

Machine Learning class

Installation Guide for Machine Learning environment

To execute lab exercises, demos, assignments, and projects, the following tools, software, and libraries need to be installed:

- Python 3.5.x
- Numpy
- Pandas
- Matplotlib
- Scikit-Learn
- PyCharm community edition
- Jupyter Notebook

When we run programs in the environment which has been set up with above libraries, we may have to install yet more libraries if the programs requires any and shows error.

All these work with Windows, Mac, or Linux. With Windows OS, your laptop must have i5 / i7 intel or equivalent AMD processor. Once the installations are complete, update the PATH environment variable. The new installations can then be detected at run time.

CellStrat recommendation for local install :-

- 1) Install with Python PIP / PIP3 commands
- 2) Install with Anaconda

Notes :

- 1) Anaconda vs PIP install :- You can use either method. Vivek (course instructor) uses PIP install. But choice is yours to make. Both work well.
- 2) Check 32-bit or 64-bit compatibility for all software below and install the right version for your computer.
- 3) Vivek runs entire environment on a Windows 64 bit i7 laptop but other laptops or desktops (i5/i7 Intel or equivalent AMD / MAC) should support everything as well.
- 4) We normally run PIP commands in Windows DOS / CMD / command prompt. Just type cmd in Windows search and you will find it. It will give you DOS-like / command line editor in windows. YOU HAVE TO OPEN CMD / COMMAND PROMPT in ADMINISTRATOR MODE IN ORDER TO PIP/PIP3 INSTALL VARIOUS LIBRARIES BELOW. Google how to do this.
- 5) You will benefit from this class the most if you install the environment before the class starts. Once the class starts, it will move forward at a fast pace. It is very important that your environment is setup before the class starts.

REMEMBER : In software you just have to try and see compatible versions. If something gets stuck, uninstall and try to find the right version.

NEW NOTES :-

11th Aug 2018

- 1) One excellent site to install anything is found by googling <https://pypi.org> pandas download or <https://pypi.org> matplotlib download etc.

- 2) Once you go to this site, click on “Download files” on the left.

CRITICAL :

- 1) ALWAYS BACKUP YOUR COMPUTER BEFORE BEGINNING INSTALLATIONS.
- 2) USE STRONG ANTI-VIRUS AND ANTI-SPYWARE TOOLS.

CELLSTRAT AND ITS INSTRUCTORS WILL NOT BE RESPONSIBLE FOR ANY DATA CORRUPTION, DATA LOSS, VIRUS INFECTION, COMPUTER CRASH, MALFUNCTION, SOFTWARE OR HARDWARE ERROR OR ANY OTHER PROBLEMS ON YOUR LOCAL COMPUTER.

Installing Anaconda Distribution 4.2.0 with Python 3.5

Remember : Vivek (your instructor) does not use Anaconda and instead prefers PIP install (provided later in this document). You can use Anaconda if you like.

Step 1: Download Python using the following links. You can choose the appropriate link based on your Operating System.

We recommend downloading Anaconda’s latest Python 3 version.

Windows: <https://repo.continuum.io/archive/Anaconda3-4.2.0-Windows-x86.exe>

Windows 64 bit: https://repo.continuum.io/archive/Anaconda3-4.2.0-Windows-x86_64.exe

Mac: https://repo.continuum.io/archive/Anaconda3-4.2.0-MacOSX-x86_64.pkg

Linux: <https://repo.continuum.io/archive/Anaconda2-4.2.0-Linux-x86.sh>

Linux 64 bit: https://repo.continuum.io/archive/Anaconda3-4.2.0-Linux-x86_64.sh

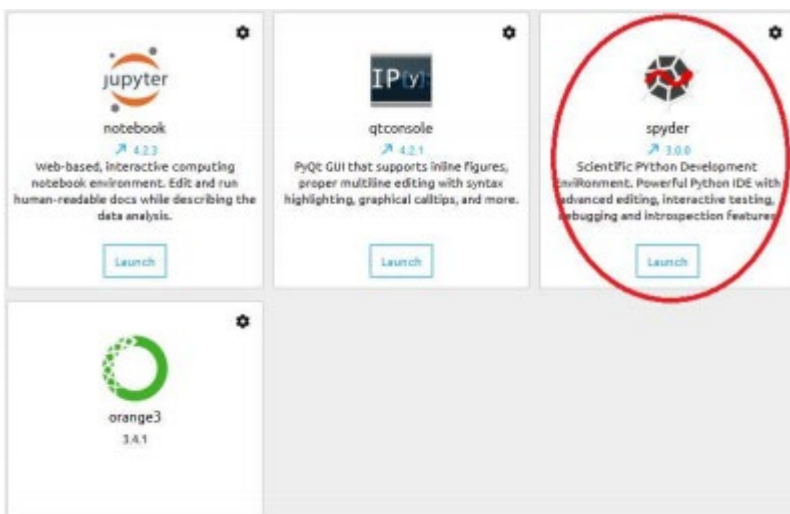
To install Python 3.5.2 click on <https://www.python.org/downloads/release/python-352/>

Step 2: Once Python and Anaconda are installed, click Windows Start button and search “Anaconda Navigator.” If Anaconda is installed properly, you will see it in the programs (marked by arrow). Click Anaconda Navigator.



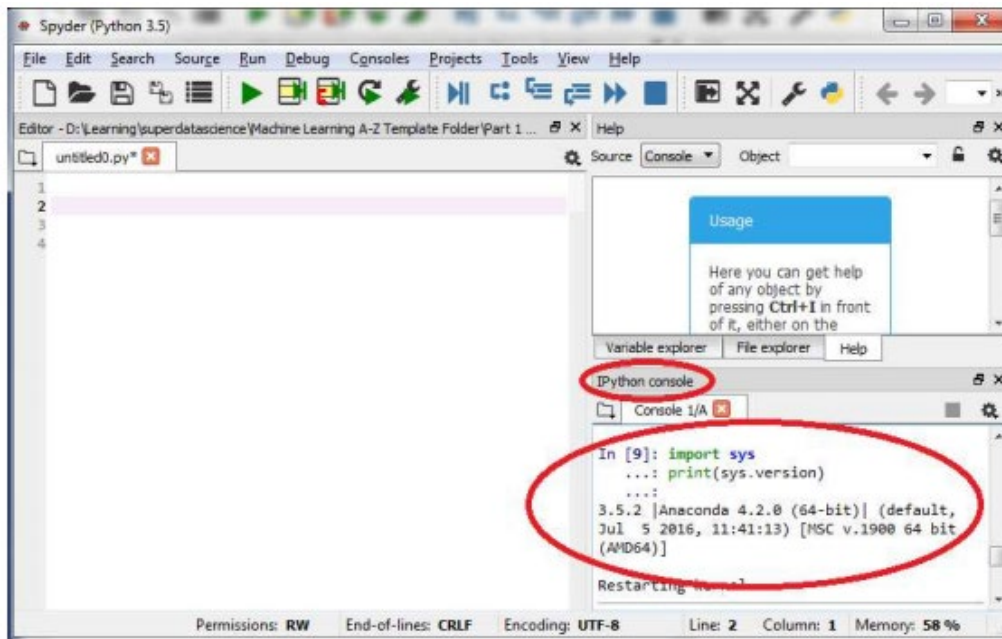
Step 3: Once you click on Anaconda Navigator, it will open a panel with some application icons. Click Launch button.

CellStrat course tutorials will require Jupyter Notebook launch along with PyCharm or Spyder !



Step 4: Once you click on Launch, it will open Spyder IDE (you can use PyCharm IDE community edition, or Spyder or even Atom – that is your choice Vivek uses Jupyter Notebook and PyCharm IDEs community edition). To confirm version of Anaconda and Python, run the following command in IPython Console.

```
import sys
print(sys.version)
```



PIP-based installs

Install PIP

```
python -m pip install --upgrade pip
```

⇒ Install PIP or upgrade if already installed

Installing NumPy

Download and install NumPy from: <http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>

Sometimes numpy+mkl may be required (program will complain). In that case, one can install numpy+mkl from the folder <http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>.

Just run :

```
pip3 install "numpy-1.13.1+mkl-cp35-cp35m-win_amd64.whl"
```

Installing Matplotlib

Download and install Matplotlib from: <https://matplotlib.org/users/installing.html>

Installing Scikit-Learn

Download and install Scikit-Learn from: <http://scikit-learn.org/stable/install.html>

Installing PyCharm community edition

Download and install PyCharm community edition from:

<https://www.jetbrains.com/pycharm/download/#section=windows>

Instead of PyCharm, one can also use Spyder, Atom, Eclipse IDEs etc. CellStrat prefers PyCharm community edition.

Installing Jupyter

Download and install Jupyter from: <http://jupyter.org/install>

Installing other software

- 1) Pandas
- 2) SciPy (<http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy>) v1.0.0 or higher.
 - a. pip install scipy-0.18.1-cp35-cp35m-win_amd64.whl OR pip3 install scipy-0.18.1-cp35-cp35m-win_amd64.whl
 - b. if you're running python3.5 32 bit on Windows choose scipy-0.18.1-cp35-cp35m-win_amd64.whl then it will automatically download
 - c. you can also try this pip3 install --upgrade scipy (to upgrade scipy to v1.0.0 which may be required for numpy+mkl install to work (program will complain then we can upgrade Scipy))
- 3) Graphviz (http://www.graphviz.org/Download_windows.php)
- 4) Pydotplus (just run pip3 install pydotplus)

OPTIONAL :-

ONE MAY ALSO NEED – (have to check when running the programming tutorials if the program asks for one or more of these) :

- 1) Cython (<http://www.lfd.uci.edu/~gohlke/pythonlibs/#cython>)
- 2) Statsmodels (just run `pip install -U statsmodels`)
- 3) OpenCV
- 4) VC++ Build tools ie MSVCP140.DLL (<https://www.microsoft.com/en-us/download/details.aspx?id=48145>)
- 5) Pyprind (just run `pip install pyprind`)
- 6) For Reinforcement Learning with OpenAI Gym, you need to run :- `pip3 install --upgrade gym`
 - a. Atari py can be installed as per <https://stackoverflow.com/questions/42605769/openai-gym-atari-on-windows/46739299>
 - b. You may also need make, Atari py and OpenGym API. Windows make utility can be installed from here : http://stat545.com/automation02_windows.html

To run a python program :

- 1) Open a command prompt (eg Windows Command prompt) and navigate to the area where the python code file is. Type :

```
>>>which python // to check whether it is picking the correct version of Python install
```

```
>>>python "hello world.py"
```

```
>>>python LinearRegression.py
```

- 2) Alternatively you can run the Python program in the IDE that you installed eg PyCharm or Atom. Open the code file in an IDE like PyCharm and run it there. You will have to attach Python interpreter which you installed earlier to the PyCharm program by using the option "Attach Python interpreter"

Working with Jupyter and github cloning :

To clone the code of Geron book "Hands on Machine Learning with Scikit-Learn and TensorFlow" from source code repository github :-

```
$ cd $HOME # or any other development directory you prefer
$ git clone https://github.com/ageron/handson-ml.git
$ cd handson-ml
```

Starting Jupyter :

You can now start Jupyter, simply type:

```
$ jupyter notebook
```

This should open up your browser, and you should see Jupyter's tree view, with the contents of the current directory. If your browser does not open automatically, visit localhost:8888.

Click on `index.ipynb` to get started!

If needed, one can save .ipynb files as .py files from within the notebook to convert iPython Notebooks to .py python files.

Troubleshooting guide :-

- 1) Import error in SKLEARN import DLL error :-

<https://stackoverflow.com/questions/30766274/error-importing-scikit-learn-modules>

- 2) Error in importing module `train_test_split`

you have the wrong version of `scikit-learn`, a similar situation was described [here on GitHub](#). Previously (before v0.18), `train_test_split` was located in the `cross_validation` module:

```
from sklearn.cross_validation import train_test_split
```

However, now it's in the `model_selection` module:

```
from sklearn.model_selection import train_test_split
```

so you'll need the newest version.

To upgrade to at least version 0.18, do:

```
pip install -U scikit-learn
```

(Or `pip3`, depending on your version of Python). If you've installed it in a different way, make sure you use another method to update, for example when using Anaconda.

In MAC (or probably Windows also in some cases), if the line :-

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/5,  
                                                    random_state = 0)
```

gives zero samples split in the Test dataset, then try replacing `test_size` as :-

```
test_size = 0.2
```

In some cases in MAC we found that, `test_size` as fraction 1/5 is not recognized and train test split did not split the dataset at all. When we replaced with 0.2 value, it worked !

<https://stackoverflow.com/questions/40704484/importerror-no-module-named-model-selection>

- (3) In sample codes of Polynomial Regression (`val = 68`), but the last two codes returned error.

It says:

ValueError: Expected 2D array, got scalar array instead: array=68.

Reshape your data either using `array.reshape(-1, 1)` if your data has a single feature or `array.reshape(1, -1)` if it contains a single sample.

Solution :-

It is due to python 3.6. It works in Python 3.5 which I use but in Python 2.6 it expects val to be 2D array.

Try this :-

```
val = [68]
```

```
OR val = [[68]]
```

OR

```
predictionlinear = linregressor.predict (val.reshape(-1,1))  
predictionPoly = linregressor2.predict (polyfeature.fit_transform(val.reshape(-1,1)))
```

OR

```
predictionlinear = linregressor.predict (val.reshape(1,-1))  
predictionPoly = linregressor2.predict (polyfeature.fit_transform(val.reshape(1,-1)))
```