Setup for Python Programming and Machine Learning

We are going to use a few tools:

- Anaconda
- Git
- Visual Studio Code
- Jupyter Notebook

Anaconda

- Go to https://www.anaconda.com/download/
- Download and install Anaconda for Python 3.7
- Settings that I used during installation
    - "Just Me" (default) – the disadvantage is that if you login using a different user in your computer, the other user ID can't use the Anaconda installed, but this options seems to be less troublesome
    - Add to PATH – although this is not recommended but since I use Anaconda as my primary Python installation this options seems to be better for me. I can call Python from command line.
- Cheatsheet: https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
- Common problems:
    - You get error messages when you run Anaconda Prompt, and things doesn't seems to work as expected
      Solution: uninstall all the Python in your machine, install Python 3.7 and then install Anaconda. Somehow this seems to work for SA 47.

Setting up separate environment (optional)

- Anaconda comes with a base environment with some pre-installed Python packages. You can see it if you run "Anaconda Prompt" from Windows Start Menu, there should be a text "(base)" on the command prompt
- Some people need to build multiple projects that requires different Python packages. Some of the differences can be incompatible with each other (e.g. Project A requires version 1 of the library but Project B can only works with version 2). Therefore Python (and Anaconda) provides a way to create multiple Python environment each with its own settings.
- Others may create a new environment just because they don't want to "pollute" the Anaconda base environment, just in case they mess up the packages.
- For this class, you can use the base environment, which has many of the required packages. In the worst case, you'll just have to reinstall Anaconda. Or you can also create a new environment for this SA module.
- If you want to create a new environment, these are the steps:
    - Launch Anaconda Prompt
    - Run the following code. "sa48" is the name of the new environment that we are creating. You can safely replace it with any other name, and we want to use Python

3 in this environment. If you need specific version of Python, you can also specify the minor version such as “3.6” or “3.7”.

```
conda create -n sa48 python=3
```

- Activate your environment. You will have to do this every time you want to use this environment.

```
conda activate sa48
```

Installing Jupyter Notebook

- From Anaconda Prompt (from your new environment or the base environment if you don’t create any new ones), run this command.

```
conda install jupyter
```

- You can test it by running the following command. This command will launch Jupyter Notebook and use the current directory as the current directory.

```
jupyter notebook
```

- We will be given a URL which we can copy and paste into our browser to access the interface.
- Typically before we launch Jupyter Notebook, we will use “cd” command to navigate to the folder where we keep our notebooks and then launch Jupyter Notebook. For example, if we keep our files at D:\sa48\python then we will use the following:

```
D:
cd D:\sa48\python
jupyter notebook
```

Git

- Go to https://git-scm.com/
- Download and install Git for Windows
- Settings that I used:
  - Use Visual Studio Code as the default editor (not the default option)
  - Use the recommended “Git from command line and also from 3rd-party software” (default option). This is the exact text in Git 2.2.0 setup. The actual text may differs in the future releases but this allow you to use Git from Git Bash as well from the normal command prompt.
  - Accept defaults for: Use OpenSSH, OpenSSL, Checkout Windows style
  - Use Windows’ default console window (not the default option)
  - Accept default for other extra options

Visual Studio Code

- Go to https://code.visualstudio.com/
- Download Visual Studio Code for Windows
- Extensions that I use

    - Anaconda Extension Pack
    - Python
- Visual Studio Code is actually not an IDE that is exclusively built for Python. One of the most powerful IDE for Python is PyCharm which has education license. However, we use VSCode so that we can see more of the setup and allow us to use any IDE in the future (even plain text editor)