Spring 2020 Co-op Report

Author: Ron Gorai

Company: ZAIS Group

Industry: Financial Services

Introduction

During the spring of 2020, I worked at ZAIS Group as a Technology and Analytics Co-op alongside two of my classmates from the Stevens Institute of Technology, Lucas and Mohammad. We worked closely with Stefan Le Noache, our immediate supervisor, and Luyao Ma, an impact investing analyst.

ZAIS is a financial services company that specializes in alternative credit and asset management. The data gathered from our work with the rest of the IT team would be utilized by some of the other teams at ZAIS.

Project Details

Our primary task was to write Python scripts to scrape ESG data from different websites. This data encompassed the different environmental, social, and governance themes of different companies. For the purpose of this project, we also used applications such as GitHub and the Microsoft Azure cloud platform.

Training

During the first week of the co-op, we spent our time learning how to use Python to scrape data from a basic website. Our group head and CTO, Sean Lensborn, introduced us to web scraping with a Python library called Scrapy. My first script recorded the ESG scores of one company on Yahoo Finance. I was soon able to scrape a whole table of data from a Wikipedia page that Sean had shown me.

Next, we were given a list of companies from another senior member of the IT team. We needed to search for Yahoo Finance for ESG data on these companies. However, since there were slight differences in the naming of companies used in the list and the naming that Yahoo's website recognizes, we decided it would be more precise to use the companies' ticker symbols rather than the names from the list. Luckily, we found a free website that would supply the ticker symbol of a given company. Unfortunately, the Scrapy script was denied access to the website, so I learned how to use Beautiful Soup, another Python web scraping library, to bypass this issue.

Before I started working on the official scrapers, I looked at and completed a tutorial Sean sent me on how to automatically upload files to Azure Data Lake Storage, which is what we would be doing with the data gathered from our scrapers.

Main Project

Soon after our introductory week, Luyao gave us a list of web pages to scrape data from. We were tasked to work on the more important ones first, and gradually make our way through the rest of the list.

I wrote scrapers for the following companies' websites from the list:

- Annual Reports
- CDP
- CSRHub
- HRC
- JUST Capital
- NRC
- PRI
- RobecoSAM
- TCFD

The most challenging out of these was CDP. This scraper was one of the first I attempted. Since the website relied heavily on JavaScript components such as iframes, the Python libraries I had used until then were unable to extract the data that I needed. I found yet another Python library called Selenium, which supports reading JavaScript, and has the power to dynamically navigate webpages and control them by actually emulating a browser of choice.

Even though my script was now able to read the data from the table I was trying to scrape, I soon realized the biggest issue that basically doomed the project for the rest of the term: the table was a dynamically growing list and updated as the user scrolled down. This meant that not all the data was available in the source code when the page first loaded, and the page even unloaded previous data as the user scrolled down. I even tried looking through the website's files with Chrome's DevTools but to no avail. I eventually decided to take a break from this script while I worked on the other important scrapers.

When I came back to it towards the end of the term, I was able to make it consistently gather the data from the table by scrolling and scraping in conjunction. However, the website seemed to have some sort of cap on scraping because no matter what I tried the script could not collect any data after exactly row 250, every single time. Since it is important data, we decided to just use this code, perhaps in the hopes that some time in the future they can find a way to bypass this hard cap.

Post-Project

After the scrapers were completed, I spent the rest of the co-op term creating components for a website drafted by Alex, from the design team. I worked with HTML, CSS, and Node and React JavaScript libraries to create them.

First, I created the empty state component, which is basically just a simple design that would be shown to the user when they are trying to access a non-existent page (similar to a 404 error). Next, I started working on an application switcher, which is a window of tiles, each responsible for launching an application, based on what the user has.

Conclusion

Being a part of the IT team at ZAIS taught me several things, both technically and professionally. Perhaps the most important skill I learned was using GitHub in my everyday life as a programmer; I learned how to integrate it into my IDE and how to utilize many of the commands to perform the tasks I needed. I also learned how much easier GitHub makes team projects by keeping track of different versions using branches. Professionally, I was exposed to multiple different types of meetings with different people and I learned how to collaborate with people that I worked directly and indirectly with.

I would like to thank Sean Lensborn and Nancy Gill for providing me with this opportunity. I found my time at ZAIS both enjoyable and instructive, and I plan to utilize the knowledge I gained from this experience to the fullest in my future endeavors.