1 Getting started

In order to solve the exercises in this book, we recommend the usage of Python. While being a widely used scripting language, Python is particularly popular in the data science and machine learning domain. The main reasons for the success of Python are its simplicity of usage, platform independence, availability of an extensive selection of libraries and a wide ML-community.

In this section, we will enumerate the required packages and give a brief tutorial of how to setup a working environment.

1.1 Required libraries/packages

To solve all exercises in this book, the following packages are required in your preferred environment:

- Python: version 3.x
- Numpy[1]: a fundamental package for scientific computing with Python. It contains:
  - a powerful N-dimensional array object
  - sophisticated (broadcasting) functions
  - tools for integrating C/C++ and Fortran code
  - useful linear algebra, Fourier transform, and random number capabilities

  In general, Numpy allows us to perform efficient operations with matrices in an easy way, similar to Matlab.
- Scikit-learn[2]: a simple and efficient toolbox for data mining and data analysis. It contains efficient implementations of many common machine learning techniques.
- Matplotlib[3]: a Python 2D plotting library. We will use it to draw figures, graphs, etc.
- (optional) Jupyter Notebook[4]: a handy web application that allows to write and execute Python code in your web browser.

[1] https://numpy.org
[2] https://scikit-learn.org
[3] https://matplotlib.org
[4] https://jupyter.org

1.2 Setting up

While any method to create a Python 3.x environment with the above packages suffices for the exercises, hereunder 2 possible solutions are elaborated in more detail.

1.2.1 Colab

Google Colaboratory [5] (or 'Colab' for short) allows the writing and online execution of Python code (Notebooks) in your browser. In contrast to other methods, no software has to be installed and no (virtual) environment needs to be created on your device. Most of the commonly used Python libraries, such as the ones named above, are directly available. Colab's working is based on Jupyter Notebook and as a consequence, such notebooks can be easily imported. It even provides free access, when enabled, to computing resources such as GPUs. The only requirement to create and import notebooks is to have a Google account. A possible drawback of Google Colab is that its usage becomes significantly more difficult when the required library or dataset is not available in the standard installed packages. Colab is only able to handle an external dataset when it is uploaded unto a Google Drive. Also the installation of external libraries is not straightforward and needs to be repeated at each runtime.

Overall Google Colab is a great and quick tool but it is not entirely suited for larger and more complex projects. Nevertheless, it is more than sufficient to solve the exercises in this book. More information can be found on the official Colab website:
https://colab.research.google.com

1.2.2 Anaconda + Jupyter Notebook

While notebooks in Colab are executed on Google servers, it is also possible to run them locally. In order to do so, the required packages should be downloaded and installed on your machine. This can easily become a complex task for large projects as each package has a list of dependent packages which could on their turn conflict with others. Anaconda [6] is able to alleviate this problem and has thus become a popular package manager in the data science domain. Anaconda directly installs all dependent packages such that the query package can work properly, while keeping track of possible conflicting requirements. Anaconda also allows us to create and maintain virtual environments, which is crucial when working on different projects in parallel. Great advantages in comparison with Section 1.2.1 is the fact that there is a larger amount of available packages in Anaconda and it is much easier to include new packages in your working environment.

[5] https://colab.research.google.com
[6] https://www.anaconda.com

Below you can find a step-by-step guide to set up your environment. Step 3 and 4 allow the creation of an environment specific for these exercises. While this is strictly speaking not necessary when no other working directories are created, it shows the procedure for possible future usage (such as the project). When executing step 3 and 4, do not forget to activate your created conda environment each time the console is restarted. This is done by executing step 4.

1. Install Anaconda: Download and install Anaconda from the link below (choose the right version for your OS):
   https://www.anaconda.com/download/

2. Open anaconda command prompt

3. (Optional) Create new anaconda environment named 'ml_labs':

   ```
   conda create --name ml_labs
   ```

4. (Optional) Enter the newly created environment:

   ```
   conda activate ml_labs
   ```

5. Install required packages by sequentially executing:

   ```
   conda install numpy
   conda install scikit-learn
   conda install matplotlib
   conda install notebook
   ```

To check if the installation was correctly executed, you can try to import the installed libraries in a Python script. This can be easily done as follows:

1. Invoke Python in your anaconda environment:

   ```
   python
   ```

2. Import sequentially the installed libraries:

   ```
   import numpy
   import matplotlib
   import sklearn
   ```

If you can import those packages without errors, your environment is ready to solve the exercises in this book. To launch Jupyter Notebook in your browser, the following steps have to be taken:

1. Open anaconda command prompt
2. (Optional) Enter the created environment:

```
conda activate ml_labs
```

3. Launch Jupyter Notebook:

```
jupyter notebook
```