



Adriano Di Florio (INFN & Politecnico Bari)

4 May 2023

Datasets



- **Option 1:** using Colab.
- **Option 2:** installing locally.
- **Option 3:** using ReCaS.

Google Colab

- Go to: <http://colab.research.google.com>

The screenshot shows the Google Colab interface. On the left, there's a sidebar with a navigation tree:

- Sommario
- Introduzione
- Data science
- Machine learning
- Altre risorse
- Esempi in primo piano
- Sezione

The main content area displays a video titled "Ti diamo il benvenuto in Colab". Below the video, there's a section titled "Cos'è Colab?" with a brief description and a bulleted list:

- Nessuna configurazione necessaria
- Accesso alla GPU senza costi
- Condivisione semplificata

There's also a section titled "Introduzione" with a note about the document being a notebook.

A central modal dialog is open, showing a list of recent notebooks:

Titolo	Aperiti per ultimi	Aperiti per primi
Un benvenuto a Colaboratory	11:45	23 gen 2020
Datasets.ipynb	10:20	21 gen 2020
Plotting.ipynb	10:19	21 gen 2020
Basics	10:19	19 gen 2020
LumDiff.ipynb	2 maggio	7 marzo

At the bottom of the dialog, there are buttons for "Nuovo blocco note" and "Annulla".

At the very bottom of the screen, there are two code cells:

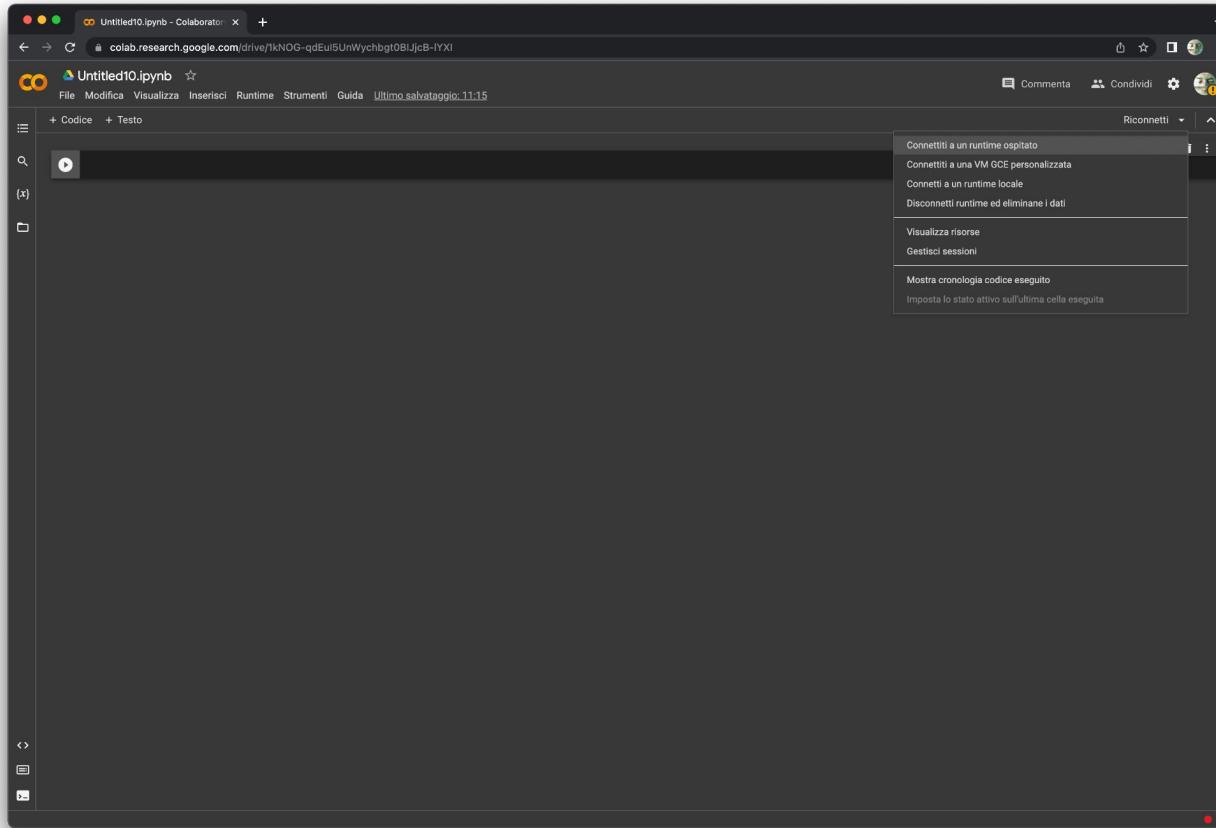
```
[1]: seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
86400
```

Per eseguire il codice nella cella sopra, seleziona con un clic e poi premi il pulsante Riprodi a sinistra del codice o usa la scorciatoia da tastiera "Comando/Ctrl+Invio". Per modificare il codice, fai clic sulla cella e inizia a modificarlo.

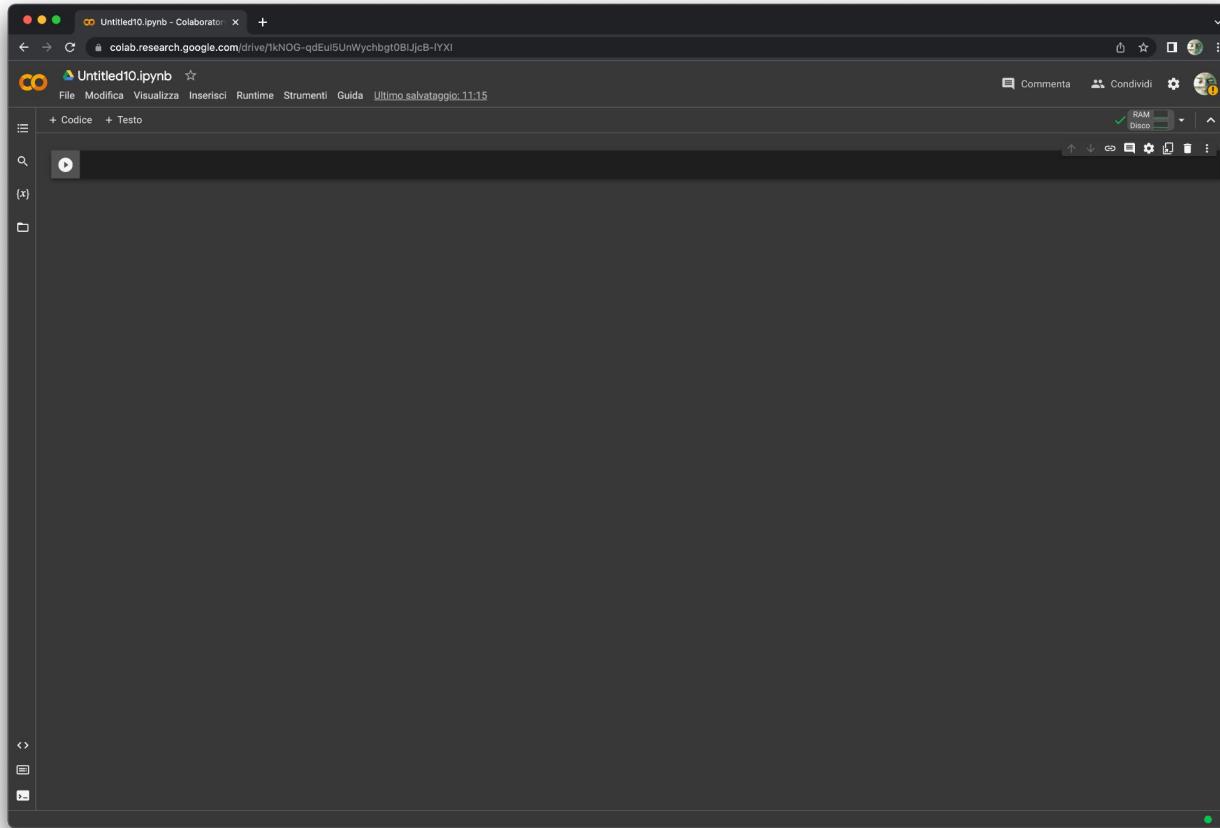
Le variabili che definisci in una cella possono essere usate in seguito in altre celle:

```
[1]: seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
524800
```

Google Colab



Google Colab



Google Colab



The screenshot shows the Google Colab interface. At the top, there's a navigation bar with a back arrow, a search bar containing 'colab.research.google.com/drive/1kNOG-qdEuI5UnWychbgt0B1jcb-IYXI', and a file icon. The main title is 'Untitled10.ipynb - Collaboratory'. Below the title is a menu bar with 'File', 'Modifica', 'Visualizza', 'Inserisci', 'Runtime', 'Strumenti', 'Guida', and a timestamp 'Ultimo salvataggio: 11:15'. On the left, there's a sidebar with a search bar, a file tree, and a code/tester switch. The main workspace is currently empty. On the right, there's a 'Risorse' panel with a message about no subscription, a 'Gestisci sessioni' button, and a 'Vuoi più memoria e spazio su disco?' upgrade offer. Below that, it shows 'Backend Google Compute Engine Python 3' and 'Inizio visualizzazione risorse: 12:03'. It displays two resource status boxes: 'RAM di sistema 0.9 / 12.7 GB' and 'Disco 26.4 / 107.7 GB'. A 'Cambia tipo di runtime' button is at the bottom of the resources panel.

Installing Locally

- Install [MiniConda](#)

The screenshot shows the official Miniconda documentation page on docs.conda.io. The left sidebar has a green header with "conda latest". It includes sections for Conda, Conda-build, and Miniconda, with "Miniconda" expanded to show "System requirements" (Latest Miniconda Installer Links, Windows installers, macOS installers, Linux installers, Installing, Other resources), "Help and support" (Contributing, Conda license), and "See if Miniconda is right for you." The main content area is titled "Miniconda" and describes it as a free minimal installer for conda. It lists system requirements for Windows, macOS, and Linux, noting compatibility with Python 3.10.10 and various architectures. A note says "On Windows, macOS, and Linux, it is best to install Miniconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Miniconda system wide, which does require administrator permissions." Below this is a section titled "Latest Miniconda Installer Links" with a table of SHA256 hashes for various platforms and architectures. The table is titled "Latest - Conda 23.3.1 Python 3.10.10 released April 24, 2023". The "Windows" row contains links for Miniconda3 Windows 64-bit and 32-bit. The "macOS" row contains links for Miniconda3 macOS Intel x86 64-bit bash, pkg, and M1 64-bit bash and pkg. The "Linux" row contains links for Miniconda3 Linux 64-bit, aarch64 64-bit, and s390x 64-bit.

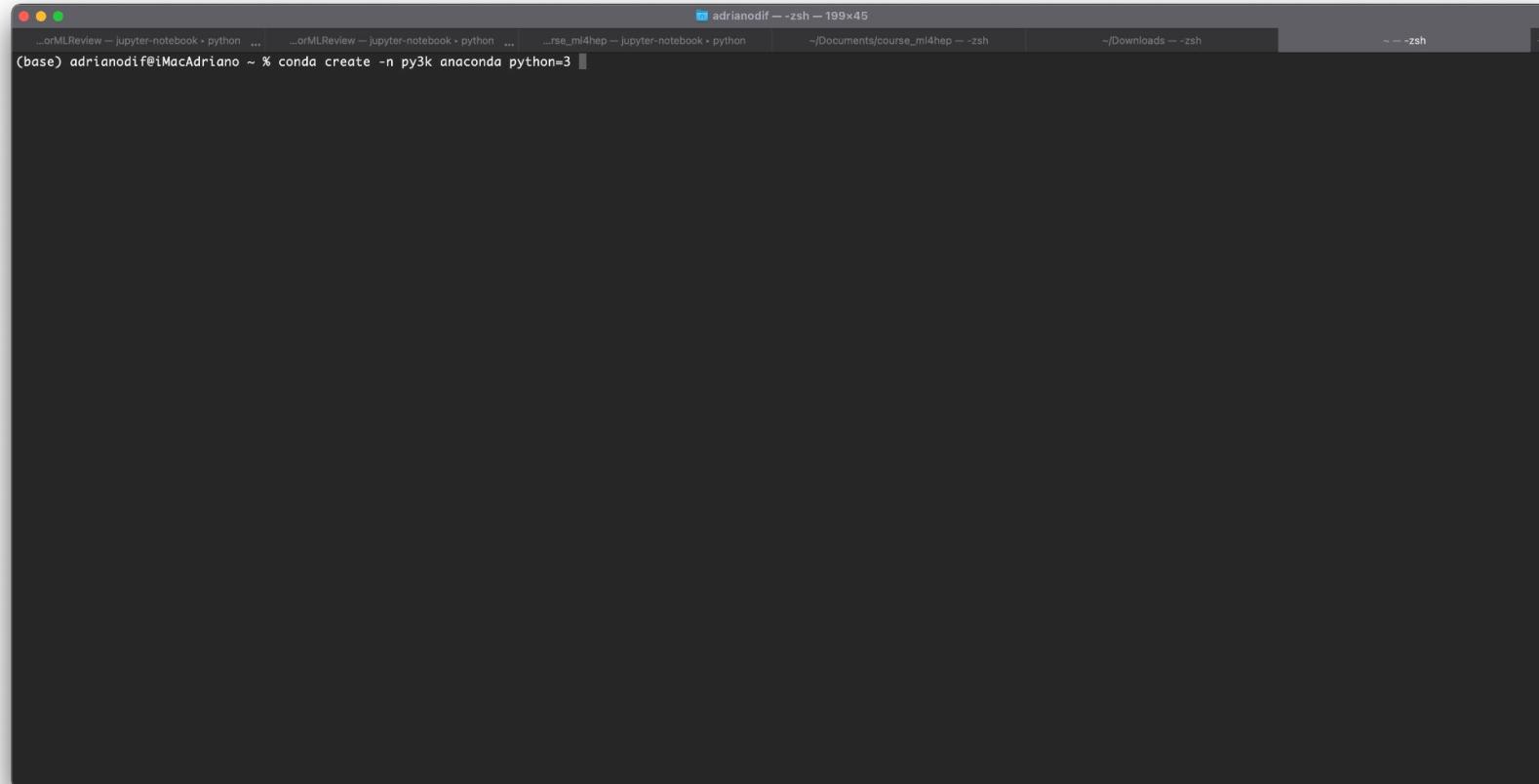
Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	387194e1f120beb52b883634e89cc67db4f7980bd542254b43d3309ea7fc395b
	Miniconda3 Windows 32-bit	4f16d6cc6c28088bea16994fb4a829119a3e3145baa0bb0a5344174ab85d5462
macOS	Miniconda3 macOS Intel x86 64-bit bash	5ab70b644b7de9d14ade110514cc90293b830c8b1ba9cfd8b9cc4153f8184
	Miniconda3 macOS Intel x86 64-bit pkg	cc3a110f1e5394f2b39726dc22551c2a19afef699c13a2566887ba796c9a5b8
	Miniconda3 macOS Apple M1 64-bit bash	9d1d12573339c4995b0b0d8a44bf0fffc32d3e1b5db478c18878e08d64
	Miniconda3 macOS Apple M1 64-bit pkg	6997472c5f99b0772eb776397f4ce322773b083a7f7b0396a2306cc7fc3b75d
Linux	Miniconda3 Linux 64-bit	ae2f29960aae7f7f7949f18aa0d17e0e5f6aa9c97487c703486f6f81f14819e5651
	Miniconda3 Linux-aarch64 64-bit	6950c7b1f4165ce9b87eda2e684837771ae7b2e0844e0aa9e915d1dedc924c
	Miniconda3 Linux-ppc64le 64-bit	b3d5e38c0542bc4f5a2f212a7939826808e8419e1459755f3cefe4763e51d18
	Miniconda3 Linux-s390x 64-bit	ed4f51a1c967e921ff5721151f567a4c43c2d88ac93ec2393c6238b8c4891de8

Windows installers

Windows

Installing Locally

- Create your **python** environment



The screenshot shows a macOS desktop with several terminal windows open in a docked interface. The active terminal window is titled "adrianodif -- zsh". The command entered is:

```
(base) adrianodif@iMacAdriano ~ % conda create -n py3k anaconda python=3
```

Datasets

- Create your `python` environment

The screenshot shows a macOS terminal window with several tabs open. The active tab displays the output of a `conda create` command. The output shows the packages being added, their builds, and their total size.

```
adrianodif — conda create -n py3k python=3 — 199x45
...orMLReview — jupyter-notebook + python ...
...orMLReview — jupyter-notebook + python ...
...rse_ml4hep — jupyter-notebook + python ...
~/Documents/course_ml4hep — zsh
~/Downloads — zsh
— conda create -n py3k python=3

added / updated specs:
- python=3

The following packages will be downloaded:

  package          |      build
-----|-----
ca-certificates-2023.01.10 | hecd8cb5_0
libffi-3.4.2           | hecd8cb5_6
ncurses-6.4             | hcec6c5f_0
openssl-1.1.1t          | hca72ff7_0
pip-23.0.1              | py311hecd8cb5_0
python-3.11.3            | h1fd4e5f_0
readline-8.2              | hca72ff7_0
setuptools-66.0.0         | py311hecd8cb5_0
sqlite-3.41.2             | h6c40b1e_0
tk-8.6.12                | h5d9f67b_0
tzdata-2023c              | h04d1e81_0
wheel-0.38.4              | py311hecd8cb5_0
xz-5.2.10                 | h6c40b1e_1
zlib-1.2.13                | h4dc903c_0

Total: 29.6 MB

The following NEW packages will be INSTALLED:

  bzip2                  pkgs/main/osx-64::bzip2-1.0.8-h1de35cc_0 None
  ca-certificates          pkgs/main/osx-64::ca-certificates-2023.01.10-hecd8cb5_0 None
  libffi                  pkgs/main/osx-64::libffi-3.4.2-hecd8cb5_6 None
  ncurses                  pkgs/main/osx-64::ncurses-6.4-hcec6c5f_0 None
  openssl                  pkgs/main/osx-64::openssl-1.1.1t-hca72ff7_0 None
  pip                      pkgs/main/osx-64::pip-23.0.1-py311hecd8cb5_0 None
  python                   pkgs/main/osx-64::python-3.11.3-h1fd4e5f_0 None
  readline                  pkgs/main/osx-64::readline-8.2-hca72ff7_0 None
  setuptools                  pkgs/main/osx-64::setuptools-66.0.0-py311hecd8cb5_0 None
  sqlite                    pkgs/main/osx-64::sqlite-3.41.2-h6c40b1e_0 None
  tk                        pkgs/main/osx-64::tk-8.6.12-h5d9f67b_0 None
  tzdata                   pkgs/main/noarch::tzdata-2023c-h04d1e81_0 None
  wheel                     pkgs/main/osx-64::wheel-0.38.4-py311hecd8cb5_0 None
  xz                        pkgs/main/osx-64::xz-5.2.10-h6c40b1e_1 None
  zlib                     pkgs/main/osx-64::zlib-1.2.13-h4dc903c_0 None

Proceed ([y]/n)?
```

Installing Locally

- Install jupyter-notebook

```
...orMLReview — jupyter-notebook + python ... ..._orMLReview — jupyter-notebook + python ... ...rse_ml4hep — jupyter-notebook + python ... ~Documents/course_ml4hep — zsh ... ~Downloads — zsh ... — zsh

openssl      pkgs/main/osx-64::openssl-1.1.1t-hca72f7f_0 None
pip          pkgs/main/osx-64::pip-23.0.1-py31hecd8cb5_0 None
python        pkgs/main/osx-64::python-3.11.3-h1fd4e5f_0 None
readline      pkgs/main/osx-64::readline-8.2-hca72f7f_0 None
setuptools   pkgs/main/osx-64::setuptools-66.0.0-py31hecd8cb5_0 None
sqlite        pkgs/main/osx-64::sqlite-3.41.2-h6c40b1e_0 None
tk            pkgs/main/osx-64::tk-8.6.12-h5d9f67b_0 None
tzdata        pkgs/main/noarch::tzdata-2023c-h04d1e81_0 None
wheel         pkgs/main/osx-64::wheel-0.38.4-py31hecd8cb5_0 None
xz            pkgs/main/osx-64::xz-5.2.10-h6c40b1e_1 None
zlib          pkgs/main/osx-64::zlib-1.2.13-h4dc903c_0 None

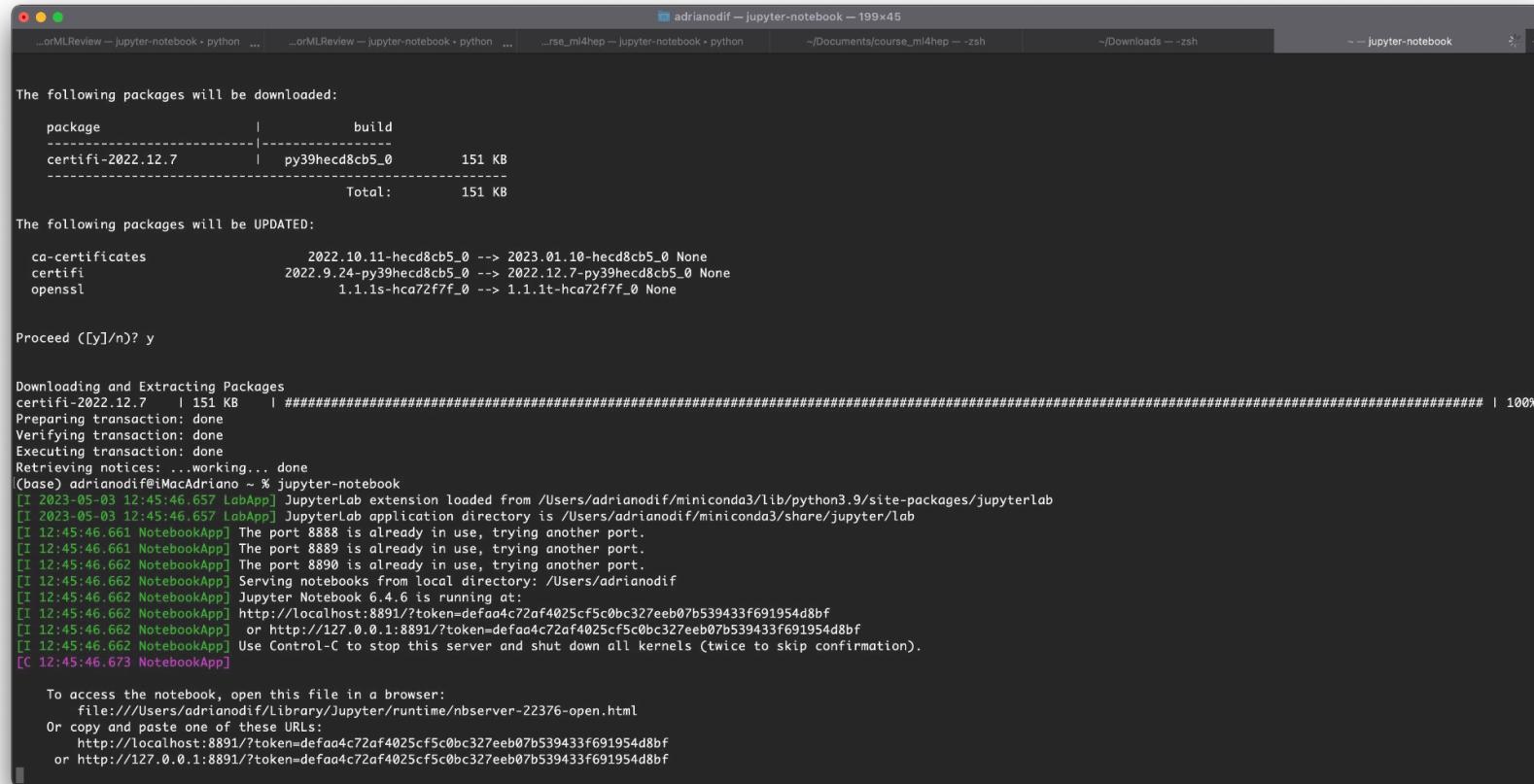
Proceed ([y]/n)? y

Downloading and Extracting Packages
xz-5.2.10           | 263 KB    | #####.....................................# | 100%
wheel-0.38.4        | 79 KB     | #####.....................................# | 100%
pip-23.0.1          | 2.8 MB    | #####.....................................# | 100%
ncurses-6.4          | 1018 KB   | #####.....................................# | 100%
tk-8.6.12           | 3.1 MB    | #####.....................................# | 100%
openssl-1.1.1t       | 3.3 MB    | #####.....................................# | 100%
libffi-3.4.2          | 127 KB   | #####.....................................# | 100%
sqlite-3.41.2        | 1.2 MB    | #####.....................................# | 100%
setuptools-66.0.0     | 1.6 MB    | #####.....................................# | 100%
tzdata-2023c         | 116 KB    | #####.....................................# | 100%
python-3.11.3         | 15.5 MB   | #####.....................................# | 100%
zlib-1.2.13           | 96 KB     | #####.....................................# | 100%
readline-8.2          | 328 KB    | #####.....................................# | 100%
ca-certificates-2023 | 121 KB    | #####.....................................# | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
# $ conda activate py3k
#
# To deactivate an active environment, use
#
# $ conda deactivate

Retrieving notices: ...working... done
(base) adrianodif@iMacAdriano ~ % conda install jupyter
```

Installing Locally

- Install jupyter-notebook



The following packages will be downloaded:

package	build
certifi-2022.12.7	py39hecd8cb5_0
Total:	151 KB

The following packages will be UPDATED:

package	old version	new version	status
ca-certificates	2022.10.11-hecd8cb5_0	2023.01.10-hecd8cb5_0	None
certifi	2022.9.24-py39hecd8cb5_0	2022.12.7-py39hecd8cb5_0	None
openssl	1.1.1s-hca72f7f_0	1.1.1t-hca72f7f_0	None

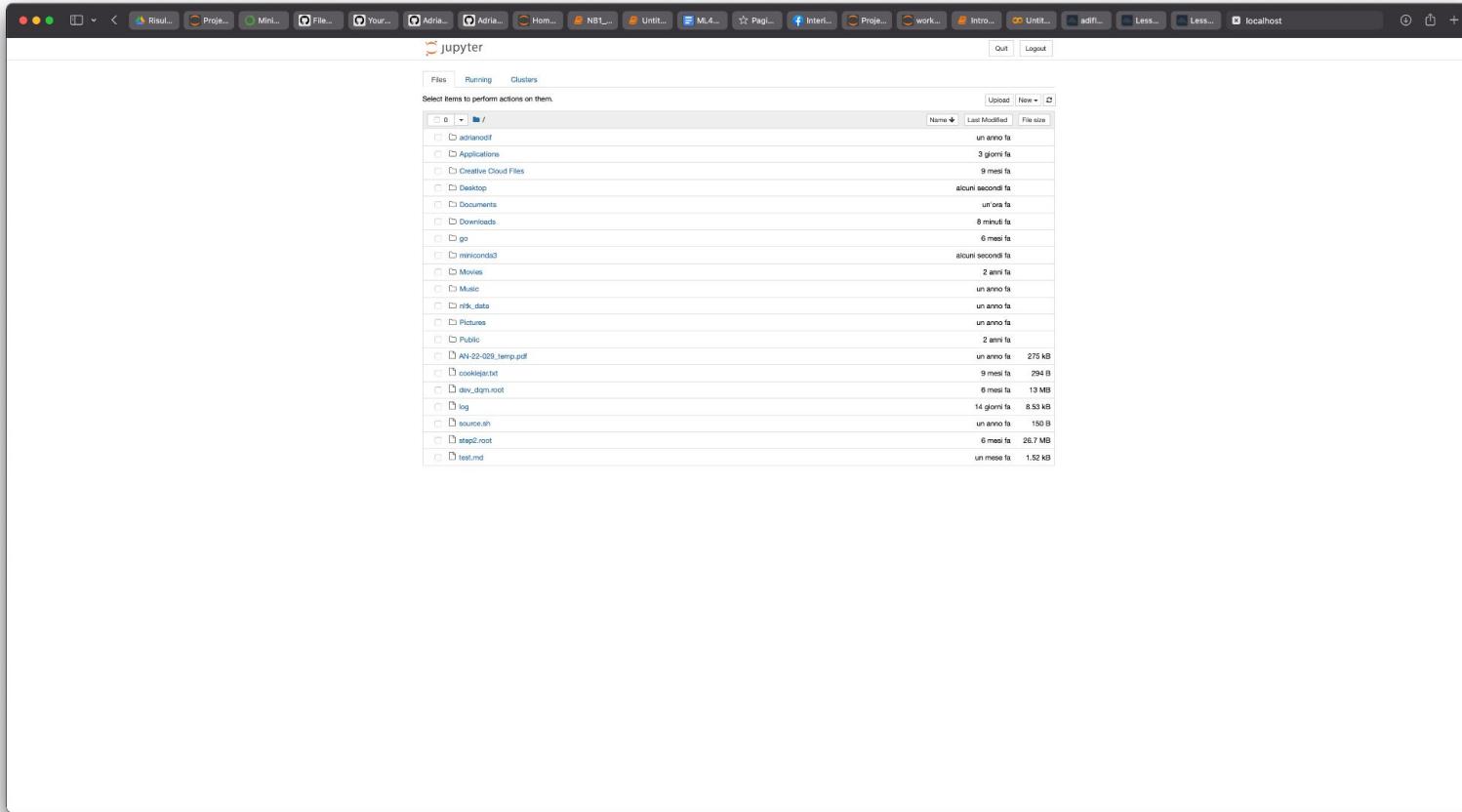
Proceed ([y]/n)? y

Downloading and Extracting Packages
certifi-2022.12.7 | 151 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done
(base) adrianodif@iMacAdriano ~ % jupyter-notebook
[I 2023-05-03 12:45:46.657 LabApp] JupyterLab extension loaded from /Users/adrianodif/miniconda3/lib/python3.9/site-packages/jupyterlab
[I 2023-05-03 12:45:46.657 LabApp] JupyterLab application directory is /Users/adrianodif/miniconda3/share/jupyter/lab
[I 12:45:46.661 NotebookApp] The port 8888 is already in use, trying another port.
[I 12:45:46.661 NotebookApp] The port 8889 is already in use, trying another port.
[I 12:45:46.662 NotebookApp] The port 8890 is already in use, trying another port.
[I 12:45:46.662 NotebookApp] Serving notebooks from local directory: /Users/adrianodif
[I 12:45:46.662 NotebookApp] Jupyter Notebook 6.4.6 is running at:
[I 12:45:46.662 NotebookApp] http://localhost:8891/?token=defaa4c72af4025cf5c0bc327eeb07b539433f691954d8bf
[I 12:45:46.662 NotebookApp] or http://127.0.0.1:8891/?token=defaa4c72af4025cf5c0bc327eeb07b539433f691954d8bf
[I 12:45:46.662 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 12:45:46.673 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/adrianodif/Library/Jupyter/runtime/nbserver-22376-open.html
Or copy and paste one of these URLs:
http://localhost:8891/?token=defaa4c72af4025cf5c0bc327eeb07b539433f691954d8bf
or http://127.0.0.1:8891/?token=defaa4c72af4025cf5c0bc327eeb07b539433f691954d8bf

Installing Locally

- Run it!

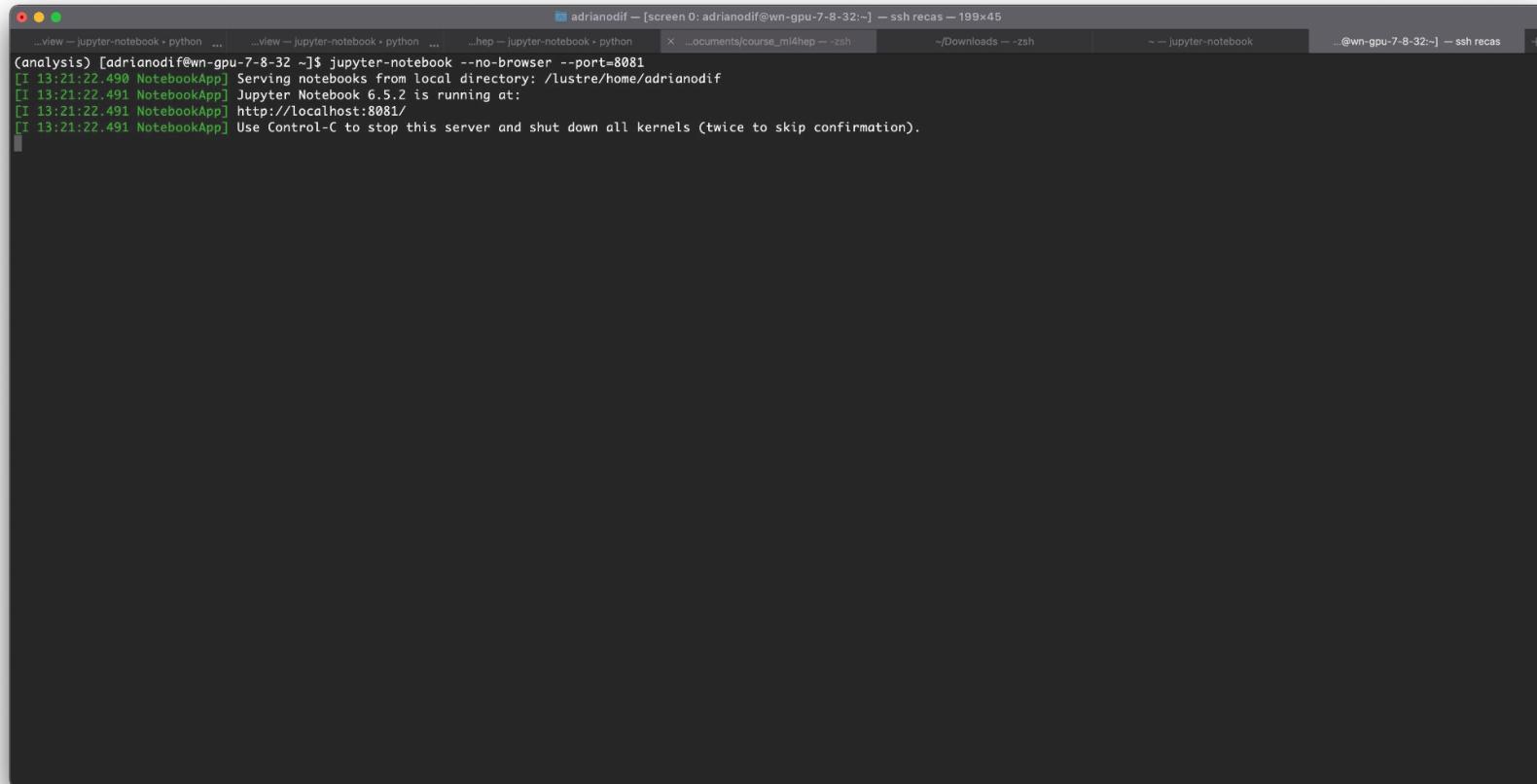


Running on ReCaS

- If you have access to recas HPC/HTC you can run it there having access to the cluster machines.
- First of all follow the instructions given for the local installation on the machine you have access to.
- Then you would need to follow the instructions [here](#) to have an **ssh** tunnel to the machine you are logged in and your local machine. There you will also find a summary of the instructions I've sketched above.
- **Warning!** Since there may be other people connected you will need to choose a port that you are ~sure is not already taken (i.e. don't use the common ones 8000 8080 8888 etc.)

Running on ReCaS

- [on remote/ssh'd machine] Start the notebook.

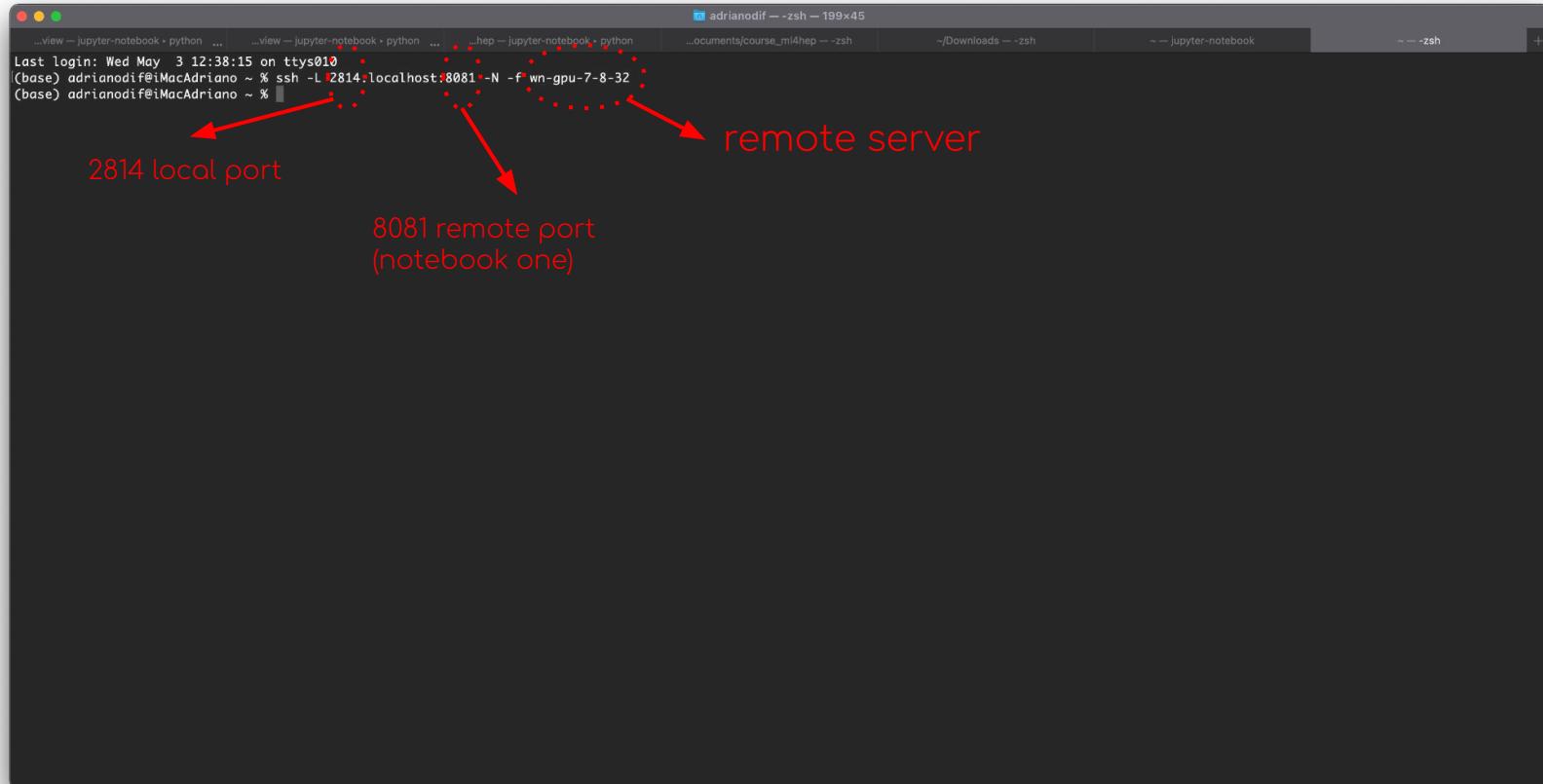


```
adrianodif ~ [screen 0: adrianodif@wn-gpu-7-8-32:~] ssh recas - 199x45
...view -- jupyter-notebook > python ... ...view -- jupyter-notebook > python ... ...hep -- jupyter-notebook > python X ...documents/course_ml4hep -- zsh ~/Downloads -- zsh ~ -- jupyter-notebook ...@wn-gpu-7-8-32:~] ssh recas + 

(adrianodif) [adrianodif@wn-gpu-7-8-32 ~]$ jupyter-notebook --no-browser --port=8081
[I 13:21:22.490 NotebookApp] Serving notebooks from local directory: /lustre/home/adrianodif
[I 13:21:22.491 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 13:21:22.491 NotebookApp] http://localhost:8081/
[I 13:21:22.491 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Running on ReCaS

- [on local machine] Tunneling from local to remote.



A screenshot of a macOS terminal window titled "adrianodif -- zsh - 199x45". The window contains a command line session:

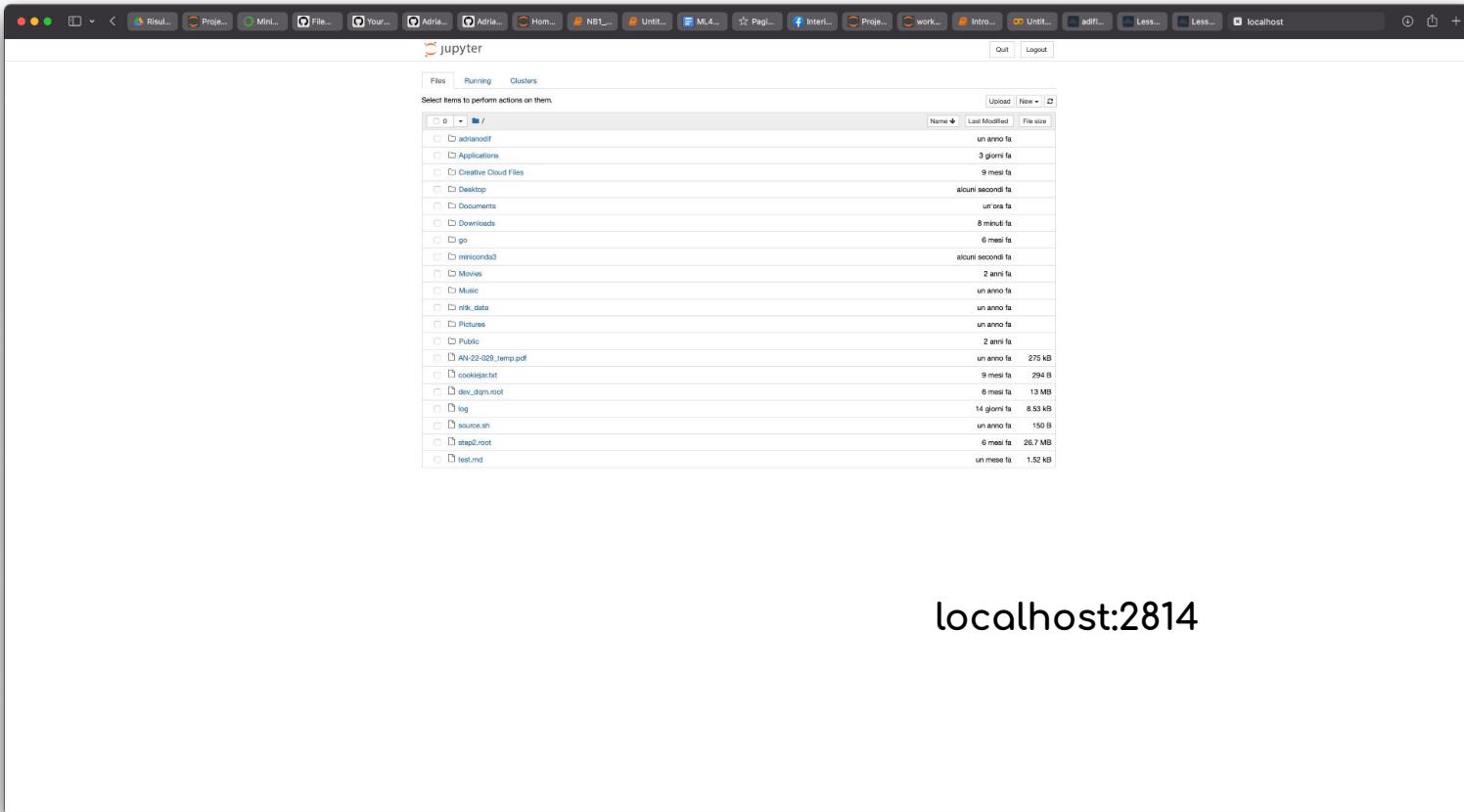
```
Last login: Wed May  3 12:38:15 on ttys010
(base) adrianodif@MacAdriano ~ % ssh -L 2814:localhost:8081 -N -f wn-gpu-7-8-32
(base) adrianodif@MacAdriano ~ %
```

Annotations with red arrows point to specific parts of the command:

- An arrow points to the number **2814** with the label **2814 local port**.
- An arrow points to the number **8081** with the label **8081 remote port (notebook one)**.
- An arrow points to the word **localhost** with the label **remote server**.

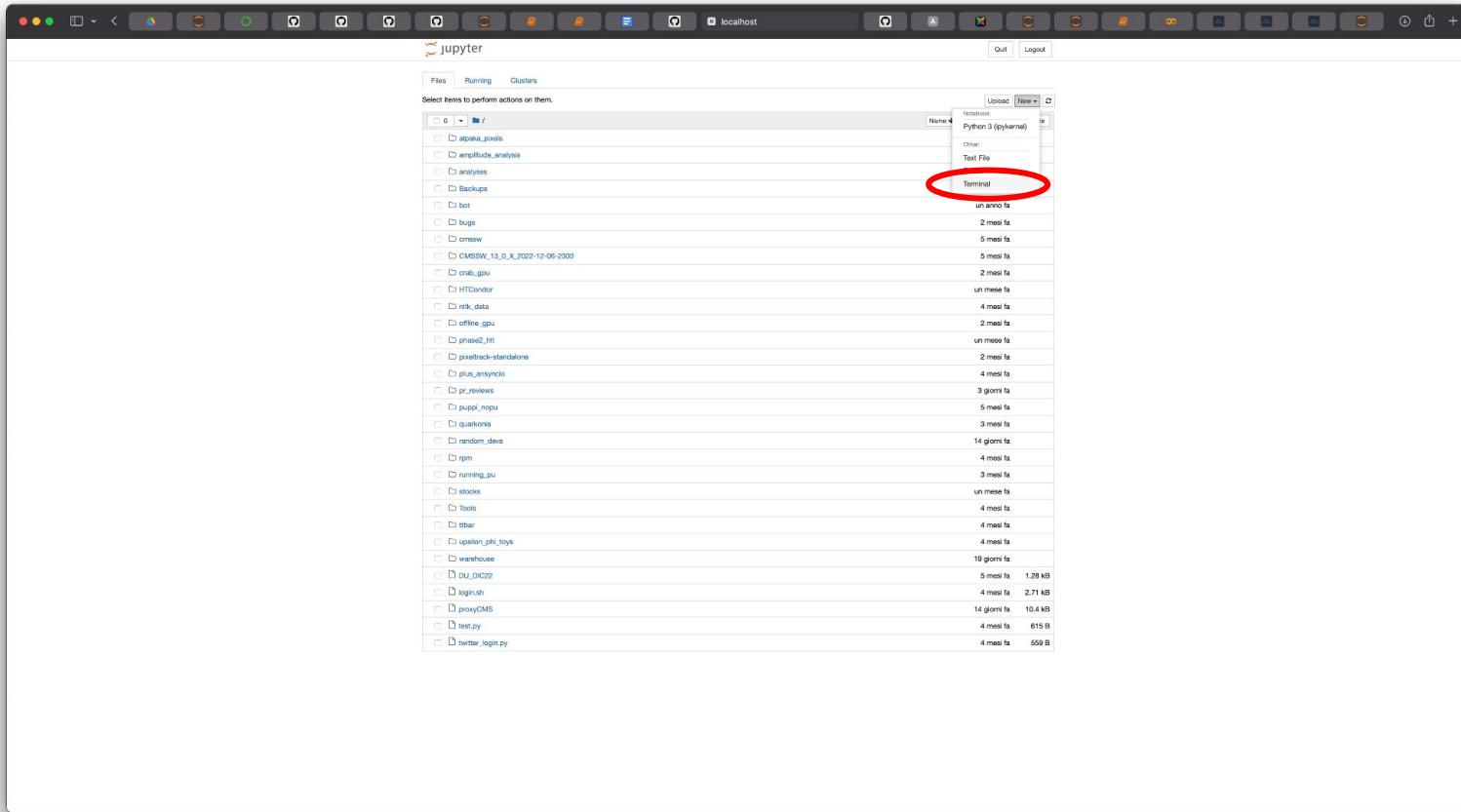
Running on ReCaS

- My notebook running on a remote machine.



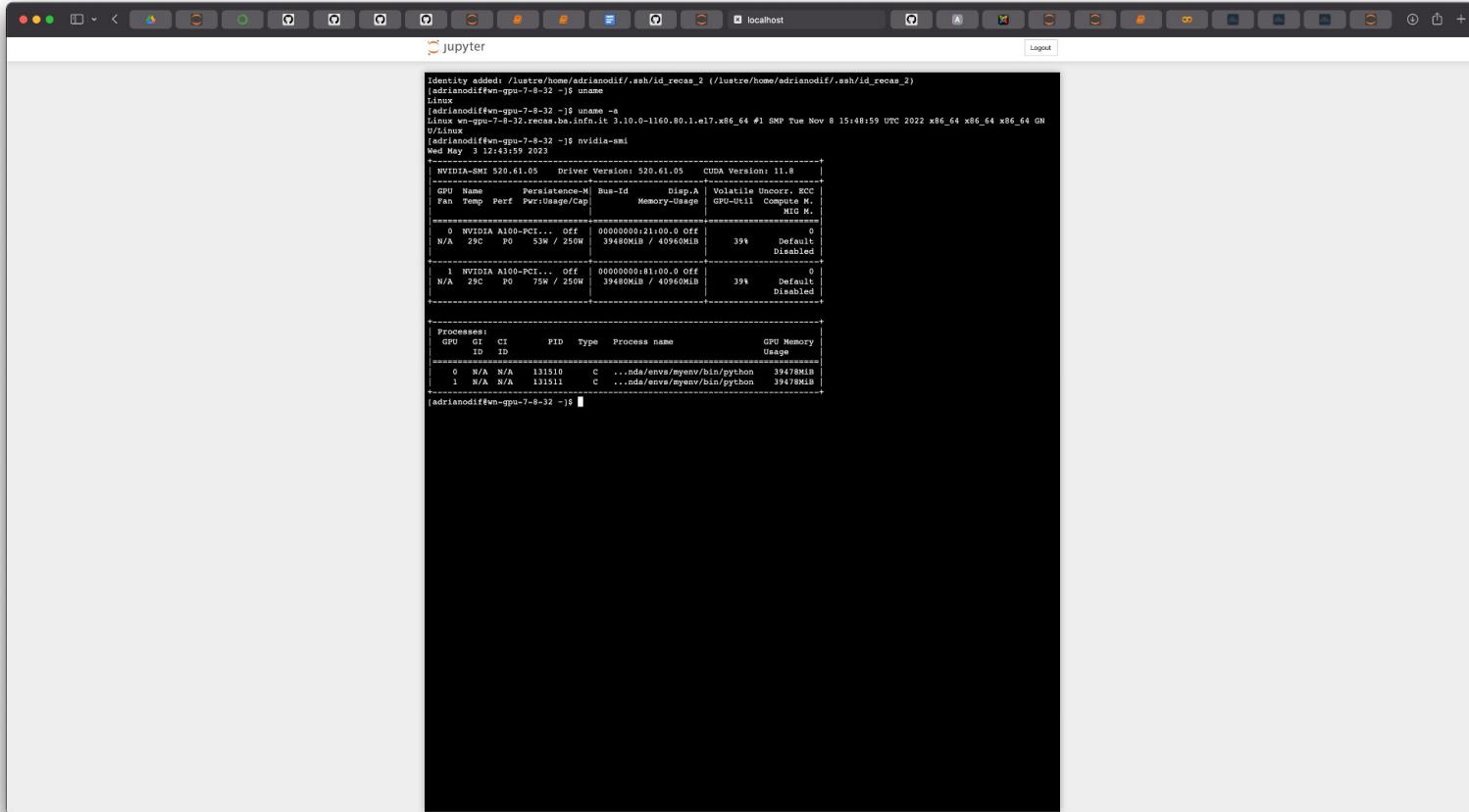
Running on ReCaS

- Let's open a terminal.



Running on ReCaS

- Running a **terminal** from a **browser** on the remote machine.





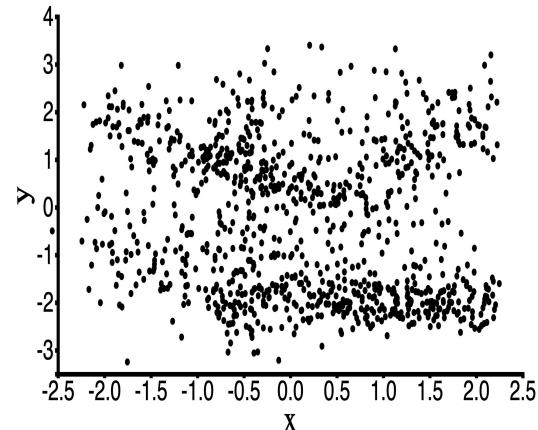
Decision Trees

Adriano Di Florio (INFN & Politecnico Bari)

Terminology

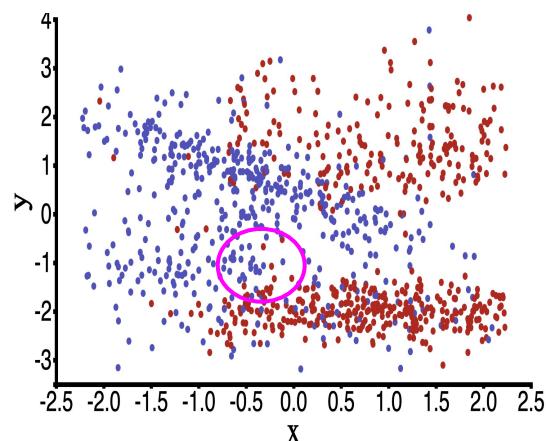
Test data

The ultimate goal of building a classifier is to be able to use it on previously unseen data and recover the correct classifications for each data sample. This unseen dataset is typically referred to as your *test data*.



Training data

In order to build/train your classifier, you will need to provide a library of examples of each target class. For supervised learning, this dataset must be pre-labelled and it is typically referred to as your *training data*.

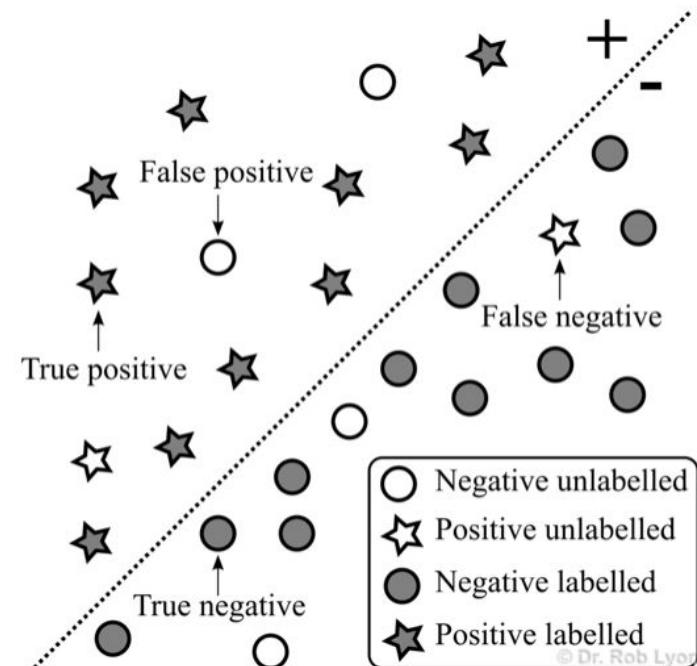


Validation data

The validation dataset is a **subset** of your training data. It is not your test data. You must not use your test data to train your classifier in any way.

Metrics

- There are a variety of ways to evaluate the performance of a machine learning model. Which one you choose should depend on the objective of your classification. Before we look at some common performance metrics we first need to define a few terms.
- Suppose we have two target classes; these could be *cat* and *mouse*, or alternatively *pulsar* and *non-pulsar*, but here I'm just going to call them *positive* and *negative*. When we apply a machine learning model to the unlabelled test data composed of these classes it fits a split that looks like this:



Metrics

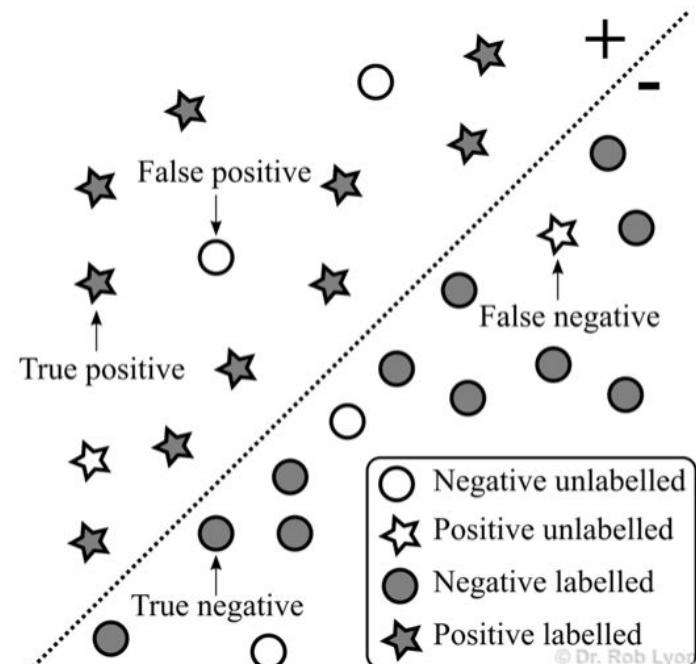
- An unlabelled (test) data sample from class one that has been correctly labelled is called a **true positive**, but a sample that has been incorrectly labelled is called a **false negative**; likewise, an unlabelled (test) data sample from class two that has been correctly labelled is called **true negative**, and a sample that has been incorrectly labelled is called a **false positive**.
- Some commonly used performance metrics are:

$$precision = \frac{T_P}{T_P + F_P}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

$$recall = \frac{T_P}{T_P + F_N}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$



© Dr. Rob Lyon

Confusion Matrix

- An unlabelled (test) data sample from class one that has been correctly labelled is called a ***true positive***, but a sample that has been incorrectly labelled is called a ***false negative***; likewise, an unlabelled (test) data sample from class two that has been correctly labelled is called ***true negative***, and a sample that has been incorrectly labelled is called a ***false positive***.
- We use these names to describe the different types of errors and hence the performance metrics of the machine learning model.

		Truth	
		Positive	Negative
Prediction	Positive	True Positive (TP)	False Positive (FP) Type I Error
	Negative	False Negative (FN) Type II Error	True Negative (TN)