DS Internship Paper

This paper will cover the process that I've gone through during my internship. I will discuss the successes that I had as well as the issues that I wasn't able to overcome, and present the final result of my work. The topic that I chose for my internship was to help identify dangerous intersections and roadways of Pike County, PA utilizing publicly available datasets in a visually approachable way that can be easily understood by the layperson. The end goal of this project would be to present local lawmakers and DOT officials with easily understood data that would enable them to make decisions as to which areas of the county should be more carefully reviewed to see if additional signage or traffic control systems should be in place in those areas.

Initially, the scope of this project was much larger. My first dataset was downloaded from Kaggle, which is an online repository of datasets of varying topics from across the world. The dataset I took from Kaggle was a collection of traffic accident data that covered the continental United States from December of 2017 to February of 2019. This dataset was massive for a collection of data that was going to be visually represented, and in this I found my first challenge. The size of the dataset I took from Kaggle was 1.2 Gb, and had 3.5 million accidents in it. I tried several methods to parse this data, most of which were unsuccessful, but I'll describe them for completeness. First, I attempted to render the full dataset using the entire continental US as my map. The library that I used for this process is called **folium**. The folium documentation can be found at https://python-visualization.github.io/folium/. Folium enabled me to create a general map of the United States with the data points plotted on it that were covered in the dataset, and I considered this to be a huge step forward towards achieving my goal. I had a map with data points on it, and utilizing the parameters of folium I was able to center the map as I chose and was also able to zoom in on the individual accidents. This is where I hit my first roadblock. The rendering of this image, even with a dedicated graphics card and nothing else running on a powerful

computer took over 10 minutes to complete. Every time a user scrolled or zoomed on this image, the image would re-render the individual data points, making the end result useless for any practical application. With an end goal of having the image available on a website for anyone to look at, this was not a workable solution. The image itself was 110 Mb, but the rendering speed was horrible. In retrospect, I believe part of this issue was an unfamiliarity with folium, because there may have been ways to lock or pre-render parts of the image so that a complete re-draw wouldn't have to happen with every mouse move of the user.

In response to this issue, I attempted several different approaches to fix it. First, I spent time attempting to manipulate the folium parameters to produce a better result, with no luck. Then, I attempted to parse the initial dataset itself, so that I might be able to focus on one specific area of the country. This was a more successful approach. By dividing the dataset in to smaller pieces, I believed that I could create a menu that allowed the user to select the area of the country they wished to view data for, and then only render that specific data. As my proof-of-concept for this approach, I needed to isolate the data for one state and see how accessible the end result was. I chose to isolate Pennsylvania.

I had never parsed a large dataset before, so in order to do this I needed to learn how. My initial approach was to split the .csv file that contained all of the data for the country into smaller .csv files that only contained data for each state. This approach was at first unsuccessful, due to a combination of my lack of experience parsing datasets, and the way that the data was organized. The data used a unique ID for each accident that was assigned based on date, irrespective of location. When I split my dataset, I ended up with 35 separate files, but the accidents for PA were dispersed throughout them. I then focused on attempting to isolate the data for PA using a loop that created a new csv with entries that returned "true" if the column labeled "state" contained "PA". This approach was completely successful, but exposed an underlying weakness in my dataset. When I selected the accidents for PA, I ended up with a csv that only contained ~9,000 accidents, which seemed like an incredibly small number of

accidents for an entire state over a three-year period. To further explore this oddity, I then used the "county" column within the csv file to isolate only the accidents that happened in Pike County (where I reside) and the csv file only contained 36 accidents. Although Pike is a small and sparsely populated county, we have more traffic accidents than that in a given month, let alone 3 years.

This led me to do a thorough investigation of my dataset, to try and understand why there seemed to be a large number of missing accidents. After many hours of manually looking through the data and attempting to match it with known accidents that had occurred in my area, I realized that the problem was at *least* two-fold. First, the dataset that I pulled from Kaggle had already be parsed to remove any accidents that had not been deemed statistically significant due to severity. All "fender benders" and other collisions that had not resulted in injury or fatality were removed from the set. This meant that data I definitely wanted to include in my end result was completely unavailable to me. The other issue was that it appeared that only certain reporting agencies were used in the assimilation of the data. Specifically, for Pike County it appeared that only accidents which were responded to by the State Police were included in the set. The State Police are frequently not called to local accidents, as there are a multitude of smaller law enforcement agencies that can be called to handle them. In my county alone there are 20+ other law enforcement agencies whose records were excluded from this set.

This led me to realize that for my purposes, my dataset was not only incomplete, but useless. This brought my project to a complete halt as I attempted to figure out a way to continue. After much research into alternative online datasets, I discovered that Pennsylvania has its own data website with much more inclusive data on a variety of topics. Specifically, I was able to find a traffic accident dataset that provided a comprehensive list of all traffic accidents in the state from 1999 to the present. This dataset, although only focused on one state, was larger than the Kaggle dataset for the entire country. After a long download process, I had a 3.5 Gb file with over 3.5 million accidents in it.

Upon a careful examination of this dataset, I saw that not only did I have plenty of accidents to use in my project, but the amount of information on each of those accidents dwarfed what I'd had available in the Kaggle dataset. There were 30 columns of data for each accident in the Kaggle set, and there were 167 columns of data for each accident in the PA set. I had information on not only when and where the accident occurred, but also such comprehensive data as whether deer were involved in the incident, the age and injury level of all passengers involved, and even the precise weather conditions and whether traffic monitoring systems were present.

Given the far-reaching information available to me, I decided to initially focus my results on a subset of columns. I chose to select the Latitude and Longitude of the accident, the date/time/year of the accident, the severity level of injuries (if any) sustained, illumination conditions, and weather conditions. I utilized the pandas and csv libraries to perform this process. Although the end result was much more useful, I realized that the map would still likely be incredibly time-consuming to render. In order to avoid this, I decided to focus my results even more precisely to just the county that I live in, Pike County. When I completed this parse of the data, I ended up with a dataset that was only 1 mb in size and contained 12,699 individual accidents. This would turn out to be my final dataset that I used in my final map.

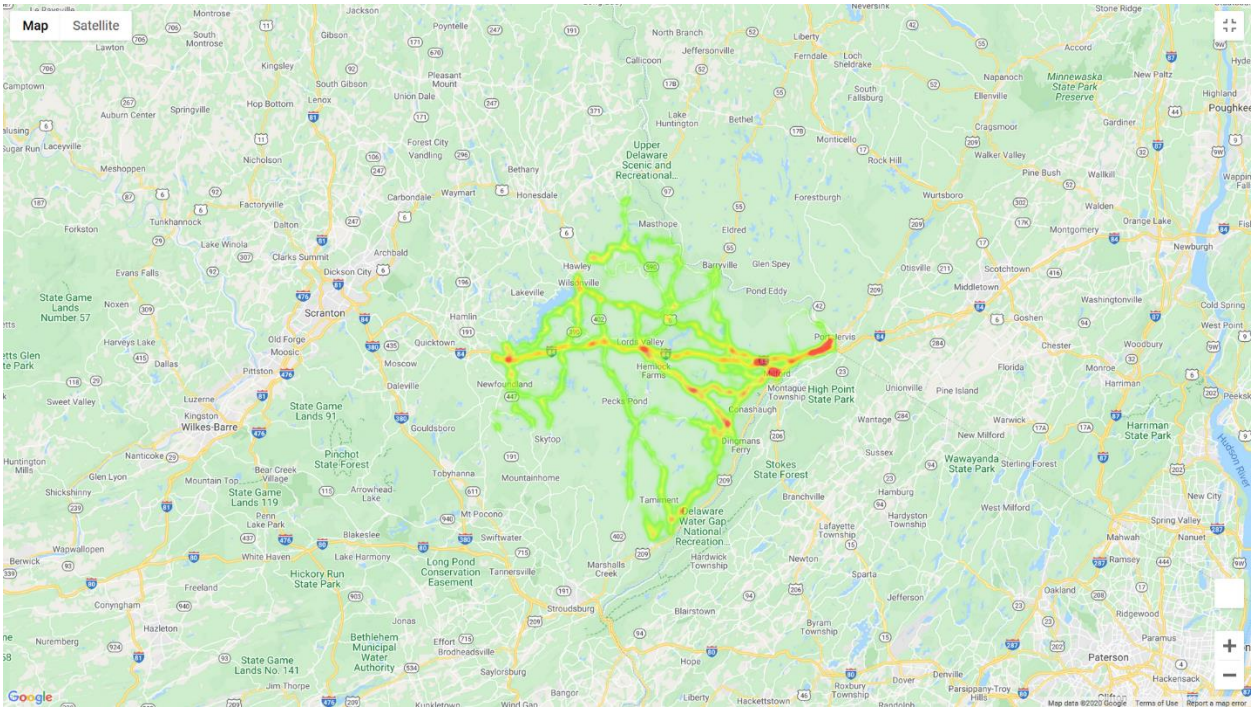
With my focused dataset, I initially used folium to give me a geographical map of the area. This produced a decent result, but it wasn't as detailed as I wanted it to be. I decided to do more research to see if there were any other APIs or libraries that might be useful. I found my answer in the Google Maps API. This API is by far the most comprehensive, flexible, and commonly used geographical data plotting API in the world. In order to use the Google Maps API, I first had to create an account for myself and submit my credit card information. There was no initial charge for the service, but Google provides a specific amount of data access for free and then will charge you after a certain point is reached. This cap resets itself on a monthly basis. For any normal, unpopular data map this cap would never feasibly be

reached, but for a data map of (for example) real-time polling numbers in the Presidential election, hits could be in the millions and that's the point where the paid-tier would kick in.

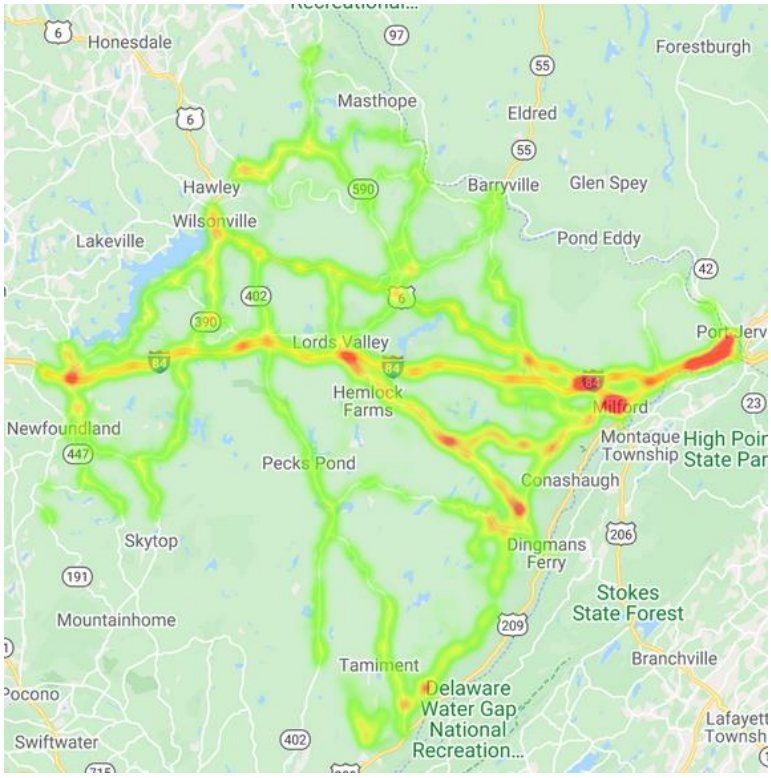
After the initial set-up process was over, I began reading the documentation for the library I would be using with Google Maps – an excellent, open-source project called gmaps. Gmaps provides a wealth of options in varying levels of complexity for data scientists who are attempting to create visually beautiful geographical data. I elected to use a heat map for this process, as I believed a heat map would best represent the data in a way that would be instantly recognizable to anyone who was attempting to see where the accidents occurred most frequently. Heat maps use increasing color intensities as more data points occur in the same locations. Gmaps allows you to choose any color gradient that you'd like for this process, and I chose a standard gradient of green-red. Individual data points would be represented by green dots, and as those dots shared the same location, they would (based off a weighting system internal to gmaps) begin to deepen in color until the highest areas of occurrence become represented by the red color.

Color choices in data science are incredibly important. It is easy to get lost in the mathematical details of data while forgetting that the reason much data evaluation is done is to allow business and safety decisions to be made by people who have no familiarity with data science at all. Giving someone a list of numbers is only useful if the person receiving it is able to understand what those numbers mean. Properly represented visual data imagery will always be more easily accessible to the layman. In this case, using color gradients that are commonly associated with traffic/safety situations (Green means go, red means stop, green means safe, red means danger) can immediately draw the audience's eye to the areas of greatest concern without having to explain anything. Even a child would be able to look at a heat map and understand that the red areas would be the "bad" areas.

Here is the final heat map for Pike County, PA showing all of the accidents that have occurred on roadways and at intersections from 1997 to the present day:



And here is a zoomed image for additional detail:



This is the point at which I was forced to end my research into this topic – far earlier than I think was necessary in order to truly make the greatest use of the data I had available to me, but I learned an incredible amount even so. I began this process with the intention of accomplishing a specific list of goals, and although I did accomplish many of those goals, there are still many more that remain incomplete. Although I was able to create an accurate heat map which visually depicts my data in a very approachable way, these are the goals which I would like to see this project accomplish in the future, should I have the opportunity to work on it again:

1. The data should be weighted in a way that allows the user to select varying levels of injury. This would enable them to see which areas of incident cause "fender benders" and which areas cause fatalities. This would enable a priority list to be put in place to make sure the areas in greatest need are the ones that are handled first.
2. Gmaps is flexible enough that I could create a GUI which would allow any number of datasets to be used and instantly generate a heat map even if the user doesn't understand code at all. With properly designed input boxes, a user could upload any .csv file and select the columns that are of greatest interest to them and the map would be instantly created based on their criteria. They could then try different configurations of the data to get the best result possible.
3. This entire project should be hosted on a website. Having an online tool to accomplish easy heat map creation would allow users who otherwise wouldn't have the ability to create meaningful data representations.
4. The data visualization from my project should be taken to local lawmakers and Department of Transportation supervisors so that when they are selecting the areas of my county most in need of repair or modification, they would have the correct data at their disposal to make those decisions.

In closing, I'd like to say that the possibilities of this project, although not endless, are far-reaching. This is a tool that can make immediate impact if properly used. I am very happy that I was allowed the opportunity to explore this idea in my own time, and I have learned a remarkable amount during the process. When I began, I'd never written a line of Python in my life. Now, I understand not only the syntax of the language, but so much more. I learned Python Notebooks, PyCharm, the proper uses of virtual environments, the libraries and API's that I mentioned earlier, and I've even selected Python as the language that I will be using in my job interviews once I graduate. Python is beautiful in its simplicity, and enabled me to focus on solutions rather than syntax. I thank ESU and Dr. Che for this internship format, as it was perfect for my busy life while still helping me to gain an incredible amount of valuable knowledge for my future as a programmer.