Part_00

February 17, 2022

1 Coding for Data Science

1.1 Module: Python (20 hours)

1.2 *Instructor: Nicolò Cesa-Bianchi*

Course page: http://cesa-bianchi.di.unimi.it/CDS/

1.3 Goals

- Focus on **combining tools** as opposed to **programming**
- How to use coding tools to understand the **geometry of data**
- Basics of data analytics, machine learning, and visualization

1.4 Syllabus

- A **short tour** of Python (no modules, no error handling, no classes)
- Linear algebra with **Numpy**
- Data analysis with **Pandas**
- Data visualization with **Matplotlib**
- Machine learning with **Scikit-learn**

Course web page: http://cesa-bianchi.di.unimi.it/CDS/

1.5 Why Python

- A full-fledged, object-oriented language with a syntax similar to other popular programming languages
- Python's libraries for data science are an **industry standard for data science applications**
- Popular language in Big Data, Deep Learning, Natural Language Processing, Collaborative filtering, and more

1.6 Jupyter Notebook

- A web-based **interactive computational environment**
- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion
- The ability to **execute code from the browser**, with the results of computations attached to the code which generated them

- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc
- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text
- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax
- The Jupyter Notebook has become a popular user interface for cloud computing, and major cloud providers (SageMaker, Colab, Azure) have adopted Jupyter or derivative tools as a frontend interface for cloud users

1.7 Basic Jupyter elements

- **Notebook server** started from the command line, `jupyter notebook`
  - This will open a web browser to the URL of the web application (by default, http://localhost:8888)
- **Dashboard:** for managing notebooks
- **Notebooks:** documents that contain the inputs and outputs of an interactive session, as well as additional text that accompanies the code, but is not meant for execution
  - The official Jupyter Notebook format is called `nbformat` and uses the open standard file format JSON
- **Kernels:** Interfaces between **notebooks** and **programming languages**. When a code cell is executed, code that it contains is sent to the kernel associated with the notebook. The results that are returned from this computation are then displayed in the notebook as the cell's output
  - Besides Python, there are kernels for a growing set of languages (e.g., Julia, Haskell, Ruby, Erlang)

1.8 Installing Python

- The Anaconda Distribution (https://www.anaconda.com/downloads) includes Python (including the data science libraries NumPy, Pandas, Matplotlb, Scikit-learn), the Jupyter Notebook, and many others.
- Install the latest version for your platform (very easy, no admin account needed)

1.9 Linux example

- Download installer (e.g., Anaconda3-2020.11-Linux-x86_64.sh) from the Anaconda website
- Open terminal in the folder where the installer is downloaded
- Make installer executable `chmod u+x Anaconda3-2020.11-Linux-x86_64.sh`
- Run installer `./Anaconda3-2020.11-Linux-x86_64.sh`

[1]:
```
!conda --version
!jupyter --version
!python --version
```

```
conda 4.9.2
jupyter core     : 4.6.3
jupyter-notebook : 6.1.4
qtconsole        : 4.7.7
```

```
ipython          : 7.19.0
ipykernel        : 5.3.4
jupyter client   : 6.1.7
jupyter lab      : 2.2.6
nbconvert        : 6.0.7
ipywidgets       : 7.5.1
nbformat         : 5.0.8
traitlets        : 5.0.5
Python 3.8.5
```

1.10 Documentation and tutorials

- Jupyter: https://jupyter.readthedocs.io/en/latest/
- Python: https://docs.python.org/3/
- Numpy: http://www.numpy.org/
- Pandas: http://pandas.pydata.org/pandas-docs/stable/
- Matplotlib: https://matplotlib.org/

1.11 Some references

- Python for Data Analysis (2nd Edition) https://www.oreilly.com/library/view/python-for-data/9781491957653/

1.12 Exam

- There is no midterm for this module.
- The section of the final exam for the Python module will consist of two exercises.
  - The first exercise asks to determine the output of three short blocks of Python/Numpy/Pandas code (variants of code seen in class).
  - The second exercise asks to briefly explain one of the machine learning topics covered in the course.

1.13 Classes

- Wednesday and Friday 8:45-10:15