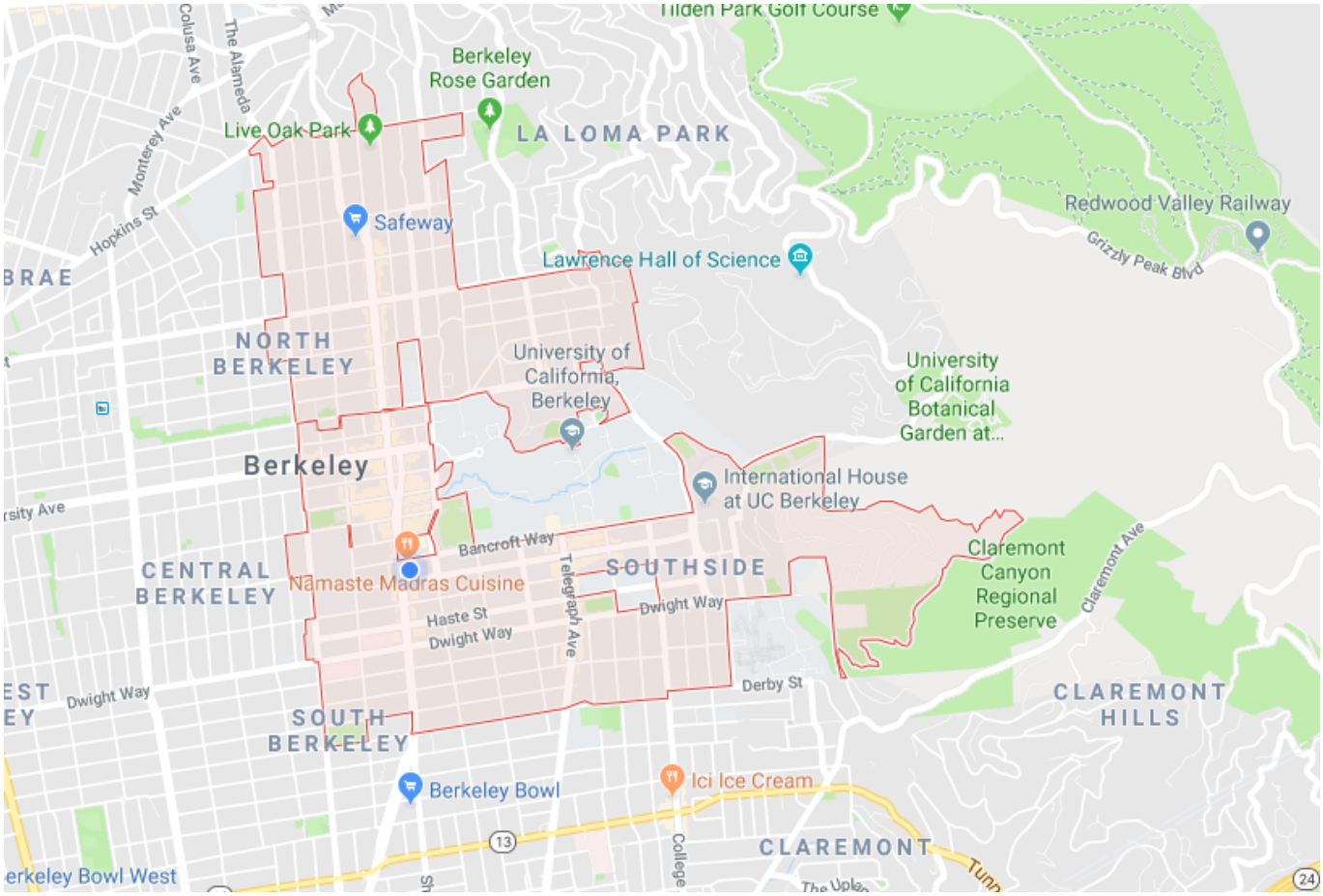
ESPM155AC Final Project

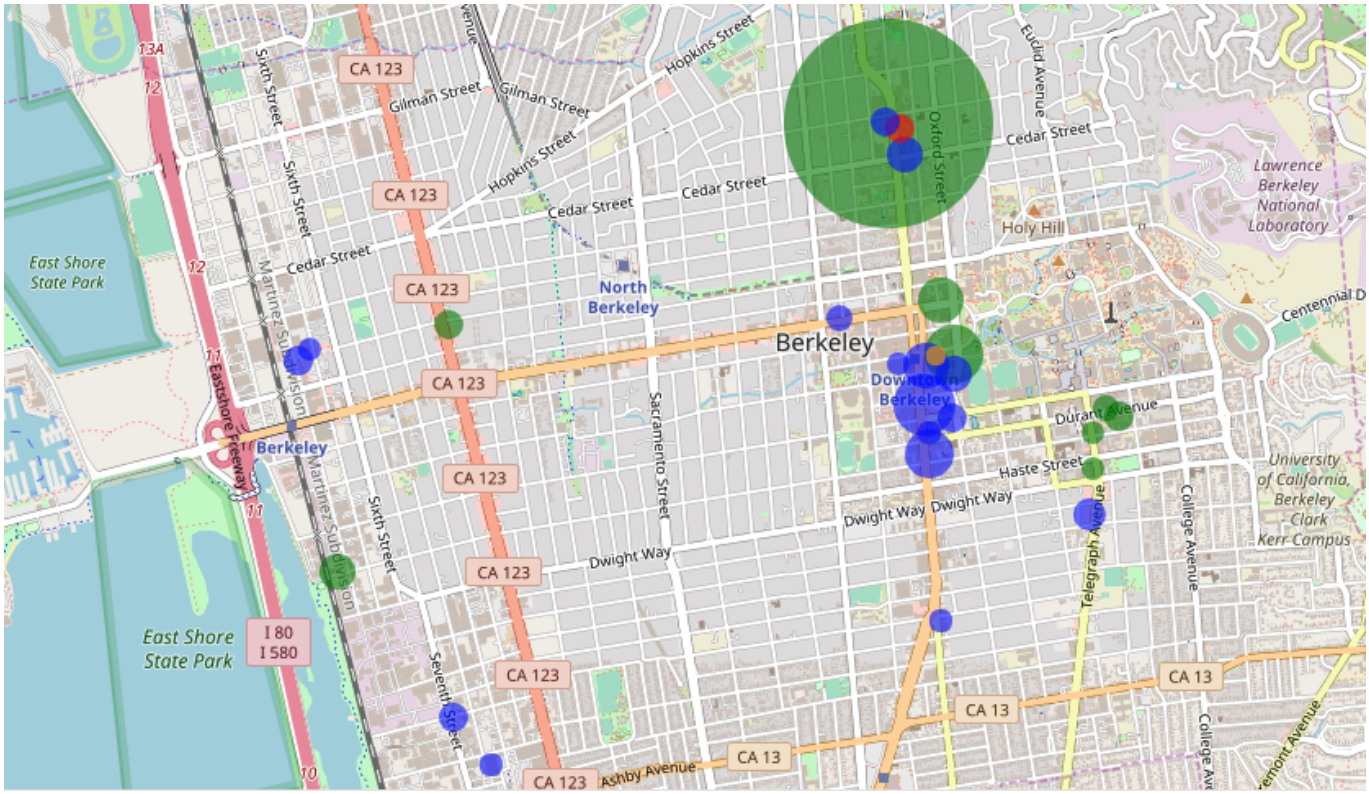
This project was an attempt to juxtapose the differences in price and healthyness from where UC Berkeley students typically lived and outside of where those students lived. We deemed the "student-living" areas to be the zip codes 94704 and 94709 whereas the rest of the zip codes corresponding to Berkeley was the population where we did not expect students to typically live.



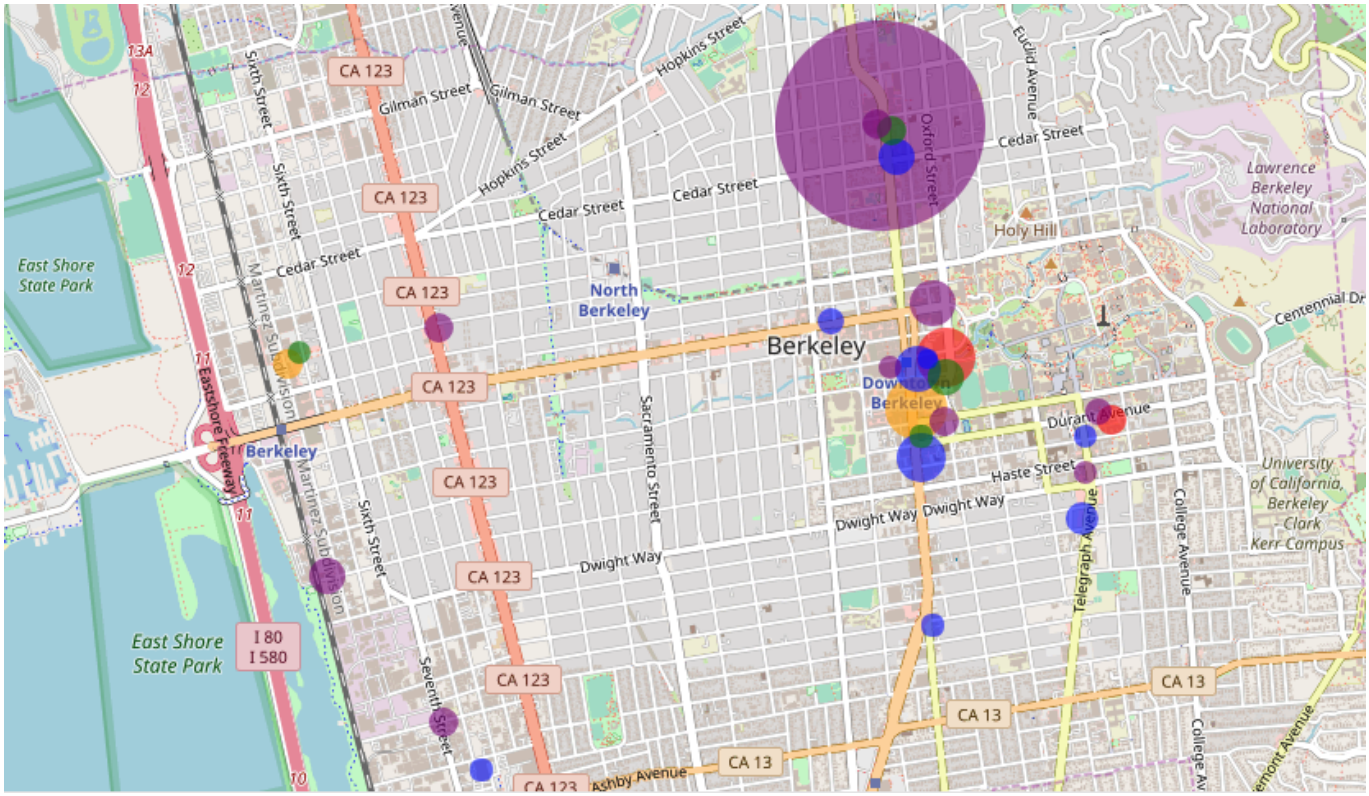
My Part

I was tasked to try to show that the prices in "student areas" were cheaper than those outside of "student areas". Since I was the only one tasked to do this, I did not have the time to collect a large enough random sample to represent the population. So I used Yelp's API to gather the data, which made my sample essentially a census, however we sacrificed more granular data since we had to switch our units from USD to Yelp's '$'. It might be worth noting that this was the first time I ever worked with an API. 1. I used the data.py file in the Projects folder to gather all the data from the Yelp API. I deleted my secret in the file since this file is for proof of work. This allowed me to creat all the CSV's for restaurant names, prices, and locations for all but one of the maps our group made. 2. I then created 4 statistical models which can all be seen here (and their corresponding histograms) in this iPython Notebook: 2 for showing the prices in student areas was lower and not due to random chance and 2 for showing there were a greater proportion of cheaper restaurants in student areas and that this was not due to random chance. **The first two of these models were**

presented. 3. I also created a formula to detect the most popular restaurants based on their rating and number of reviews on yelp. This was to help practice with API's, make something cool with maps (worth noting I've never made a map before in my life), and help my other team members by creating a CSV for popularity so they can also make their own maps. This can be seen through this iPython Notebook. Unfortunately, this does iPython Notebook does not show images (unlike the previous one because it's a different application), so I'll paste the maps of the 27 most popular restaurants I made below. The radius represents the popularity factor of the restaurants. The first map is colored by prices (green = 1 '$', blue = 2 '$', orange = '$', and red = 4 '$').



This next map is colored by healthiness. To determine the healthiness factor, we selected the most expensive and least expensive main items on the menus. We gave each item 2 points, one based on number of fruits and vegetables and the other based on how fried the food was. 0 for no fruits or vegetables, 1 for some fruits or vegetables, 2 for many fruits or vegetables. 0 for lots of fried food, 1 for some fried food, and 2 for very little to no fried food. The sum of the four numbers was the restaurants health metric. The other group members calculated this number for these restaurants. The colors are as follows: green = 8, blue = 7, purple = 5-6, orange = 3-4, red = 1-2.



These maps were not used for our presentation or deliverable in favor of prettier maps made by other group members and more widely accepted map-making tools. These maps were created using Data 8's Data Science Library which can be found here: http://data8.org/datascience/. 4. I also attempted to predict the price of a restaurant when given the location of the given restaurant. I used a linear based approach which calculated the distance away from UC Berkeley and predicted the price given the previous data we had before. However, this approach would never work because prices would asymptotically approach some fixed point (not considering outliers like Chez Pannisse). Furthermore, the data was not granular enough to do a regression analysis like this. I made the data less granular by grouping all the restaurants by distance from the middle of UC Berkeley and taking the average prices (still using Yelp's '$') which led to a better fit. However, the problem of asymptotics still arises. I did not have time to attempt a logistic regression unfortunately, but this can be added as future work. All the work for this regression attempt can be found through this iPython Notebook. **This was not shown in the presentation or deliverable.**