had, jvo

Lab 3: Linear regression with Python and Jupyter Notebook

Submit:
- Your notebook (within a zip file) to http://deei-mooshak.ualg.pt/~jvo/ML/submissions/
- Up to October 10, 2024

1. Introduction

In this lab we will learn how to use Python and Jupyter Notebook[1] to implement simple Machine Learning (ML) algorithms. Jupyter Notebook runs locally but the user interface is accessible through a web browser.

2. Installing Python and Jupyter Notebook

We will need to install Python, some packages, and Jupyter Notebook:

1) Install the latest version of Python, suitable for your operating system, from:

https://www.python.org/downloads/

2) Install a virtual environment. This is optional for some operating systems. However, for some Linux distributions, it is required:

https://pypi.org/project/virtualenv/

3) Go to the command line and use pip to install the following Python packages to support numeric and ML development:

```
pip3 install --user numpy
pip3 install --user matplotlib
pip3 install --user plotly
pip3 install --user pandas
pip3 install --user keras
```

Note the --user switch to install in the user home directory. If willing to install system wide, omit this switch.

4) Go to the command line and use pip to install the Jupyter Notebook and Jupyter Lab:

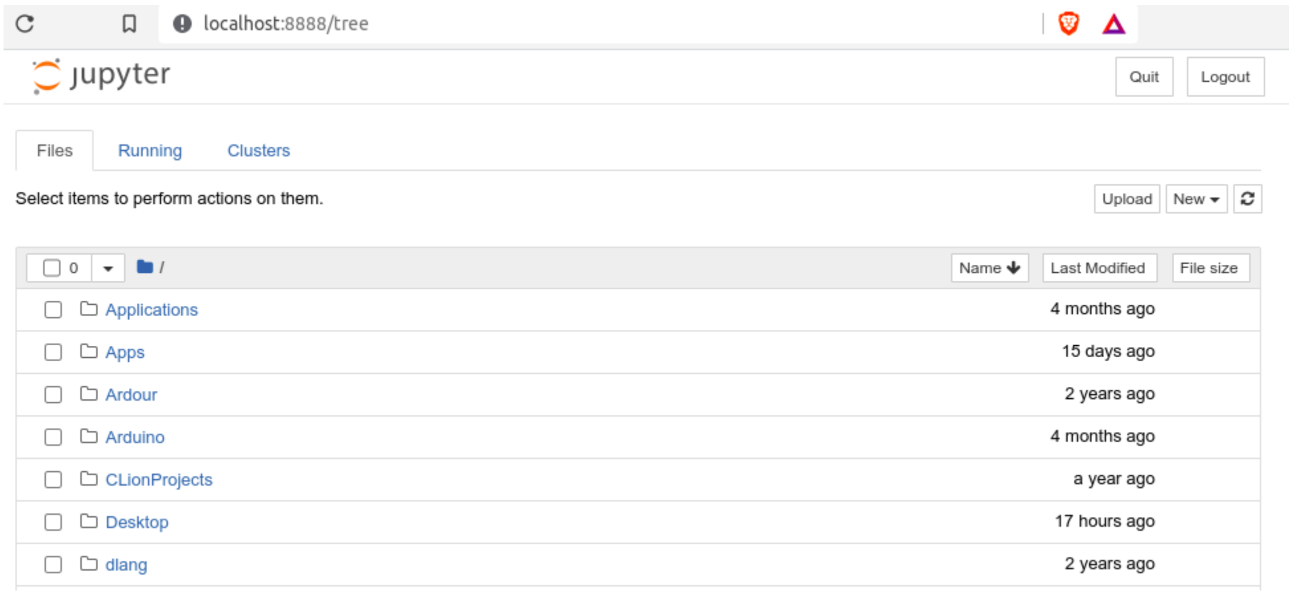
```
pip3 install --user notebook
pip3 install --user jupyterlab
```

3. Using Jupyter Notebook

In the last step we install Jupyter Notebook and Jupyter lab. The latter is a newer version of the notebook. We will learn how to work with the Notebook first since the basics are the same.
Go to the command line a type:

```
jupyter notebook
```

The Jupyter Notebook opens in your browser, showing a view of the current folder. From there you can open a file or create a new one by pressing the new button.



Select items to perform actions on them.

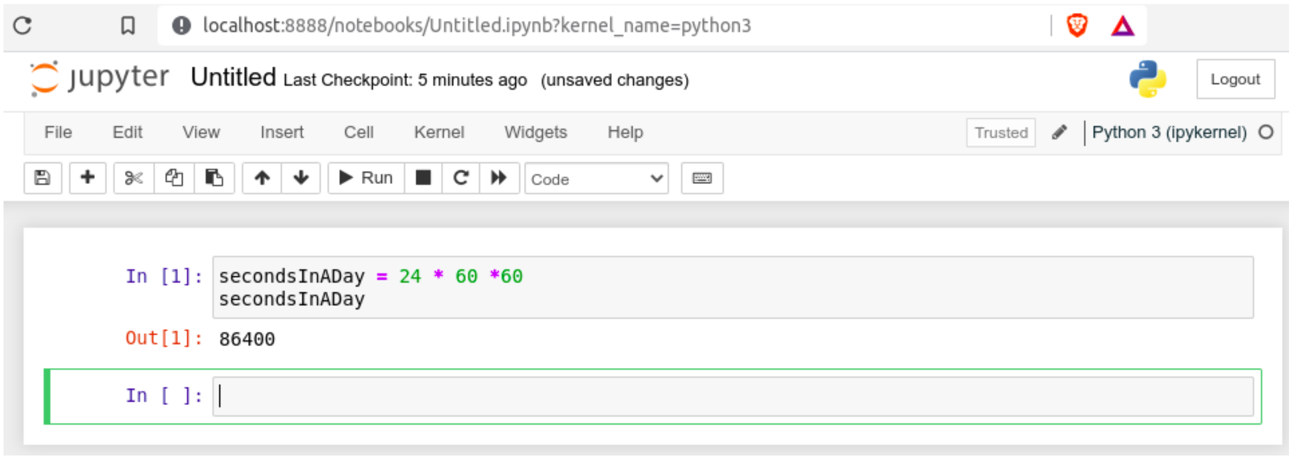
	Name	Last Modified
	Applications	4 months ago
	Apps	15 days ago
	Ardour	2 years ago
	Arduino	4 months ago
	CLionProjects	a year ago
	Desktop	17 hours ago
	dlang	2 years ago

Press the new button to create a new file, and choose Python 3 (ipykernel).

The basic execution unit is a cell. We can type Python code in a cell and then execute this cell only. We can also execute the code in all the notebook cells at once. Move the mouse to the cell, and type the following code:

```
secondsInADay = 24 * 60 *60
secondsInADay
```

Now go to the **run** button and press it. The cell code is executed. The output of the cell is printed below the cell. It prints the value of the variable in the last line in the cell:


localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
jupyter
Untitled Last Checkpoint: 5 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: secondsInADay = 24 * 60 *60
        secondsInADay

Out[1]: 86400

In [ ]:
```

The file name is in the upper left corner, with the default name **Untitled**. By clicking on this name we can change it. We can save, load or create a new file from the file menu.

Now follow the Real Python tutorial below to learn the basic notebook operations:

https://realpython.com/jupyter-notebook-introduction

After completing the tutorial above, follow the one below. This second tutorial from Google shows more basic operations and also some oriented to scientific computation and chart plotting:

https://colab.research.google.com/github/bebi103a/bebi103a.github.io/blob/main/lessons/00/intro_to_jupyterlab.ipynb

Note that this tutorial is implemented as a Jupyter Notebook itself. When the cells are run, they are NOT executed locally but in the Google Colab cloud.

We will not need to use the Google Colab for now, but we can use it to run this tutorial. Alternatively, we can go to File → Download and save the notebook to our local drive to run it locally. We can choose to download in 2 formats:

intro_to_jupyterlab.py	Plain Python runnable with Python interpreter
intro_to_jupyterlab.ipynb	Jupyter Notebook runnable version

4. A concrete regression problem

Distributed with this tutorial is a Jupyter Notebook named **regression-intro.pynb** and the dataset file **demodataset.csv**.

Download the notebook and the dataset file, and inspect and execute all the code cells in the Jupyter Notebook. Then add to the end of the notebook your answers to the following questions:

a) Compute θ_0^* and θ_1^* for the line of best fit:
   i) using scipy.stats.linregress()
   ii) implementing the linear regression model below using **numpy** functions:

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\theta_1 = \frac{SS_{xy}}{SS_{xx}} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

b) Superimpose the line of best fit to the dataset.

c) Predict *y* for *x* = 20.27. What is the error when compared to the *y* value in the dataset? Superimpose on the graphic obtained in b)

d)
   i) Compute $J(\theta_0^*, \theta_1^*)$ for the dataset.
   ii) Express the cost $J(\theta_0, \ldots, \theta_n)$ in vector notation.

iii) Compute $J(\theta_0^*,\theta_1^*)$ in vector notation, for the dataset.
iv) What are the advantages of expressing and computing in vector notation?

e) For θ_0^* and θ_1^*, plot the residuals *vs* the independent variable *x*. Briefly comment on what you observe.

f) For θ_0^* and θ_1^*, proof that the sum of residuals is zero.

g) Prove that the derivative of the cost function in order to θj is:

$$\frac{\partial}{\partial\theta_j}J(\theta_0,\ldots,\theta_n) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \quad , \quad j = 0, 1\ldots n$$

h) Express in vector notation the following gradient descent updating expressions:

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$
$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

i) Apply gradient descent and linear regression to the dataset. Find θ_0^*, θ_1^* and $J(\theta_0^*,\theta_1^*)$.

j) Compare the results obtained in i) to those found in a) and d) i).

k) Plot $J(\theta_0,\theta_1)$ as a function of the number of iterations. Briefly comment on what you observe.

Bibliography

[1] Jupyter notebook. https://jupyter.org/