

Setting up your computer for the labs

Please read fully before starting

Setting up the environment

This only needs to be done once

Miniconda — Conda document +

docs.conda.io/en/latest/miniconda.html#latest-miniconda-installer-links

Conda latest Search docs

Conda
Conda-build

Miniconda

- System requirements
- Latest Miniconda Installer Links
- Windows installers
- macOS installers
- Linux installers
- Installing
- Other resources

Help and support

Contributing

Conda license

Read the Docs v: latest

Latest Miniconda Installer Links

Latest - Conda 4.10.3 Python 3.9.5 released July 21, 2021

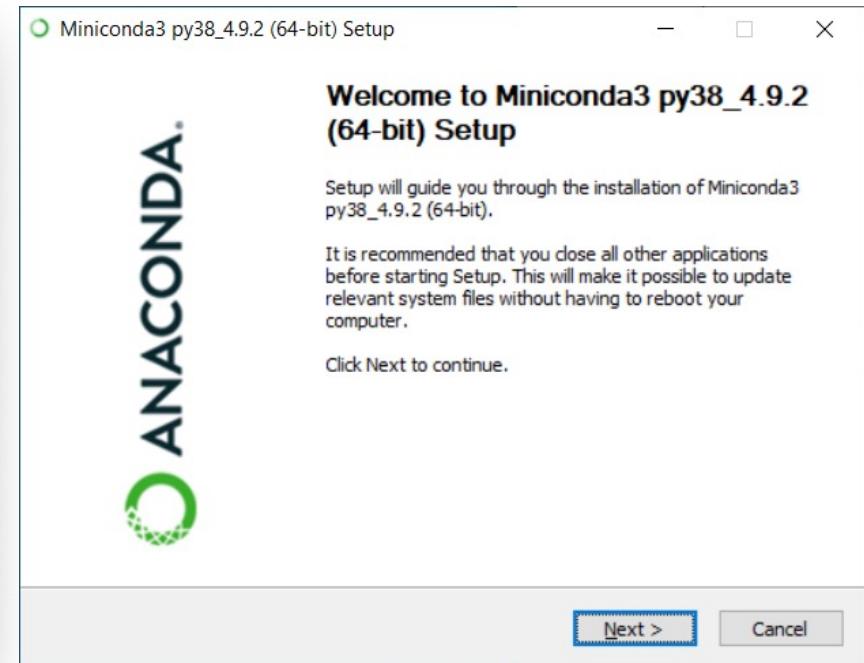
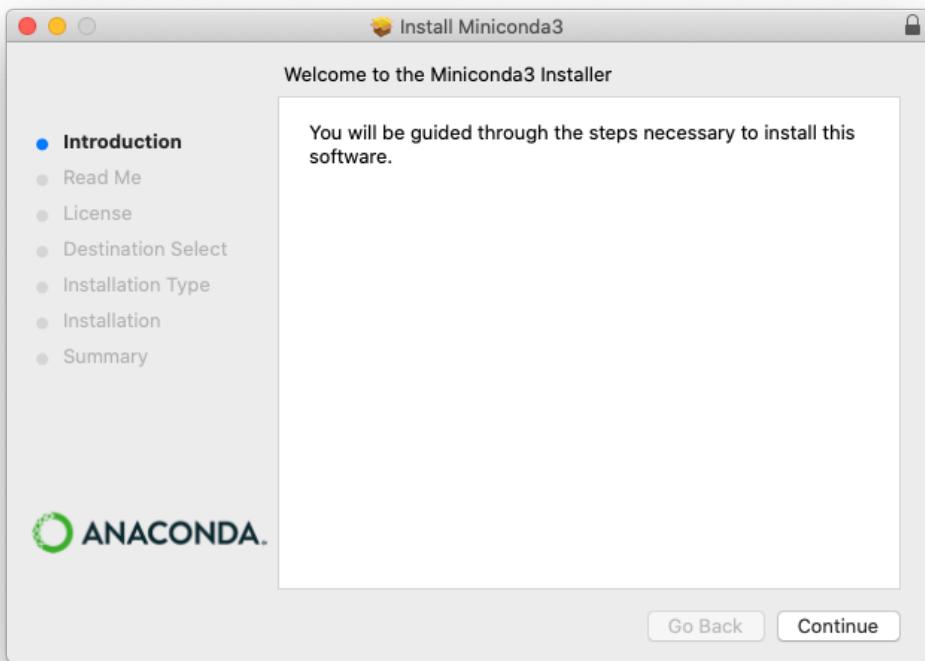
Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	b33797064593ab2229a0135dc69001bea05cb56a20c2f243b1231
	Miniconda3 Windows 32-bit	24f438e57ff2ef1ce1e93050d4e9d13f5050955f759f448d84a40
MacOSX	Miniconda3 MacOSX 64-bit bash	786de9721f43e2c7d2803144c631
	Miniconda3 MacOSX 64-bit pkg	8fa371ae97218c3c005cd5f04b1
	Miniconda3 macOS Apple M1 64-bit bash (Py38 conda 4.10.1 2021-11-08)	4ce4047065f32e991eddb63b3
Linux	Miniconda3 Linux 64-bit	1ea2f885b4dbc3098662845560b
	Miniconda3 Linux-aarch64 64-bit	4879820a10718743f945d88ef14
	Miniconda3 Linux-ppc64le 64-bit	fa92ee4773611f58ed9333f977d
	Miniconda3 Linux-s390x 64-bit	1faed9abecf4a4dd4e0d8891fc

Windows installers

Windows

Python version	Name	Size	SHA256 hash
Python 3.9	Miniconda3 Windows 64-bit	58.1 MiB	b33797064593ab2229a0135dc69001bea05cb56a20c2f243b1231
Python 3.8	Miniconda3 Windows 64-bit	57.3 MiB	8940cdd621557bc55743d6bb4518c6d343a4587127e76de808fb0
Python 3.7	Miniconda3 Windows 64-bit	55.8 MiB	9c031506bfcb0428a0ac46c9152f9bdd48d5bd8aa83046691bf8e0
Python 3.9	Miniconda3 Windows 32-bit	55.3 MiB	24f438e57ff2ef1ce1e93050d4e9d13f5050955f759f448d84a40

Download the right version of Miniconda for your OS.



Install Miniconda. The default settings should be fine.

Neural Networks and Learning x +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2022VT_8A/CourseDocuments/Forms/AllItems.aspx?id=%2Fsites%2FLisam_TBMI26...

SharePoint Search this library

Lisam / Neural Networks and Learning Systems (TBMI26 2022VT 8A) Adjust course room Courses and programs

NN Neural Networks and Learning Systems (TBMI26 2022VT 8A) Private group

Home + New Upload Edit in grid view Share ... All Documents

Course plan Course documents Collaborative workspace Members and groups Schedule Teams Submissions Signup Lecture videos Edit

Course documents > Assignments > environment_files

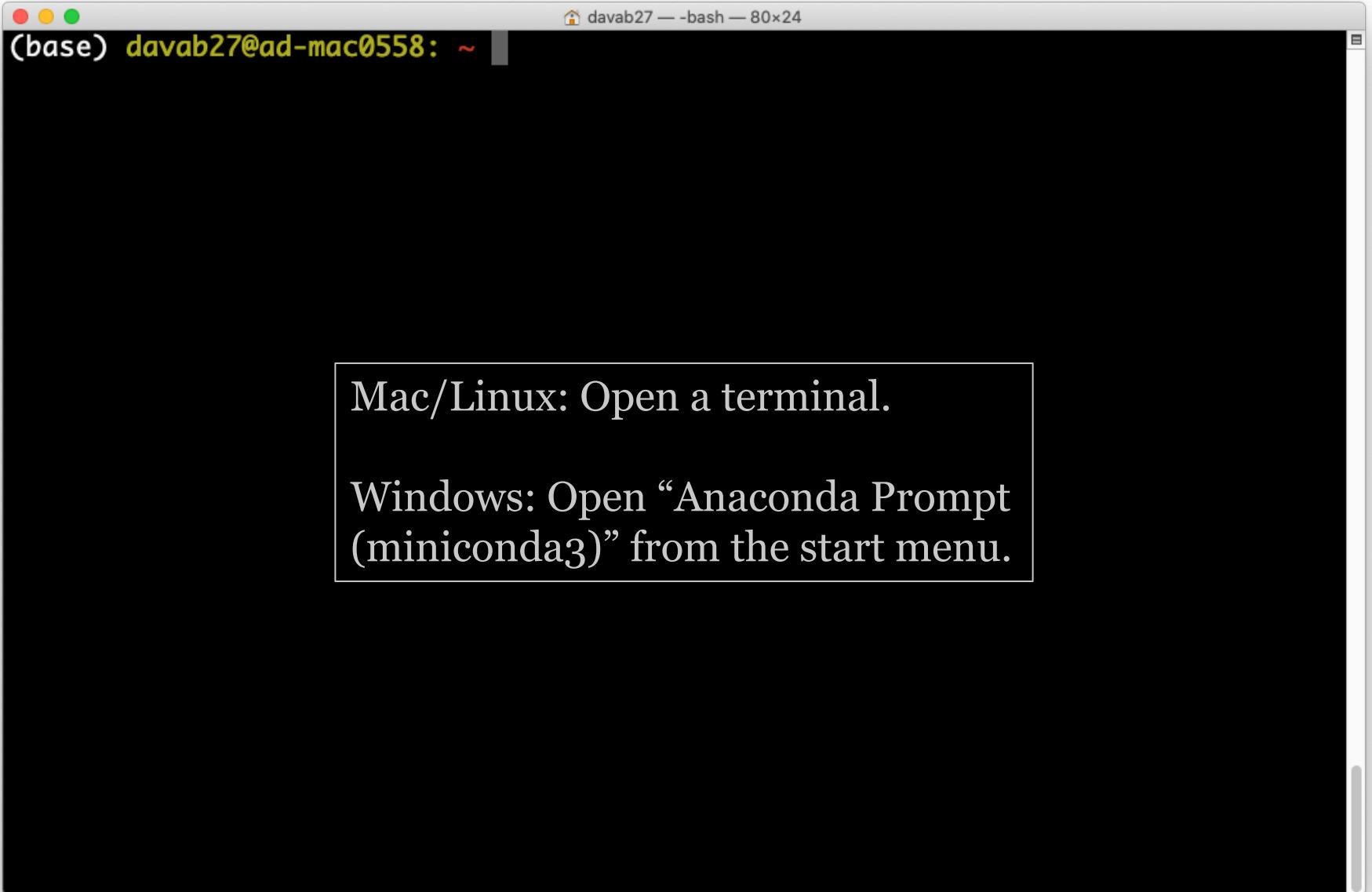
Name	Modified	Modified By	Version	Description
tbmi26_env.yml	21 minutes ago	David Abramian	1.0	
tbmi26_tf_env.yml	21 minutes ago	David Abramian	1.0	
tbmi26_tf_gpu_env.yml	21 minutes ago	David Abramian	1.0	

Return to classic SharePoint

Download the right yml file for your situation:

- Default environment for labs 1, 3, 4. Recommended!
- Usable also for lab 2 if you don't have an Nvidia GPU.
- Usable also for lab 2 if you have an Nvidia GPU.

Only the last of these will take advantage of a GPU.
The rest will run on the CPU.



```
 davab27 — bash — 80x24
(base) davab27@ad-mac0558: ~ conda config --set auto_activate_base false
(base) davab27@ad-mac0558: ~ conda deactivate
davab27@ad-mac0558: ~
```

Mac/Linux: Without going into explanations, run the commands above. You will not have to do this again.

Windows: Skip this and go to next step.

```
[davab27@ad-mac0558: ~ conda env create -n tbmi26 -f Downloads/tbmi26_env.yml ]
```

We create the conda environment used in this course by running the command above.

- Name of the conda environment.
- Path to the yml file downloaded earlier.

```
[davab27@ad-mac0558: ~ conda env create -n tbmi26 -f Downloads/tbmi26_env.yml ]
```

Collecting package metadata (repodata.json): done
Solving environment: -

This will download and install many packages and will probably take a while.

```
matplotlib-inline-0. | 12 KB    | ##### | 100%
bleach-4.1.0        | 123 KB   | ##### | 100%
importlib_metadata-4 | 12 KB    | ##### | 100%
jinja2-3.0.2        | 110 KB   | ##### | 100%
pip-21.2.2          | 1.8 MB   | ##### | 100%
prompt-toolkit-3.0.2 | 259 KB   | ##### | 100%
argcomplete-1.12.3  | 35 KB    | ##### | 100%
freetype-2.11.0     | 589 KB   | ##### | 100%
giflib-5.2.1         | 70 KB    | ##### | 100%
debugpy-1.5.1        | 1.7 MB   | ##### | 100%
mkl_fft-1.3.1        | 161 KB   | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate tbmi26
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

davab27@ad-mac0558: ~

We now have an environment with Jupyter and all the other packages we will need.

Neural Networks and Learning x +

liuonline.sharepoint.com/sites/Lisam_TBMI26_2022VT_8A/CourseDocuments/Forms/AllItems.aspx?id=%2Fsites%2FLisam_TBMI26...

SharePoint Search this library

Lisam / Neural Networks and Learning Systems (TBMI26 2022VT 8A) Adjust course room Courses and programs

NN Neural Networks and Learning Systems (TBMI26 2022VT 8A) Private group

Home + New Upload Edit in grid view Share ... All Documents

Course plan Course documents Collaborative workspace Members and groups Schedule Teams Submissions Signup Lecture videos Edit

Course documents > Assignments

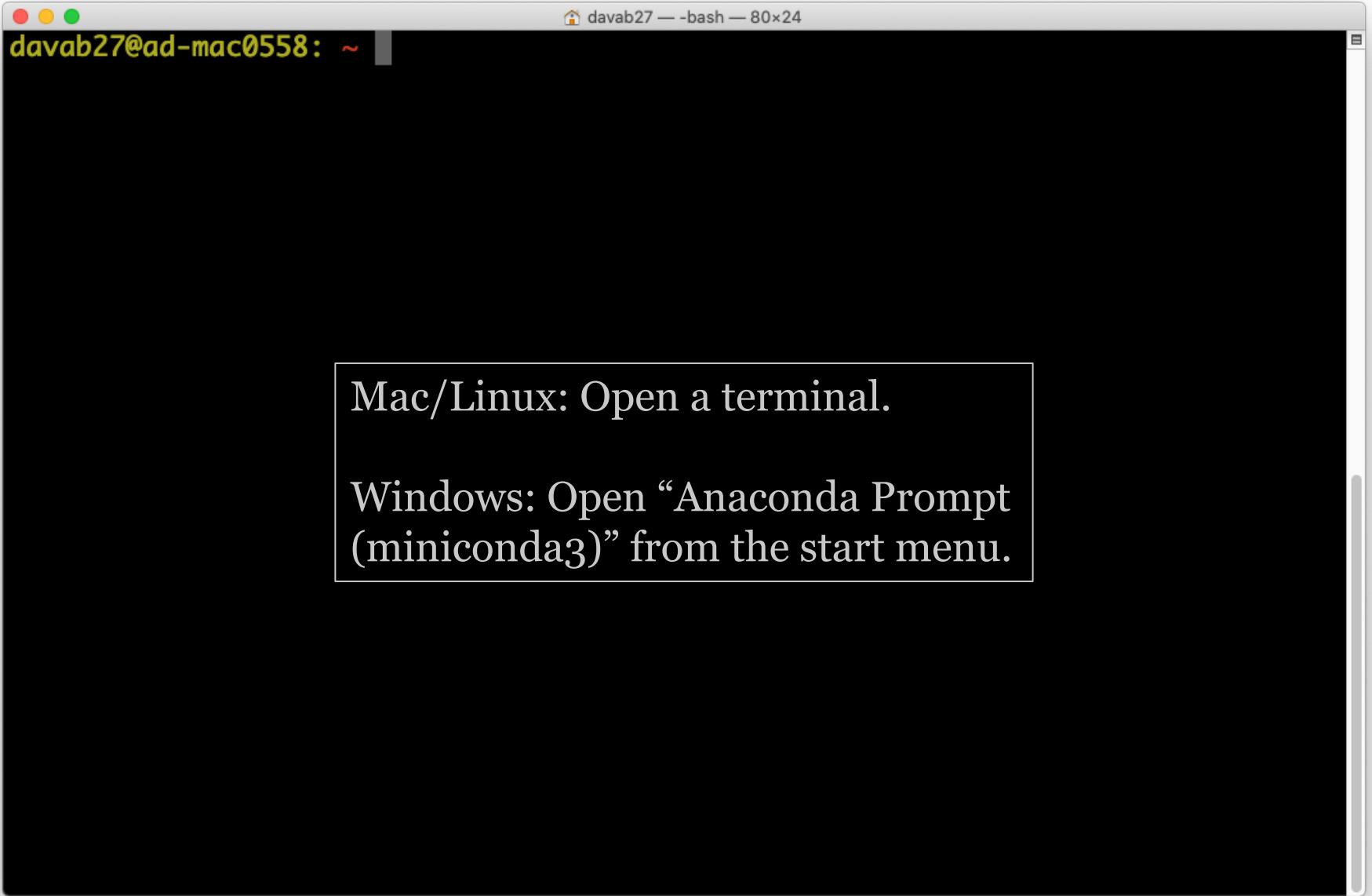
Name	Modified	Modified By	Version	Description
environment_files	21 minutes ago	David Abramian	1.0	
Lab_2	4 days ago	David Abramian	1.0	
Assignments.zip	4 days ago	David Abramian	1.0	
ComputerCompendium.pdf	4 days ago	David Abramian	1.0	

If you don't have the assignment files already, download them from Lisam and extract them somewhere.

Return to classic SharePoint

Launching the labs

Do this every time you want to work on the lab



Mac/Linux: Open a terminal.

Windows: Open “Anaconda Prompt
(miniconda3)” from the start menu.

davab27 — bash — 80x24

```
[davab27@ad-mac0558: ~ conda activate tbmi26  
(tbmi26) davab27@ad-mac0558: ~ ]
```

Activate the conda environment.

Assignments — bash — 80x24

```
[davab27@ad-mac0558: ~ conda activate tbmi26
(tbmi26) davab27@ad-mac0558: ~ cd Downloads/Assignments
(tbmi26) davab27@ad-mac0558: Assignments ]
```

Navigate to the folder
containing the lab files.

```
(tbmi26) davab27@ad-mac0558: Assignments jupyter lab
[I 11:59:55.695 LabApp] JupyterLab extension loaded from /Users/davab27/opt/miniconda3/envs/tbmi26/lib/python3.7/site-packages/jupyterlab
[I 11:59:55.695 LabApp] JupyterLab application directory is /Users/davab27/opt/miniconda3/envs/tbmi26/share/jupyter/lab
[I 11:59:55.700 LabApp] Serving notebooks from local directory: /Users/davab27/Downloads/Assignments
[I 11:59:55.700 LabApp] Jupyter Notebook 6.4.6 is running at:
[I 11:59:55.700 LabApp] http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
[I 11:59:55.700 LabApp] or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
[I 11:59:55.700 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:59:55.734 LabApp]
```

To access the notebook, open this file in a browser:

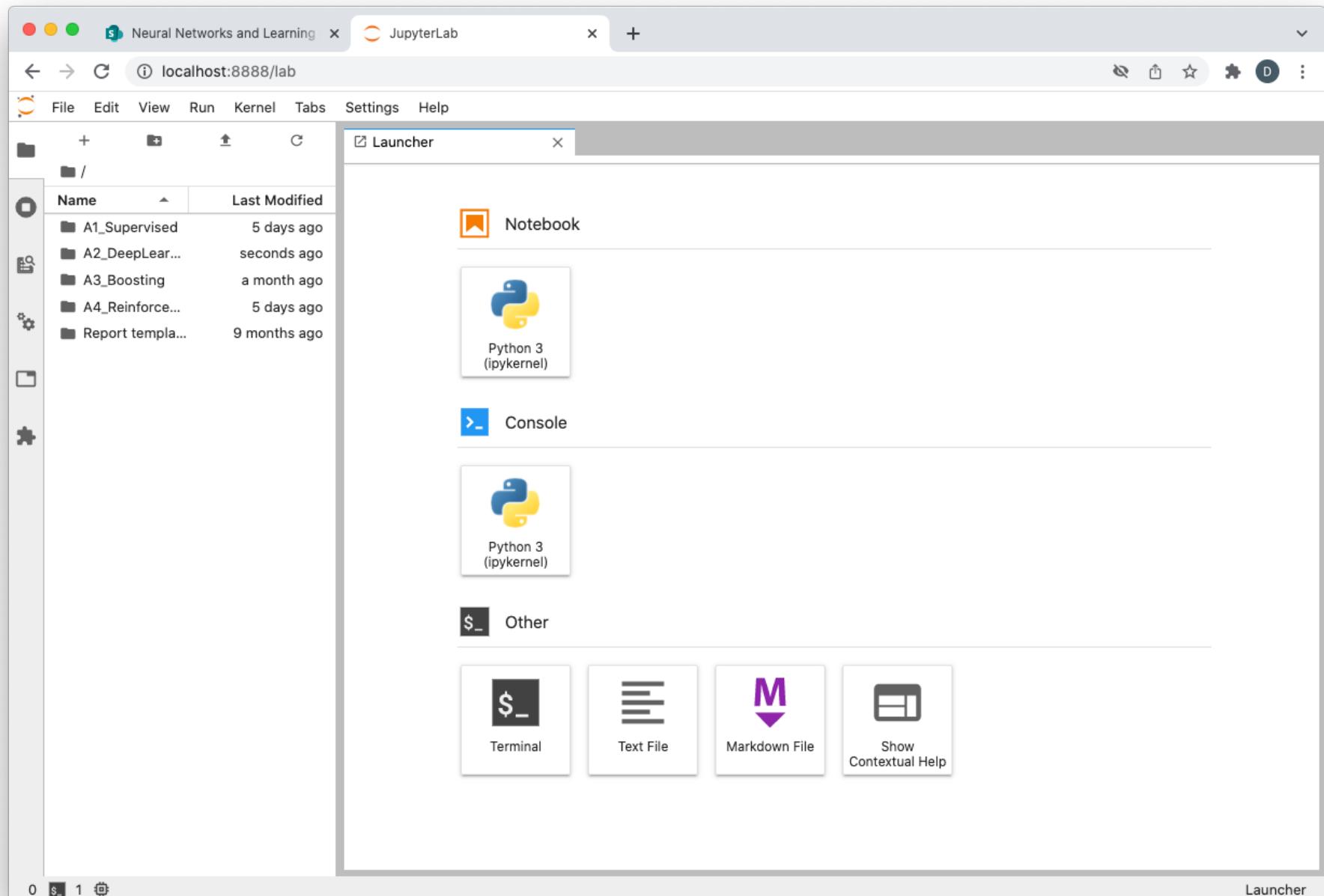
file:///Users/davab27/Library/Jupyter/runtime/nbserver-17472-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663

or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663

Start Jupyter with the given command.



Jupyter will launch in a browser window automatically.

```
[I 12:00:06.814 LabApp] Build is up to date
[I 12:00:07.735 LabApp] Kernel started: 48f5dfed-97b8-4af3-ab55-6070c7fbubb5, na
me: python3
[W 12:01:52.272 LabApp] Notebook A1_Supervised/main_kNN.ipynb is not trusted
[I 12:01:52.653 LabApp] Kernel started: eff4d800-4a83-4209-b7bd-302f07cdf914, na
me: python3
[I 12:01:55.938 LabApp] Starting buffering for eff4d800-4a83-4209-b7bd-302f07cdf
914:dc4a55bb-3203-4c1e-95f1-664999d40786
[I 12:02:01.786 LabApp] Starting buffering for 48f5dfed-97b8-4af3-ab55-6070c7fb
bc5:25db727f-3035-4c78-9060-dcb4dcc1d7cf
^C[I 12:02:13.651 LabApp] interrupted
Serving notebooks from local directory: /Users/davab27/Downloads/Assignments
2 active kernels
Jupyter Notebook 6.4.6 is running at:
http://localhost:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc8663
or http://127.0.0.1:8888/?token=a68afb91c23e79756cc74e014519701ca146a60bb4cc866
3
Shutdown this notebook server (y/[n])? ^C[C 12:02:13.864 LabApp] received signal
2, stopping
[I 12:02:13.865 LabApp] Shutting down 2 kernels
[I 12:02:14.069 LabApp] Kernel shutdown: 48f5dfed-97b8-4af3-ab55-6070c7fbubb5
[I 12:02:14.070 LabApp] Kernel shutdown: eff4d800-4a83-4209-b7bd-302f07cdf914
[I 12:02:14.071 LabApp] Shutting down 0 terminals
(tbmi26) davab27@ad-mac0558: Assignments
```

You can shutdown Jupyter by pressing
[Ctrl + C] twice in the terminal.

Install Jupyter variable inspector

This is strongly recommended!

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

SEARCH

WARNING

INSTALLED

DISCOVER

@jupyterlab/apputils JupyterLab - Application Utilities
Install

@jupyterlab/translation JupyterLab - Translation services
Install

@jupyter-widgets/jupyterlab-manager The JupyterLab extension providing Jupyter widgets.
Install

@jupyterlab/git A JupyterLab extension for version control using git
Install

@jupyterlab/debugger-extension JupyterLab - Debugger Extension
Install

main_kNN.ipynb

Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy, pl
from classifiers import kNN

plt.rcParams['figure.facecolor']='white'

## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



Mode: Command ✓ Ln 1, Col 1 main_kNN.ipynb

Open the Extensions tab.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

inspector

WARNING

INSTALLED

No entries

SEARCH RESULTS

@lckr/jupyterlab_variableinspector
Variable inspector extension for JupyterLab
Install

@almond-sh/jupyterlab_variableinspector
Variable inspector extension for JupyterLab
Install

main_kNN.ipynb

Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy,
from classifiers import KNN

plt.rcParams['figure.facecolor']='white'

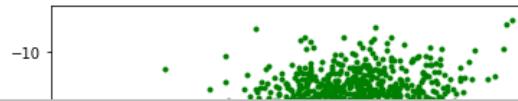
## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



0 1 Python 3 (ipykernel) | Idle Mode: Command ⚡ Ln 1, Col 1 main_kNN.ipynb

Search for and install the variable inspector extension (make sure to get the indicated one). This will take some time.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

inspector

A build is needed to include the latest changes

Ignore Rebuild

WARNING

INSTALLED

@lckr/jupyterlab_variableinspector Variable inspector extension for JupyterLab Uninstall Disable

SEARCH RESULTS

@almond-sh/jupyterlab_variableinspector Variable inspector extension for JupyterLab Install

main_kNN.ipynb

Python 3 (ipykernel)

```
[1]: # This configures the notebook to automatically reload code when it is changed in imported functions
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingSamples, calcConfusionMatrix, calcAccuracy,
from classifiers import KNN

plt.rcParams['figure.facecolor']='white'

## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

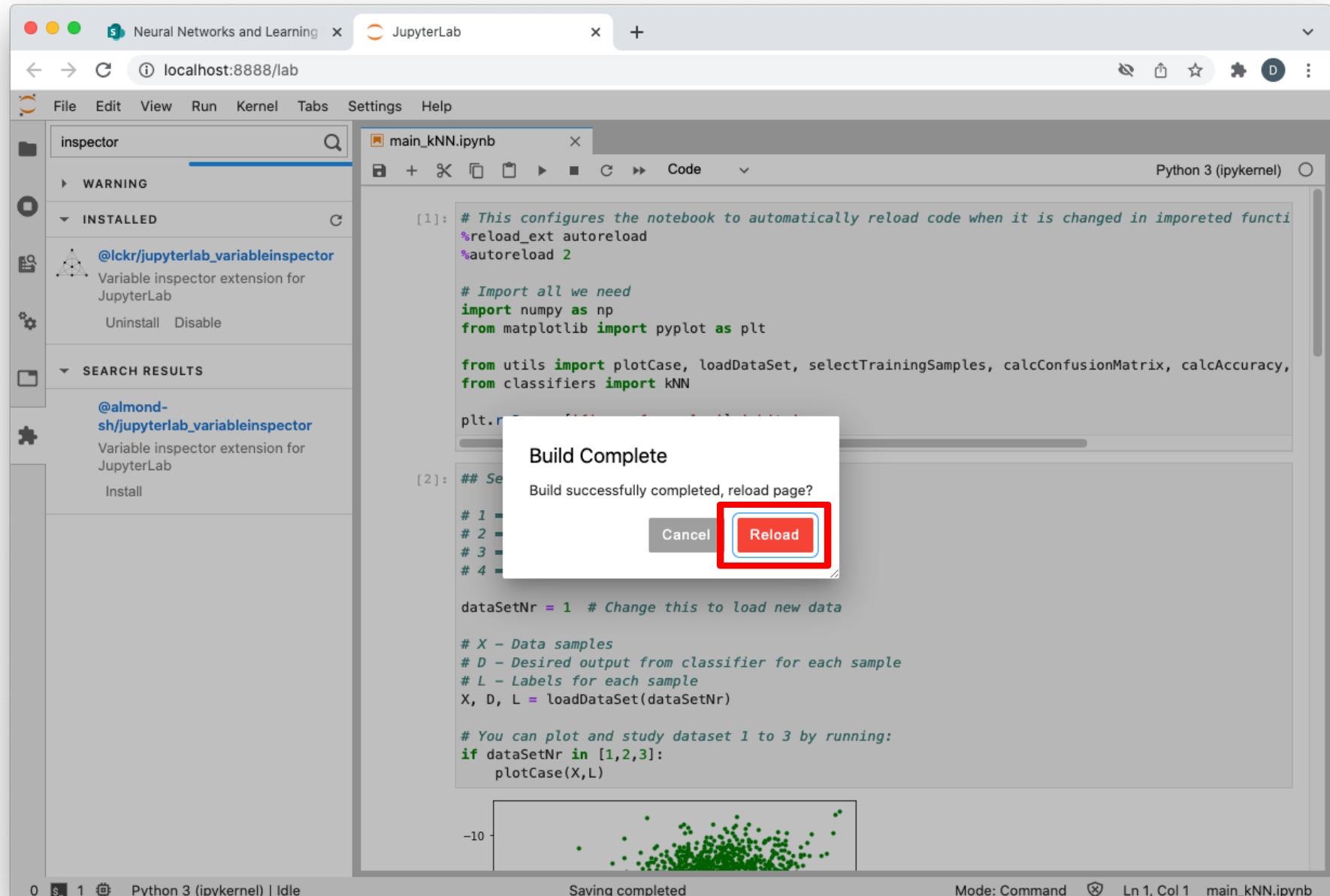
# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



Mode: Command Ln 1, Col 1 main_kNN.ipynb

After installation you will need to rebuild Jupyter. This will take some time.



Reload the page.

Neural Networks and Learning x JupyterLab x localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

SEARCH

WARNING

INSTALLED

- @lckr/jupyterlab_variableinspector Variable inspector extension for JupyterLab
Uninstall Disable

DISCOVER

- @jupyterlab/apputils JupyterLab - Application Utilities
Install
- @jupyterlab/translation JupyterLab - Translation services
Install
- @jupyter-widgets/jupyterlab-manager The JupyterLab extension providing Jupyter widgets.
Install
- @jupyterlab/git A JupyterLab extension for version control using git
Install
- @jupyterlab/debugger-extension

main_kNN.ipynb

```
[1]: # This configures the notebook to automatically reload code on changes
%reload_ext autoreload
%autoreload 2

# Import all we need
import numpy as np
from matplotlib import pyplot as plt

from utils import plotCase, loadDataSet, selectTrainingData, accuracy
from classifiers import kNN

plt.rcParams['figure.facecolor']='white'

[2]: ## Select which data to use:

# 1 = dot cloud 1
# 2 = dot cloud 2
# 3 = dot cloud 3
# 4 = OCR data

dataSetNr = 1 # Change this to load new data

# X - Data samples
# D - Desired output from classifier for each sample
# L - Labels for each sample
X, D, L = loadDataSet(dataSetNr)

# You can plot and study dataset 1 to 3 by running:
if dataSetNr in [1,2,3]:
    plotCase(X,L)
```



Cut Cells X

Copy Cells C

Paste Cells Below V

Delete Cells D, D

Split Cell ⌘ ⌘ -

Merge Selected Cells ⌘ M

Create New View for Output

Clear Outputs

Clear All Outputs

Enable Scrolling for Outputs

Disable Scrolling for Outputs

Undo Cell Operation Z

Redo Cell Operation ⌘ Z

Restart Kernel...

New Console for Notebook

Show Contextual Help ⌘ I

Show Log Console

Open Variable Inspector

Shift+Right Click for Browser Menu

Mode: Command ↵ Ln 1, Col 1 main_kNN.ipynb

You can right click anywhere to open the variable inspector.

The screenshot shows a JupyterLab interface with two tabs open: "Neural Networks and Learning" and "JupyterLab". The "JupyterLab" tab is active, displaying the URL "localhost:8888/lab". On the left, there's a sidebar with a search bar and sections for "WARNING", "INSTALLED" (containing "@lckr/jupyterlab_variableinspector"), and "DISCOVER" (listing extensions like "@jupyterlab/apputils", "@jupyterlab/translation", "@jupyter-widgets/jupyterlab-manager", "@jupyterlab/git", and "@jupyterlab/debugger-extension"). The main area is titled "Variable Inspector" and shows a table for inspecting variables from the file "main_kNN.ipynb".

	NAME	TYPE	SIZE	SHAPE	CONTENT
	D	ndarray	32000	2000 x 2	array([[0.99, -0.99], [0.99, -0.99], [0.99, -0.99], ..., [-0.99, 0.99], [-0.99, 0.99], [-0.99, 0.99]])
	L	ndarray	2000	2000	array([1, 1, 1, ..., 2, 2, 2], dtype=uint8)
	X	ndarray	32000	2000 x 2	array([[-0.90412345, 3.26041993], [0.09863015, 4.35740643], [-0.45150727, 2.41539265], ..., [0.22865578, -7.56594632] ...)
	dataSetNr	int	28		1

At the bottom, it says "Mode: Command" and "Ln 1, Col 1 main_kNN.ipynb".

This will allow you to examine the defined variables and their shapes (the “size” column is not so useful, as it gives the size of the variable in bytes).

The screenshot shows a JupyterLab interface with several panels:

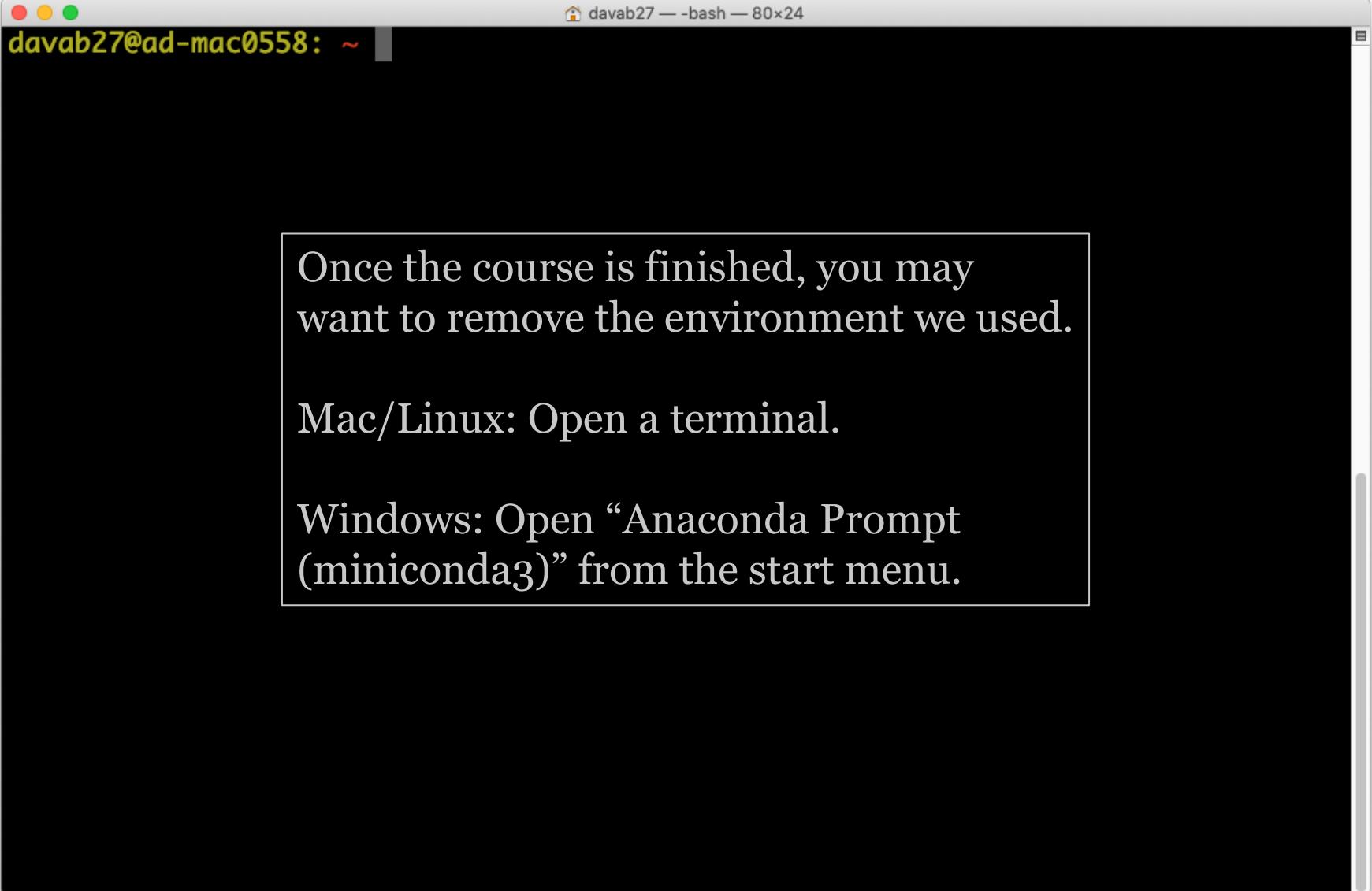
- Left Sidebar:** Contains a search bar and sections for **WARNING**, **INSTALLED** extensions (@lckr/jupyterlab_variableinspector, @jupyterlab/apputils, @jupyterlab/translation, @jupyter-widgets/jupyterlab-manager, @jupyterlab/git), and **DISCOVER** extensions.
- Code Editor:** A tab titled "main_kNN.ipynb" is active, showing Python code for k-Nearest Neighbors classification. It includes imports for numpy, matplotlib.pyplot, and classifiers, and defines variables X, D, L, and dataSetNr.
- Variable Inspector:** A panel titled "Inspecting 'Python 3 (ipykernel)'". It lists variables with their type, size, shape, and content:

NAME	TYPE	SIZE	SHAPE	CONTENT
D	ndarray	32000	2000 x 2	array([[0.99, -0.99], [0.99, -0.99], [0.99, -0.99], ..., [-0.99, 0.99], [-0.99, 0.99], [-0.99, 0.99]])
L	ndarray	2000	2000	array([1, 1, 1, ..., 2, 2, 2], dtype=uint8)
X	ndarray	32000	2000 x 2	array([-0.90412345, 3.26041993], [0.09863015, 4.35740643], [-0.45150727, 2.41539265], ..., [0.22865578, -7.56594632] ...)
dataSetNr	int	28		1
- Output Panel:** Shows a scatter plot of data points.
- Bottom Status Bar:** Displays "Saving completed", "Mode: Command", "Ln 6, Col 9", and the file name "main_kNN.ipynb".

You can also view it side-by-side with the code by dragging the tab to the right.

Deleting the environment

Do this if you want once the course is finished

A screenshot of a Mac OS X desktop environment. In the top-left corner, there are three colored window control buttons (red, yellow, green). To the right of them is the window title bar, which includes a small house icon, the text "davab27 — -bash — 80x24", and a close button. Below the title bar is a dark terminal window. Inside the terminal, the text "davab27@ad-mac0558: ~" is displayed in white, followed by a black command-line prompt. A large, semi-transparent rectangular box is overlaid on the terminal window, containing the following text:

Once the course is finished, you may want to remove the environment we used.

Mac/Linux: Open a terminal.

Windows: Open “Anaconda Prompt (miniconda3)” from the start menu.

```
 davab27@ad-mac0558: ~ conda env remove -n tbmi26
Remove all packages in environment /Users/davab27/opt/miniconda3/envs/tbmi26:
davab27@ad-mac0558: ~
```

Remove the conda environment. This will not delete the downloaded packages.

```
[davab27@ad-mac0558: ~ conda env remove -n tbmi26
Remove all packages in environment /Users/davab27/opt/miniconda3/envs/tbmi26:
[davab27@ad-mac0558: ~ conda clean --all
Cache location: /Users/davab27/opt/miniconda3/pkgs
Will remove the following tarballs:
/Users/davab27/opt/miniconda3/pkgs
-----
parso-0.8.3-pyhd3eb1b0_0.conda          70 KB
ipykernel-6.4.1-py37hecd8cb5_1.conda    191 KB
decorator-5.1.0-pyhd3eb1b0_0.conda       14 KB
argcomplete-1.12.3-pyhd3eb1b0_0.conda    35 KB
jupyter_core-4.6.3-py37_0.conda          70 KB
importlib-metadata-4.8.2-py37hecd8cb5_0.conda 39 KB
wcwidth-0.2.5-pyhd3eb1b0_0.conda        26 KB
pyrsistent-0.18.0-py37hca72f7f_0.conda   93 KB
scipy-1.7.3-py37h8c7af03_0.conda        15.5 MB
matplotlib-3.5.0-py37hecd8cb5_0.conda    28 KB
pip-21.2.2-py37hecd8cb5_0.conda         1.8 MB
pyparsing-3.0.4-pyhd3eb1b0_0.conda       81 KB
giflib-5.2.1-haf1e3a3_0.conda            70 KB
blas-1.0-mkl.conda                      6 KB
```

Delete the conda packages used in the environment,
freeing up some hard drive space.