NIL Deals Final Project Write Up

Emily Overend

ECON 8320

Scraping the NIL College Athletes Website

For this final project, I chose to scrape data from both the NIL College Athletes and the On3 NIL 100 websites. I started this project by scraping the NIL College Athletes website first. Looking at this website, I knew I would need to figure out how to extract each individual value for Sponsor in each row and figure out how to append each individual sponsor value as a separate row in a new data frame with the same athlete name, university, and sport. I knew that was going to be a struggle, but I decided to scrape this website first anyways. Looking back now, I wish I started with the On3 NIL 100 website instead, and that is something I would do differently in the future- I would evaluate each website or other websites more carefully before deciding to scrape it.

During my first time scraping the NIL College Athletes website, I started by finding the tags for each cell of the table: I would find the <td> tag and the class attribute to get the names of all the athletes, and then I would find the <td> tag and the class attribute to get the sponsors, and I would find the same <td> tag and class attribute to get the university and sport for each row. After finding all the necessary tags and class attributes I needed, I worked on extracting the text from each of these tags and putting them in a list. After creating individual lists for the name, sponsors, university, and sport, I knew I would need to put these lists together into one data frame, and I also still needed to fix the sponsors so that each sponsor value would be its own row in the new data frame. But then I remembered you can scrape an entire table from a website.

Scraping the NIL College Athletes Table

I started a second attempt at scraping the NIL College Athletes website where I parsed the web page with BeautifulSoup and changed the information from HTML to LXML, and then I looked for the <table> tag and the class attribute to scrape the entire table. I decided to do a second attempt at scraping this website because I thought scraping the entire table would be more efficient, and I also thought that maybe it would be easier to fix the sponsor values issue. I was able to scrape the entire table for the first page of the website, but still wasn't able to figure out the code to fix the sponsors. I've included the code for this with my submission.

After scraping the entire table, I decided to go back to the first attempt at scraping the NIL College Athletes website. I created the lists studNames, sponsors, univ, and sport, and turned each of those lists into data frames. Then I joined the univ and sport data frames together to create the univSport data frame, and then I joined sponsors and univSport to the studNames data frame to make one data frame consisting of all those columns. After this, I worked on scraping data from the other pages and combining the data from other pages. When I was following the getting data from other pages step with the GitHub example, I learned that you need to create a root object that contains the url of the website, you need to use the <div> tag and the class attribute in parsed.find() because it would not scrape simply using the tags like how it shows in the GitHub, and I learned about the function urljoin() to join the href link to the root.

Scraping the On3 NIL 100 Website

I started scraping the On3 NIL 100 web page because the data was easier to collect and I needed more quantitative data to perform analyses and make graphs with. I scraped the school level (college or high school), athlete's name, ratings, status (commit, FR in college, SO in college, ect.), the total number of social media followers, and the On3 NIL valuation. I wish I looked at this website first and focused most of my scraping with this website because I could have scraped the top athletes for different sports at the college or high school level, and then after scraping all of the information I wanted, I could have performed analyses and made comparisons for each sport, school level (college or high school), and gender. But because I was running out of time to finish this project, I only scraped information on the top 100 athletes regardless of sport, gender, and school level. After scraping all of the information and creating the function, I performed a multiple regression with Valuation as the dependent variable and school level, rating, status, and social media followers. This multiple regression produced a R-square value of 1.00 and an adjusted R-square value of "nan" and I knew something was wrong with this regression, so I removed ratings as a predictor variable and the new results show a R-square of 0.655 and an adjusted R-square of 0.616.

Creating the valchange() Function

The data type for the values for number social media followers and valuation were objects and had other characters attached to them. The social media values had either M or K after the number, and valuation had a $ in front of the value and either M or K after it. To get rid of the M or K, I created a function with If/Elif/Else statements that checked to see if the object had an M or a K after it, dropped the M or K, and then multiplied the value by either 1000000 for a numbers in the millions or 1000 for numbers in the thousands. Then it turned those values into floats. To get rid of the $ symbol in front of the valuation values, I used the .replace() function that replaces '$' with a space (' '). I then created loops to extract the data for these columns, which first pulled the entire text for each row, then dropped the $ symbol for valuation (this step was skipped for social media follower count), then applied the function to the text (which did not include the $ sign for valuation), then converted the text to a float object, and finally appended the float value to the lists. I thought this part of my project was interesting and fun to do because I've never created my own function and called it within another function that I also created. When I was first thinking of a process that could drop the $, M, and K, I initially thought I would have to do a loop within a loop, but I'm glad I created a separate function that could easily be applied within one loop because it made things less complicated. I'll be looking to make more functions and calling them within loops and other functions in the future.

Conclusion

In my project proposal, I stated that I would collect NIL data regarding the student's school, sport, sponsorship, and social media account. I was able to collect the student's school, sport, the company's name that is sponsoring the athlete, but I was unable to collect the social media handle (the number of social media followers was not listed) or the value of the sponsorships from the NIL College Athletes website. For the On3 NIL 100 website, I was able to scrape data on the school level (college or high school), athlete's name, ratings, status (commit, FR in college, SO in college, ect.), the total number of social media followers, and the valuation. By scraping these two websites, I was able to collect data regarding the student athlete's name, school, sport, sponsorship, valuation, and information about their social media accounts (specifically the number of followers), which I stated I would scrape these data in my project proposal. Unfortunately, I did not have time to scrape the student enrollment count/size of the school, the amount of funding the school's athletic department receives, or the student's gender. These data would have been really interesting to collect and evaluate in a multiple regression, and maybe in the future I'll come back to this project and try to scrape these data.

By completing this project, I became more familiar and more comfortable with creating loops, functions, Pandas dataframes, and scraping websites. I learned that with some websites you need to include more attributes (such as the class attribute) along with the tag as parameters in functions in order to extract the data you want to scrape, I should start by scraping the entire table instead of scraping each individual column, I can create functions (not just loops) and call those functions in loops and other functions, and I also learned that I need to evaluate websites more carefully and also do some computational thinking before deciding to scrape them. Recognizing these things would help me scrape websites more efficiently. Furthermore, I should solve problems by the process of computational thinking before jumping into the problem because computational thinking would also help me code more efficiently as well.