Machine Learning
Logistics

- Contact: francois.landes@u-psud.fr;
- **Tuesdays**, 8:30 – 12:45, Jussieu, Room 114 T22/23
  Typically, 2h Lecture, 15 min break, 2h tutorial
- Sometimes, pen-and-paper exercises (“TD”), most of the times, hands-on lab session (“TP”).
- Everything is at https://gitlab.inria.fr/flandes/pcs-ml

- Needed: **install** *python3, jupyter, scipy, numpy, matplotlib, scikit-learn* (+ *seaborn, pandas*, if possible)

Goals

What you should know *by the end of the term*

Know the basics of **ML vocabulary**

1. **Know** a couple of standard algorithms (be able to write their pseudo-code, explain their functioning)

2. Be able to code an algo (implement it) by **reading its doc** (documentation $\simeq$ book chapter)

Also: Make good **habits**, understand the standard **pipeline**

Goals

In the *long term*

- Learn **life-long fundamentals** that will not be outdated (obsolescent) in a couple of years
- Know the fundamentals enough so that you may **go beyond them** – to understand **newer paradigms**, you need to know about the previous one !

Goals
Detailed description

In 3 words: **inside** the **black boxes – let's do the maths !**

- This course is **algorithms-oriented, i.e. we will sketch the great principles of ML, but focus on how algorithms work in practice, including all necessary mathematical aspects**.
- Assuming a knowledge of fundamental maths notions (Bayesian inference, Algebra, Analysis, some optimization), we will cover the **inner workings of ML algorithms in detail.**
  Beyond their technical implementation, we will also **explain their theoretical foundations (mathematical definitions, limits, when and why they fail or work, etc).**
- The course will be **supported by pen-and-paper sessions and lab sessions** in groups of ~20, where we will re-code and play with algorithms, using Python.
- Note ! An **important part of the course material will be dispensed through the black/white/digital-board.** You are supposed to be **taking notes**, either **individually or in groups**.

Outline of contents

- See the readme of the class at: https://gitlab.inria.fr/flandes/pcs-ml

Grading: **Project**

- A **bibliographical+coding project** (in line with class' content)
  - from **selected papers**, understand a **new method** (actually "old" of course, but not seen in class)
  - **master it:** [relate with concepts seen in class]
    either **re-code** algo
    or **test** algo **on** some **tasks** and **be critical**
    (depends on the project choice)
  → **summarize your understanding to the rest of the class**
- **Expected written work:**
  - very short **report** (4 to 6 pages)
  - **slides/material for the oral presentation** (sent 2 days ahead)
- **Oral presentation**: <u>**(most of the grade)**</u>
  - duration: 10-15min, depends on practical constraints

Pre-requisites

- The class ***Computational Science***, by Martin Weigt, which implies:
  - Good knowledge in **python**, **numpy**
  - Experience with **scikit-learn**

  And also, some stuff you know:

- Basic **Algebra**
- A bit of **optimization** (mostly Gradient Descent)
- **Bayesian** calculus
  (at least MLE, maybe MAP?)

Bibliography
books

GO SEE:

http://lptms.u-psud.fr/francois-landes/machine-learning-resources/

[BEST] Classics:

- *Pattern Recognition and Machine Learning*, Christopher **Bishop**, 2006
  (more advanced, rather general)
- *Information Theory, Inference, and Learning Algorithms*, David J.C. **MacKay**
  (more theoretical, excellent if you enjoy probabilities)
- P Mehta, M Bukov, CH Wang, AGR Day, C Richardson, CK Fisher, D Schwab, ***A high-bias, low-variance introduction to Machine Learning for physicists***, Physics Reports (2019) 1-124.
- Scikit-learn’s doc: https://scikit-learn.org/stable/user_guide.html