Final Project Reflection

Jee Hyun Kim & TJ Kim

0. Alterations to Project Proposal

After working on the project, we decided to change our MVP and final output. Our initial goals mostly included:
- Creating a single axis spectrum graph with a selectable characteristic to map colleges onto it.

We wanted to alter that goal into something else:
- Mapping the final college locations onto a map of the US

For the following reasons:
- People did not understand when we attempted to explain our initial idea.
- It allowed us to emphasize the data more graphically and visually appealing.
- Helped us learn Bokeh, a new a slightly challenging library to implement initially.

1. Assessment evidence and interpretation

(Copied from the README from our github repo for DataScienceCTW)

Below are the files you should check for evidence of work.

Development of algorithms and documentation

new_data_csv_book.ipynb: Ipython notebook with walkthrough and documentation through taking initial data csv downloaded from collegescorecard, and filtering out data columns we deem unuseful. Output is "new_college.csv" file, with filtered out columns.

alg_finalized_1_1.ipynb : Ipython notebook with walkthrough and documentation on algorithms used to filter out colleges through user preference input.

Exploring how the college major changed over the years

majoroveryears.ipynb: Ipython notebook with data exploration and visualization of colleges over 1996 - 2013. There is a focus on growth and decline of college major rates throughout the years. Although we altered our minimum viable product, we still wanted to stay loyal to our original goal and graph data on axis.

Final Code

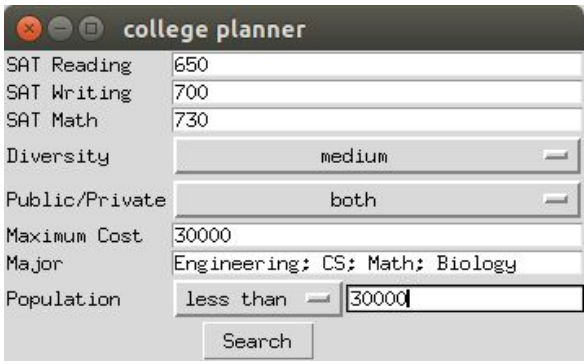
college.py: Equivalent to alg_finalized_1_1 in py file. This allows other py files to import it and use it. Holds algorithms that filters colleges out based upon user input.

college_map.py: Uses Bokeh library to generate map with selected schools printed onto map of US.

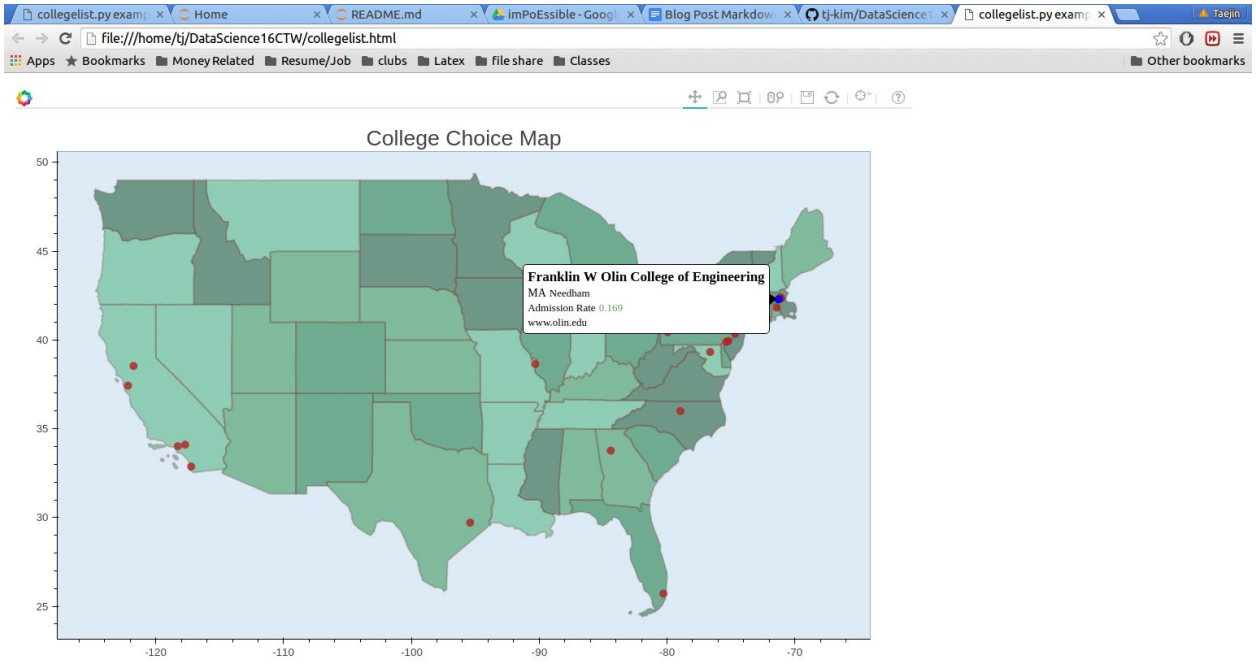
college_gui.py: Uses Tkinter library to build a graphic user interface for users to to input their preferences and information. This file imports college_map.py and college.py for final using.

Final Output: Example run of our tool

Here we use the college finder ourselves. Initially we run with python college_gui.py. This opens up a tkinter GUI, college planner. The users input their information/preferences accordingly.



After we hit the search button, a new tab in the internet opens! It is a map of the US with the colleges marked on it as dots. From here, users can hover over the dots (which turns the dot blue) and to get more information about each college such as name and website. An example screenshot is presented below.



Entering different inputs and searching again opens up another browser with new set of colleges corresponding to the new inputs.

2. Changing the world: Do you think your project has the potential to change the world? If not, why? If so, what are the next steps to make this happen?

We believe that this project had a positive impact for our specific user group. Although we have used many college search tools, none of them were exactly what we wanted. It captured some elements of what we are looking for in school, such as choosing the majors, but not some others, such as an option for diversity or choosing a variety of majors we are interested in. Also, none of them had a very visual output where students could have a graphical understanding of how far they would be traveling. This tool also delves deeper into other student preferences such as majors and racial diversity. Although the tool does not give the student lots of data on each school, we intentionally did that to prevent the overload of data. The student will instead receive a cursory introduction to different schools where they can research further into.

The next step to enhance the impact on the world would be to publish the work and bring the separate components together. Currently, the tool can only be run from a python script, and the interface, output, and further graphical analysis of college major over the years are all separate files. Bringing this together and giving the user one compact page may enhance experience greatly. Another step we can take is to make it into a webapp using js.

Learning goals: Did you learn the things that you wanted in this project? If not, why? If so, why do you think you were successful?

Previous Learning Goals:
Learning to create effective visualization which is aesthetically pleasing
Develop own algorithms to rank schools along our single axis

We did not learn all of the things we wanted, mainly due to the change in our minimum viable product. We did learn how to create a visualization that is pleasing using bokeh. This process was overall manageable, as the bokeh library, had very handy tools available for use. We were not able to create the final output on a single axis graph, as we decided to map it on the US map. However, we did develop an algorithm to filter out colleges based upon user preference. The filter process worked fairly well, as we ran tests with different users, and many got the college pool they did actually prefer and may have applied to in their senior year of high school. Overall, the project was a great learning experience, and exposed us to a part of data science that was not strictly connected to machine learning.

The success: entering our high school preferences / information generated a list of colleges with Olin in it.

Self Assessment of Team (Reflection and Team Analysis)

Category	Score	Score Explanation
Usefulness of the visualisation is to high school students, given the limitation in data	9/10	We created a very cursory, and introductory guide to the college applicant. Also an in depth analysis of majors over time gave insight on trends in academia. We did not stick to our initial MVP and pivoted, although still having components of both data visualization and tool usefulness.
Code readability	9/10	We think that the code generally has high skim value. However, it could be improved for college_gui.py. It can be arranged and documented to be more readable. Documentation for college_gui was not as thorough as others, as user can look at documentation of Tkinter, rather than our code.
Code efficiency	4.5/5	Whenever we saw repeated code, we developed a function to streamline the process. The one exception for this was when we used Tkinter to develop the GUI. Each input instance in the GUI was similar, with slightly unique attributes, so it was difficult to implement a common function and the code got long and repetitive. We also trimmed down the original dataset to load faster and to streamline the process.
Code documentation	5/5	We pair programmed and after the pair programming, in order to ensure we were on the same page, the other member who did not code wrote the documentation for the functions or the general code. After we were done with coding, we went through the documentation to ensure that the codes were documented
Writing Quality (New Added Category)	10/10	From the project proposal through to the ipython notebooks, the github README, and the final reflection, our team has had a great emphasis on writing. We believed that communication of our process was key to those who wanted to look at and understand our work later. Our logic is very thorough, and user friendly.
Teamwork	10/10	It was a great pair programming experience. We had great communication, shown in how we both divided our work and worked separately at times, but also came together for pair programming when we were developing new code and ideas. We enjoyed working together as a team. Even though we had some different ideas on what we want to do, we discussed and collaborated well.