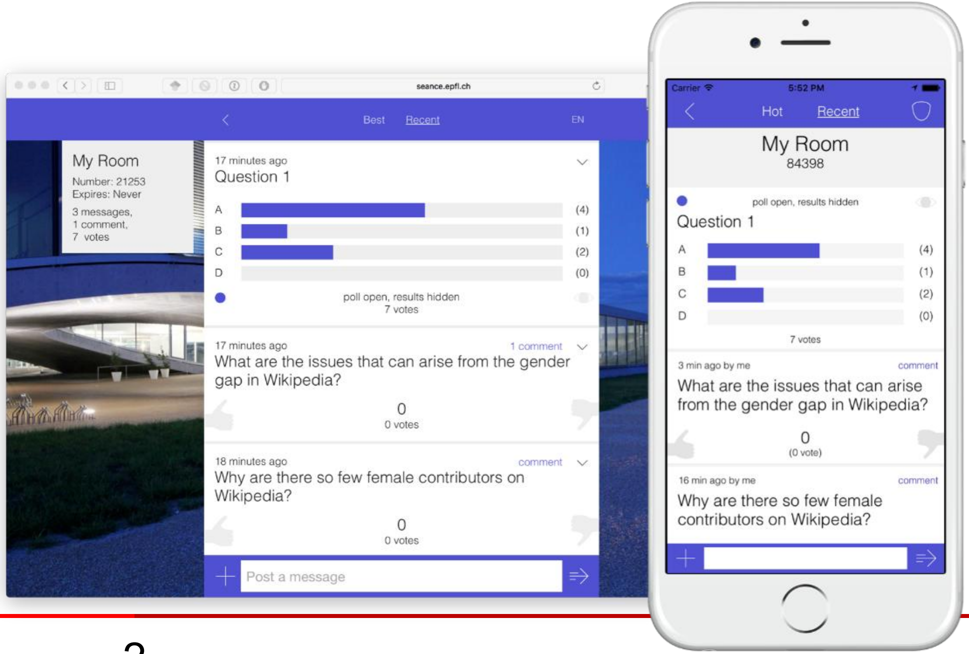
Getting started

Machine Learning for Behavioral Data (CS-421)

February 17, 2026

SpeakUp

https://go.epfl.ch/mlbd-speakup



My Room

Number: 21253
Expires: Never

3 messages,
1 comment,
7 votes

17 minutes ago
Question 1

A	(4)
B	(1)
C	(2)
D	(0)

poll open, results hidden
7 votes

17 minutes ago 1 comment
What are the issues that can arise from the gender gap in Wikipedia?

0
0 votes

18 minutes ago comment
Why are there so few female contributors on Wikipedia?

0
0 votes

Post a message

My Room
84398

poll open, results hidden
Question 1

A	(4)
B	(1)
C	(2)
D	(0)

7 votes

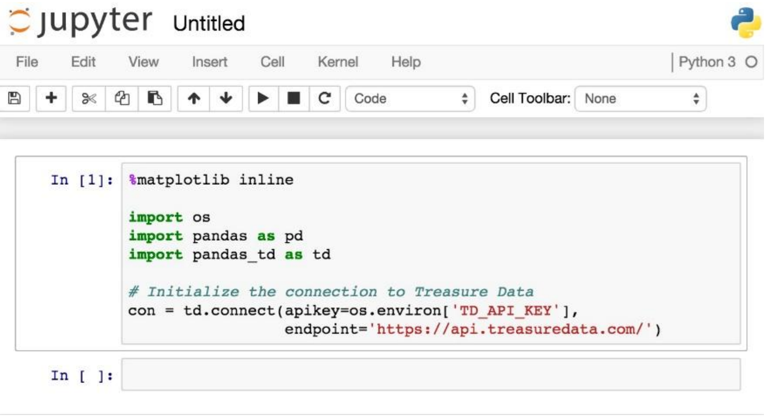
3 min ago by me comment
What are the issues that can arise from the gender gap in Wikipedia?

0
(0 vote)

16 min ago by me comment
Why are there so few female contributors on Wikipedia?

Jupyter

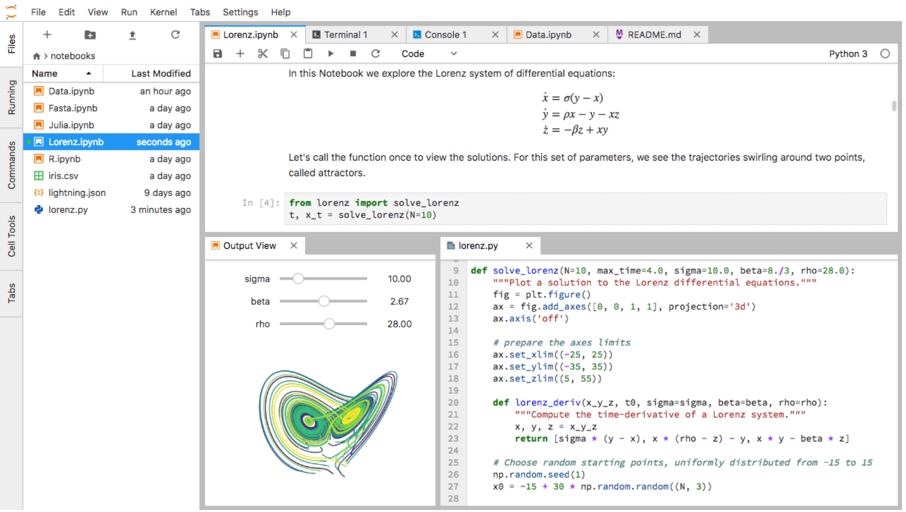
Jupyter notebook



```
In [1]: %matplotlib inline

        import os
        import pandas as pd
        import pandas_td as td

        # Initialize the connection to Treasure Data
        con = td.connect(apikey=os.environ['TD_API_KEY'],
                         endpoint='https://api.treasuredata.com/')

In [ ]:
```

Tutorial: https://www.dataquest.io/blog/jupyter-notebook-tutorial/

JupyterLab



notebooks

Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=10)
```

sigma 10.00

beta 2.67

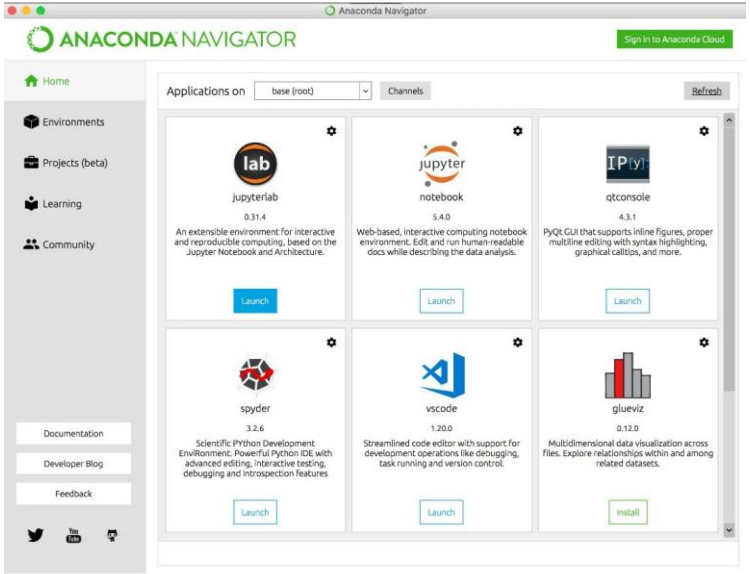
rho 28.00

```
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim((5, 55))

    def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
        """Compute the time-derivative of a Lorenz system."""
        x, y, z = x_y_z
        return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

    # Choose random starting points, uniformly distributed from -15 to 15
    np.random.seed(1)
    x0 = -15 + 30 * np.random.random((N, 3))

```

Why JupyterLab: https://towardsdatascience.com/jupyterlab-a-next-gen-python-data-science-ide-562d216b023d

Anaconda (local env)

- You have the full control
- Works offline
- https://www.anaconda.com/products/individual



Applications on base (root)

jupyterlab 0.31.4
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

notebook 5.4.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

qtconsole 4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

spyder 3.2.6
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

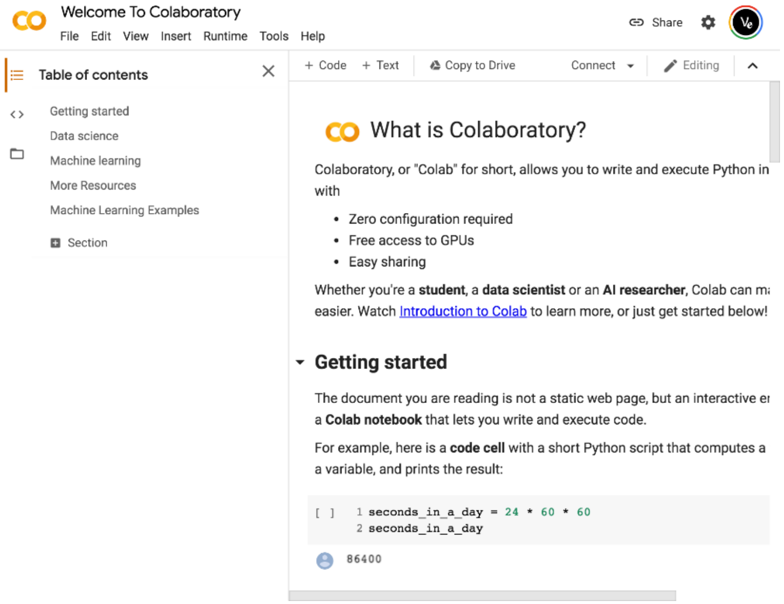
vscode 1.20.0
Streamlined code editor with support for development operations like debugging, task running and version control.

glueviz 0.12.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.

- **Tutorial**: https://www.edureka.co/blog/python-anaconda-tutorial/

Google Colab (online env)

- Ready environment
- Uses Google's infrastructure
- Collaborative functionality
- Requires Google account
- https://colab.research.google.com/



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can m easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive e a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a a variable, and prints the result:

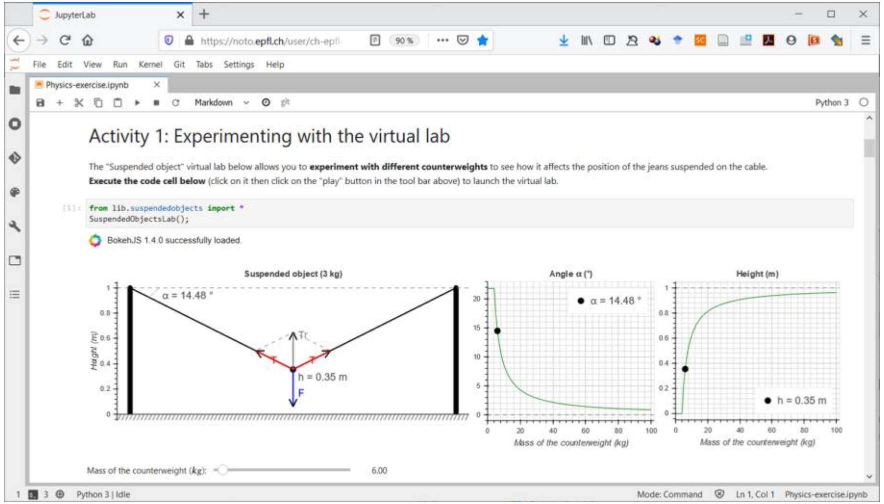
```
[ ]  seconds_in_a_day = 24 * 60 * 60
     seconds_in_a_day
```

```
86400
```

- **Video**: https://www.youtube.com/watch?v=inN8seMm7UI

EPFL Noto (online env)

- Ready environment
- Login with your Gaspar
- https://noto.epfl.ch/



JupyterLab
https://noto.epfl.ch/user/ch-epfl
90 %
File Edit View Run Kernel Git Tabs Settings Help
Physics-exercise.ipynb
Markdown
git
Python 3

Activity 1: Experimenting with the virtual lab

The "Suspended object" virtual lab below allows you to **experiment with different counterweights** to see how it affects the position of the jeans suspended on the cable.
Execute the code cell below (click on it then click on the "play" button in the tool bar above) to launch the virtual lab.

```
[1]: from lib.suspendedobjects import *
     SuspendedObjectsLab();
```

BokehJS 1.4.0 successfully loaded.

Mass of the counterweight (kg): 6.00

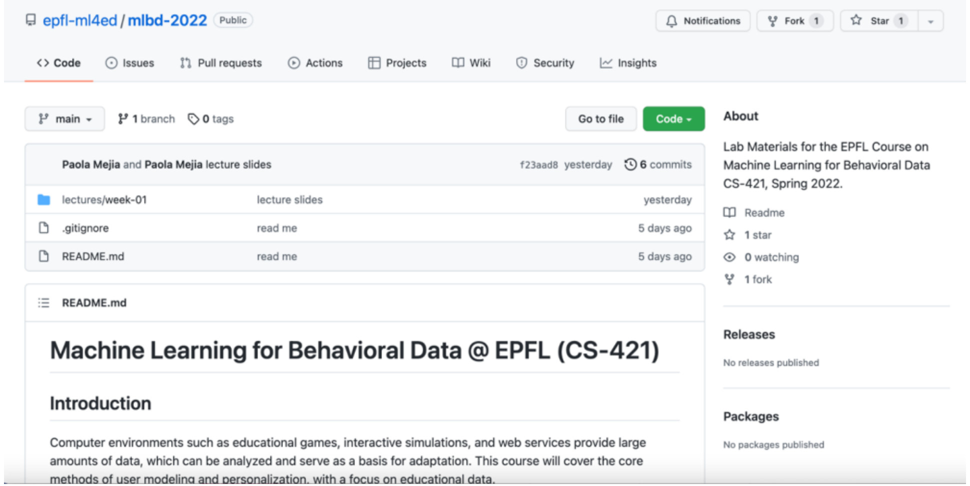
Noto

- Using Noto:
  - Go to https://noto.epfl.ch/
  - Login with your GASPAR
  - Go to Git → Clone
  - Clone the course repository: https://github.com/epfl-ml4ed/mlbd-2026

GitHub

- Share files and code
- Version control (git)

- **Tutorial**: https://www.edureka.co/blog/how-to-use-github/



Paola Mejia and Paola Mejia lecture slides f23aad8 yesterday 6 commits

lectures/week-01	lecture slides	yesterday
.gitignore	read me	5 days ago
README.md	read me	5 days ago

About

Lab Materials for the EPFL Course on Machine Learning for Behavioral Data CS-421, Spring 2022.

Machine Learning for Behavioral Data @ EPFL (CS-421)

Introduction

Computer environments such as educational games, interactive simulations, and web services provide large amounts of data, which can be analyzed and serve as a basis for adaptation. This course will cover the core methods of user modeling and personalization, with a focus on educational data.

(Demo)

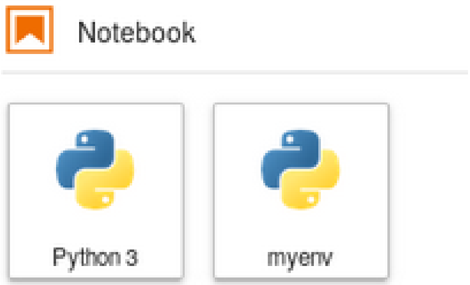
Setting up the environment

- Set up an environment on which you can
  - Run Jupyter notebooks in Python
  - Connect to course repository: https://github.com/epfl-ml4ed/mlbd-2026

- We will use https://noto.epfl.ch/
  - But you are free to use whatever you want (e.g. Anaconda, Colab etc.)
  - It's your responsibility to have a working environment
- **Task**: Pull course's GitHub repository

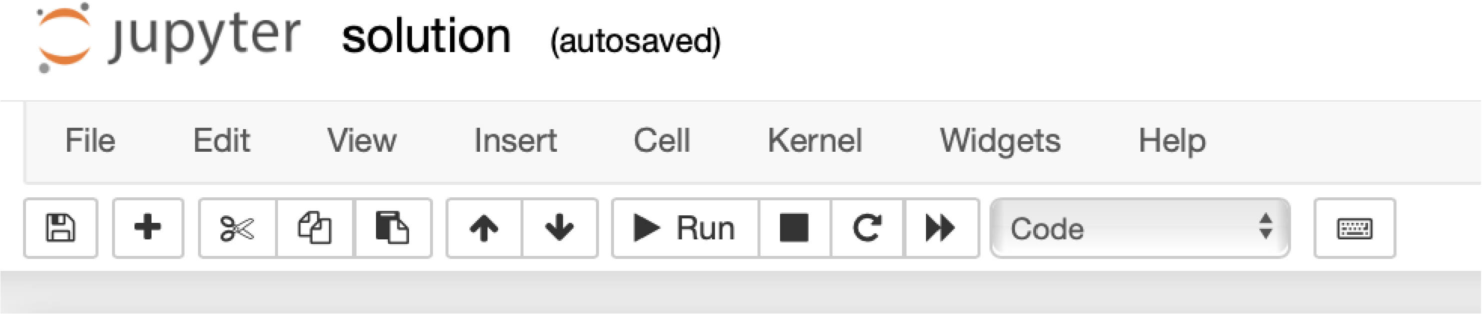
Anaconda

- Virtual environment:
  - https://janakiev.com/blog/jupyter-virtual-envs/
  - Create virtual environment: `python -m venv myenv`
  - Activate virtual environment: `source myenv/bin/activate`
  - add to Jupyter: `python -m ipykernel install --user --name=myenv`



Basic functions

Colab intro.



Git Intro

1. Basic git tutorial (add, commit, status).
2. Github introduction.
3. Branches (team work).

Git | Hello World

- New directory for Git repository
  - `mkdir gitdemo`
  - `cd gitdemo`
- Now we're inside our new folder. Time to make it a proper Git repo:
  - `git init`
- Now we're inside our new folder. Time to make it a proper Git repo:
  - `git init`
- You'll see Initialized empty Git repository in `/path/to/your/repo/.git/`. What's that `.git`? If you list all files in your directory (`ls -a`), you'll see a new hidden `.git/` directory. That's where Git stores the information about this new repository.

- Time to add some files.
  - `touch new.txt`
  - `echo "Hello, World!" > new.txt`
- You'll have a new file, `new.txt`
- But this isn't just any old folder; it's Git repository! Git has tracked that we have a new file. Enter the following command:
  - `git status`

`Why can’t you see the file?`

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Git | Hello World

- ➔ `git add new.txt`
- ➔ `git status`
- ➔ Git knows about our file now. Time to commit our changes to Git's history.
  - ◆ `git commit -m "Add new.txt"`

The `-m` flag provides a commit message. Such a message is required for all commits.

- ➔ let's make some changes.
  - ◆ `echo "Foobar!" >> new.txt`
- ➔ This adds a new line (again, no text editor needed) to our `new.txt`.

`How can you see the changes?`

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Git | Hello World

- ➔ `git add new.txt`
- ➔ `git status`
- ➔ Git knows about our file now. Time to commit our changes to Git's history.
  - ◆ `git commit -m "Add new.txt"`

The `-m` flag provides a commit message. Such a message is required for all commits.

- ➔ let's make some changes.
  - ◆ `echo "Foobar!" >> new.txt`
- ➔ This adds a new line (again, no text editor needed) to our `new.txt`.

`How can you see the changes?`

- ➔ `git status`
- ➔ `git dif new.txt`

`How can` **`add`** `the changes?`

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Git | Hello World

- ➔ `git add new.txt`
- ➔ `git status`
- ➔ Git knows about our file now. Time to commit our changes to Git's history.
  - ◆ `git commit -m "Add new.txt"`

The `-m` flag provides a commit message. Such a message is required for all commits.

- ➔ let's make some changes.
  - ◆ `echo "Foobar!" >> new.txt`
- ➔ This adds a new line (again, no text editor needed) to our `new.txt`.

`How can you see the changes?`

- ➔ `git status`
- ➔ `git dif new.txt`

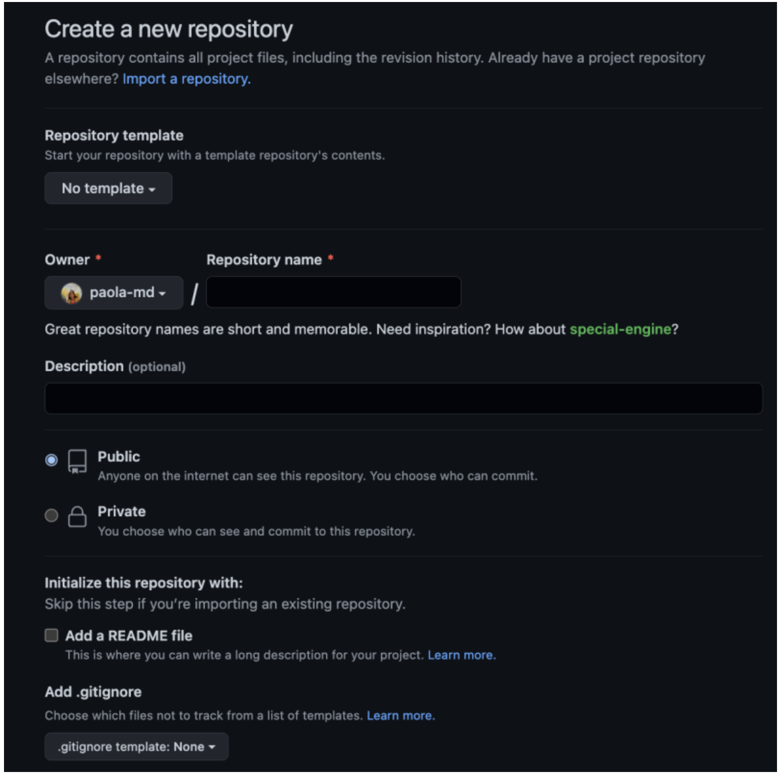
`How can` **`add`** `the changes?`

- ➔ `git add new.txt`
- ➔ `git commit -m "adds changes"`

`How can you` **`push`** `to github?`

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Github | Hello World



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * **Repository name** *

paola-md ▾ /

Great repository names are short and memorable. Need inspiration? How about special-engine?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ **Add a README file**
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None ▾

- `git branch -M main`
- `git remote add origin https://github.com/paola-md/test.git`
- `git push -u origin main`

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Github | Challenge

Try solving the tasks on your own an raise your hand if you need help.

Instructions:

1. Create a team of three and decide who is person A, B and C.
2. Person A: **Fork** the course's repo (https://github.com/epfl-ml4ed/mlbd-2026) and add B and C as collaborators.
3. B and C: **Clone** the forked repo.
4. A, B and C: **Create a branch** <person>-challenge-<number>. For example: a-challenge-1.
5. A, B and C: In your branch solve the corresponding task in https://github.com/epfl-ml4ed/mlbd-2026/blob/main/lectures/week-01/git-challenge.py

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Github | Challenge

6. A, B and C: Create a **pull request** with your changes.
7. B: **Merge pull requests**.
8. C: **Pull** changes and run challenge.py locally.

Full tutorial: https://opensource.com/education/16/1/git-education-classroom

Milestone M1

Available on **Moodle**

Fill out with team and start-up preference + the confidentiality form

Deadline: Tuesday, Feb 25th, 23:59

Feedback

We are actively looking for feedback to improve

https://go.epfl.ch/mlbd-feedback

Questions?