Daniel Moeller
March 9, 2017

COSC/MATH 4931: Midterm Milestone Report

All the documentation and code referenced in this report can be found in the following github repository:

https://github.com/danmoeller/ncaa-bball-attendance

This report assumes the reader understands what this project is at a high level and has a casual knowledge of college basketball. If there is any doubt to either of these please visit the README.md or the introduction PowerPoint in the doc directory.

To start this project I began with where I knew the bulk of the data I would need would reside, ESPN. ESPN is the self-proclaimed "worldwide leader in sports" and has the most robust and complete set of information for nearly any sport. Unlike many of my fellow classmates, this data is not readily available through an easily exportable dataset. There is a free API that ESPN offers to stay up to date with a sport, team, or player; but it is not designed to pull the historical data I was interested in, which is where much of my need lies. This led to the realization that some form of web scrapping would be required. After researching others that are pulling similar data from ESPN (*surprise someone on the internet has done it first*), I decided to use a popular open source python framework called Scrapy. [3] Even for a new python developer, Scrapy turned out to have a very small learning curve with most of the issues coming from sifting through large quantities of html tags to understand a specific page's structure.

After a few quick tutorials, I set out to write my first spider, or page crawler, for a single college basketball game [1]. This process was simple enough to start, with a lot of trial and error around how the data came out the other side. Through this process I also was able to identify data points that I had previously thought were inaccessible such as

gambling information, television network coverage, time of day, and even the names of the referees officiating the game. The real challenge began when I started looking at pulling different games with the same spider. While each game page on ESPN's website is relatively similar, there are many games that are either missing data or have different data formats. This led to much work within the spider to check the possible ways the data could be presented in order to ensure the data pulled is correctly represents the actual information. In similar fashion I created a spider for crawling a schools game schedule for a year as well as a spider that gathers all the identifiable information around each school.

It was at this point that the issue surrounding how a team is identified became very apparent. Depending on the source of the information, even within ESPN itself, each school can be identified by their name, mascot, nickname, abbreviation, or some form of identification number. This variability in isolation would be but a small hill for a climber. However, the issue lies with the fact that different sources have different abbreviations for longer team names. For example, in my exploratory analysis I have seen Texas A&M University-Corpus Christi also written as Texas A&M-Corpus Christi, TAMUCC, A&M-Corpus Christi, and A&M-CC. In an attempt to join game information from ESPN to aggregated attendance numbers from the NCAA, I discovered over half the schools in division 1 basketball have names that do not directly match between the two entities. This currently serves as a problem I am still facing and exploring possible solutions. Two that have set themselves apart from the others are using a python library for fuzzy string matching such as fuzzywuzzy or generating a master table as a go for more robust joining [2]. The first solution would obviously be less intensive, but the second option would guarantee accuracy.

This first issue I encountered really open my eyes to how singular my project had become. I was planning on pulling all my data at once and then runny some analysis on it. I realized that if I could make the data extraction repeatable and customizable, I could gather data from which every school for any year I desired. This led to the creation of an extraction shell script that would be the brunt of my data collection and wrangling. This script is designed to pull game specific data for a given school identifier and a given season. If the identifier is unknown, it can be searched for using the lookup script. The extraction works by first crawling the schools schedule of games and storing it in a file. Then it iterates through said file of games and uses the game identifier to crawl each of the games data. All of this information is stored together and sorted out to remove any erroneous values. Once multiple seasons of data have been gathered, they can be combined through the combine script. This script is relatively simple and more or less appends each season file and filters out the header lines. Thus there is a very barebones way of extracting any data that is desired. I have future plans for this script to be more flexible around which data it pulls. The plan is to add features around pulling data for entire conferences worth of data, adding an option to include or exclude neutral court games, and excluding games not against conference opponents.

With a rudimentary data collection process I began to do some exploratory analysis to identify general trends to help me with my direction. I settled on using a Jupyter Notebook to this analysis because of its robust interface and the language consistency with the web scrapping [4]. Since my data collection process is more nimble than static, I didn't want to waste too much time gathering data in case I needed more information from the site. Thus I have just left a few data sets in the data directory in case you wanted to look at

the graphs. This part of the project is where things really started to come together and I gained understanding in what my data was and what it was not. I decided to graph the attendance of each game against features that have been identified by others as critical to determining attendance [6]. Along with expected trends, I began seeing glitches in my data. I was identifying games that had no attendance recorded, games with the wrong stadium set as its capacity, the time of the game was defaulting to midnight, and most importantly betting lines were regularly unavailable. These are all issues that have still yet to be solved. I have set out and collected more data regarding the betting lines, but many of the other issues do not have a simple solution.

These are the types of things that are of high priority on my backlog. Due to the vast quantity of records I am pulling myself, tidying this data up in a repeatable way could take me most of the semester. I am working towards prioritizing what issues are really detrimental, and what can be ignored because of the minuscule effect. The beauty of my project's framework is that the actual actions are programmable. Thus each change to the process is immediately seen in the results. I need to prioritize the actual analysis high enough however that I do not put it off until too late. Luckily in terms of the general idea around my report, I am not the first person to look at what determines attendance. I have been using similar papers from Syracuse University as we as here at Marquette University to help guide my analysis [5][6]. It does appear many of these reports are a form of regression, which I am more comfortable with, but I plan to continue to look for opportunities to use other new techniques to help answer similar questions.

I am very excited about the work already completed and still to come for this project. It has always been hard for me as a young developer to identify a project I would

like to pursue whole-heartedly. This class has given me the motivation to tackle a project by myself that is bigger than just a few weeks.

References

1. Duke, J. (2016) *How to Crawl A Web Page with Scrapy and Python 3*. Retrieved from https://www.digitalocean.com/community/tutorials/how-to-crawl-a-web-page-with-scrapy-and-python-3
2. Fuzzywuzzy (15.0) [Software]. (2017). New York: SeatGeek. Retrieved from https://github.com/seatgeek/fuzzywuzzy
3. Howell, P. (2015). NCAA_MBB. [Software] Retrieved from https://github.com/pathow/NCAA_MBB
4. Jupyter (1.0) [Software]. (2017). Project Jupyter. Retrieved from https://github.com/jupyterhub/jupyterhub
5. Mueller, K. (2013). *Marquette Basketball Attendance Analysis*. Retrieved from http://epublications.marquette.edu/cgi/viewcontent.cgi?article=1052&context=cps_professional
6. Paul, R., Weinbach, A., Plaut, J. *Information, Prediction Markets, and NCAA Basketball Attendance: The Big East Conference*. Retrieved from http://buildingthepride.com/jobie/uploads/JOBIEV19B31.pdf