

MLPy Workshop 0

January 12, 2026

1 Machine Learning in Python - Workshop 0

Darren Lim

As with any other programming language, the best way to learn Python and its machine learning libraries is to play with them, so follow the steps below and ask a tutor if you get stuck.

If you are reading this document then you have most likely been able to obtain and open the assignment in Jupyter on Noteable or locally (or via Colab, Deepnote, etc).

Once you have reached the end of this worksheet you should try generating a PDF of the notebook. For future workshops, you will submit the pdf on Gradescope (linked on Learn).

1.1 Jupyter notebooks

A **Jupyter notebook** is a literate programming tool that allows you to combine text, typeset maths, images, and code (and its output) together in one document. Jupyter notebooks are edited and viewed in a web browser.

A Jupyter notebook consists of several **cells**, which can be of 2 main types: * **Markdown cells**, like this one, contain text formatted using Markdown. They can be edited by double-clicking on them. Markdown syntax is straightforward – you can double-click on the Markdown cells in this notebook to view the source text. Markdown syntax also supports LaTeX typesetting for maths, both inline using \dots , e.g. $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, and in display mode using
$$\dots$$
, e.g.

$$\frac{\partial f}{\partial y} = 2e^{-x} \cos(y).$$

- **Code cells**, like the one below, in which we can type and run Python code interactively. They are indicated by **In []**: on the left hand side. Note that cells will be executed in the order you run them (as indicated by the number in the square brackets on the left, to ensure the reproducibility of your document is always a good idea to clear all cells and rerun the *entire* document).
-

1.1.1 Exercise 1

Run the code cell below, you should see the text `This is a code cell!` printed afterwards.

```
[1]: print('This is a code cell!')
```

This is a code cell!

1.1.2 Exercise 2

Below is a markdown cell, try adding some text to it and try using some markdown formatting syntax.

Hello, World!

1.2 Checking the Python Environment

First we want to make sure that we are using a current version of Python - run the cell below and check to make sure that the version begins with a 3. If you are using Noteable this should definitely be the case. If you are running a local install of Python and Jupyter then this is very important to check before this week's workshop.

```
[2]: from platform import python_version
      print(python_version())
```

3.12.3

Now we can check that you have some of the core libraries necessary for this course installed with up-to-date versions. This is done by checking your installed versions versus those listed in `requirements.txt`.

If the cell above runs without any errors or exceptions than all the needed packages are installed. If there is a error, check the very last line to see what if any package is missing. Any missing or outdated packages should be installed / updated and you can then retry running the check above. See example below to install nbconvert (which will be using the render the notebook).

If you have any difficulties with any of the above checks please post to Piazza with the specific issues / errors you are seeing.

```
[3]: # One of the packages that we will need
      #!pip install nbconvert

      # You can also install specific versions:
      #!pip install nbconvert==7.16.6
      # Or upgrade via:
      #!pip install --upgrade nbconvert
```

```
[4]: # You can check the current version of package:
      !pip show nbconvert
```

Name: nbconvert

Version: 7.16.6

Summary: Converting Jupyter Notebooks (.ipynb files) to other formats. Output

formats include asciidoc, html, latex, markdown, pdf, py, rst, script.
nbconvert can be used both as a Python library (``import nbconvert``) or as a
command line tool (invoked as ``jupyter nbconvert ...``).

Home-page: <https://jupyter.org>

Author:

Author-email: Jupyter Development Team <jupyter@googlegroups.com>

License: BSD 3-Clause License

- Copyright (c) 2001-2015, IPython Development Team
- Copyright (c) 2015-, Jupyter Development Team

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Location: /home/darrenljs/MachineLearninginPython_UoE/.venv/lib/python3.12/site-
packages

Requires: beautifulsoup4, bleach, defusedxml, jinja2, jupyter-core, jupyterlab-
pygments, markupsafe, mistune, nbclient, nbformat, packaging, pandocfilters,
pygments, traitlets

Required-by: jupyter, jupyter_server

1.3 Rendering to PDF

After completing each worksheet, you will be expected to render the document to PDF and turn the resulting in using Gradescope. We will be using the nbconvert tool to render the notebook. This is accomplished by running the cell below, if everything works then `mlp-week00.pdf` should appear in your workshop folder. You should always try opening this file and reviewing the content to make sure everything looks correct (content, format, etc.).

If you are having errors in converting to pdf, then a quick solution is to export to html and then convert to pdf in your browser.

Before generating the PDF, please **change ‘Student 1’ and ‘Student 2’ at the top of the notebook to include your name(s).**

```
[5]: !jupyter nbconvert --to pdf mlp_week00.ipynb
```

```
[NbConvertApp] Converting notebook mlp_week00.ipynb to pdf
```

```
[NbConvertApp] Writing 35878 bytes to notebook.tex
```

```
[NbConvertApp] Building PDF
```

```
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
```

```
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
```

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
```

```
[NbConvertApp] PDF successfully created
```

```
[NbConvertApp] Writing 52640 bytes to mlp_week00.pdf
```

Finally, note that this rendering process will contain all of the current cells and their output values - to ensure the reproducibility of your document it is strongly encouraged that you clear all cell output in your notebook and rerun everything in order.

For Noteable and local Jupyter:

- Select the Kernel menu and click **Restart & Clear Output**
- Select the Cell menu and click **Run All**

If your rendered PDF is missing some or all cell output values, make sure to save your notebook and then rerun the cell above.