Initial Requirements

Data Scraping and Database Management

- Woo Jung

User Requirements

Overview

The data scraping and database management portion will not have any direct user interaction as this will be done all behind the scenes. However, in the data scraping and database management step, the user can expect the financial and sentiment data scraped from sources on the Internet and stored into a database in the cloud for the following data analysis step.

System Requirements

Overview

This portion of the project is namely data scraping and database management. These names are broad and somewhat vague. An apt umbrella term for this portion of the project would be data preparation. Data preparation consists of the following steps: data exploration, data scraping, and data storage. Data exploration consists of exploring our choice of data to make sure our system is still capable of scraping and storing the data. Data scraping consists of the physical scraping (collection) of the data from online sources. Data storage consists of storing our collected data into a database.

Data Exploration

Data exploration will be done in person before the actual data scraping. In this step, we hope to figure out whether our online sources are using JavaScript to load their website and what format our data is presented in. This can be done using a normal browser and disabling JavaScript. This will tell us whether a website loads completely with or without JavaScript. This is an important step as this is the deciding force behind which Python libraries we will be using for data scraping. Also figuring out which format our data is presented in is helpful as it will allow us to export the scraped data in the most effective file format.

Data Scraping

Data scraping will be done using the Python BeautifulSoup4 and Selenium packages. BeautifulSoup4 will be our primary means of extracting data from our webpage. It is a Python library which provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree. However, BeautifulSoup4 has the working constraint of only being able to read the HTML file upon visiting a particular webpage. This means BeautifulSoup4 is not an applicable library for reading websites built completely using JavaScript as most of the website's HTML is loaded after the initial request. In this case, we need a physical browser to visit the webpage and wait for the Javascript to load the rest of the HTML and then scrape the website. This can be done using the Selenium library. Selenium is a package that allows us to spin up any kind of browser-whether that is Chrome, Chromium, Firefox, Safari, or anything browser on the market. It provides us with an API to control these browsers using Python. Thus, using BeautifulSoup4 and Selenium, we are able to effectively scrape financial and sentiment data from various websites.

Data Storage

Data storage is the step where we store the collected data from the data scraping step. The scraped data will be stored as either a CSV (Comma-Separated Values), TSV (Tab-Separated Values), or a different file format if any data contains a comma or a tab which could disrupt the file format. We should

have figured out which file format we will be using in our initial data exploration step. We will have a relational database in a cloud server and we will feed the data file to our table using SQLite3.

Maintenance/Updates

The previous steps of data exploration, scraping, and storage are only relevant to the initial data collection and storage. However, our project requires that we collect and update our data on a daily basis. This means that a major portion of our data scraping and database management step is maintenance and updating our relational database with new information as time passes. This will be done on a set interval - most likely a 12 or 24 hour interval. This is because the stock market opens at 9:30am EST and closes at 4:00pm EST everyday. This means there are constant updates for 6 ½ hours on a daily basis. This implies that we can scrape the day's worth of data after 4:00pm EST and we should theoretically have the full days' worth of data. The daily data scraping will be similar to the initial data collection. The storage will be adding Tuples of data to our relation online. Our server will be online 24/7 running our data collection and storage script so we automatically have the most up-to-date data each day.