Statistical and Mathematical Methods for Machine Learning

Course Presentation

A.Y. 2024 - 2025

Davide Evangelista

davide.evangelista5@unibo.it

Useful Informations

- **Exam:** Check professor Elena Loli Piccolomini course presentation

- **Office Hour:** Wednesday hour 9am-12am. Reserve an appointment via email at:
  davide.evangelista5@unibo.it

- **Books:**
  - Jupyter Notebooks on Virtuale
  - My personal website (check following slides)
  - Optimization: *Numerical Optimization* di Nocedal e Wright

Lectures slides

- The material for the course is available at my website:

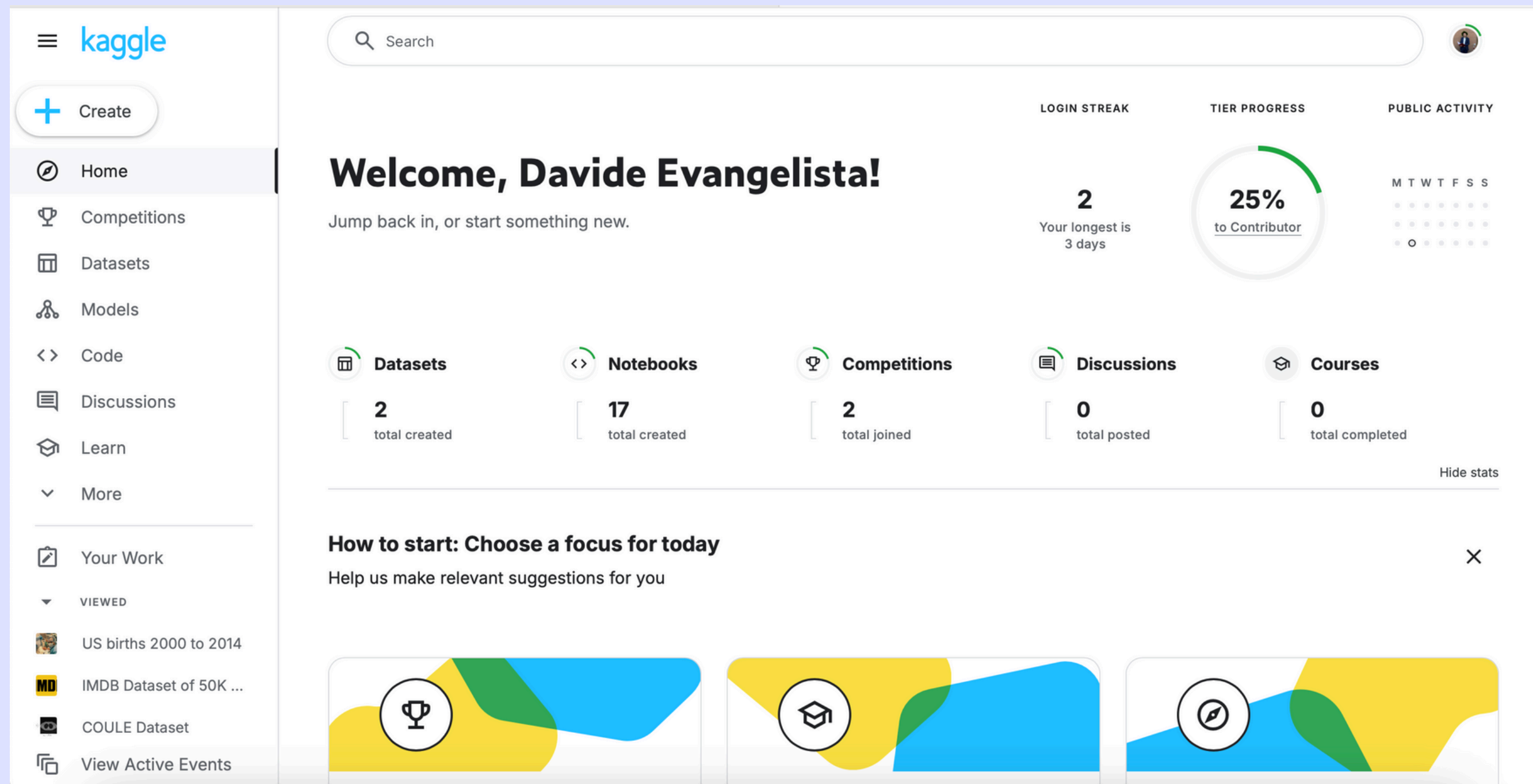https://devangelista2.github.io/statistical-mathematical-methods

- Also available on Virtuale

- **Note:** Lectures are new and may contain errors. If you spot any error, please tell me by e-mail so that I can improve them.
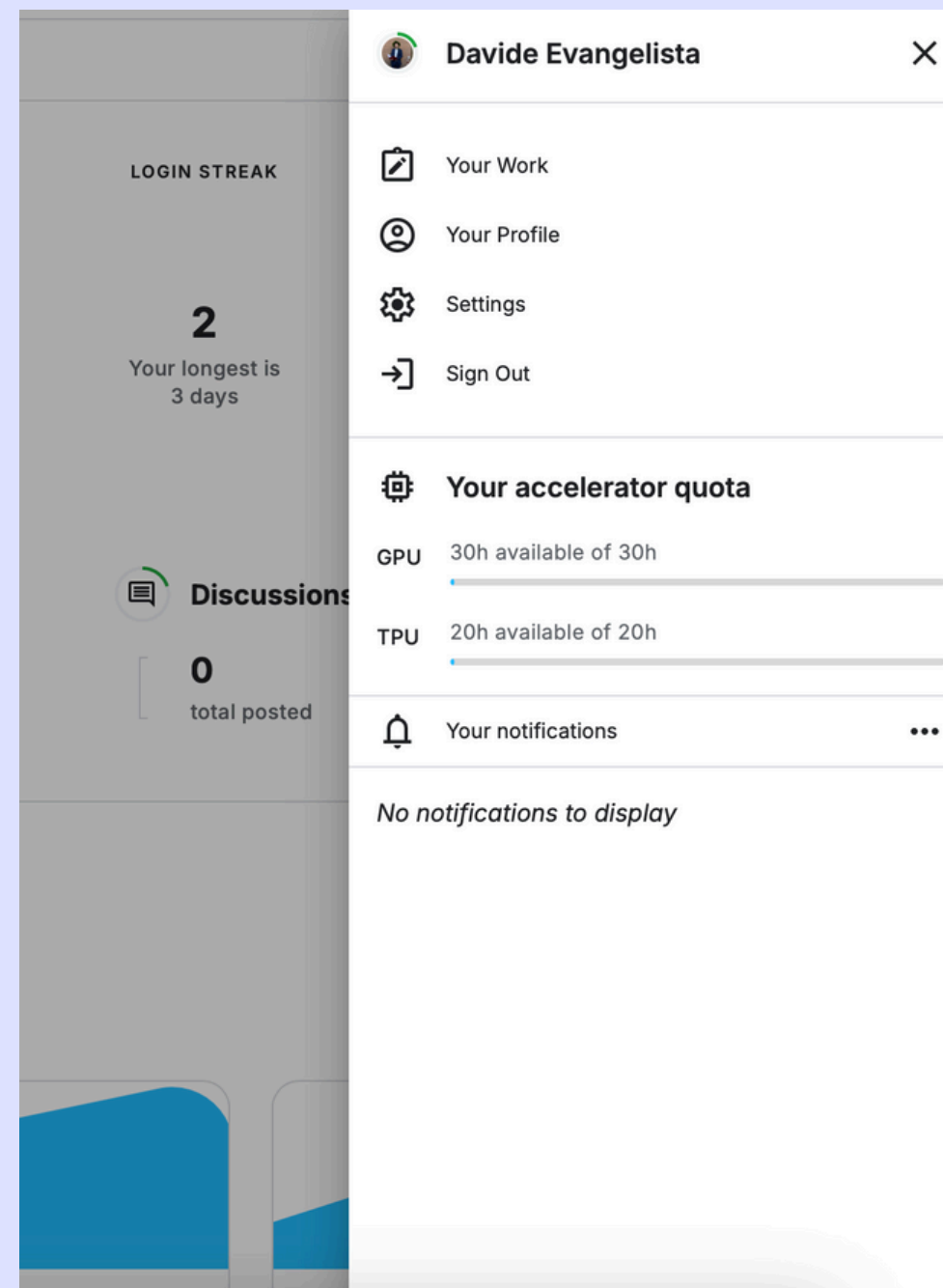

SCAN ME

Useful website (for datasets)

- **Kaggle:** https://www.kaggle.com

Useful website (for datasets)

- **Kaggle:** https://www.kaggle.com

Login

Useful website (for datasets)

- **Kaggle:** https://www.kaggle.com

Dataset info

Python installation on your personal device (only if you need to install it on your device)

Python Installation

Python Installation

Anaconda Installers

Windows

Python 3.12

⬇ 64-Bit Graphical Installer (912.3M)

Mac

Python 3.12

⬇ 64-Bit (Apple silicon) Graphical Installer (704.7M)

⬇ 64-Bit (Apple silicon) Command Line Installer (707.3M)

⬇ 64-Bit (Intel chip) Graphical Installer (734.7M)

⬇ 64-Bit (Intel chip) Command Line Installer (731.2M)

Linux

Python 3.12

⬇ 64-Bit (x86) Installer (1007.9M)

⬇ 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)

⬇ 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

Python Installation

Anaconda Installers

Windows

Python 3.12

⤓ 64-Bit Graphical Installer (912.3M)

Mac

Python 3.12

⤓ 64-Bit (Apple silicon) Graphical Installer (704.7M)

⤓ 64-Bit (Apple silicon) Command Line Installer (707.3M)

Linux

Python 3.12

⤓ 64-Bit (x86) Installer (1007.9M)

⤓ 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)

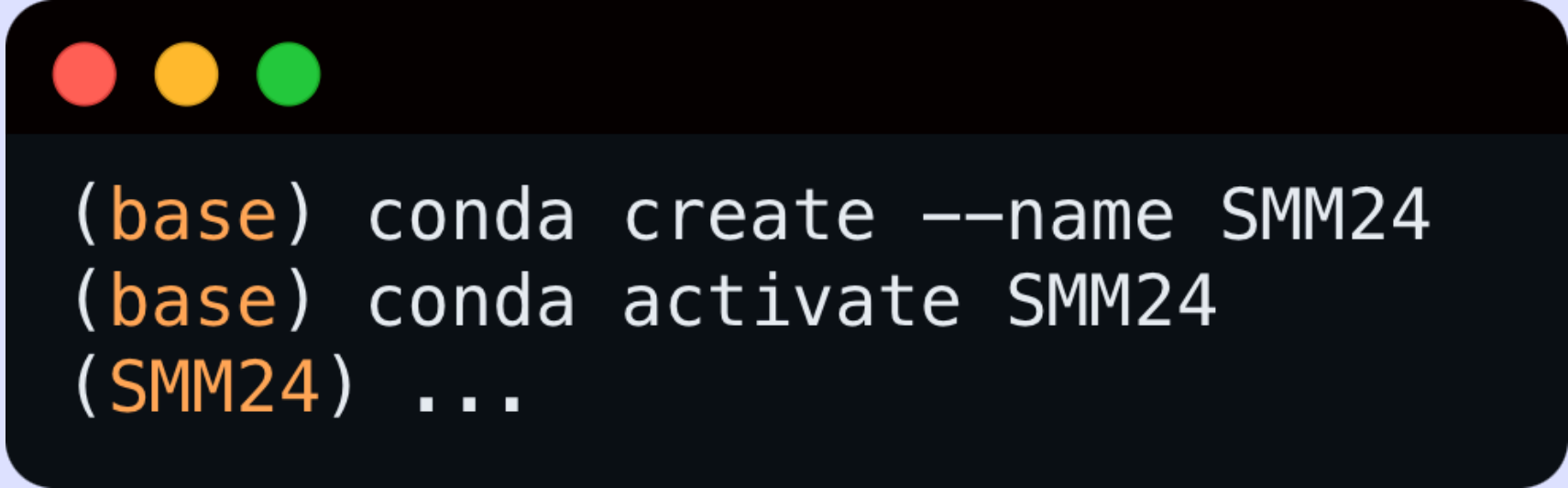⤓ 64-bit (Linux on IBM Z & LinuxONE)

Creating Virtual Environment

*A **virtual environment** in Python is an isolated workspace that allows you to install and manage packages for specific projects without affecting the global Python installation or other projects. This ensures that different projects can have different dependencies, package versions, and configurations.*

You shoud **always** work on virtual environment, so that you can easily fix any compatibility issues without interfere with others existing Python installations.
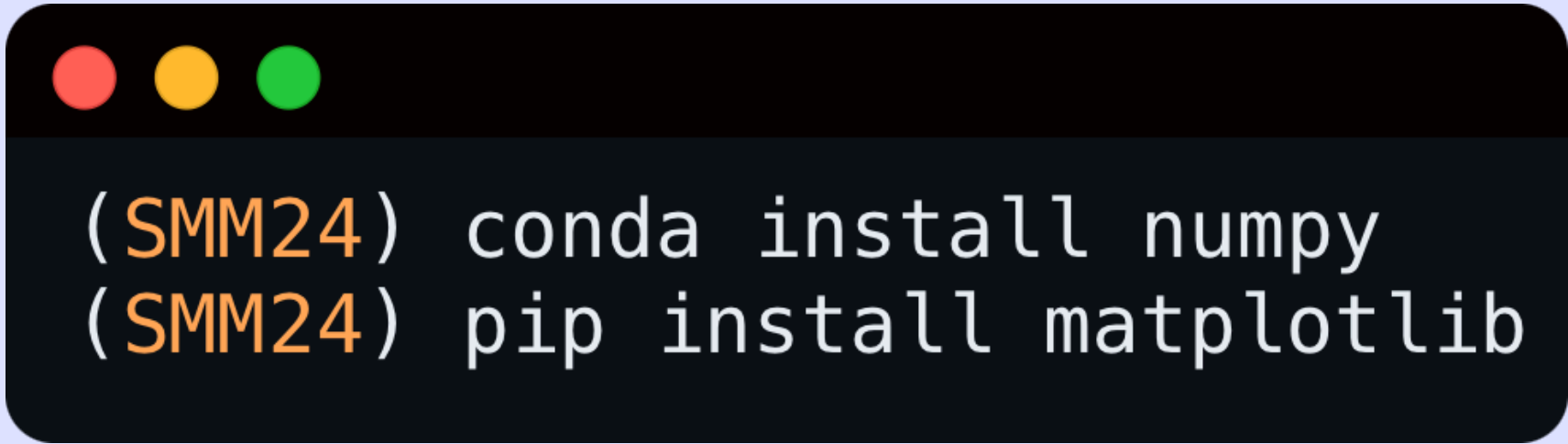
Creating Virtual Environment

- On terminal (or CMD, for Windows users):

```
(base) conda create --name SMM24
(base) conda activate SMM24
(SMM24) ...
```

← crete venv
← activate venv

- To install packages:

```
(SMM24) conda install numpy
(SMM24) pip install matplotlib
```

Creazione Virtual Environment

- Packages **required** for the course:
  - `numpy`
  - `scipy`
  - `pandas`
  - `matplotlib`
  - `scikit-image`
  - `scikit-learn`

- Install with **pip** or **conda**

Code Editor

- Types of Python files: `.py` > script

  `.ipynb` > notebook (Jupyter notebook)

- Editor: `VSCode (Visual Studio Code)`

  `PyCharm`

  `Spyder`

Visual Studio Code
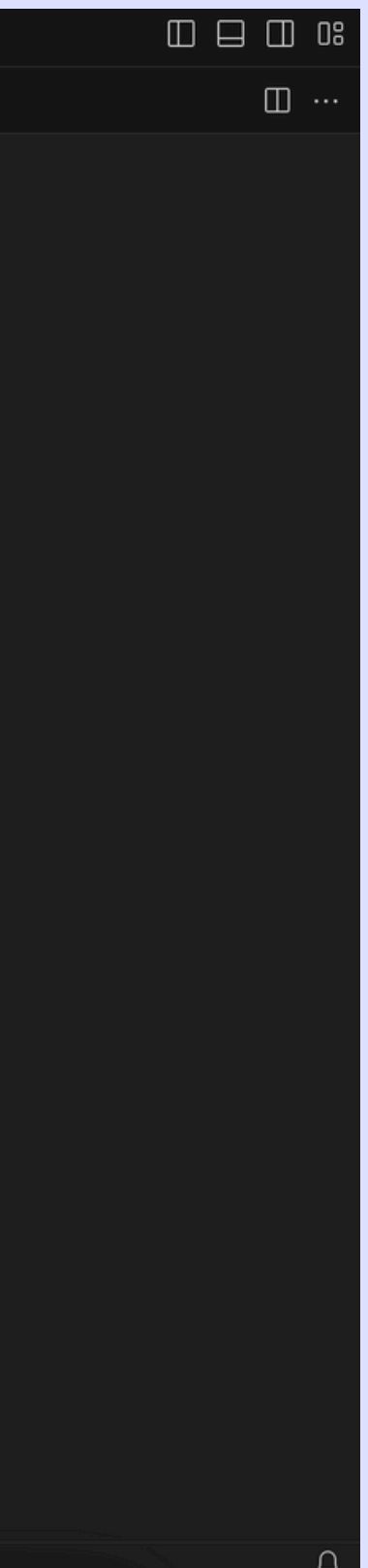
Download page

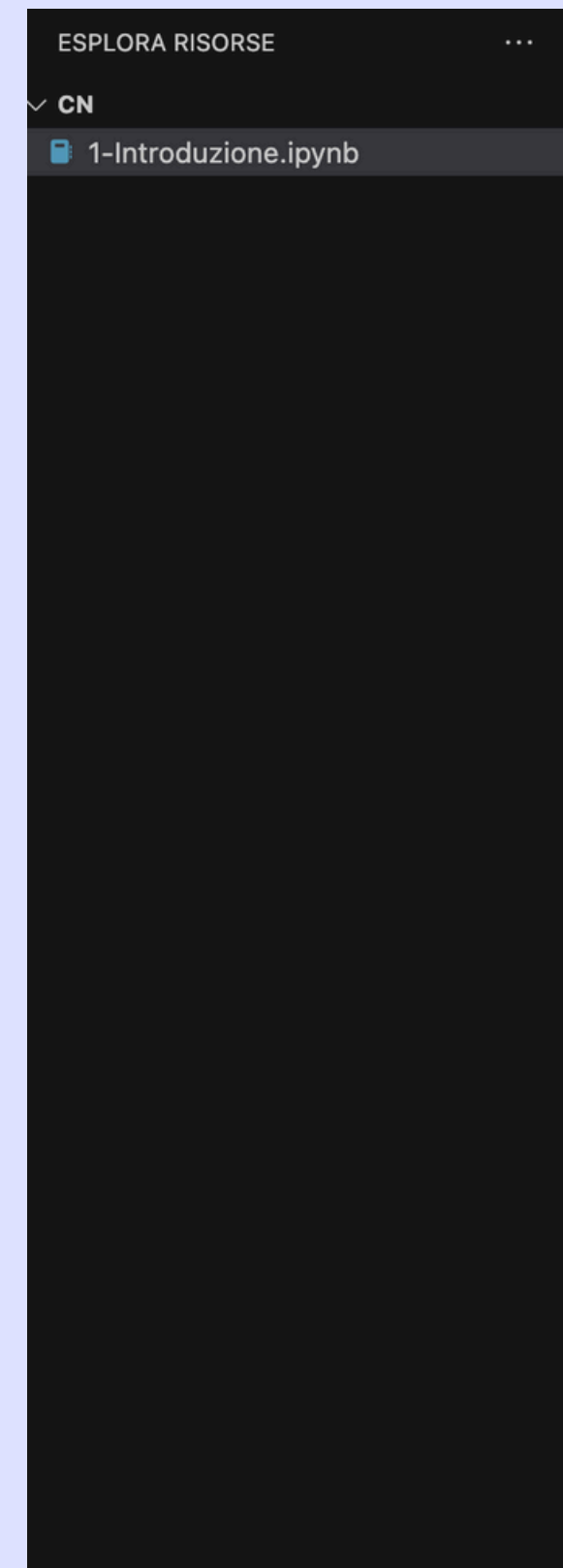# Plugins

Plugins

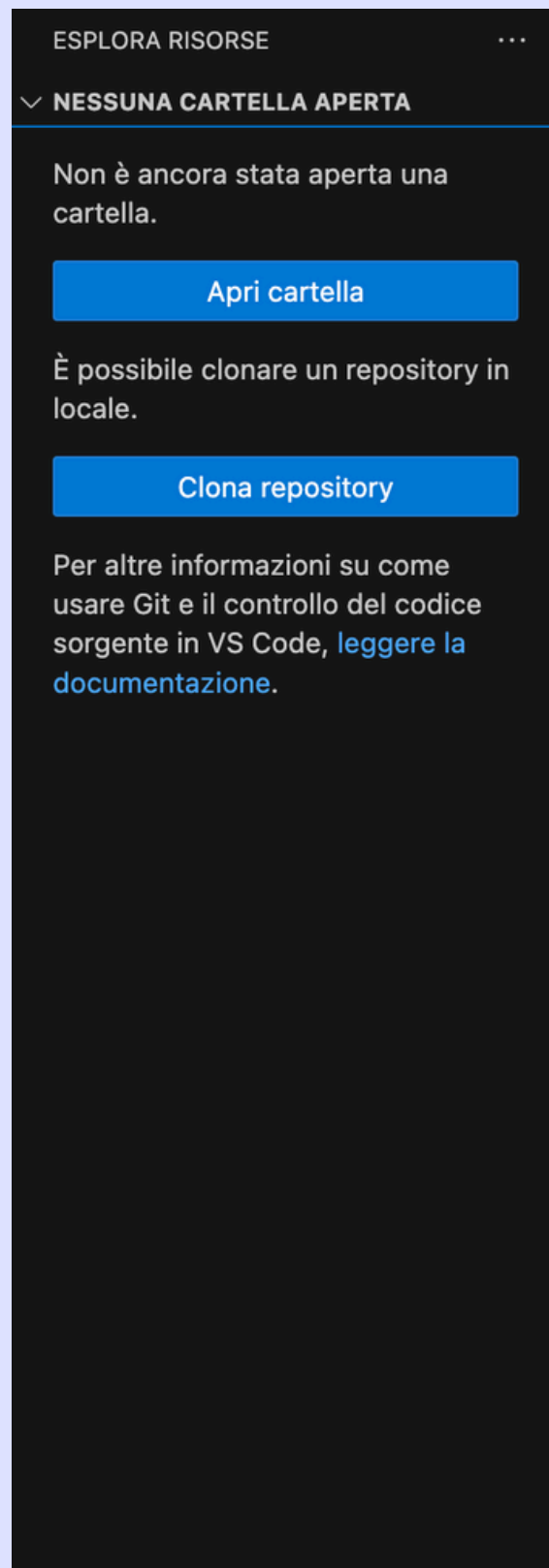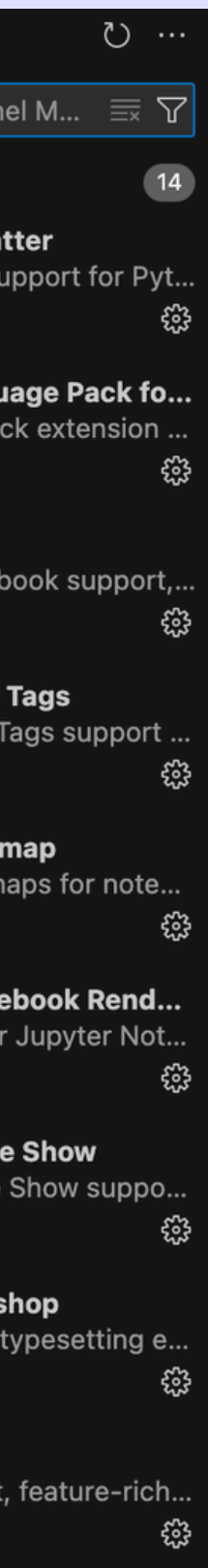Required plugins

`Python`

`Jupyter`
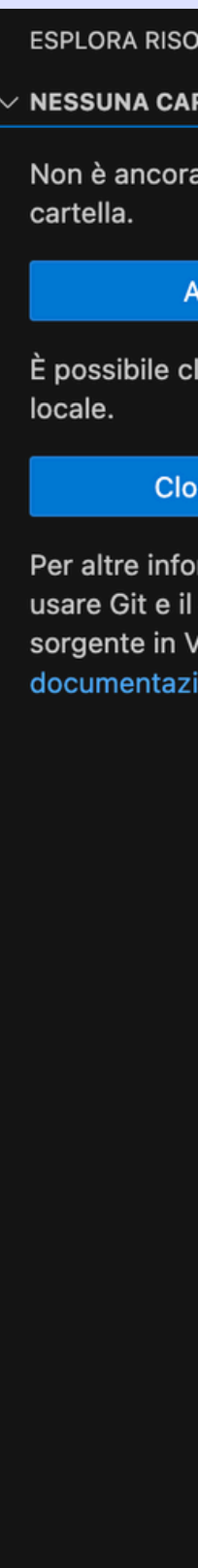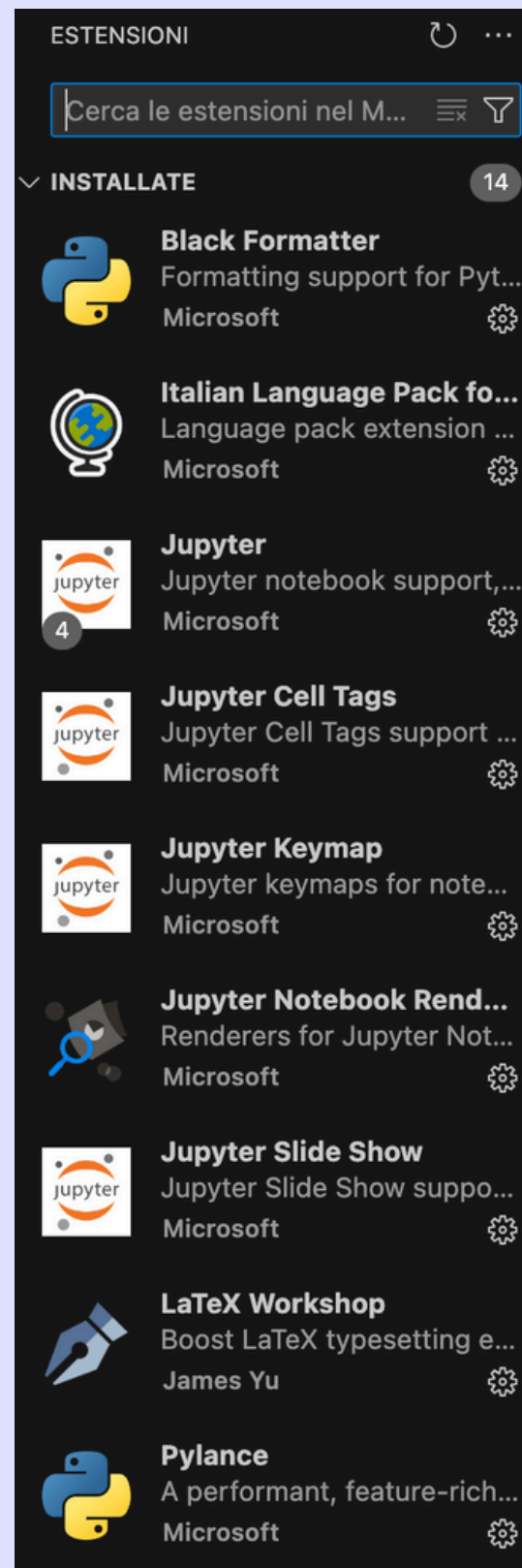
Plugins

Always open the folder on which you are working on, so that the predefined PATH will be setted as the current PATH.

Conclusions

In case of problems, please contact me by e-mail