This is a living document containing a number of technologies one should familiarize oneself with to get cranking with machine learning. This includes tech for distributing and installing open source software, as well as useful open source machine learning libraries and datasets.

**PART 0: PYTHON PREREQUISITES**

In order to successfully navigate this course you will need to be proficient with the Python programming language, on the level of [this Python course from CodeAcademy.](https://www.codecademy.com/learn/python) Make sure you have a working familiarity with [basic data structures in Python](http://www.thomas-cokelaer.info/tutorials/python/data_structures.html) before diving into these libraries.

In this course we will be using Python for all of our machine learning tasks. We will - in particular - be using [Python Jupyter notebooks](https://conferences.oreilly.com/jupyter/jup-ny/public/cfp/537) - a great tool that makes investigating and writing code much easier. [Click here for a video introduction to Jupyter notebooks.](https://www.youtube.com/watch?v=HW29067qVWk)

**PART 1. GET JUST WHAT YOU NEED TO GET STARTED**

What follows is a minimal but complete set of instructions for downloading the required libraries and files for the first Layer of the course. If you can follow the instructions below, you’re all set to go. Descriptions of developer tools, libraries, and datasets follow afterwards. After finishing the installation instructions below you will need to learn about several of these items in Parts 2 and 3 of this document.

------------------------------------------------------------------------------------

**Step 1: Download Anaconda from**

<https://www.continuum.io/downloads>

**Step 2: Clone the Layer 1 github repo**

[https://github.com/jermwatt/](https://github.com/jermwatt/udacity_cap1)

**Step 3: from the terminal**

-cd into the directory

-type jupyter notebook

and the course materials will open in a web browser automatically

------------------------------------------------------------------------------------

Note: in order to install or update additional packages using anaconda, you will need to use the terminal.

**PART 2. GREAT OPEN SOURCE MACHINE LEARNING LIBRARIES AND DATASETS**

1. Libraries - open sourced and reputable computation and machine learning functionality that lets you quickly get started and test ideas.
   1. **Scikit-learn** - [extremely popular library](http://scikit-learn.org/stable/) containing great overall functionality, a jack-of-all trades machine learning library. Contains versions of gradient boosting (ala Xgboost) and neural networks (ala Tensorflow)
   2. **Pandas** - [data loading / manipulation / transformation library](http://pandas.pydata.org/). Very handy. [See here for a very detailed tutorial in Jupyter notebook form.](https://bitbucket.org/hrojas/learn-pandas)
   3. **Numpy** - [numerical linear algebra library](http://www.numpy.org/) - super handy.
   4. **Xgboost** - [one of the best tree-based algorithms](http://xgboost.readthedocs.io/en/latest/) out there, has helped many competitors win Kaggle (machine learning) competitions
   5. **Deep learning libraries** - there is a growing number of these, we will be using an abstraction of Google’s open source deep learning library [Tensorflow](https://www.google.com/webhp?sourceid=chrome-instant&rlz=1C5CHFA_enUS709US713&ion=1&espv=2&ie=UTF-8#q=tensorflow) called [Keras](https://keras.io/) since it is very easy to use.
   6. **dlib** - one of my personal favorites - [an extensively commented C++ machine learning library](http://dlib.net/) with a good number of Python bindings.
2. Datasets - what’s a machine learning algorithm without data?
   1. [Kaggle’s open data repository](https://www.kaggle.com/datasets)
   2. [CrowdFlower](https://www.crowdflower.com/data-for-everyone/)
   3. [UCI machine learning datasets](http://archive.ics.uci.edu/ml/)
   4. [Amazon Web Services datasets](https://aws.amazon.com/public-datasets/)
   5. [And the list goes on….](http://www.kdnuggets.com/datasets/index.html)

**PART 3. DEVELOPER TOOLS a.k.a. Tools that help you find, install, and configure (machine learning) software**

This part of the document contains descriptions and links to some very powerful developer tools for getting started *using* open source machine learning functions and libraries. If you have never used these technologies it is worth your while to take some time to become at least moderately familiar with them.

Why invest time to learn how to use these? Because there is a lot of great open-source machine learning software out there and being able to use it will help you play around, experiment, grow your intuition, and build things faster. In addition, because you will learn how many of these algorithms work in this class you can save yourself development time in using reputable open source libraries, with the confidence that you can ‘pop the hood’ at any time you want and adjust the code to work better for whatever problem you are facing. However without these sorts of tools you may a) never hear about new libraries or b) might not be able to get them up and running on your own machine. So you you need at least a working familiarity with how these technologies work. Learning these sorts of technologies is the price you pay for using open-source, and the price is well worth it.

Note that this list is by no means exhaustive - nor is it fixed as newer / more popular technologies are developed in the years to come some of these will likely be edged out - but for now this list is perfectly suitable for our course since we will be writing a good amount of code ourselves. Going forward you will want to remain as agile as possible when it comes to using machine learning - and that means regularly diving into new and better ways to access and spin up open source software on your own machines.

1. **The got-to-haves** - having working familiarity with these are a **must**
   1. **Basics of the command line user-interface**. Only the basics for now - how to change directories, move / copy files, remove files, list your current directory, change permissions, and list the contents of a directory. Other functionality will be required as the course progresses, but we can cover these commands together as we move along. Note: you do not need to know how to write scripts for this class, just the basic commands for now.
      1. Linux / Unix / OSX / AWS EC2 instance users can [see this resource](http://linuxcommand.org/lc3_learning_the_shell.php) for further info
      2. Windows users can [see this resource](http://www.cs.princeton.edu/courses/archive/spr05/cos126/cmd-prompt.html) for further info
   2. **A github account** - [click here](http://www.github.com) to start your account. This is like a dropbox / box / icloud /etc., but for code. A lot of people / industries host their code on this website, and often it's made public. There are other similar services out there, but for the sake of simplicity you should first learn this one. In particular you need to be able to
      1. Create a repository using the website / desktop interface, upload and download files, etc., [click here for an overview of how to setup](https://www.youtube.com/watch?v=KOyKPlByeAw)
      2. Create a repository using your command line terminal - update a repo, push, pull, etc., [click here for an overview of how to perform these commands](https://www.youtube.com/watch?v=0fKg7e37bQE)
   3. **The Anaconda Python installer.** This is a great Python installation package for one of the main machine learning libraries we will use in the course - scikit learn. You can use one of their graphical installers by [clicking here.](https://www.continuum.io/downloads)
2. **Nice-to-haves -** these are recommended tools that you will not absolutely need for the course, but are very useful for development, and it is recommended that you investigate their use.
   1. [**Docker**](https://docs.docker.com/engine/installation/) - this virtual-environment tool allows you to share complete libraries, fully installed in a complete Linux OS. In short this allows you to share exactly what you are working on with other people, saving them the hassle of having messy installation and configurations. [See here for an overview.](https://www.youtube.com/watch?v=YFl2mCHdv24)  If you want to use this in concert with Jupyter notebooks, follow the install instructions below:

------------------------------------------------------------------------------------

**Step 1:** Install Docker on your machine. You can see install instructions for every OS by [clicking here](https://docs.docker.com/engine/installation/) and [download install scripts for Linux Ubuntu here](https://github.com/jermwatt/various_install_scripts)

**Step 2:** From the terminal of your machine, type the following command

docker run -it --rm -p 8888:8888 jupyter/datascience-notebook

to pull a docker image containing an Anaconda install, Jupyter notebook, and more. [See this github repo for a description of different image installs available](https://github.com/jupyter/docker-stacks).

**Step 3:** Paste the following address into your browser (Chrome and Firefox work best)

<http://localhost:8888/>

And you should then be able to access all contents. You can then pull a github repository into your container and start playing, or pull other Docker images containing

------------------------------------------------------------------------------------

* 1. [**Amazon Web Services (AWS) EC2 instances**](https://aws.amazon.com/) - virtual servers that can make developing and deployment of (machine learning) software much smoother than otherwise. For one, your code is always running in the cloud. This means - for example - that you don’t need to wait around for your home machine to finish executing a complicated algorithm while you do other important work. In addition Amazon Machine Images (AMIs) - like Docker images contain a complete operating system + any installed files are commonly used to quickly share machine learning code. See e.g., here for how [to setup an EC2 instance.](https://www.youtube.com/watch?v=h9-UUJ_rjp0) Once you setup an EC2 instance you can e.g., [install Docker](https://github.com/jermwatt/various_install_scripts.git) and run a Jupyter notebook Docker image (as described above) directly from your instance, or [install Jupyter directly on the instance.](https://gist.github.com/iamatypeofwalrus/5183133)
  2. **OS specific installers** - there are OS-specific package managers for every OS out there. MacOS has [Homebrew](http://brew.sh/), Linux has [apt-get](https://help.ubuntu.com/12.04/serverguide/apt-get.html), etc., Some languages / communities have created their own managers as well e.g., Python’s [pip](https://www.google.com/search?q=pip&rlz=1C5CHFA_enUS709US713&oq=pip&aqs=chrome..69i57j69i60l5.374j0j4&sourceid=chrome&ie=UTF-8) installer.