Scott Ricketts

Project Writeup

ECON 8320: Tools for Data Analysis

Spring 2023

**Brief Project Summary**

In this project, I used Beautiful Soup to scrape [www.nilcollegeathletes.com](http://www.nilcollegeathletes.com) for information about Name, Image, and Likeness (NIL) deals for college athletes. From this site, I collected each athlete’s name, sponsorships, university, sport, social media links, and gender. The data was collected into two separate Pandas data frames and then merged into a single data frame to form a complete data set on which to conduct analysis. The complete data set contains 8,515 rows and 7 columns. The Plotly library was used to create visualizations with the data to help with the analysis.

**Results**

The data set contains 8,348 unique athlete names, 66 sponsors, 502 universities, and 36 different sports. Additionally, there are 5,546 male athletes in the data set and 2,967 female athletes. The athlete with the most sponsors is Jashon Hubbard, a wrestler from Ohio State, who has 12 sponsors. Interestingly, the men and women in the data have approximately the same number of sponsors to athlete ratio. For men, the ratio is .9138 sponsors per athlete, while women are at .9097. These numbers are less than 1 because the site has some athletes listed as having an NIL deal without providing a name for the sponsor. It is unclear from the website if these athletes had deals and then lost them, or why these athletes continue to be listed on the site. While the site does appear to be still adding new deals, a quick glance at some of the data reveals a few inaccuracies, so it’s possible that those athletes without a deal listed are on there with no sponsor due to error by the group who maintains the site.

Barstool Sports is the top sponsor in the data, sponsoring 7,514 athletes. This is more athletes than all the other sponsors combined, with the next closest, Rhoback and Liquid I.V., both sponsoring 30 athletes each. Clearly, additional research is needed to better understand why Barstool Sports far outpaces other sponsors. A quick glance at their page on the nilcollegeathlete website reveals that “[e]ach Barstool Athlete gets a t-shirt, sweatpants, and hoodie as part of the deal.” Perhaps the explanation is as simple as Barstool’s NIL contribution to each athlete is relatively low from a fiscal perspective, thus allowing for the company to sponsor more athletes.

Ohio State University has 98 athletes with NIL deals, which is the most in the data. The median number for universities is 11 athletes with deals. There are 16 outlier universities are that are above the upper limit of 54 athletes with sponsorship. Of those 16 outlier schools, the Big Ten conference has is represented by 6 schools, including the top 3 positions. Only one SEC team, Alabama, is a high outlier. Schools not in the Power Five football conferences make up most of the rest of the list. Three of those schools are in the Big East, known primarily as a basketball conference, which may account for their high sponsorship level.

Not surprisingly, football is the sport with the most athlete sponsorships at 1,527, approximately 33% more than baseball, which is in the second spot. However, basketball, the other revenue-generating college sport, only has the sixth most sponsorships. Non-revenue sports baseball, soccer, lacrosse, and track and field are in the 2 through 5 positions. This is interesting because most of these sports do not offer their players full scholarships, so there may be a relationship between the lack of financial support for the athletes via traditional means and making up that difference in NIL deals. Similarly, when looking at the athletes with the most sponsorships, only 2 of the top 10 are football players, zero are basketball players, and the rest of the list is comprised of athletes in non-revenue sports.

**Methods**

As mentioned above, the data was collected by scraping the website and placing it into two data frames that were then merged. For the first data frame, Beautiful Soup was employed to scrape the <https://nilcollegeathletes.com/athletes> page, gathering each athlete’s name, sponsors, university, and sport. Each page has approximately 20 athletes on it and code was written to check for a “Next Page” link at the bottom to iterate over those pages as well. Additionally, it was important to put each sponsor on its own row even if that meant having an athlete’s name and other info appear multiple times. This was so I could use a Python groupby method in my code to calculate and plot accurate counts. While organizing the data this way made the final data frame longer, I found it easier to access those individual sponsors. When I was coding this, I knew that I need a ‘for loop’ inside of another, but I keep ending up with one row per athlete with either only the first sponsor or a list of all the sponsors in the same row. After talking with Dr. White, I realized the first loop need to collect the multiple sponsors in a list while the second loop would actually create the multiple rows by iterating over the list of sponsors. The appended data was then placed into a Pandas data frame.

The second set of data also began on <https://nilcollegeathletes.com/athletes> and iterated through the rows and the pages similarly to the first data set. However, when iterating, the code was finding the “More Info” link for each athlete to go to their specific page. Once on the athlete specific page, the code grabbed the athlete’s name, Instagram, and Twitter links and appended each in a list. REGEX was also used to look for the word “He” in the “About” section. If “He” was found, a 1 was appended to a gender list. If it wasn’t found, a 0 was appended in the list. After the code looped through all the athletes, a data frame was created using that list.

The two data frames were then combined into a single data frame using Pandas merge. Because both data frames had the athletes’ names, I was able to join on that field. I selected an outer join because I didn’t want to lose any data from either data frame. Once the data frames were merged, I exported it to a .csv file which I then imported as a new data frame. I did this so I knew that I had the data I would need for analysis without having to rerun my code. The code for creating the second data frame took about an hour to run, and I wanted to avoid having that down time.

Once I had all the data in a single data frame, I explored it by using the Python count method to see what each of the columns looked like. I did not use the dropna method because I didn’t want to lose an entire row if a value was missing, particularly considering not every athlete had social media accounts listed. For example, in this data, there are only 6,634 unique Twitter links out of 8,348 athletes. From there, I used the unique method to find counts of each row’s unique data, and the I used the Pandas groupby method to organize the data by gender. I also used the groupby method to organize the data for visualizations. I create some bar charts, boxplots, and histograms with this data using the Plotly Express library. The insights gleaned from those visualizations have been discussed above and the visualizations themselves are in my PowerPoint deck.

**Difficulties**

There were several learning opportunities for me in this project. The first thing I learned is to keep the scope of my project smaller rather than larger. When I looked back at my Project Proposal, I can see that I over-promised. Instead of just proposing to scrape one of the websites, I proposed to scrape both NIL sites and even bring in some information scraped from the NCAA’s website. It turns out that simply scraping nilcollegeathletes.com was more than enough challenge for my coding skills. Additionally, proposing a full dashboard in plotly was, again, too ambitious. It’s not that creating an app using Dash seems overly difficult, but with the time constraints on this project and my struggles to simply scrape a single site, sticking to some charts in plotly was the better idea.

I also learned a great deal about decomposing the project in the planning stage, as well as debugging the code as I progressed through the project. My lack of decomposing goes together with overpromising in the proposal: I just sort of jumped in without really laying out the best strategy for solving the various parts in the web scraping aspect of this project. Had I really scoped out how I should have written the code, I would not have made the mistake of scraping column-wise instead of row-wise. Related to that is the fact that I tend to write bigger chunks of code at one time than I should, which makes it more difficult to debug. If I had properly decomposed the steps, it would have been much easier for me to see where I had made mistakes. The upside is that spending so much time debugging ended up being useful because it forced me to look at my code in smaller blocks to figure what worked and what didn’t. These were tough lessons to learn, but they are certainly useful.

**Future Projects**

There are a few things that I would do differently if I were to take on a similar project in the future. As mentioned above, I would write my code to scrape row-wise instead of column-wise. Again, this is a mistake I made by just jumping in before I really understood everything I needed to for the project. Had I slowed down, I would have been able to use more of the code from class as starter code on this project and that may have caused fewer issues for me as I progressed. The upside is that at least now I have code for both row- and column-wise scraping available if I need it.

Another thing I would do in the future is try to put the entire code into one function instead of two. This would allow me to avoid the extra step of using pandas ‘merge’ to combine the two data frames. In other words, a single function would keep all the needed data in one place. I did consider attempting one function here, but because I was having so many problems getting my code to work, I just kept the two separate so as not to increase the level of difficulty.

Finally, I think one additional thing I would do on this project if I had more time is to take it a step further by navigating to the individual deals page and grabbing the value of the individual sponsorships. Not all the deals have the amount, but it would be interesting to look at the ones where there is a number to see if any patterns emerge from that data. A step beyond that would be to scrape the social media posts where the athlete announces their deal and maybe use some natural language processing to analyze those.