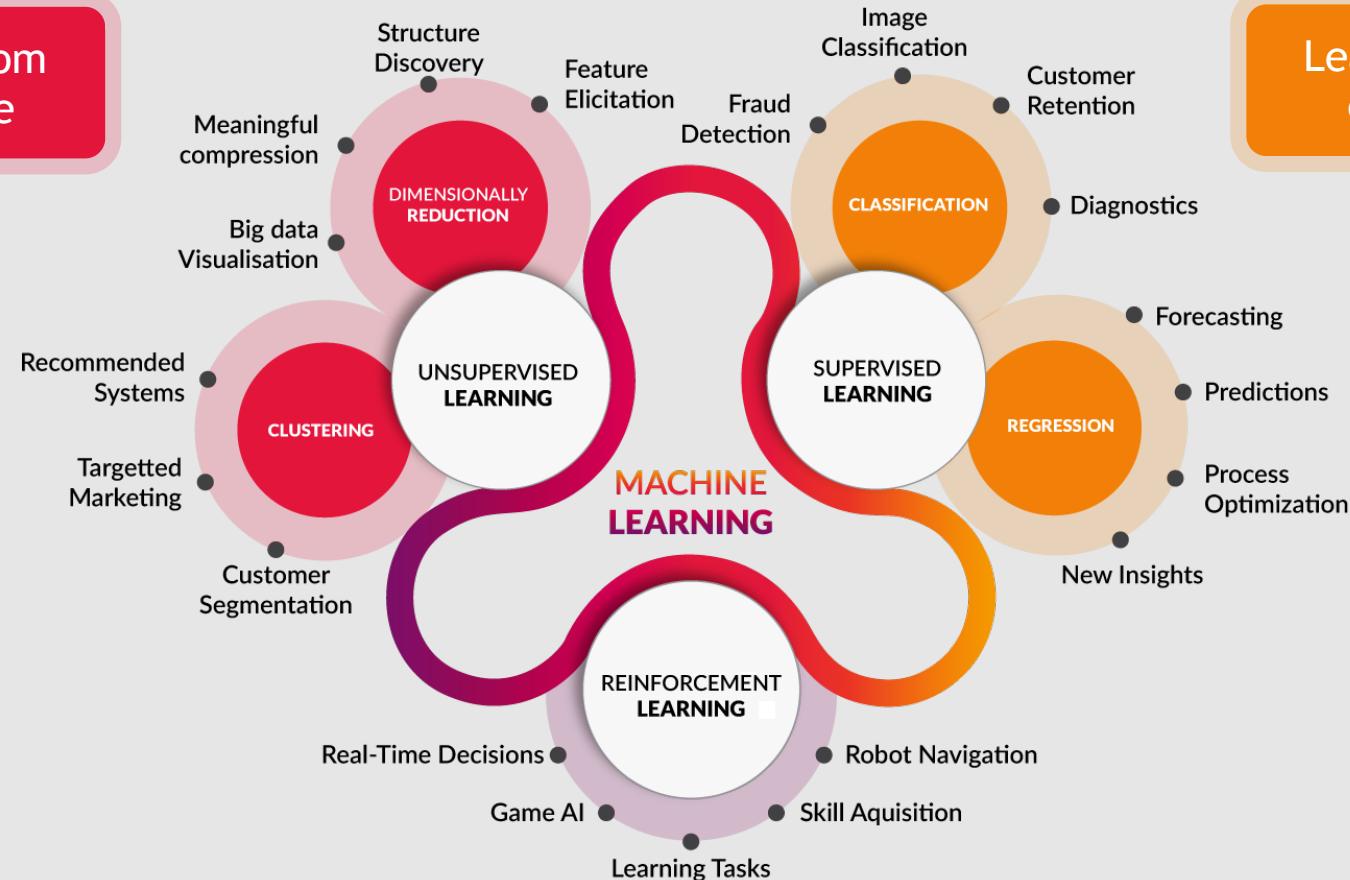
broadly defined, but anchored in chemistry



Machine Learning

Machine learning

Learning from data alone



Learning from examples

Introduction to deep learning



<https://fidle.cnrs.fr>

Version: Monday, June 17, 2024



Programme

1



History,
Fundamental
Concepts

Basic
Regression
DNN

Basic
Classification
DNN

2



Hight
Dimenisional Data
CNN

3



Demystify
mathematics
for neural networks.

4



Training
strategies
Evaluation



Sparse
data (text)
Embedding

5



Sequences data
RNN

6

20 Séquences
du 17 novembre
au 14 mai 2023



SAISON
22/23

11



Variational
Autoencoder
VAE

12



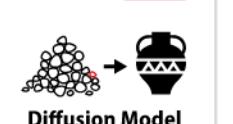
Project session
«My project in 180 s»

13



Generative
Adversarial Networks
GAN

14



Diffusion Model
Text to image

15



AI, Law, Society
and Ethics

16

New!
Model and training
optimization
Resource efficiency

17



Jean-Zay
GPU acceleration

18



Physics-Informed
Neural Networks
PINNs

19



Deep Reinforcement
Learning
RL

20



What will be
tomorrow's AI
Review & perspectives!

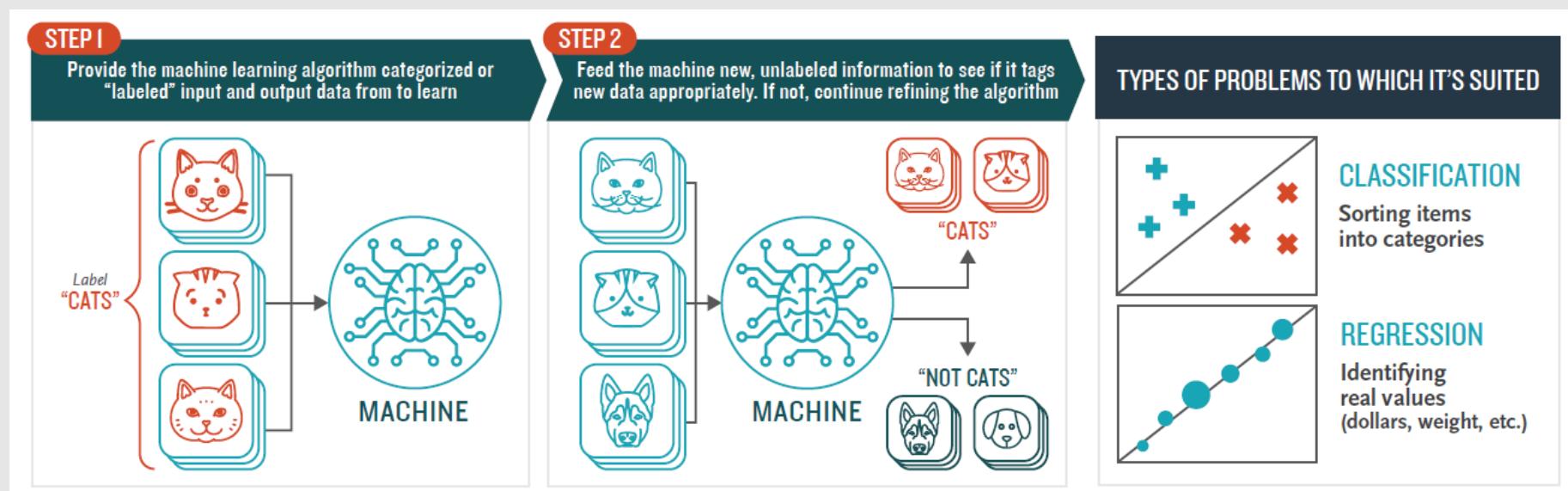
Supervised learning

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



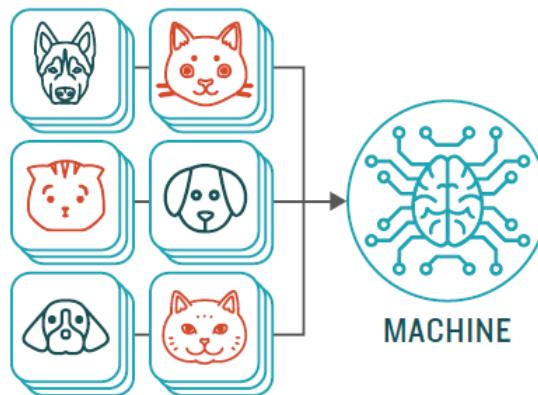
Unsupervised learning

Learning from data alone

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

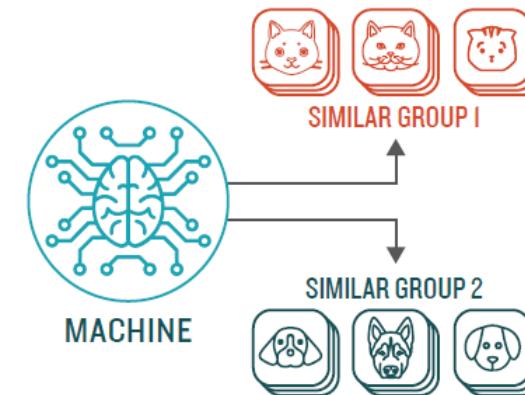
STEP 1

Provide the machine learning algorithm uncategorized, unlabeled input data to see what patterns it finds



STEP 2

Observe and learn from the patterns the machine identifies

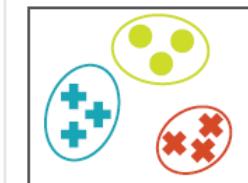


TYPES OF PROBLEMS TO WHICH IT'S SUITED

CLUSTERING

Identifying similarities in groups

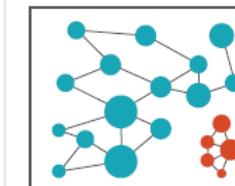
For Example: Are there patterns in the data to indicate certain patients will respond better to this treatment than others?



ANOMALY DETECTION

Identifying abnormalities in data

For Example: Is a hacker intruding in our network?



How to develop home-made ML tools?



mathematica



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

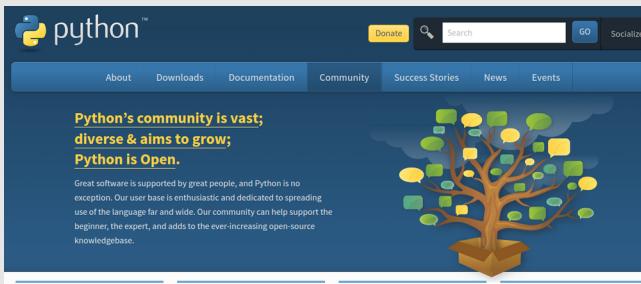
Core philosophy:

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

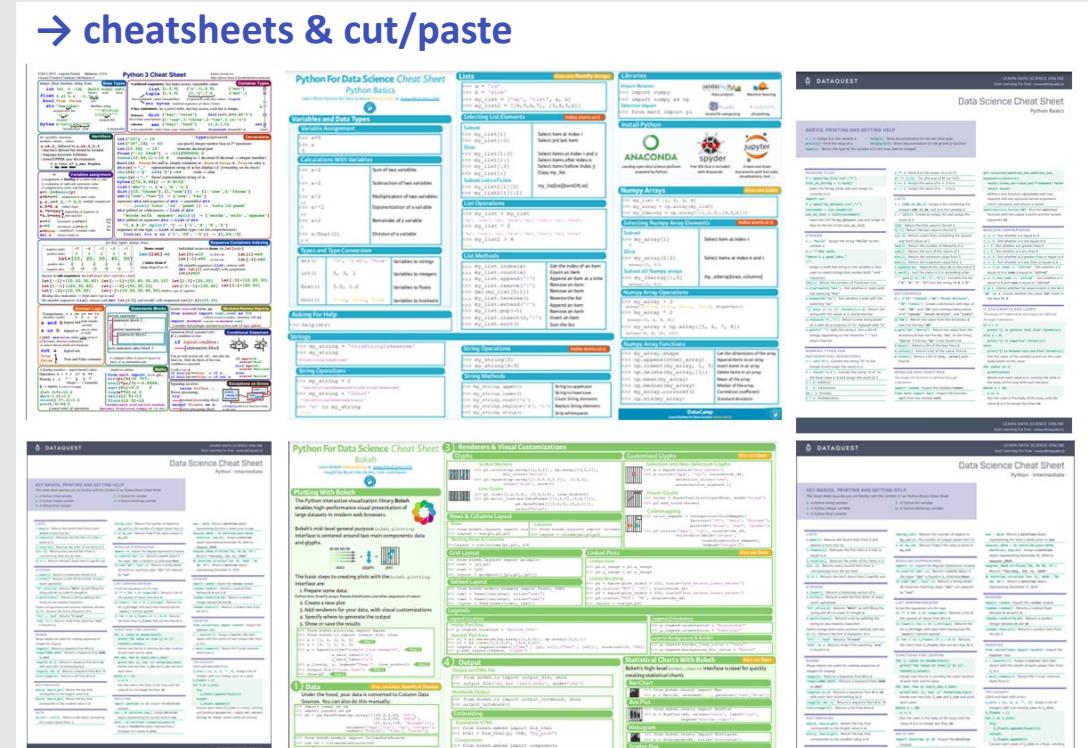
~ 250 additional [libraries](#) are available
- data science
- machine learning
- modern scientific computation
- visualization

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



The image displays four separate screenshots of Python cheat sheets from the website [DataScienceCheatSheet.com](https://datasciencecheatsheet.com). Each screenshot is a full-page reference guide for a specific Python topic:

- Python 3 Cheat Sheet**: A comprehensive guide for Python 3, covering topics like Data Types, Operators, Control Flow, Functions, Classes, and Modules.
- Python For Data Science Cheat Sheet - Python Basics**: A guide for data science basics, including NumPy arrays, DataFrames, and various data manipulation operations.
- Data Science Cheat Sheet - Beginner**: A general introduction to data science concepts and tools.
- Data Science Cheat Sheet - Intermediate**: An advanced guide for data science, covering topics like Machine Learning, Deep Learning, and Data Visualization.

Uneasy?

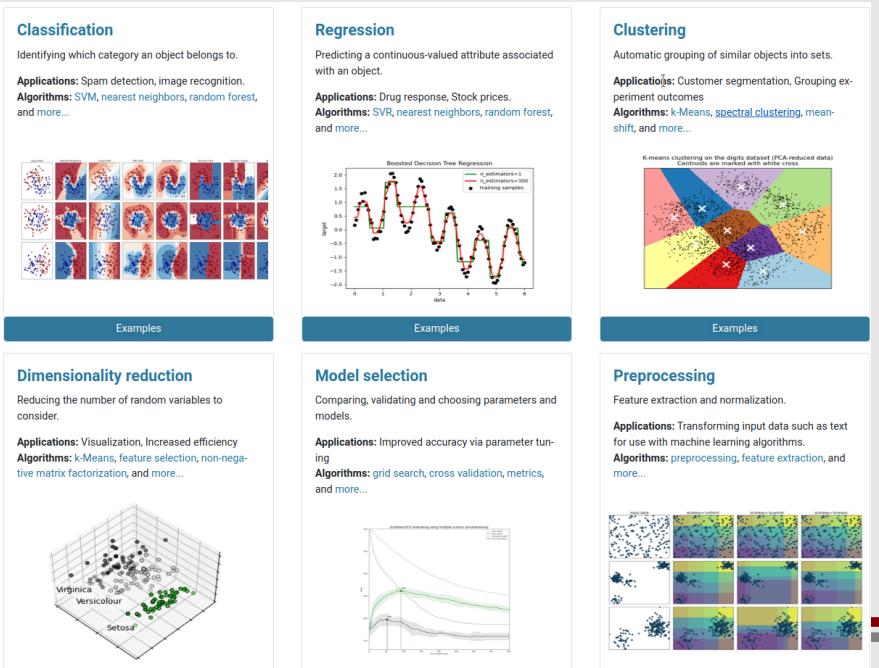
- yes and no
- important initial investment
- worth the effort



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)



Classification
Identifying which category an object belongs to.
Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

Regression
Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

Dimensionality reduction
Reducing the number of random variables to consider.
Applications: Visualization, Increased efficiency
Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

Model selection
Comparing, validating and choosing parameters and models.
Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...



Keras
<https://keras.io/>

High Level Deep Learning Application Programming Interface (API)

By François Chollet (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)

PyTorch

<https://pytorch.org/>

From Torch library

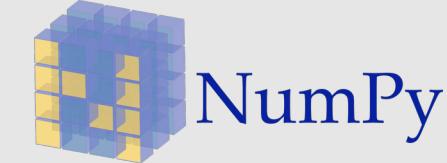
Supported by Facebook

BSD licence

(permissive free software license)

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

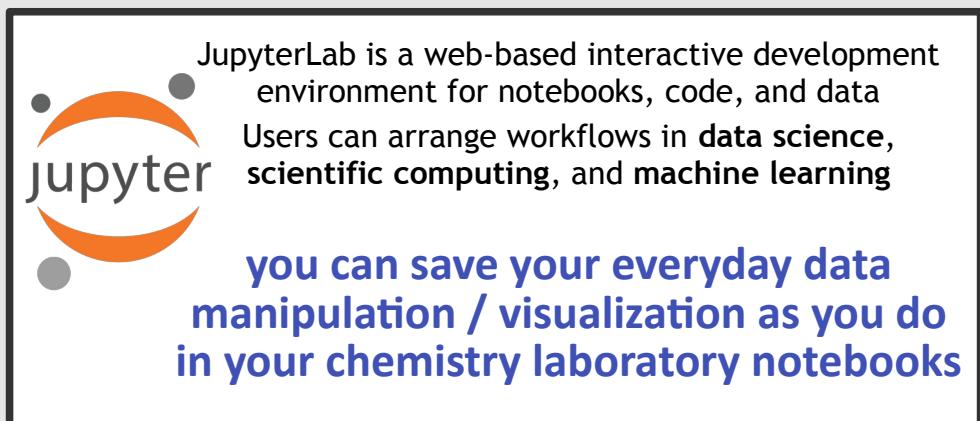
Widely used in the field of AI research





The screenshot shows the Anaconda website homepage. At the top, there's a navigation bar with links for ANACONDA, Products, Pricing, Solutions, Resources, Partners, Blog, Company, and Contact Sales. Below the navigation, a green banner reads "Individual Edition is now ANACONDA DISTRIBUTION". Underneath, it says "The world's most popular open-source Python distribution platform". A central callout box for "Anaconda Distribution" offers a "Download" button for Linux (Python 3.9), additional installers for Windows and macOS, and icons for GitHub, Docker, and Conda.

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