

Machine Learning for researchers (300001030)

Jupyter notebooks

Javier Vales Alonso

January 2020

1. Notebooks: what are they?

As stated in the syllabus, the course's goal is understanding the main ideas underlying in machine learning (ML), and studying how the algorithms (the *maths*) of these methods work. Besides, we also provide coding examples in *notebooks*, illustrating how these methods can be used in actual ML projects.

A *notebook* is a document which contains both *code*, *text* and *plots*. It is an excellent way to provide detailed explanations about the algorithm operation and to execute and modify this demo code on the fly. Also, notebooks support mathematical notation and it is very easy to write formulas and maths. Thus, they are ideal in scientific and academic contexts.

The code inside the notebooks can be written in different programming languages, but the most standard is to use Python (specifically, Python 3 is used in the notebooks provided). Let us remark that in this course it is not necessary to develop code, though students willing to do so for the project are welcome, and feedback will be provided on that code.

2. How can I run them?

To run the notebooks it is first necessary to have a suitable Python installation that has support for Jupyter notebooks. This process is well documented in many online tutorials, for example, in [this link](#). Of the different methods proposed, it is recommended to use the *anaconda* manager to install python and library packages. For that, follow the links and the instructions provided in the corresponding section.

Once *anaconda* has been installed it is recommended to create a new *conda* environment with python 3.6 (for example, named *mlr* with command `conda create --name mlr python=3.6`) and then install the following recommended packages:

- `ipython`
- `jupyter`
- `numpy`
- `matplotlib`
- `pandas`
- `scikit-learn`

- `tensorflow`
- `keras`. You can install all the packages above by first activating the environment with command `conda activate mlr` and then issuing the command:

```
conda install jupyter numpy matplotlib pandas scikit-learn tensorflow keras
```
- `jupyter_contrib_nbextensions`. This is a recommended package that allows to use extensions in the Jupyter notebook (e.g., autosaving, translating, etc.). It has to be installed from a different channel with the command:

```
conda install -c conda-forge jupyter_contrib_nbextensions
```

Then, in this same environment execute the *Jupyter* server (see section “How to Use Jupyter Notebooks” in the tutorial). You can download the `toy.ipynb` file from AV and load it from the browser page opened by Jupyter. This toy notebook shows a couple of *cells* (document sections in Jupyter’s terminology). The cells contain some text with maths, and the also pieces of sample code. Try to execute a code cell by selecting it and pressing CTRL+Intro (Linux, Windows) or CMD+Intro (MAC), or just by tapping on the **Run** button. If you have used Mathematica, you will note the similarity.

Last, but not least, when code is executed and you see the output you have to have in mind that this code is *not* actually executed in the browser, but by the Jupyter server which is running in the terminal. This way, the Jupyter process can be run (and often happens) on a server or on cloud and the notebooks accessed from terminals at the office or home.

3. Not working?

Don’t panic! If after reading the tutorials you are unable to view and run notebooks contact me and we will try to fix it asap!