FPML – Foundational Principles of ML
François Landes & Manon Verbockhaven

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

- This course **is the theoretical counterpart of HoML** (Hands-on ML, applying ML to concrete projects, which is more hands-on than this course).
  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 black/white/digital-board.** You are supposed to be **taking notes**, either **individually or in groups**.
  To adjust for covid-related constraints, motivated **students are encouraged to self-organize to type a set of notes**, which we may proofread, to then share with the class.
  (although the class will be recorded, **I recommend taking notes**: a video does not replace good notes).

Pre-requisites

- (Maths for DS): **Basic Linear Algebra** (we'll do very quick reminders but won't spend much time on it)
- (Applied Stats): **M**aximum **L**ikelihood **E**stimate and related notions
- (Scientific prog.): it was advised to follow it: we'll use **numpy** heavily
- (Optimization): **Gradient Descent** mostly (we'll discuss it) – it's advised to take it as well. More advanced notions are very useful to understand SVMs
- (Hands On ML): it's a good complement to this class, very good to master sklearn. Here we'll look inside the algos of sklearn.

Goals

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

Know the basics of **ML vocabulary**

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

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)

3. Be able to **analyze critically** typical (classic) **experimental phenomena**, be able to make the good decisions

4. Given a problem (task), guess the relevant class of methods
(this is the slightly Hands-on part)

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 !

Grades / Evaluation

MCC (grades):

- **Session 1**: 0.3 CC + 0.7 EE (*Controle Continu*, *Examen Écrit*)
  - EE 70% **Limited time written exam.** (Limitless) documents will be allowed.
  - CC 30% **5 min quizzes at the beginning of each class**
- **Session 2:** 1.0 EE (2nd chance exam)
  - EE 100% New written exam (replaces previous grades)

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

How to get ready ?

- Written exam (70%) - *December 17th, 3h-long, pen-and-paper*
  - what you need to know: **points 1,2,3,4 in slide #4**
  - prepare: work at constant pace on tutorials (+read corrections)
  - documents allowed (*all* of them, books if you want ! – but *no internet*)
- Quizz : *Fridays, 2pm, 5 to 10 min quizzes online (MCQ & the like)*
  - https://ecampus.paris-saclay.fr/course/view.php?id=28064#section-5
  - be in class on time (easy !)
  - review last week material (lectures, tutorials, tutorials corrections), making sure you understood everything
  - easy points to score !

Python

IMPORTANT – make sure you have un **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)

Outline of contents

Approximate and Tentative program of the semester (or term, really)
(1 subject ≠ 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).

- **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, simulatneously rather practical yet complete)
  https://github.com/yanshengjia/ml-road/blob/master/resources/

  Version **en Francais**:

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

Course Material
(see gitlab)

- **Slides** like this
- Writings on the blackboard (take notes)
- **Annotated slides** like this
  (by me, or by generous students?)
- 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 (Some)
- **Past exam~~s~~** : only 1, from last year.
- Quizz solutions (ecampus) -
  https://ecampus.paris-saclay.fr/course/view.php?id=28064#section-5

FPML – Foundational Principles of ML
François Landes & Manon Verbockhaven

- francois.landes@u-psud.fr;
- https://gitlab.inria.fr/flandes/fpml
- Fridays, 13:30 – 17:15
- Typically, 1h30 Lecture, 15 min break, 2h 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 B107, always at 13:30**

- 5 november: **2 large** TP, **E201-E202**
  1 CM 13h30-15h
  1 TP 15h15-17h15
- The rest: to be announced