1. Project code:

Link to GitHub repo: https://github.com/yanzhuo2001/SI507_final_project

README file:
https://github.com/yanzhuo2001/SI507_final_project/blob/main/README.md

Required Python packages: Flask; pandas; requests; folium; networkx; scikit-learn.

2. Data sources:

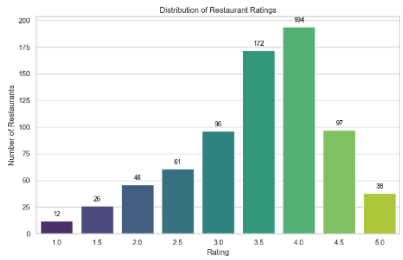
Origin: Yelp's API (api_key = 'xjBZe7_IvwLXcaUPxWrfcvMT-Xw1rUxNlx4PSdvHx6jSzUpYYZyHLo3SzXGIlWR8261_pGwuoTo8hhAZEeioWQIhB9k1freKyv8vlMMSihaQZyakMhlPHXO_c_ZXYx'; url = 'https://api.yelp.com/v3/businesses/search')

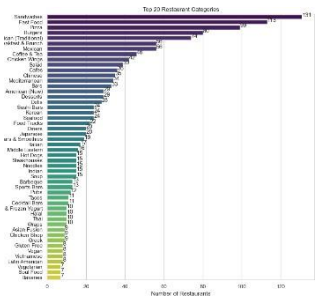
Data format: JSON file

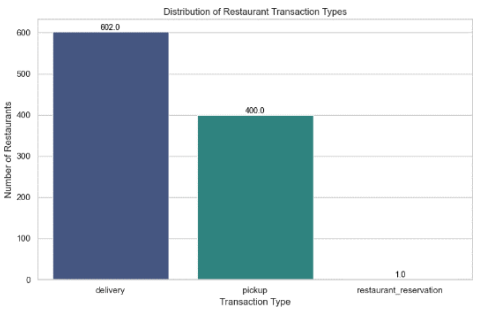
Access method using caching: Utilized Python's requests library to access Yelp's API. Implemented pagination to retrieve multiple records, with a limit of 50 results per request, up to a maximum of 1000 results. And finally saved this results into a json file “ann_arbor_restaurants.json”.

Data summary: 1003 records available for Ann Arbor on Yelp and successfully retrieved 1003 restaurant records.

Key fields in records: Each record contains essential information such as restaurant name, categories, rating, location coordinates, price, review count, and Yelp page URL. Some distribution graphs are attached below:







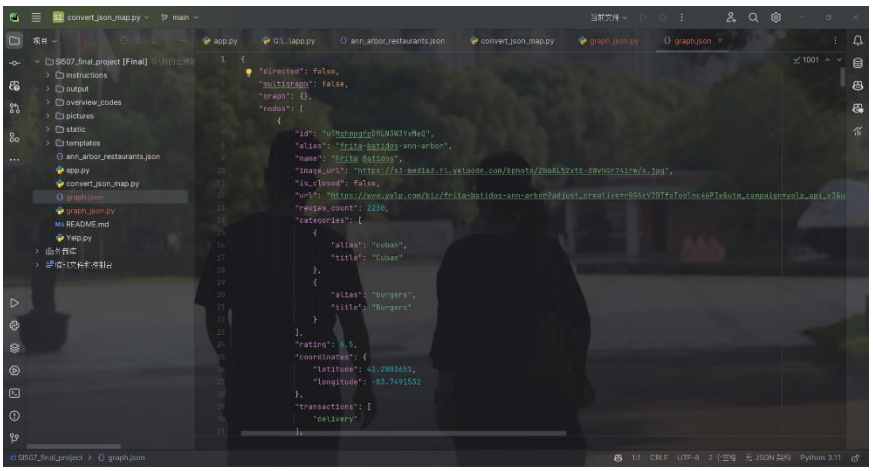
3. Data Structure:

Describe your Data Structure (graphs or trees): Graphs (Details have been added to the README file on GitHub).

A python file that constructs my graph from my stored data using classes (or some other method): graph_json.py.

JSON file with your graphs or trees: graph.json

A stand alone python file that reads the json of my graphs or trees: convert_json_map.py
Screenshots showing your data and data structures:



4. Interaction and Presentation Options:

High-level, plain-English description of the user-facing capabilities of your project:

1. Search specific restaurants and filter out certain types of restaurants in Ann Arbor, Michigan.
2. Display the result restaurants in the form of paginated list and user-interacted graph.
3. Allow users to go to corresponding Yelp page directly by clicking names on the list or graph.

Interactive and presentation technologies used: Flask

Brief instructions for how a user would interact with your program:

1. Run Yelp.py to get the json file ann_arbor_restaurants.json. (The users may need to get a new Yelp API key by themselves and replace my key with it on the Yelp Developers website)
2. Run graph_json.py to get the graph json file graph.json.
3. Run app.py to start the local user and get a web page, then the users can perform the actions of search and filter. (The users may need to get a new Google API key by themselves and replce my key with it on the Google API website)
4. The users can either search the name of restaurants directly, filter out certain types of restaurants as they like, or do both at the same time.
5. If the users want to reset all the values, they don't need to refresh the whole page. Instead, they just need to click the "Reset" button.

6. If the users want to see the details of certain restaurant in the result list, they can just click its name, and it will redirect them to the corresponding page on the Yelp website.
7. If the users want to see the details of certain restaurant on the map, they can just click the position mark and then its name, and it will redirect them to the corresponding page on the Yelp website.

4. Demo Link:

Link to demo video:
https://sjtu.feishu.cn/minutes/obcn6yw5199p1si218i8393p?from=from_copylink