Prerequisites

Course materials on GitHub

The course materials are available on Canvas, but the sources are also on GitHub, so that you always have the latest updates. To download them, you first need to install git (if you haven't already).

You can 'clone' the course as follows from the command line (you can also use a GUI)

```
git clone https://github.com/ML-course/master.git
```

To download updates, run `git pull`

For more details on using git, see the GitHub 10-minute tutorial and Git for Ages 4 and up.

Alternatively, you can download the course as a .zip file. Click 'Clone or download'. Download individual files with right-click -> Save Link As...

Python

You first need to set up a Python environment (if you do not have done so already). The easiest way to do this is by installing Anaconda. We will be using Python 3, so be sure to install the right version. Always install a 64-bit installer (if your machine supports it), and we recommend using Python 3.8 or later.

If you are completely new to Python, we recommend reading the Python Data Science Handbook or taking an introductory online course, such as the Definite Guide to Python, the Whirlwind Tour of Python, or this Python Course. If you like a step-by-step approach, try the DataCamp Intro to Python for Data Science.

To practice your skills, try the Hackerrank challenges.

Virtual environments

If you already have a custom Python environment set up, possibly using a different Python version, we highly recommend to set up a virtual environment for this course so that it does not affect your existing environment. This is not strictly needed if you use a fresh Anaconda install.

Install virtualenv with `pip install virtualenv`, then go (cd) to the folder for this course. Create a new environment, e.g. 'mlcourse' with `virtualenv mlcourse`.

Activate the environment with `source mlcourse/bin/activate` or `mlcourse\Scripts\activate` on Windows. To deactivate the virtual environment, type `deactivate`.

Required packages

Next, you'll need to install several packages that we'll be using extensively. You'll need to run these commands on the command line.

Basic packages

If you are using Anaconda, you already have the basic packages. You can skip to the next step. If you want to install these yourself using pip, do the following

```
pip install --upgrade pip
pip install -U numpy scipy matplotlib pandas
```

Additional packages (install with pip)

Pip is the Python Package index, which helps you to install the latest versions of the additional extra libraries that we'll need:

```
pip install --upgrade pip
pip install -U scikit-learn graphviz mlxtend category_encoders
fancyimpute imblearn GPy pods
```

The -U option updates all packages so that you'll have the latest versions. Note: In any case, make sure to use scikit-learn 0.24 or later, and pandas 1.0 or later.

For a few plots, you also need to install the graphviz C-library:

- OS X: use homebrew: `brew install graphviz`
- Ubuntu/debian: use apt-get: `apt-get install graphviz`.
- Installing graphviz on Windows can be tricky and using conda / anaconda is recommended.

Installing TensorFlow

To install *TensorFlow 2*, follow these instructions for your OS (Windows, Mac, Ubuntu). While installation with `conda` is possible, they recommend to install with `pip`, even with an Anaconda setup. We recommend using TensorFlow 2.2 or later.

Installing OpenML

OpenML is used to easily import datasets and share models and experiments.

```
pip install -U openml
```

For Windows, you need to have a C++ Compiler installed. If the above install fails, you may need to install this first. Download and install Visual C Build Tools

You'll also need an OpenML account to upload data. If you don't have one, go ahead and create one.

Installing Jupyter

As our coding environment, we'll be using Jupyter notebooks. They interleave documentation (in markdown) with executable Python code, and they run in your browser. That means that you can easily edit and re-run all the code in this course.

If you use Anaconda, Jupyter is already installed. If you use pip, you can install it with

```
pip install -U jupyterlab ipywidgets
```

If you are new to notebooks, take this quick tutorial, or this more detailed one. Optionally, for a more in-depth coverage, try the DataCamp tutorial.

Running the course notebooks

Run jupyter lab from the folder where you have downloaded (cloned) the course materials.

```
jupyter lab
```

A browser window should open with all course materials. Open one of the chapters and check if you can execute all code by clicking Cell > Run all. You can shut down the notebook by typing CTRL-C in your terminal.

Alternative environments for running the notebooks

GOOGLE COLAB

Google Colab allows you to run a notebook on Google Drive (with limited GPU support): https://colab.research.google.com/notebooks/gpu.ipynb A more detailed tutorial can be found here (you won't need PyTorch for this course, but we do recommend learning it): https://towardsdatascience.com/fast-ai-lesson-1-on-google-colab-free-gpu-d2af89f53604

There are limitations (obviously): right now GPU usage is limited to 12h and RAM is shared among multiple users. Using a GPU for a longer period of time may temporarily lock you out.

Note: You need to upload your course notebooks to colab yourself (File > Upload Notebook). You can install additional packages from within notebooks with '!pip install package'. See the introduction video for this course for more details.