

# T2A: Foundational Principles of ML

## Pre-requisites

- (T1A, Applied Stats): **Maximum Likelihood Estimate** and related notions
- (T1B, Maths for DS): **Basic Linear Algebra**
  - Note that T1A and T1B are **mandatory**, i.e. you **must** attend them to be allowed to follow T2A (FPML=this)
  - Check out **previous years exam** to see the level of math-mastery that is expected (*of course some of it is covered in this course, but you need solid basics*).
- We'll use **numpy** heavily.  
We expect you to know some **scientific programming**.
- (T2B, Optimization): **Gradient Descent** mostly (we'll discuss it) – **it is very advised to take it** as well if you plan to learn a lot of ML.  
More advanced notions are very useful to understand SVMs for instance.
- (T2D or T2E, Hands On ML with sklearn): it's a good complement to this class, very good to master sklearn.  
Here we'll look *inside* the algos of sklearn.

# T2A – FPML

## Goals

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

Know a bit of the **ML vocabulary**+standard pipeline

1. **Know** a couple of standard algorithms

*(from the Loss, be able to derive the pseudo-code, explain how they work)*

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

Also, to some extent :

3. Given a **problem** (task) or an **issue** (learning going wrong), explain simple phenomena, guess the solution

# T2A – FPML

## 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- Learn **life-long fundamentals** that will not be outdated (obsolescent) in a couple of years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** (with other classes) – to understand **newer paradigms**, you need to know about the previous one !

# T2A – FPML

## Foundational Principles of ML

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

- This course **is the theoretical counterpart of T2D-HoML/T2E (Datacamp)**. FPML 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 blackboard**. You are supposed to be **taking notes**, either **individually or in groups**.

Motivated **students are encouraged to self-organize to type a set of notes**, which we may proofread, to then share with the class.

# Grades / Evaluation

MCC (grades coefficients):

- **Session 1:** 0.3 CC + 0.7 EE (*Contrôle Continu, Examen Écrit*)
  - EE 70% **Limited time written exam.**  
documents will be allowed (6 pages max).
  - CC 30% **5-10 min quizzes at the end of each class** (grade average: 4 best grade out of 5)  
+ [probably not] a small homework coding exercise
- **Session 2:** 1.0 EE (2nd chance exam)
  - EE 100% New written exam  
(replaces previous grades)

Advice: Quizz is easy → more easy points in 1<sup>st</sup> session → try to get it the 1<sup>st</sup> time.

# How to get ready ?

- **Written exam (70%)** - *December 20<sup>th</sup>, 3h-long, pen-and-paper exam*
  - what you need to know: **points 1, 2 (a bit of 3) in slide #2**
  - prepare: work at constant pace on tutorials (+read corrections)
  - **!/: documents allowed: 6 pages of notes** (If typed, typed by yourself, not stupid copy-paste from anywhere)
- **Quizz (30%)**: *Fridays, 5 to 10 min quizzes online (MCQ & the like)*
  - *on ecampus – adress: TODO*
  - be in class on time (easy !)
  - review last week material (lectures, tutorials, tutorials corrections), making sure you understood everything
  - easy points to score !

# Outline of contents

**Approximate and Tentative program** of the semester (or term, really)

(1 subject  $\neq$  1 session, some are longer, some shorter)

If you get bored with the basic subjects, **please ask questions, interact**, and we can do more ! Also it's good to really master the basics in deep (no pun intended).

Not in chronological order (see the gitlab, it's organized by week-session)

- **Linear Regression** and related models: coding from scratch, basic notions + Gradient Descent
- **Perceptron**, Single Layer Neural Network : coding from scratch  
*Toy examples / MNIST*
- [Generic]: **train/validation/test** (extremely important !), Cross Validation
- **PCA**, from scratch (knowing algebra and np.linalg.eig )  
*Image compression*
- [Generic] **Feature maps, Kernels** (not from scratch, probably)
- [Generic] **Regularization**
- **SVM**, ~from scratch (knowing Lagrange multipliers)  
*Classification*
- **Naive Bayes**, from scratch (knowing Bayesian Inference)  
+ also **using a Prior** (i.e. real bayesian computation)  
*Image classification*
- [probably no time for this] **EM**, from scratch (knowing Bayesian Inference)  
*image clustering*
- [optionnal] **Decision trees**, ~from scratch, (knowing Entropy, Mutual Information)  
*Categorical data clustering*
- [Generic, Optionnal] **Metrics** (MSE, MAE, ROC AUC)

# 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)
- Your friends: **sci-hub** (papers) and **lib-gen** (books) or **book-zz** (books)  
(sometimes blocked from outside the university)

## Simple + exists in French:

- *Hands On Machine Learning with Scikit Learn and TensorFlow*, **Aurélien Géron** (not too hard, simultaneously rather practical yet complete)  
<https://github.com/yanshengjia/ml-road/blob/master/resources/>

## Version en Français:

- *Introduction au Machine Learning*, **Aurélien Géron**



# Course Material

(see gitlab)

- **Slides** like this
- Writings on the blackboard (take notes)
- There will be **no official lecture notes** !  
But, you can ***make your own*** (collective) notes.  
(I can take the time to proofread them if you give me clean notes)
- **Jupyter** notebooks (subjects)
- Jupyter notebooks (corrections)
- **Pen-and-paper** (subjects)
- Pen-and-paper (corrections)
- **Past exams** : 3 of them, esp. 2023 and 2022.
- Quizz (ecampus) – address: **TO COME**

# FPML – Foundational Principles of ML

## François Landes & Nicolas Béréux

- [francois.landes@u-psud.fr](mailto:francois.landes@u-psud.fr); [nicolas.bereux@inria.fr](mailto:nicolas.bereux@inria.fr)
- <https://gitlab.inria.fr/flandes/fpml>
- Fridays, 9:00 – 12:30 (or 12:45...)
- Typically, **1h30 Lecture**, 15 min break, **~2h Tutorial (TD/TP)**
- MCC: 0.3CC+0.7EE
- Needed: **install** *python3, jupyter, scipy, numpy, matplotlib, scikit-learn* (+ *seaborn, pandas*, if possible)

# Where is the class ?

- Lecture: **always in B108, always at 9:00**
- **1 Lecture (“CM”) 9h-10h30** Room: **B108**  
**1 Tutorial (“TP”) 10h45-12h30-45** Rooms: **????**
- See the calendar

<https://calendar.google.com/calendar/u/0/embed?color=%2309ecca&color=%23cd74e6&src=j10ll862qf53pdck5bj7u5ebek@group.calendar.google.com&src=k45ke3q7314b07uodmf1epq7f4@group.calendar.google.com>

# Python

IMPORTANT – make sure you have an **updated version of python3 and jupyter-notebook**, with at least **numpy, scipy, matplotlib** installed. Shortly we will also need **sklearn (scikit-learn)**, possibly **pandas**. **Seaborn** is always nice to have (I am not an expert of it).

- Alternative Solution 1: Use <https://jupyterhub.ijclab.in2p3.fr/> . Use your **institutional (Paris-Saclay, typically) account to connect for the first time**. This will open a work session of jupyter-notebook, that runs on the cloud, or more precisely, on the servers of the LAL (Linear Accelerator Laboratoire). You can click on the blue button on the top right corner, « upload », to import a notebook file onto the cloud, and then edit and run it online. Your files are saved over time there.
- Alternative Solution 2 (worse): same thing but using instead <https://colab.research.google.com/notebooks/intro.ipynb> (bad point: it's google, you need an account + data privacy is bad)