1/10/2019 Getting Started

**Getting Started with Python**

The purpose of this tutorial is to help you get started installing and using Python. Although there are many programming languages available for implementing data mining algorithms, we choose Python for several reasons. First, it has emerged as one of the most popular programming languages for data science in recent years. Second, it has extensive libraries and ecosystems available to support the collection, preprocessing, mining, and visualization of data, which is beneficial for rapid prototyping of code. Third, it is open source, allowing anyone with access to a computer to download and execute the example code given in this tutorial without the need to install expensive software.

In this tutorial, we first provide instructions on how to install and setup Python and Jupyter notebook on your machine. We then illustrate a few simple commands to help you get started executing Python code from a Web browser via the Jupyter notebook.

**1. Installing and Setting Up Python**

Although there are many Python installations available, one of the easiest way to install Python on your machine is by using a pre-packaged distribution such as Anaconda from Continuum Analytics (http://www.anaconda.com (http://www.anaconda.com)). In this section, we describe the installation procedure for the Anaconda distribution of Python packages.

http://www.cse.msu.edu/~ptan/dmbook/tutorials/tutorial0.html 1/15

1/10/2019 Getting Started

**1.1. Installing Python from Scratch**

First, download the Python base installation package from https://www.anaconda.com/download (https://www.anaconda.com/download). All the examples given in this tutorial were tested on Python version 3.6 installation, then launch the Anaconda command prompt and the following interface will be displayed: Follow the installation instructions as per the below url. Please select the correct version for the OS you have.(eg:- if you use apple laptop use MAC OS). Remember to install python 3.6 version only.

**1.2. Updating and Installing Additional Python Library Packages**

After installing the base Python distribution packages, you should check to see if any of the installed packages are outdated. If so, you should update them. To search for outdated package, type the following on Anaconda command prompt:

C:\Users\Default> conda search --outdated C:\Users\Default> conda update --all

You can also type the conda update command without searching for outdated packages. To check all the packages installed on your machine, type the following command:

C:\Users\Default> conda list

You can also use 'conda list' to check whether a specific package has been installed. For example, to check whether a specific package name is installed on your machine, you can type the following:

C:\Users\Default> conda list package\_name

To install a specific package, type the following:

C:\Users\Default> conda install package\_name

where package\_name is the name of the Python library package you would like to install. Some Python packages may not be part of Anaconda's distribution. In that case, you can still install the package using the 'pip' command:

C:\Users\Default> pip install package\_name

**2. Jupyter Notebook**

You have two options to run Python codes.

First, use Spyder IDE installed with Anaconda. To see how to open spyder please below link.

<https://docs.anaconda.com/anaconda/user-guide/getting-started/>

You can also use a Web-based user-friendly environment called Jupyter notebook to write and execute your Python program. I will conduct the workshop using Jupyter Notebook and Google Collaboratory(I will explain this Collaboratory setup in the workshop)

To launch the notebook, type the following command on the Anaconda prompt/windows command center(cmd)/Mac Terminal:

For example in windows: Open cmd then type below command and press enter

C:\Users\Default > jupyter notebook









