Data Science Professional Practicum (DSCI 560)

Laboratory Assignment 1

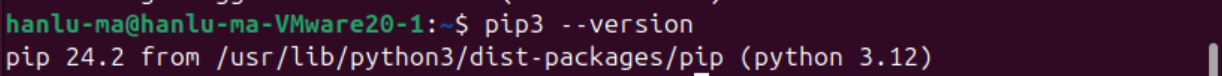
1. Installation and Setup
   1. Install VMware   
      VMware is downloaded and is waiting to be launched fully through the steps in the next section.  
      A screenshot of a computer

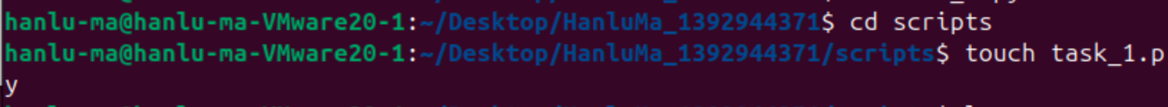
      Description automatically generatedA screenshot of a computer

      Description automatically generated
   2. Download Ubuntu ISO Image
      1. Create a new virtual machine by using Ubuntu ISO image as the installation medium, install Ubuntu, and launch the virtual machine  
         A screenshot of a computer

         Description automatically generated
   3. Install Python on Linux
      1. Update packages list fully  
         A screenshot of a computer screen

         Description automatically generated
      2. Install python3 and confirm its installation  
         A screenshot of a computer program

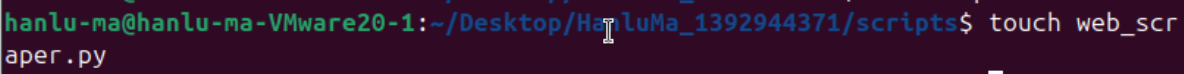
         Description automatically generated
      3. Install pip and verify its installation  
           
         
2. Get Familiar with Linux and Python
   1. Playing around with Linux Terminal
      1. All the requested directories, subdirectories, and files are created and located as requested.   
         A screenshot of a computer program

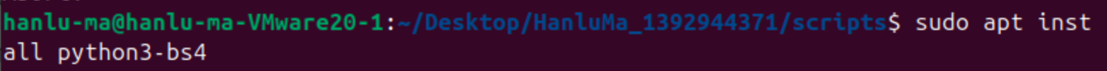
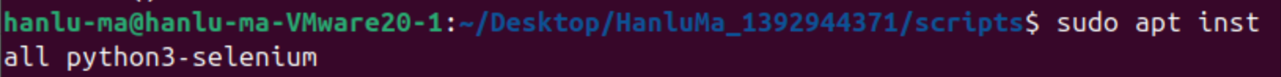
         Description automatically generated  
         A computer screen with blue text

         Description automatically generated
   2. A basic Python Script
      1. Program the basic python script for ‘task\_1.py’  
         A computer screen with blue text

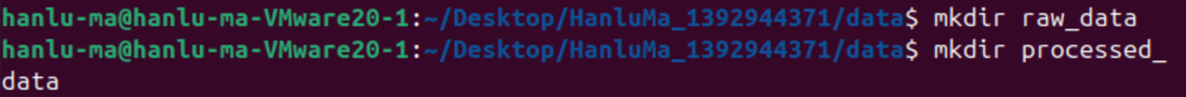
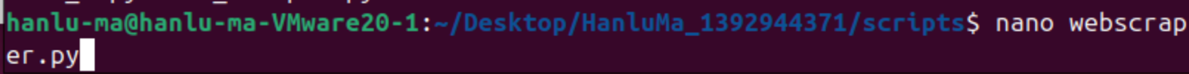
         Description automatically generated  
         A computer screen with text

         Description automatically generated
      2. Run and test the python code, which shows the desired output.  
         A computer screen with blue text

         Description automatically generated
   3. Python Web-scraping Task
      1. Create ‘web\_scraper.py’ file  
         
      2. Install libraries of requests, bs4, and selenium. I only installed requests and bs4 at first. However, when I start working on the data filtering script, I realized that the market data was not being parsed properly. After doing some research online, I learned that the market data was not parsed due to the highly dynamic nature. Therefore, I installed ‘selenium’ as this is the common tool.  
         A screenshot of a computer

         Description automatically generated  
           
         
      3. Analyze the HTML structure and find the corresponding tags for the market banner and latest news. Specifically, the market banner and all the relevant details (i.e., symbol, stock position, change pct) are all shown within the div with the class of ‘MarketBanner-main’. The latest news and all the revelant details (i.e., timestamp, title, link) are all shown within the div with the class of ‘LatestNews-isHomePage LatestNews-isIntHomePage’.  
         A screenshot of a computer

         Description automatically generated  
         A screenshot of a computer

         Description automatically generated
      4. Create folders under data folder  
         
      5. Write the ‘web\_scraper.py’ script to scrap the webpage.  
           
         Initially, I parsed the webpage and obtained the content through ‘get’ method in requests and BeautifulSoup.  
         A screenshot of a computer program

         Description automatically generated  
         After realizing that the market data cannot be parsed, I start to utilize selenium to do web-scrapping. However, after trying out all the suggested methods in piazza, I still encountered obstacles in connecting to Firefox and Chrome after working around with numerous setups and parameters. Therefore, even though I did not reach my expectation to parse the webpage by using selenium, I still want to include screenshots of my code for web-scrapping by using Firefox and Chrome as the browser driver to showcase my efforts and tires.  
         ‘web\_scraper\_firefox.py’:  
         A screenshot of a computer program

         Description automatically generated  
           
           
         ‘web\_scraper\_chrome.py’:  
         A screenshot of a computer program

         Description automatically generated  
         A computer screen with text

         Description automatically generated
      6. Rationale of the scripts:  
         For ‘requests’ web-scrapping method, ‘get’ method in requests library prompt us to send HTTP requests to the website and receive the HTML content as the response. For ‘selenium’ web-scrapping method, the script retrieves the HTML content after JavaScript has executed and the content is fully loaded. Then, ‘BeautifulSoup’ library helps to parse the HTML content and extract the desired content. Eventually, the script saves these data into files.
      7. Outcome of the web-scrapping (i.e., use the requests library):   
         A computer screen with white text

         Description automatically generated
   4. Data Filtering Task
      1. Rationale of the script  
         On the high level, we read the content of the stored file previously through ‘BeautifulSoup’. Then we locate and filter the data we want. Specifically, we zoom into the sections of HTML, find the parts (i.e., div, href, title, etc) with corresponding key words in class, and store relevant details within these parts into variables. Eventually, we iterate through every element in these variables and store them sequentially into csv.  
           
         A screenshot of a computer program

         Description automatically generated  
         A computer screen with many white and green text

         Description automatically generated
      2. Execution and output  
         Because requests library does not work well with the dynamic sections, we did not obtain the market data successfully. However, except this characteristic, the methods of parsing market data and news data are highly similar. We successfully obtained the news data, which also reflects the feasibility of the script.  
         A screenshot of a computer

         Description automatically generated  
           
           
         parts of news data:  
         A screenshot of a computer

         Description automatically generated  
         market data:  
         A close-up of a card

         Description automatically generated