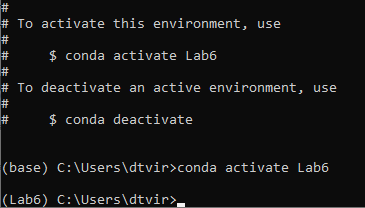
**InLab**

1. **Objectives**

* To explore web scraping in Python
* To demonstrate basic web scraping using the BeautifulSoup library
* To understand data visualization with matplotlib library
* To visualize datasets using different kinds of graphs

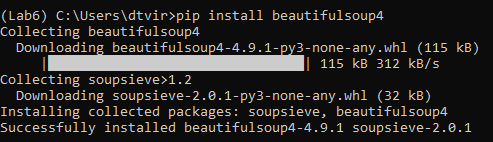
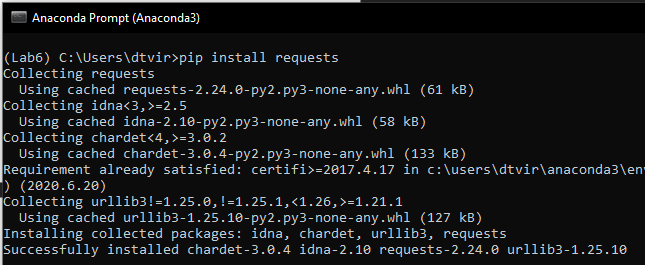
1. **Steps performed with screenshots of tools used**

**Part 1. Web Scraping**

****

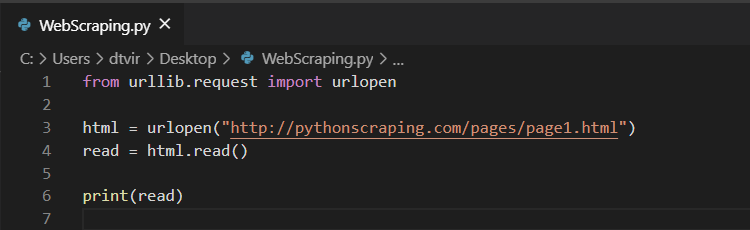
**Figure 1.** Anaconda virtual environment

For this experiment, we used Anaconda Prompt to create a virtual environment named Lab6. We activated the environment using the command *conda activate Lab6* so that we can perform the experiment on the said virtual environment.



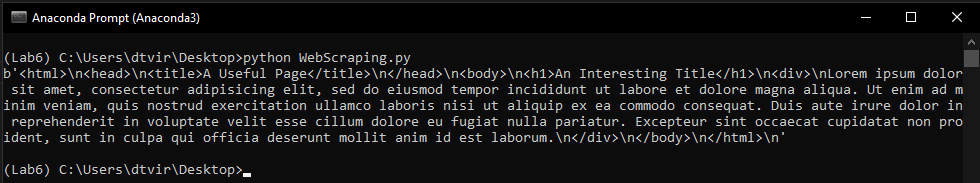
**Figure 2.** Installing required libraries

Before proceeding to the web scraping experiment, we installed the necessary libraries first. These libraries are requests and beautifulsoup. We used the command pip install to install these libraries.



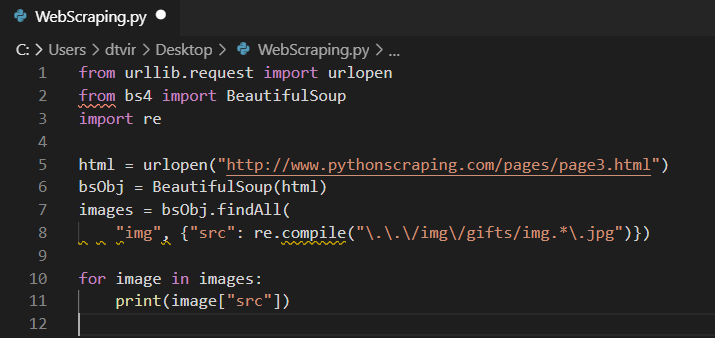
**Figure 3.** Simple web scraping code

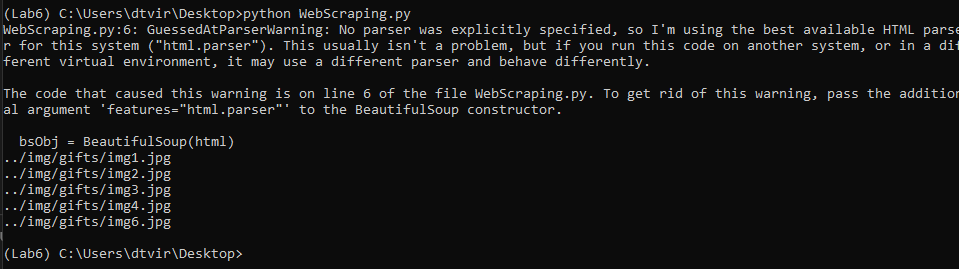
We opened VS Code in the Anaconda Prompt and then we entered these lines of codes for web scraping. It gets the data from the website: <http://pythonscraping.com/pages/page1.html>



**Figure 4.** Running the web scraping program

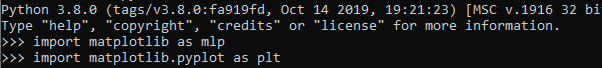
We navigated to the path where the program is saved and run it on the Anaconda Prompt. The data that can be found on the web page is printed out on the terminal.



**Figure 6.** Using BeautifulSoup Class

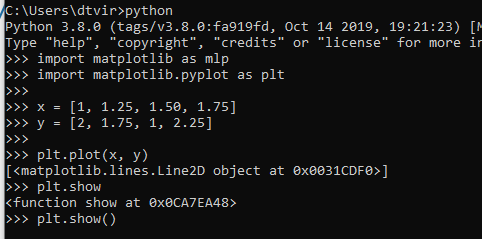
In this web scraping code, we used BeautifulSoup to create an instance which can be used to acquire the img tag and src on a web page. The sample run of the program was provided below the code.

**Part 2. Data Visualization with matplotlib**

****

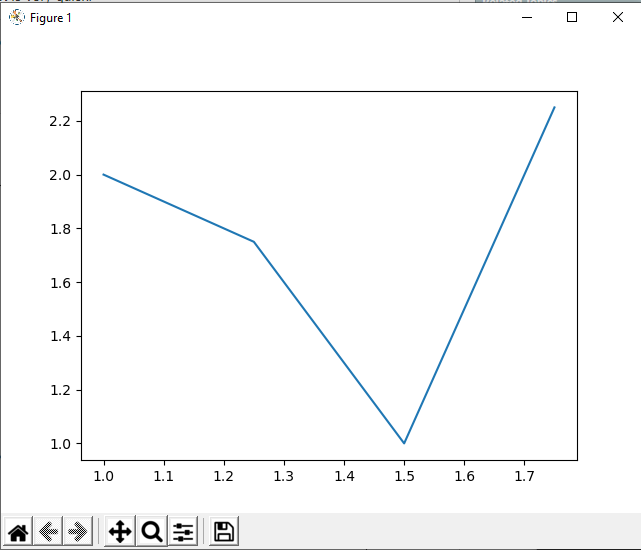
**Figure 1.** Using matplotlib in an interactive shell

Since matplotlib is already installed in my system, I can directly import it on my shell. If matplotlib is not yet installed, it can be installed using the command *pip install matplotlib*

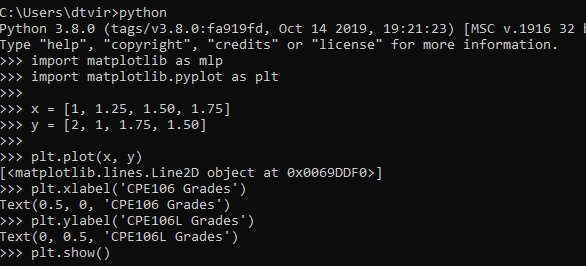


**Figure 2.** Sample commands in matplotlib

In this part, we created an array of integers named x and y. We plotted them using plt.plot() command and displayed the graph using plt.show(). The sample graph is shown below.

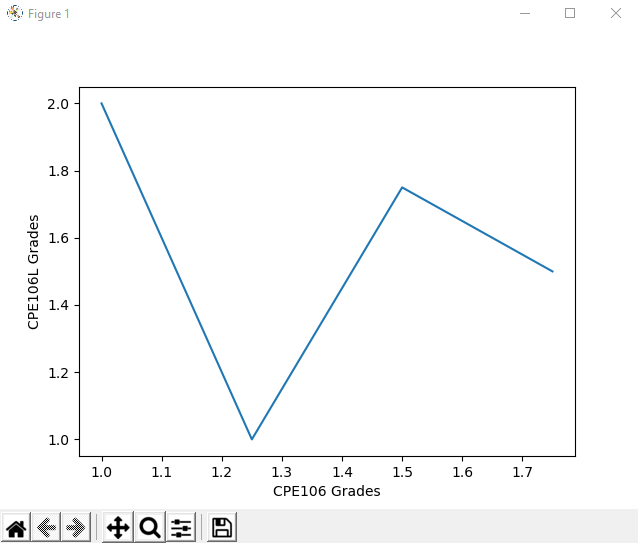


**Figure 3.** Sample run of the matplotlib commands



**Figure 4.** Sample commands in matplotlib

In this part, we introduced a new command which is the plt.xlabel() and plt.ylabel(). It basically adds label on the x-axis and the y-axis of the graph. The sample run of the code is displayed below.

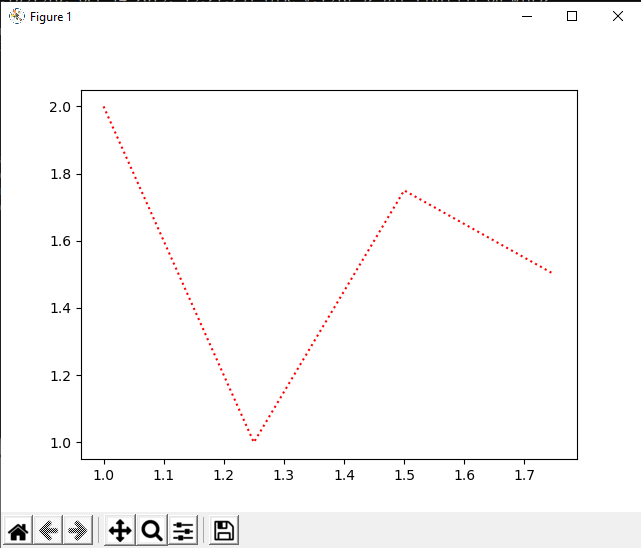


**Figure 5.** Sample run of the script

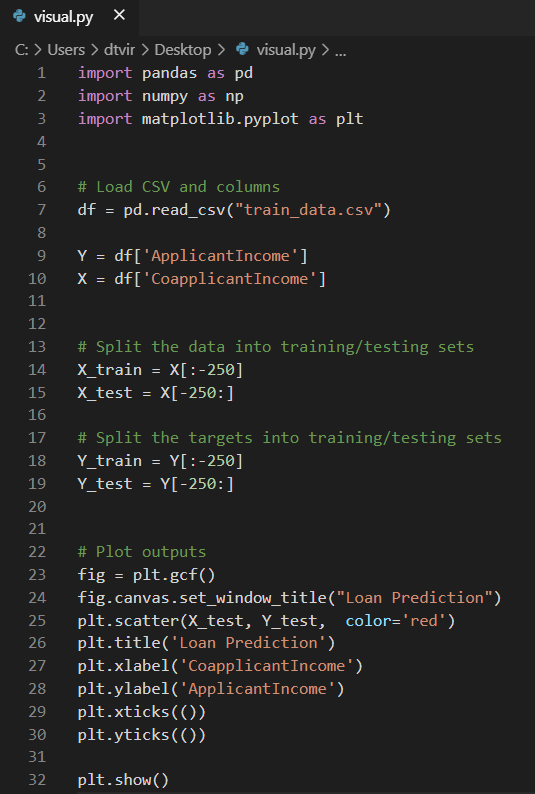


**Figure 6.** Sample commands in matplotlib

We used the same commands in Figure 4 and added a new parameter in the plt.plot() command. The third parameter controls how the graph will look like. There are several options and it can be found in the matplotlib documentation. For this example, we used ‘:r’ as the third parameter and it means dotted lines and in color red. The sample run is displayed below.

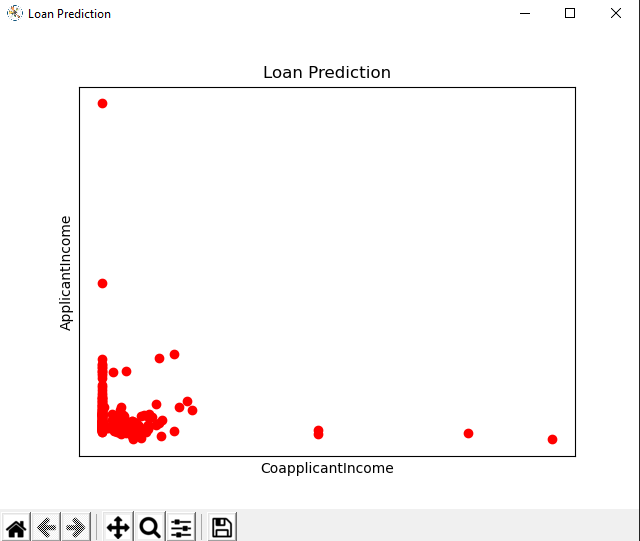


**Figure 7.** Sample run of the script



**Figure 8.** Sample python script using matplotlib

This is an example of python script that utilizes matplotlib. It uses several other libraries such as pandas and numpy. We used pandas to read into the dataset named train\_data.csv. In this script, we will used matplotlib to graph the applicant income and the co-applicant income in the dataset. It sets the window title to “Loan Prediction”. It also utilizes scatter plot and set the color to red. The label of the x-axis is the co-applicant income and for the y-axis, the label is applicant income. The sample run of the code is displayed below.



**Figure 9.** Sample run of the python script