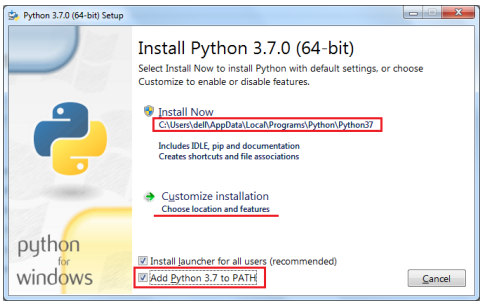
Python Setup
Introduction to Machine Learning
ML_2022: Machine Learning

Location: https://people.sc.fsu.edu/~jburkardt/classes/ml_2022/class00/lecture/setup_lecture.pdf



A screen you may see when setting up Python

Python Setup

You need to install the Python programming language on your computer to do the lab exercises.

- *getting a terminal window*
- *check whether Python is already available on your machine;*
- *install Python on your machine;*
- *check the version of Python;*
- *check that Python is working;*
- *create Python scripts, run them, and save them;*

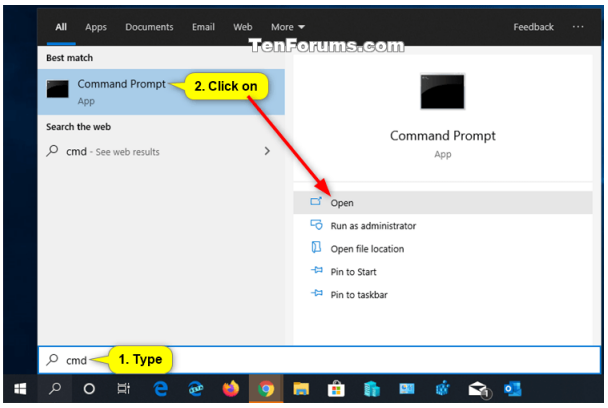
1 Getting a terminal window

My preferred way of using the Python language is to work inside a terminal window. Because icon-based work is so common, users of Windows and MacOS might not even be aware that there is a command window available, and indeed, there are icon-based options for Python as well, which we will consider in a moment. Even so, a command window is necessary at least in order to install the program, so this is where we will start.

Getting a terminal window depends on your operating system

- On Windows, the latest version includes **Windows Terminal**; You can go to the Menu, search for `cmd`, and press Enter or click.
- On MacOS, there is a **Terminal** application, which may hidden in the Applications/Utilities menu. Clicking on the icon will start the program;
- on Linux, you may click your mouse in a blank area of the screen, to see a menu that includes the words **Open Terminal**;

In each case, the command window is an initially empty screen. You can type a command on the current line, and see the results printed out on the next. You can continue working in this way, until you are finished and close the window.



Finding **cmd** *to open a command window on Windows.*

2 Is Python3 already installed?

To see if you have any version of Python installed on your computer, issue the following command in your terminal window:

```
1  which python
```

If the response is a blank line, you have no version installed at all. Otherwise, you will be given the location of a program. On my system, the response is

```
1  /usr/bin/python
```

Now that I know `python` is a legal command, I can ask for the version:

```
1  python –V
```

When I run this, I see

```
1  Python 2.7.15+
```

which means that my computer has at least one version of Python, and that if I type `python` without specifying a version number, I will get a version of Python2. To guarantee that I get version 3, I will want to use the command `python3`. So I can use the command `which python3` to check if it's there, and if so, the command `python3 -V` to get the complete version information.

Exercise:

1. Is the `python` command available on your computer?
2. If so, what version of Python does it use?
3. If `python` gets version 2, then is the `python3` command also available?

3 Preparing for this class

This class will involve many programming examples, exercises, and homework. The programming language will be Python, version 3.0. Along with Python, we will need associated libraries `numpy`, `scipy`, `matplotlib`, `PIL`, `pandas`, `seaborn` as well as the specialized machine learning library `scikit-learn`.

If you are a computer whiz, you may be able to download and install these applications on your own. If you have a Linux or Macintosh system, you may already have some of these installed by default. However,

in any case, it is best to set up your programming environment by downloading Anaconda, and its package manager `conda`.

Go to the site

`https://www.anacondacom/products.individual`

and download a copy of Anaconda for your operating system.

Check that things are properly installed. On Linux or Macintosh, open a terminal window. On Windows, an Anaconda powershell prompt should appear in your Start menu. Click on that to open a terminal window.

Within the terminal window, you can verify that the appropriate packages are now available. To do this, you will run a Python session. If things are going well, then each of the following commands should execute silently. If there is a problem, you will get an error message like *Command not found*, or *No module named 'numpy'*. Assuming there are no such warnings, then the `quit()` command will terminate the Python session, and we have confirmed that we have all the packages we need.

```
python3
>> import numpy
>> import scipy
>> import matplotlib
>> import pandas
>> import PIL
>> import seaborn
>> import sklearn
>> quit()
```

You should try to become familiar with basic programming in Python. If you have no programming experience at all, this may take some time to get used to. If you have experience in programming in C, C++, Fortran, Java, or other procedural languages, then a lot of the features of Python will be familiar. If, better yet, you have worked with an interactive language like MATLAB or R, then Python will seem very similar, and it will just be a question of "translating" how you do things.

Unless you are already familiar with Python, you should refer to the online tutorial available at

`https://docs.python.org/3/tutorial/`

4 Installing Python3 if you don't already have it

The website https://realpython.com/ has numerous tutorials on Python. In particular a useful step-by-step guide, *Installing Python on Windows, macOS, and Linux*, is available at https://realpython.com/installing-python/

Briefly, to download Python3, go to https://www.python.org/downloads/ and look for the latest release, which will be Python 3.7.4 or later.

Note that, when you download Python3, you automatically have access to **idle**, a handy editor for creating, correcting, and running Python scripts. From the command line, you can type **idle**, or to create or modify a specific Python file, **idle myprog.py**.

The **pip** program can be used to install and update Python packages. On Windows and macOS, this will be available automatically as part of your Python3 installation. On a Linux machine, you may need to install this program yourself - note that on Linux the program name is **pip3**. The necessary command on Ubuntu Linux is:

```
sudo apt-get install python3-pip
```

Along with Python3, you will need the following additional packages now:

- **numpy**, a numerical library;
- **scipy**, a scientific programming library;
- **matplotlib**, a graphics library;

Eventually, we will also want:

- **tensorflow**, needed for the keras package;
- **keras**, a neural network package we will need eventually, at `https://keras.io`;
- **cvxopt**, an optimization package at `https://cvxopt.org/`;

5 Try a tutorial

If you are unfamiliar with Python, find a book or online resource, and learn enough so that you think you understand how to create and run a simple Python script.

For example, you can go to `www.python.org` and look at the *Get Started* item. Under the section called "Learning" is a pointer to many online tutorials.

Whatever resource you choose, concentrate on finding the simple "Hello, world!" example, and an example of a `for` loop that runs from 1 to 10. Read enough so that you think you, too, could write and run such an example.