Tyler Pooley

ECON 8320: Tools for Data Analysis

May 10th, 2023

**Current NIL Landscape in College Athletics**

My semester project focused on the wild landscape that is the current NIL environment in college athletics. Many have referred to it so far as the “wild west”, and that’s true because there have been many fast changes to college athletics without guardrails in place and without a way to analyze the effects of the changes. As someone who has not made up his mind as to the benefits and drawbacks of NIL, I was interested in gathering data in an attempt to help assess the present landscape.

The data collection included information regarding the athlete’s name, school, sport, and sponsor. This is different than what I originally set out to do which was grab data focused on sponsorship type and the monetary value of them. The site I used to gather data was not set up well for finding monetary value; actually, many deal records contained no information regarding monetary value or the financial terms of the deal. I therefore had to pivot from the original plan of assessing if the amount of money players receive by school affected their overall winning percentages and focusing primarily on obtaining the data in a clean format with what information was actually available. The monetary value was not there for most deals, but the sponsor was almost always there. For that reason I focused on grabbing the sponsors’ names for each athlete and pairing that with the athlete’s name for each athlete-sponsor pair. Doing it this way enabled me to get a clean count of those relationships as well as see any information regarding common sponsor-school connections or sponsor-sport connections since each row of data was exactly one athlete, school, sponsor, and sport.

The data collection strategy was sound, but I struggled mightily in the implementation. I learned a lot, though, about how to approach this type of exercise in the future. An example is implementing a user agent to access websites to get data. Early on, I was unable to access the site because I failed to pass a user agent as part of my code, enabling the site to immediately know I was not a browser and denying me access. Implementing a user agent fixed this and I no longer encountered that issue. I now know how, without a user agent, a website can easily know and block you, so it’s always good to use one.

Another challenge I faced was getting the code to correctly go to the subsequent pages to continue grabbing the athletes’ NIL data. I know nearly nothing about HTML, so while the issue likely is a result of that lack of knowledge rather than something truly unique about the page’s design, it appeared as if the tag format of the “Next (page)” button was different than I was used to seeing. I tried many iterations of code to get to the next page but with no success. I ended up changing my approach to one that is, in my opinion, less ideal but achieve my goal. I added components to the code related to page number that told the code to go to each page number in this range of numbers, grab the data from each one, and then continuously append the data until completed. The downside of this approach is that a page number range must be supplied which means either entering it manually or finding a way to grab the page number of the last page and make that the last page. All that being said, having to make that small adjustment is minimal to accomplishing the objective. From this obstacle I learned having even a basic understanding of HTML would be beneficial in knowing the components of it and, ultimately, knowing where to the Python code to go.

Those were some valuable lessons learned, but I also learned about two valuable methods that helped achieve what I wanted- a clean data frame from which further analyses could be conducted. The website had each athlete listed and then, along with the name, a list of each sponsor with whom they have deals. I wanted the data, however, to be pulled such that each row was the name of the player, name of one sponsor, name of the school, and name of the sport. In other words, even if the same athlete had multiple sponsors, each would be on a new row. In reading about the best way to go about this, I learned about the “split” and “explode” methods. These methods, in conjunction, tell Python to “explode” each of the list elements (i.e. sponsor names) into individual components and then split that data over multiple rows. This was just a couple lines of code that achieved exactly what I needed and would be valuable in the future because this type of instance is common on websites.

Another successful strategy I used was building the code in a small way and then expanding it as needed as the project progressed. I started by establishing the import statements, establishing the code to access the site, and grabbing the first page’s data rather than the entire set of pages. This enabled me to get the code set up quickly, and from there the only remaining piece was getting the data from the remaining pages.

There were results both good and bad about my project, but going forward, I would approach one important aspect of the project differently. I would attempt to define the scope of the project in a more realistic way. I went into this project with larger expectations than what I was actually able to achieve. There is undoubtedly an aspect of this type of project that you don’t fully know all the challenges that will appear, but it would have made sense to spend more time assessing what would be required to do everything outlined in the proposal to determine if it is feasible.

I believe I executed parts of this project well and other parts not so well. If I had done a better job researching and planning the scope, the project would have gone better because I was effective in starting the coding in small chunks and learned ways to succinctly do what I needed. Putting all that together would have enabled me to achieve my objective much faster, but now I have these learnings to build off of in the future.