HES-SO
Engineering and Architecture
Faculty
MSE

Lecturers: Stefano Carrino & Simon Ruffieux

Multimodal Processing, Recognition and Interaction

17 September 2019

Initiation to NumPy, Pandas & scikit-learn

In this practical work, you will install the Anaconda framework and the *PyCharm IDE* and then you will go through a few tutorials about fundamental packages for scientific computing with Python: *NumPy* and *Pandas*. Finally, you will use *scikit-learn* to implement a basic ML solution.

You do not have to hand back this practical work. However, the installation and hands-on will prepare you for the practical work that takes place next weeks.

Note:

1. *We tested all the operations on Windows, although it should also work on Mac and Linux OS, we have not tested it!*
2. *For the installation, we are using the Anaconda framework and PyCharm, you are free to adopt any other solutions (e.g. pip for Python, Atom as IDE,* ***Jupyter Lab****, etc.)*
3. *However, Python 3.6+ is required*

Installation & setup

1. Programs installation
   - Download and Install the Anaconda framework (Python 3.7+) on your machine
     - https://www.anaconda.com/distribution/
   - Download and Install PyCharm Community Edition IDE on your machine
     - https://www.jetbrains.com/pycharm/
     - Since 2019 you can install PyCharm with an Anaconda plugin. More info: https://www.jetbrains.com/pycharm/promo/anaconda/
     - If you are interested - You can obtain a license with @hefr or @edu.hefr.ch mail for the professional version (the free version is enough for this course)
       - See https://www.jetbrains.com/student/

2. Configure PyCharm and test installing new packages
   - Now that the IDE and dev framework are installed, you must instruct PyCharm to use anaconda's python interpreter:
     - You can have a look at the following link if you are lost: https://medium.com/@GalarnykMichael/setting-up-pycharm-with-anaconda-plus-installing-packages-windows-mac-db2b158bd8c

- You should be able to install new package with the pip commands in a terminal (note that you can also install new packages from pyCharm directly)
  - `pip3 freeze`: show all installed packages
  - `pip3 install pandas`: install Pandas (& NumPy as bonus dependence)
  - `pip3 install matplotlib`
  - `pip3 install scipy`
  - `pip3 install scikit-learn`: install scikit-learn

Hands-on

This practical section will demand you to work with three technologies:

- NumPy
- Pandas
- scikit-learn

2.1 Numpy

Guided by the examples here (https://docs.scipy.org/doc/numpy/user/quickstart.html), solve the following questions.

a) Create an n-dimensional (2 x 3) array of random [0-1] floats (optionally, use the `reshape` method)

b) Create the following matrix, called *a*:

```
[ 0,  1,  2,  3
  4,  5,  6,  7
  8,  9, 10, 11 ]
```

(optionally use the `reshape` method)

c) Select the second element of the third column, and set his value to zero

d) Print the first row (do not use a loop)

e) Print the second column (do not use a loop)

f) Save in a matrix *b*: the first 2 elements of the first 2 columns of matrix a. Modify the first element of *b* (e.g. set it to 9), then print *a.* What do you notice?

g) Returns the indices of the maximum value i) of the whole the matrix *a*, along ii) the horizontal axe iii) the vertical axe

2.2 Pandas

2.2.1 PANDAS Series

a) What is a Pandas `series`, what are the differences with a NumPy `ndarray` and a Pandas `dataframes`?

b) Show how to create a Pandas `series`,

   a. from a one-dimensional Numpy `ndarray`; use chars as indexes

   b. From a Python dictionary

2.2.2 PANDAS DataFrames

Download the dataset US - 500 records from https://www.briandunning.com/sample-data (or from the Moodle) and load it in a Pandas DataFrame (`data = pd.read_csv(...)`)

2.2.1 Single selections using *iloc* and *DataFrame*

a) Select the first row of the data frame (e.g. row with name *James Butt*) - GOAL: as output we want a `Series` data type (to check: `print(type(data.iloc(...))`)

b) Select the first row of data frame (*James Butt*) BUT this time we want a `DataFrame` data type output.

c) Select the last row of data frame (*Chauncey Motley*)

d) Select the first column of data frame (first_name)

e) Select the second column of data frame (last_name)

f) Select the last column of data frame (web)

2.2.2 Multiple columns and rows selections using the *iloc* indexer

a) Select the first five rows of the dataframe

b) Select the first two columns of data frame with all rows

c) Select the 1st, 4th, 7th, 25th row + 1st 6th 7th columns.

d) Select the first 5 rows and 5th, 6th, 7th columns of data frame (county -> phone1).

2.2.3 Selecting data using *loc*

a) Before starting, set the column "last_name" as index (inplace=True)

b) ... then analyze the results of the following instruction:

```
df.loc[['Rim', 'Perin'], ['first_name', 'address', 'email']]
```

c) Select rows with index values 'Antonio' and 'Veness', with all columns between 'city' and 'email'

d) Select same rows, with just 'first_name', 'address' and 'city' columns

e) Analyze the results of the following instruction:

```
data.loc[data['first_name'] == 'Erick']
```

f) Compare the following instructions, which is the difference?

```
data.loc[data['first_name'] == 'Erick', 'email']
```

vs

```
data.loc[data['first_name'] == 'Erick', ['email']]
```

g) Select rows with first name *Erick* and all columns between 'city' and 'email'

h) Select rows where the *email* column ends with 'hotmail.com', include all columns

i) Select rows with *first_name* equal to some values, all columns (hint: `isin`)

j) Select rows with first name *Erick* AND *aol.com* email addresses

k) Select rows where the company name has 4 words in it. A lambda function (if you are new to Python see http://book.pythontips.com/en/latest/lambdas.html) that yields True/False values can also be used.

2.3 scikit-learn (tutorial)

It is finally time to implement your first ML solution using scikit-learn (http://scikit-learn.org/)

scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

This first tutorial (here) will guide you step by step in the implementation (and evaluation!) of a first machine learning solution. You will implement a classifier called “Random Forest” and you will use it on the well-known “Iris dataset” (this dataset is already included in scikit-learn, so you don’t need to download it).

You can find the full tutorial here:
https://chrisalbon.com/machine_learning/trees_and_forests/random_forest_classifier_example/