Final Report INFO 4310 HW 4

Melissa Avila, Eric Li, Benjamin Stevens

Dataset

We chose to use the Boston Yelp dataset, provided in the assignment description. We originally chose the Boston Yelp dataset because we were interested in the best ways to visualize/recommend a topic as subjective as food (plus generally preferred the topic over the home markey). We specifically chose the Boston dataset over the other Yelp dataset because of the inclusion of the review snippets, we didn't want to limit ourselves in case the reviews ultimately became interesting/relevant to our visualization.

Since the dataset lacked price range information, we extracted that information ourselves from the Yelp website (to be used in our filtering and recommendation weightings).Other general weaknesses of the dataset include a general lack of restaurants and records, the exclusion of all reviews (albeit, snippets were provided), and a lack of information regarding restaurant hours, whether a restaurants allows for carryout/delivery, and other restaurant-specific info.

Our visualizations cater to people who may be unfamiliar with restaurants in Boston (and employs a sort of AirBnB strategy of exploration for our main visualization) and to those who have already eaten at certain Boston restaurants and are seeking recommendations for other based on where they've been. More specific needs of our consumers we are hoping to address include finding restaurants in their vicinity/in a general area, exploring restaurants similar to ones they are familiar with, and only viewing restaurants based on the factors (price, rating, etc) and filters they find most important.

Challenges

For data explorers trying to view the data in the csv format it originally came it, it's extremely difficult to conceptualize the relative location of different restaurants as well as how they compare on different factors such as pricing, number of reviews, and rating. Moreover, filtering would be cumbersome, even if loaded into excel, and it would be more difficult for the users to understand the entirety of the dataset, as they would be limited to a few records at a time.

If users were simply given points of restaurants on a static map, than of course point density would become problematic, but also it would be difficult for users to see changes and apply filters to the dataset.

Interaction Affordances

1. **Interaction:** Dynamically update restaurants on the map that match a user's indicated filter.
    a. **User Need it satisfied**: this interaction expedites the user desire for exploration as well as allows users to search the data according to their own preferences and tastes/wants. The responsive nature of the filters indicate to the user that their choices and preferences matter and are constantly being taking into account.
    b. This interaction addresses the challenges associated with the static map aforementioned in the previous section.
2. **Interaction**: Click on a restaurant on the map to get more details and see a forced layout with other recommended restaurants (based on the user's choice).
    a. **User Need it satisfies:** This interaction satisfies the user's need to drill-down on a piece of information and retrieve more detailed information on a restaurant. Moreover, the incorporation of the second visualization strengthens the user's exploratory whims.
3. **Interaction:** Click on a restaurant from the dropdown (on force-directed layout vis) and see a forced layout diagram displaying recommended restaurants based on the user's choice.
    a. **User Need it satisfies:** Find restaurants similar to ones users have visited. This interaction caters both to users who simply want to explore restaurants and to users who want recommendations based on places they've already been to, without having to do extensive research online.
4. **Interaction:** Changing the weightings of the different recommendation factors in the force-directed layout vis
    a. **User Need it satisfies:** This interaction provides the user with agency to decide what is most important in a recommendation. For example, users can specify exactly how much more important rating is in the recommendation decision compared to other factors such as price similarity (when recommending restaurants based on the user's choice).
5. **Interaction:** Click on a point in the forced layout diagram to see details about the restaurant.
    a. **User Need it satisfies:** Point on a diagram are useless without more contectural and specific information. This interaction allows the user to see information about the restaurants recommended so that they can explore/pursue recommended restaurants.
6. **Interaction:** When viewing the details of a point, the user has the option to make the point the new center of the forced layout by clicking "Make this the Center."
    a. **User Need it satisfies:** This interaction satisfies the users curiosity in exploring restaurants they are unfamiliar with as they can continue to explore restaurants with more unexpected category combinations.

Visualization Description

The purpose of the visualization is to allow users to explore restaurants in the Boston region, and to recommend restaurants based on other restaurants the user has eaten at/enjoys. The primary parts of the visualization are a map of Boston with points representing restaurants, filterable by a number of factors, and a forced layout diagram with the center point being the restaurant the user is familiar with and surrounding points the restaurants the tool recommends based on the user's choice and weighting factors. Recommendations in the forced layout are determined by weighting category similarity, rating, number of reviews, and price-point similarity into a recommendation function. The weightings can be changed and specified by the user, but the default reflects what we believed was an accurate representation of significant factors and ratings in restaurant similarity/likability.

Specific Features

- Move around the map to explore restaurants in the Boston region
    - Restaurants can be filtered by price and rating
        - When filters are applied, the map dynamically updates
    - Clicking on a neighborhood on the map will zoom in on the neighborhood
    - Clicking a restaurant on the map will change the visualization to a force-directed layout diagram (with the option to switch back to the map view on the upper left corner)
- Select/type a restaurant in the dropdown to produce a force-directed layout diagram displaying restaurants recommended based on the user's choice
    - The force-directed layout diagram shows restaurants recommended to the user based on the user's initial choice/selection
        - The center point on the diagram is the restaurant the user chose for which recommendations are being made against
        - Distance of surrounding nodes is determined by the degree of recommendation, closer nodes are more highly recommended to meet the user's interests and tastes
        - Color of node is indicative or rating (since most restaurants are rated a 4 or a 5, there is overall little color variation)
    - Hover over a point in the diagram to see the restaurant name
    - Click on point for more details on the restaurant
        - Clicking on a point provides details concerning the name, price range, website, location, average rating, and number of reviews
    - Make a point on the diagram the new center by clicking "Make this the Center" in the point description
    - Change weightings of recommendation factors
        - The users can update the weightings of the different factors used in the recommendation function: price and category similarity (to the chosen point), ratings, and number of reviews
    - Switch back to the map visualization by clicking it in the upper left-hand corner

Issues and Trade-offs

1. Map Visualization
    a. May appear slightly dense for consumers who are unsure of the type of restaurant they want to eat at and who are completely unfamiliar with the Boston area. And, while this design works for the small dataset we have, this design does not work for very large datasets as it may lead to an overpopulation and difficult to interpret clustering of points.
    b. For consumers with sight disabilities that use screen readers, the tool is nearly impossible to use.
    c. The map responds a lot slower on some computers/web browsers than we planned, so this may be frustrating for some users.
2. Force-Directed Layout Recommendation Tool
    a. In the force-directed layout having users pick from a restaurant in the drop down caters to users who already know what restaurants they like and requires the users to have eaten at a restaurant in Boston (which appears in the dataset)
        i. However, this aspect of visualization was made specifically to cater to users who have already eaten at a Boston restaurant because it relies on the information associated with this restaurant to recommend other, similar, restaurants
    b. The different restaurant options may be overwhelming when viewed in the tiny drop-down and it may also be hard to locate a specific restaurant if the user does not remember the exact name/type
    c. This visualization is also more difficult to readily interpret it is an uncommon visualization the average consumer does not often encounter and thus may require more training and time to understand than a user is willing to tolerate.
        i. To accommodate for this, we included a description under the visualization.
3. Had we more time, we would have connected the filters in the map visualization to the recommendation visualization so that user's could more exactly specify what type of restaurant they were seeking and sought to be recommended
4. Moreover, when creating the filters we ran out of time and ran into a tons of issues so had we more time, we would have included filtering by neighborhood and number of reviews

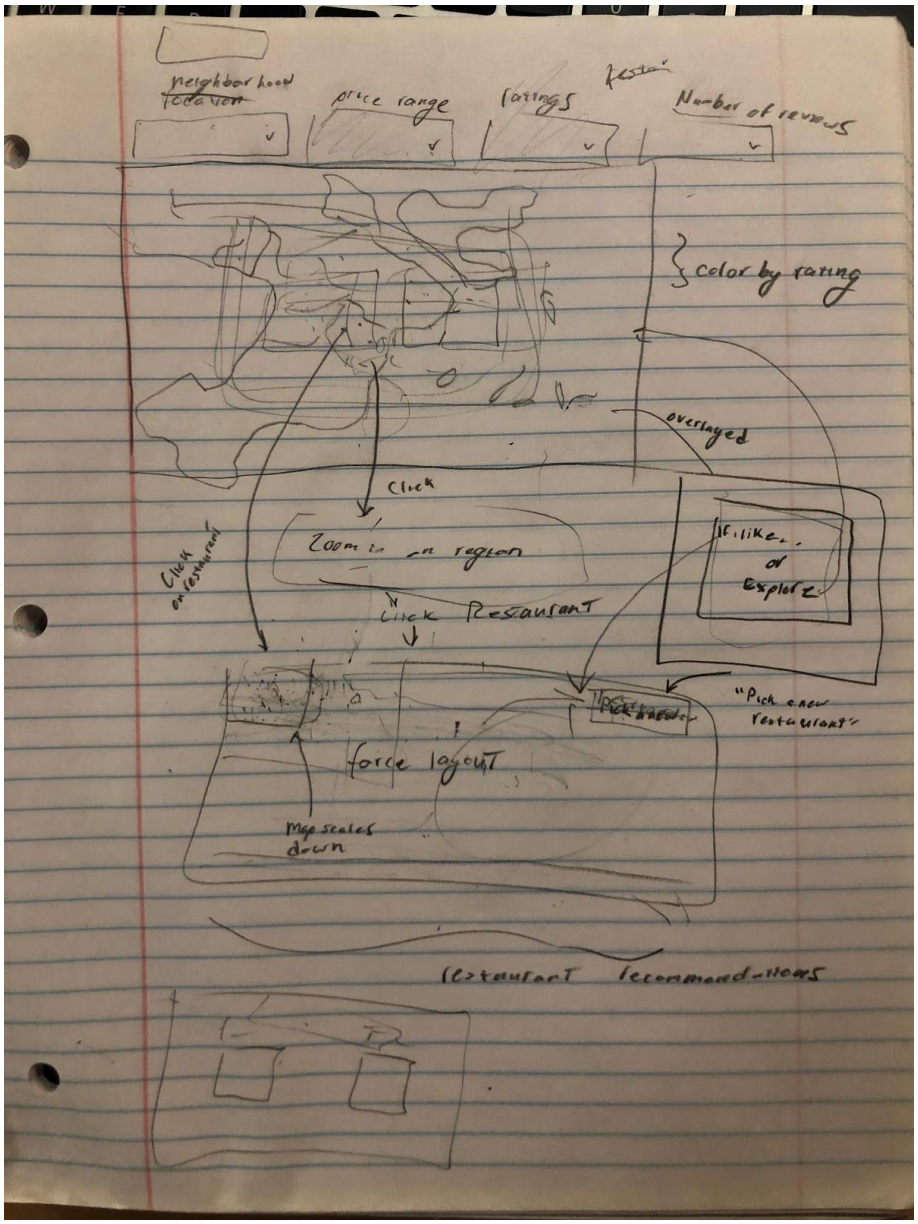
Development Process

Originally, we were going to center our project around the second, force-directed layout visualization which was inspired by the subreddit “If you like blank” (link to subreddit). Essentially we wanted to create a recommendation system that bases its analyses primarily on existing user tastes. Moreover, originally during the ideation process, we planned to have a second force-layout diagram visualization in which users could explore the different category tags in a forced layout manner so that they could see all the restaurants in the dataset. While, this second force-directed layout idea was cut because of fluidity and implementability, we developed the second map-based visualization to satisfy a more exploratory approach to the dataset. While brainstorming the map idea, we were unsure of whether to keep the first force-directed graph for spatial reasons, but then realized we could connect the two visualization by directing the map visualization to the force-directed graph visualization when a restaurant point was clicked.

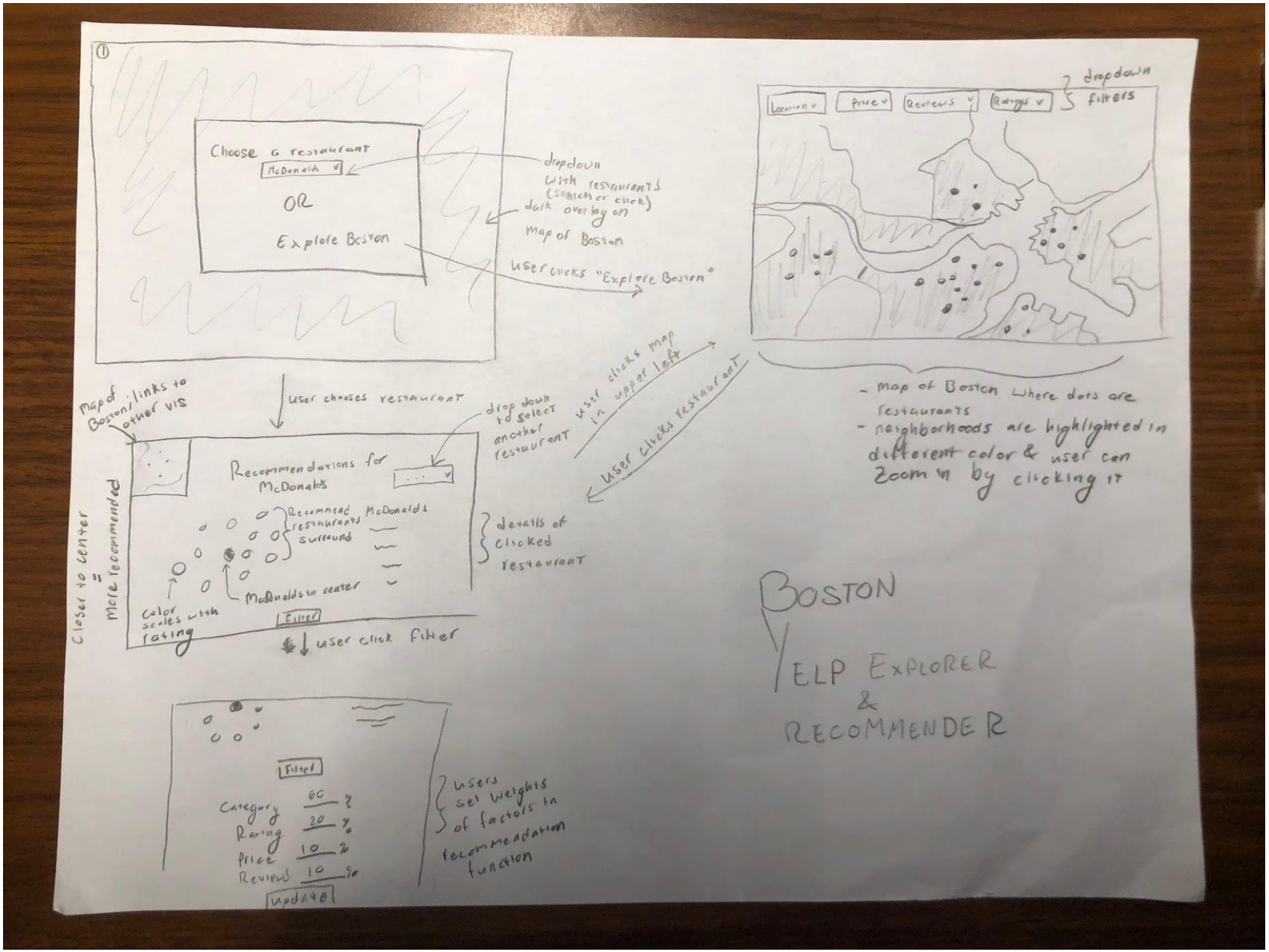
Our second ideation challenge was easing user access (as originally our map idea made the force-directed graph visualization visible only after a restaurant click) to these two very different approaches to the dataset, which we decided to solve through the introduction of a modal (opening upon page load). The modal gave users the option to choose between either the force-directed graph or map visualization after the initial page load, so that way we would cater to users who either wanted to explore, or receive recommendations based on restaurants they were already familiar with.

Our last challenge was facilitating switching between the two visualizations in a fluid manner. We discussed putting the force-directed layout visualization in a modal, but decided it would look unnatural. Ultimately we decided that when the user selected a restaurant on the graph, the map would shrink (yet still remain visible) in the top left corner in the graph, which the user can click to return to the map visualization.

Below is our initial storyboarding of our designs combining the force-directed layout recommendation visualization with the map exploration visualization:

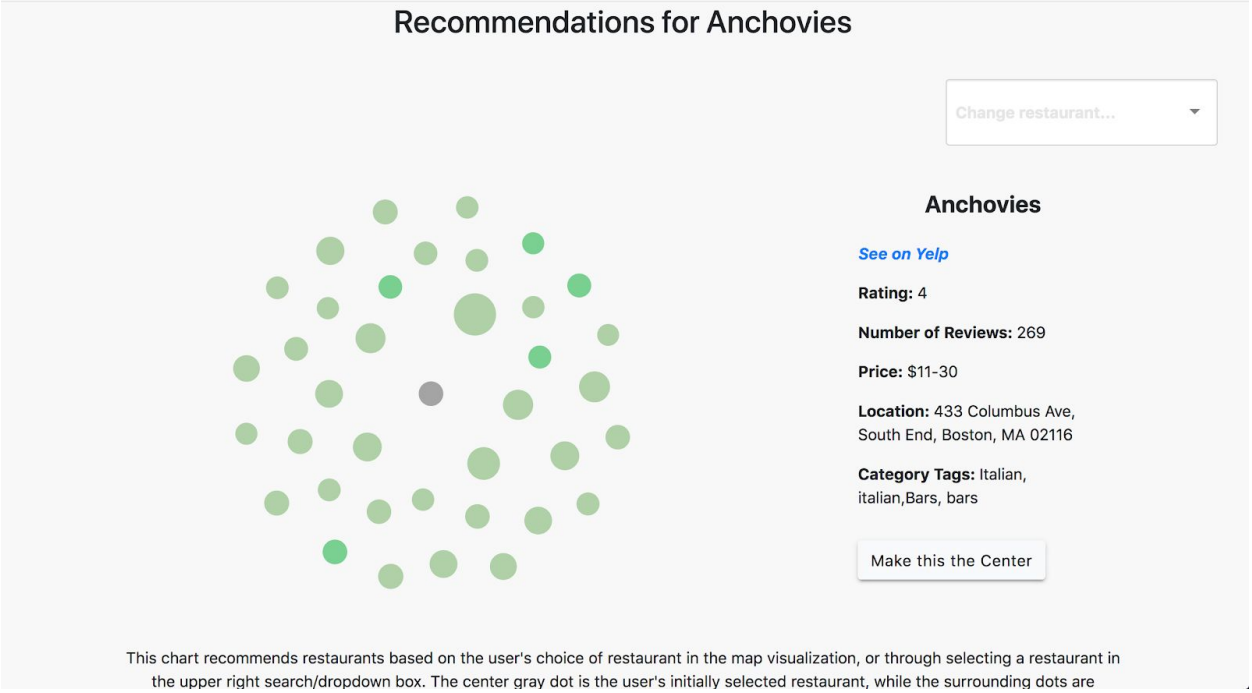

neighborhood
location
price range
ratings
Number of reviews
color by rating
overlayed
Click
Zoom in on region
Click on restaurant
Click Restaurant
If i like..
or
explore
"Pick a new restaurant"
force layout
Map scales down
restaurant recommendations

And a cleaned-up storyboard:



More Detailed Process

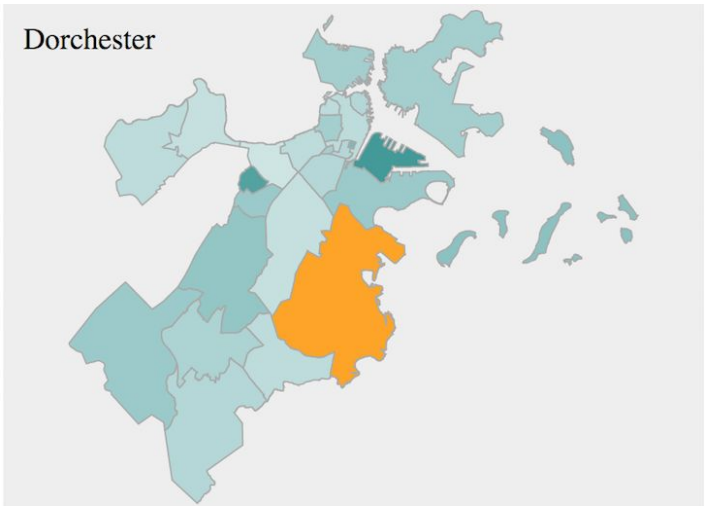
As mentioned above, we originally started with the idea of the force-directed layout and completed that before the map portion of our project

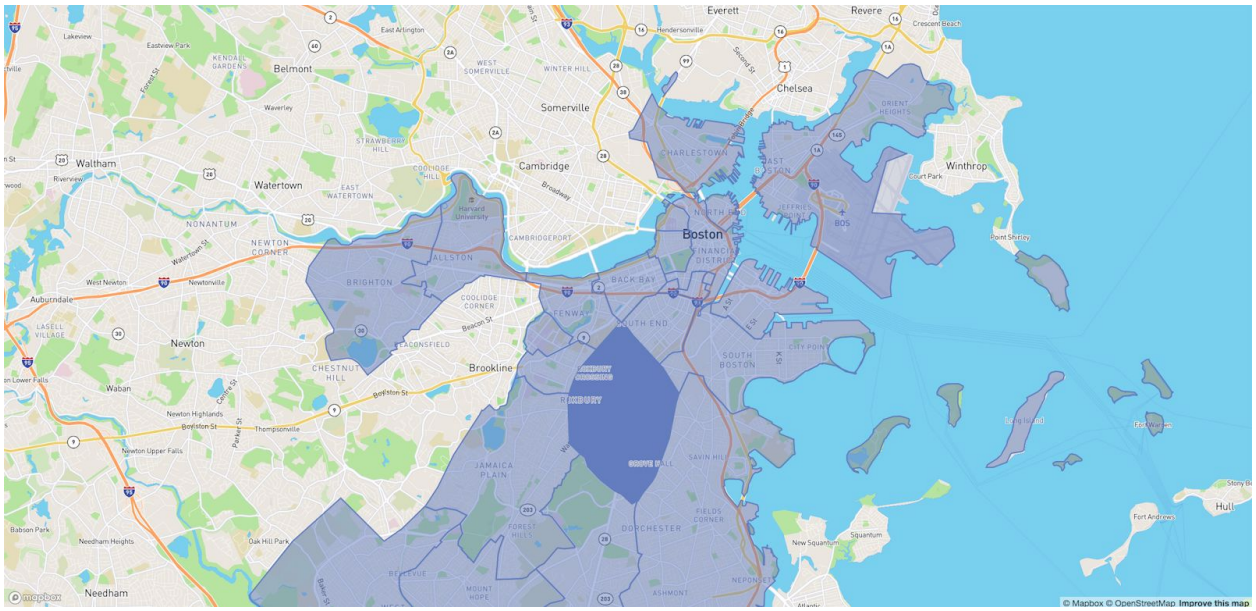


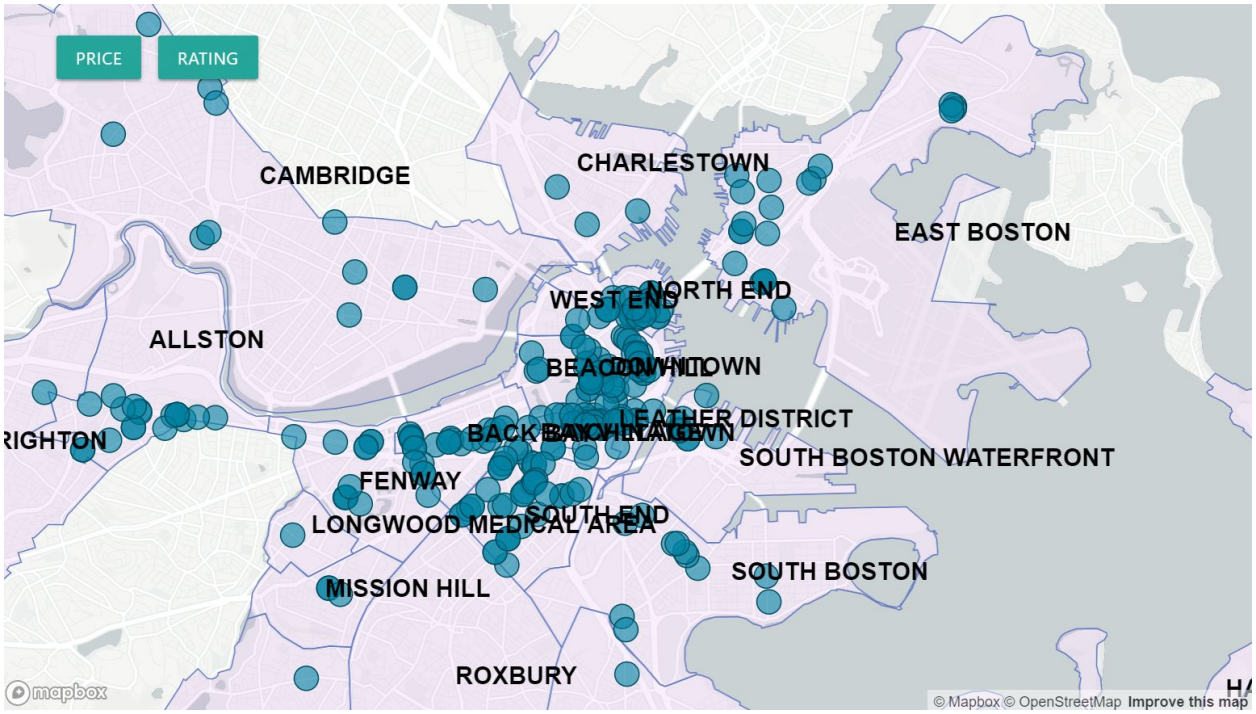
In the development of this visualization, our primary difficulties involved getting the force-directed layout points to transition smoothly and not stutter on the onset. A lot of fiddling was done with simulation.alpha and the force charge to make the points appear/transition smoothly and accurately.

After realizing we needed a more exploratory tool, we decided to create a map of Boston with points of the restaurants overlaid, filterable by price, ratings, reviews, and location. Early on we encountered a lot of problems using MapboxGL and overlapping D3 onto MapboxGL. Early map iterations used only D3, emphasizing exploration of neighborhoods. Very close to the deadline when we were combining the two visualizations, we underwent A LOT of integration problems as the map portion was coded in D3 version 3 and the force-directed layout was in D3 v4. Moreover, indentation problems with the code and sublime made it very difficult to know when one function ended and the other started.

The images below are the design changes we made, in order, as we were developing and designing the map:


Dorchester


Everett
Revere
Chelsea
Somerville
Belmont
Waltham
Watertown
Cambridge
Boston
Winthrop
Newton
Brookline
Needham
Hull
© Mapbox © OpenStreetMap Improve this map



Work Contribution

- Melissa Avila
    - *Scraped, filtered, pre-processed, and parsed data* - 4hrs
    - *Developed forced layout diagram* - 6hrs
    - Added details (& JS functionality) and movement to force layout diagram - 1.5hrs
    - Integrated information into force layout diagram - 2hrs
    - Added drop-down - .5hr
    - Significant styling - 1hr
    - Wrote recommendation function integrated into force layout .5hr
    - Added ability to change weightings of different factors in recommendation function - 1.5hrs
    - Writing Rationale - 1.5hrs
- Eric Li
    - Built final filters for map points
    - Made map dynamic with shrinking
    - Added final zoom to selected neighborhoods
    - Saved the day with an iframe - 10mins
- Benjamin Stevens
    - Added initial ‘filter by’ dropdowns for category and neighborhood - .5hr

- Collecting, formatting, and learning geojson - 1.5hrs
- Initial exploration of map as second vis through Mapbox and D3 - 5hrs
- Initial selectable neighborhoods and zoom to fit - 1.5hrs
- Added neighborhood labels in map centers to align with GeoJSON - 1.5hrs
- Added hover effects to restaurant map points - 1.5 hrs
- Merging code - 1hr