mlp-week00

January 12, 2021

1 Machine Learning in Python - Workshop 0

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 the assignment via Deepnote or opened in Jupyter on Noteable or locally.

Once you have reached the end of this worksheet you should try generating a PDF of the notebook and submit it on Gradescope (linked on Learn) using the Workshop 0 assignment.

1.1 1. Jupyter notebooks

A **Jupyter notebook** is a literate programming tool thats 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 `$...$`, e.g. $f : \mathbb{R}^2 \to \mathbb{R}$, and in display mode using `$$...$$`, 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.

$A^2 = b + c$

1.2 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 Deepnote or 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())
```

```
3.7.6
```

Now we can check that you have all 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`.

```
[5]: import pkg_resources
     ok = pkg_resources.require(open('requirements.txt', mode='r'))
     print(ok)
```

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.

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

1.3 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 encounter an error with this step, please post to Piazza with details on the environment you are using and the error messages that occured.

```
[ ]: !jupyter nbconvert --to pdf mlp-week00.ipynb
```

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 Deepnote you can do the following:

- Open the command palette (`Cmd + p` on Macs and `Ctrl + p` on Windows & Linux) and select `Clear all outputs`
- Open the command palette again and select `Run all cells`

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.