Fundamentals of Machine Learning

Fondements de l'Apprentissage Machine

Lab 0 - Basics to know BEFORE you start
Bases à connaître AVANT de commencer

Ioannis Mitliagkas, Guillaume Rabusseau

Teaching Assistants:

Reyhane Askari, Thomas George, Meraj Hashemizadeh, Philippe Laferrière,

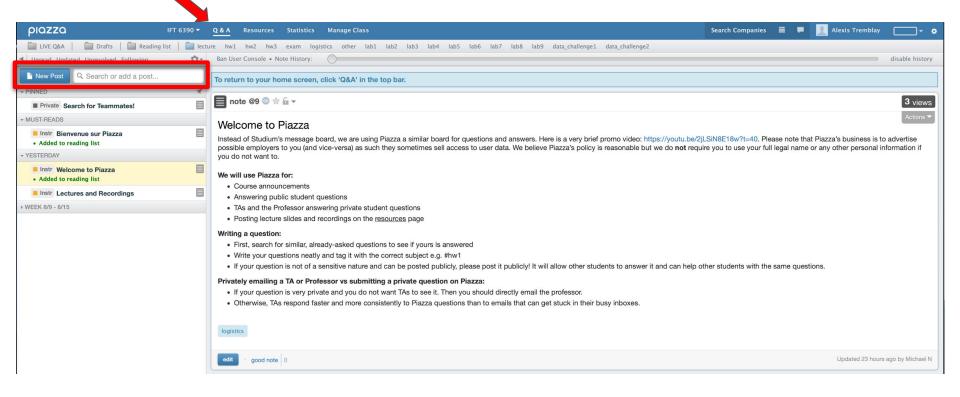Viet Nguyen, Michael Noukhovitch, Mandana Samiei, Alexis Tremblay, Vikram Voleti

Contents

1. Piazza

2. Gradescope

3. Overleaf (Latex)

4. Miniconda (Package manager)

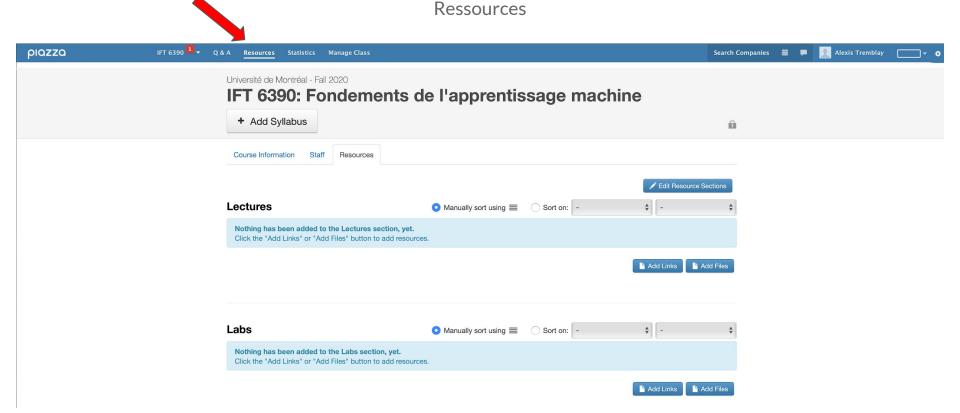5. `python hello_world.py`

6. Jupyter

7. Google Colab
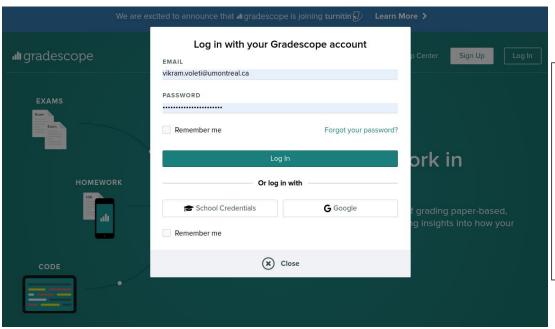
1. Piazza

Announcements - Forum - Annonces

1. Piazza

Resources
Ressources

All assignment submissions will be on gradescope.com

Toutes les soumissions seront sur gradescope.com

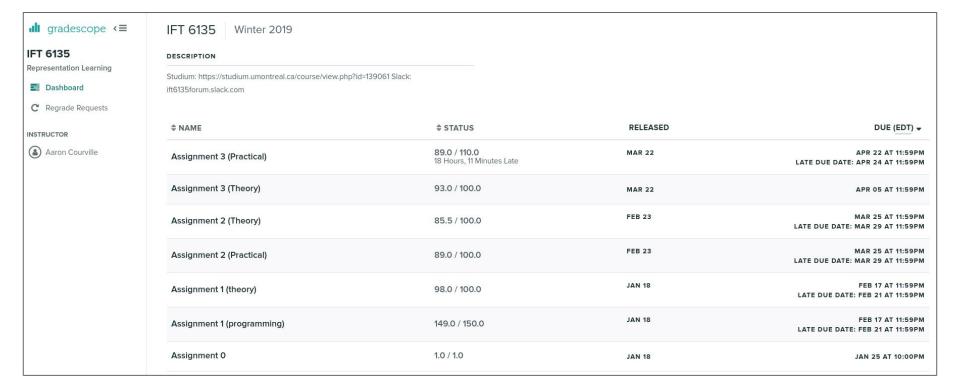- **Find all your assignments here! (with their due dates, status, etc.)**
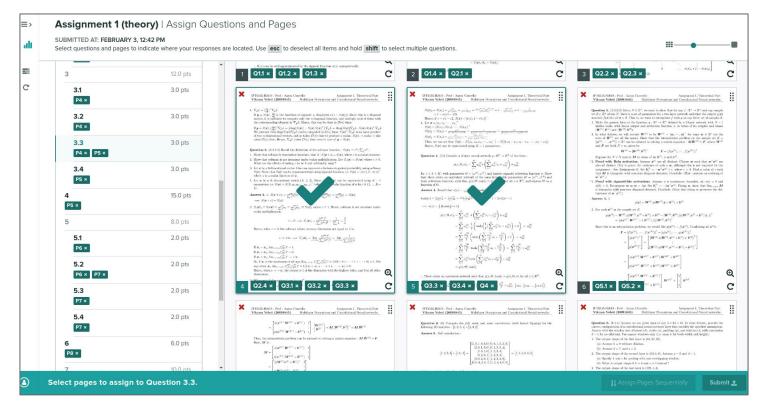 Trouvez tous vos devoirs ici! (avec leurs dates d'échéance, statut, etc...)

- **Submission**: for <u>each</u> question, select <u>all pages</u> with the answer (hold shift)

  Pour chaque question, sélectionnez toutes les pages correspondant à votre réponse (en appuyant sur shift)

Mila — Université de Montréal

- **TAs can point out errors or comments**

  Les TAs peuvent spécifier vos erreurs et vous faire des retours dessus.

- **"Regrade requests"** can be submitted <u>within a 2 week deadline</u>

  Sur Gradescope, vous pouvez demander qu'on recorrige certaines de vos réponses (justifier pourquoi) jusqu'à 2 semaines après publication des notes.

- If you're enrolled in this course, you should have received an invitation to sign up sent to your @umontreal.ca email (yes even you have one, McGill infiltrators)
  Si vous êtes inscrits à ce cours, vous avez dû recevoir une invitation pour vous enregistrer sur Gradescope sur votre boîte e-mail @umontreal.ca

- Using Gradescope is easy, but some of you will still get it wrong! Sad!
  L'utilisation de Gradescope est facile. Faites-le bien !

- To remedy this, we have an easy first "test" assignment to get you used to it.
  - <u>It is still for marks</u> but not as many, and we will not punish you for trouble with gradescope.

Un devoir 0 vous sera donné pour vous habituer à utiliser Gradescope. Vous ne serez pas pénalisés pour une mauvaise utilisation de la plate-forme.

- We <u>will</u> punish you for all assignments afterwards e.g.
  - not all answer pages selected for a question
  - not adding all your group members to a group submission

Pour les devoirs suivants, des points vous seront retirés si vous utilisez mal la plate-forme,

(si vous oubliez de sélectionner des pages pour une réponse à une certaine question,

ou si vous oubliez d'ajouter tous les membres de votre groupe à la soumission)

- There is a **written** assignment that you _should_ do in LaTeX, but all you _need_ to submit is a PDF.

  Il y a une partie écrite des devoirs, que vous devriez idéalement faire sur LaTeX. Vous **devez** soumettre un PDF. (N'oubliez pas de sélectionner les pages pour chaque question)
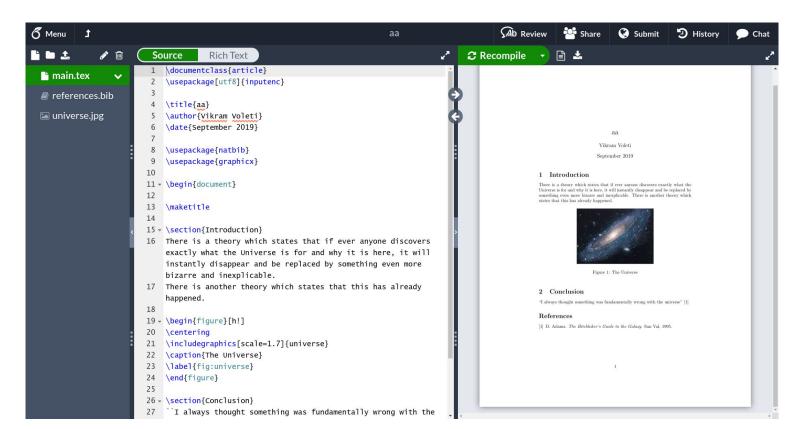
- There is a **programming** assignment that you _should_ be using version control to keep track of (e.g. Github), but all you _need_ to submit is a python file.

  Il y a une partie programmation des devoirs, pour laquelle vous devriez idéalement utiliser Github. Vous **devez** soumettre un fichier python.

3. Overleaf (Latex)

overleaf.com

- **LaTeX** (pronounced **"lay-tech"**) : Make a PDF using a program!

  LaTeX: Créez un PDF avec un programme!

- Extreme control, easy to use

  Contrôle extrême. Facile à utiliser.

- Latex CheatSheet: https://wch.github.io/latexsheet/

https://docs.conda.io/en/latest/miniconda.html

What is Miniconda?

Miniconda is the "mini" version of conda.

Ok, what is conda?

Conda is an open source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads, and switches between environments on your local computer.

Miniconda est la version "mini" de conda. Conda permet de gérer vos environnements et vos librairies. Facile à installer, à utiliser, et permet de faire des mises-à-jour rapides de vos libraires. Il est facile de créer des environnements et d'alterner entre les différents environnements sur votre machine.

https://docs.conda.io/en/latest/miniconda.html

Why do you need it?

To run stuff in Python, many codes require Python packages (such as `numpy`, `matplotlib`) that come pre-installed with good functionalities! (For example, `matplotlib` can be used to visualize several kinds of data, like 2D points, 3D meshes, images, etc.)

Miniconda lets you install Python packages easily, and manage "environments".

Pour faire tourner votre code Python, vous aurez besoin de plusieurs librairies (qui viennent avec un tas de fonctionnalités très utiles). Miniconda vous permet d'installer ces librairies facilement, et de gérer vos environnements.

4. Miniconda

Mila Université de Montréal

https://docs.conda.io/en/latest/miniconda.html

How to use it?

- Install Miniconda using the link above

  Installez Miniconda en utilisant le lien ci-dessus

- Make a new virtual environment for your project: `conda create -n proj1 python=3.7`

  Créez un nouvel environnement pour votre projet

- Activate this virtual environment: `conda activate proj1`

  Activez cet environnement virtuel

- Install required Python packages, such as `numpy` and `matplotlib`: `conda install numpy matplotlib`
- Use it! (Launch `python`)

  Installez les librairies Python nécessaires, puis utilisez votre environnement.

4. Miniconda

How to use it?

- Install Miniconda using the link above

- Make a new virtual environment for your project: `conda create -n proj1 python=3.7`

- Activate this virtual environment: `conda activate proj1`

- Install required Python packages, such as `numpy` and `matplotlib`: `conda install numpy matplotlib`
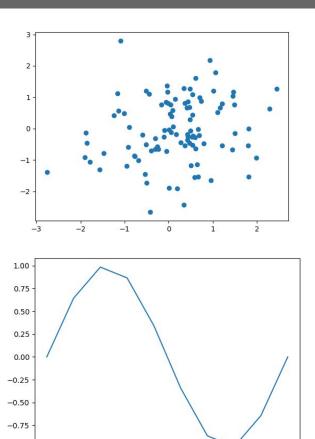
- Use it! (Launch `python`)

Good practice: Make a new virtual environment for each project - no conflicting packages among projects!

Pour éviter d'avoir des conflits entre vos librairies, créez un nouvel environnement virtuel pour chaque projet :)

```python
import matplotlib.pyplot as plt

import numpy as np


print("Hello World")


pts = np.random.randn(100, 2)
```
```
sudo mount -t ntfs-3g -o remove_hiberfile /dev/sda2
/media/2A36EE3236EDFEA7
```

```python
plt.show()


x = np.linspace(0, 2*np.pi, 10)

y = np.sin(x)

plt.plot(x, y)
```

https://jupyter.org/install.html

`pip install jupyter`

- Interactive Python "notebook"

- Very user-friendly for logging work flow of code

- Can plot graphs (and other visual data) within notebook itself!

https://colab.research.google.com/

- Python notebook + GPU

Thank you.