MCC’s Course Enrollment Data:

An Event Analysis

Chris Tapia

Dr. Tahir

CIS291M

August 3, 2019

Say, you’re a college professor and you may be trying to figure out why you haven’t had as many students enrolled this semester. You may be wondering what happened or if there was something that you could have done to improve the enrollment. The goal of my project was to grab the student enrollment data from Manchester Community College so that, when compared to the college’s event, they can see what is driving the enrollment of students to specific classes. A project where I write a program that takes the data from the college’s classes that are being held in their respective semester. More of a focus on the numbers of enrollment week over week and comparing it to things that the school does to drive more enrollment (signs, events, etc.). The end result was something beautiful. The developed the project so the user is presented with options to produce a tailored representation of their data in the forms of a horizontal bar graph saved as a PDF, a new CSV file showing hard numbers and growth percentage, as well as a couple of highlights in the summary. Anyone can run this and follow the three prompts and at a glance or in detail of a course’s enrollment data.

On the surface, one begins to question, how do you go about doing that? It’s starts with cleaning up the data and then pulling what you need from what you don’t need. Once I received the data, I knew I needed to clean it up; it was messy. I asked myself, should I just clean up this existing file? No, I needed to start fresh, on a clean piece of canvas. It makes a huge difference when you are pulling out something from a mess and start to put the pieces together. Upon the start of this class, I had been learning data science with Python online and thus I immediately knew I needed to use Pandas and NumPy. It just so happens that I hadn’t gotten to learning about Pandas in the course yet.

I have already learned much already with Python and file manipulation from my other classes. I knew how to create CSV files and how to import lists, but this would have to be a step up from that. With many columns and needing to pull specific information, I proceeded to look up the Pandas docs and learn of ways to clean up data and get what I need. I learned how to import an existing CSV file to a Pandas Data Frame. You’re able to choose your columns and reorganize it so you can create another CSV with that data and go onto producing graphs. At first, I developed the program to produce all the classes in a chosen subject. Then there was a need to dig deeper. I then implemented statistics for the courses in that subject. As I dug deeper, I realized I needed to see the course itself and compare it to how it’s been doing over X amount of days. I had some difficulty learning the Pandas modules and using it with Python and its own data structures. One of the more difficult situations were using Pandas group by function to group the data by subject or course title while simultaneously getting the sum of all enrollments by those course titles/ subjects (“Group By: Split-Apply-Combine”,1). Among the places researched more on Pandas grouping mechanics was from Hugo Bowne-Anderson. Hugo stated that, “*Groupby objects are not intuitive. They do, however, correspond to a natural the act of splitting a dataset with respect to one of its columns (or more than one, but let's save that for another post about grouping by multiple columns and hierarchical indexes). The split-apply-combine principle is not only elegant and practical, it's something that Data Scientists use daily, as in the above example.*”(Bowne-Anderson,1). After working on this project, what Hugo wrote is true. They’re definitely not intuitive but, once you figure them out, they can be the best tool you have when working with a Pandas Data Frame. The other thing that helped was my background in sales.

Coming from a sales and business background, my approach this project was very analytical and business oriented. The way I saw it, the College needs to hit goals, just like any business. Those numbers are a direct reflection of how well they’re doing. Seeing it this way helped, especially since I’ve made myself previous sales trackers to aid in my own personal success. I knew the pieces of data that were important for the everyday person to view who wants to know “how they’re doing”. It doesn’t have to be overcomplicated with a bunch of nonsense numbers. That’s why I focused on three areas: growth percentage/net additions, graph visual representation for an at-a-glance view, and standard deviation.

In sales, percentages and net additions are basics. That’s you’re bread and butter. Those are real numbers. If a professor can view how many more enrollments, they gained in an 10-20-day period, that can drastically change that professor’s or even the college’s approach to marketing that class. More signs, encourage students to take it when teaching other classes, etc. It’s quick numbers. The graph was the primary goal, because graphs are the best way to visualize data for anyone who may not fully understand the data. I focused on building a horizontal bar graph using Matplotlib to customize and plot the data. It pops right up when the program is run and also gets saved as well as sent to the user’s email. The third thing I had mentioned was standard deviation. Standard Deviation was something I picked up from one of my other classes. Standard deviation tells us the spread of the data. The larger the standard deviation, the more spread out our data is from the center while on the other hand, the smaller the standard deviation, the more the data is clustered around the mean. I found this very useful because when you produce the numbers for a course’s enrollment, you also know if there has been a large or small of overall change in the data. Standard deviation is measured by a number, when the number is smaller or closer to zero, it means the data (numbers) are all relatively close. This is great for an at a glance view. Because standard deviation is measured in a number, it’s not very user friendly. Therefore, I wrote an if-statement producing if there was a lot of change in the overall data selected. That also gets sent in the body of the email to the user. (Lynn,1)

The last pieces of piece of the program is having the CSV of the data, the graph as a PDF and the highlights or standard deviation message sent in an email. In one of my other classes, I developed a web scraping program that sends you all the new shopping deals on the popular sites via email. That project was extremely valuable because I love that email function. I was able to take what I learned then and apply it here as well. When you run the program, one of the first inputs the user gets asked to enter one or more email addresses separated by commas while also choosing not to as well. I then split this into a list of emails and use a bot account I made for testing to send the emails with its attachments. The key here is that the user can opt out of it if they’re looking at a few different sets of data, but if the user wanted to send it to a colleague or even all the employees, they could with minimal effort.

I utilized my skills to problem solve and write efficient algorithms. I really focused on constructors and ensuring I don’t overload a function with too much code so it’s easier for me or another developer to debug and rewrite. I also utilized my knowledge in data structures to make this program as one big class that can be used to make countless constructors if needed, parsing through large sets of data and reconfiguring it to provide a statistical analysis. This project taught me the importance of docstrings so that others can understand my code and help shave time off from figuring it out. If I were to do things differently, I would start with creating my main Class and \_\_init\_\_ for the program. I didn’t at first because I was focused on figuring out how to do things with the data, but as the code got longer and more complex, it would have saved me so much time. To add the class, I had to add “self” to certain variables which cleans up the code but also took more time to find them all and figure out the main variables that were going to be used. Next time, I would start with the main Class and its variables. This course enrollment program has taught me a great deal since it’s one thing to learn all these concepts but another to apply all that you’ve learned into one project. This gave me a great real-world example of what it’s like to be a developer. In the end, I developed an analysis that helps others, makes their lives easier and in turn help a business grow.

Works Cited

Bowne-Anderson, Hugo. “Groupby, Split-Apply-Combine and Pandas.” DataCamp Community, 26 Sept. 2017, [www.datacamp.com/community/tutorials/pandas-split-apply-combine-](http://www.datacamp.com/community/tutorials/pandas-split-apply-combine-) groupby.

“Group By: Split-Apply-Combine.” Group By: Split-Apply-Combine - Pandas 0.25.0 Documentation, pandas.pydata.org/pandas-docs/stable/user\_guide/groupby.html.

Lynn, Shane. “Summarising, Aggregating, and Grouping Data in Python Pandas.” Shane Lynn, 10 Apr. 2019, [www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-](http://www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-) python-pandas/.

Programmer, Python, director. What Is Pandas? Why and How to Use Pandas in Python. YouTube, 24 May 2018, youtu.be/dcqPhpY7tWk.