Project Documentation

Fall 2021

SI 507 Final Project

Wenjie Wu

Project code:

- Link to my GitHub repo: https://github.com/collinswu/Si507_Final_Project
- README: see the GitHub repo
- Required packaged:
  - secrets: file contains API key and client ID
  - requests
  - json
  - plotly.express
  - matplotlib.pyplot
  - pandas
  - BeautifulSoup
  - webbrowser
  - yaml

Data sources:

- Yelp Fusion API:
  - URL: https://www.yelp.com/developers/documentation/v3/business_search
  - formats: JSON
  - To access the data, you need to follow the instruction to register for your private API key to authenticate requests. The register link is: https://www.yelp.com/developers/documentation/v3/authentication. The data is cashed as a JSON file.
  - Summary of data:
    - Number of records available: depends on which city you chose. If you choose Ann Arbor, there are 20 records. But if you choose a larger city, there may have more records.
    - Number of records retrieved: depends on which city you chose. If you choose Ann Arbor, there are 20 records. But if you choose a larger city, there may have more records.
    - Description of records: here is the list of important fields and what they represent:

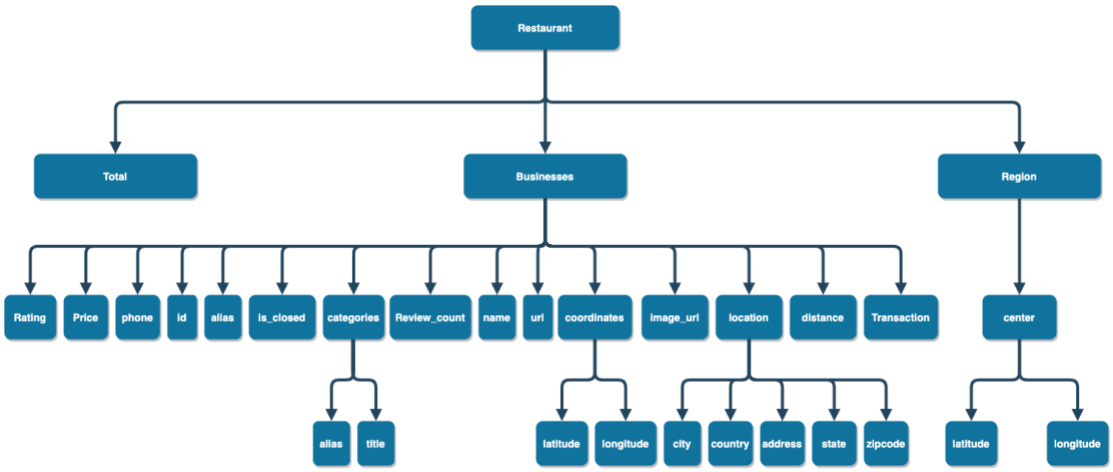
        - businesses: List of business Yelp finds based on the search criteria.
        - id: Unique Yelp ID of this business.
        - name: Name of this business.
        - url: URL for business page on Yelp.
        - review_count: Number of reviews for this business.
        - categories: List of category title and alias pairs associated with this business.
        - rating: Rating for this business (value ranges from 1, 1.5, ... 4.5, 5).
        - transactions: List of Yelp transactions that the business is registered for. Current supported values are pickup, delivery and restaurant_reservation
        - price: Price level of the business. Value is one of $, $$, $$$ and $$$$.

- Wikipedia Website: List of United States cities by population
    - URL: https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population
    - Format: HTML
    - To access the data, you need to scrap the relevant data from the website by using the BeautifulSoup. I stored the data inside the python file as a dictionary.
    - Summary of data:
        - Number of records available: there are 326 cities on the list.
        - Number of records retrieved: 326 cities are retrieved.
        - Description of records: The list contains the 326 incorporated places in the United States (excluding the U.S. territories) with a population of at least 100,000 on April 1, 2020, as enumerated by the United States Census Bureau. Five states—Delaware, Maine, Vermont, West Virginia and Wyoming—have no cities with populations of 100,000 or more. Here is the list of important fields and what they represent:
            - Ranking: population ranking based on 2020 census
            - City: name of the city
            - State: state of the city located
            - Population: number of population based on 2020 census
            - Change: changes of population compared to the number of population in 2010
            - Area: 2020 land area in square miles

Data Structure:

- README describing the Data Structure: See GitHub
- A python file that constructs trees from stored data using functions: final.py
- A JSON file that contains review information of restaurants with trees: review_<cityName>.json
- A JSON file that contains information of restaurants with trees: yelp_<cityName>.json
- A stand-alone python file that contains API information named secrets: secrets.py
- Screenshots showing your data and data structures:
  - Restaurant data: (Sample)


```
{
  "total": 8228,
  "businesses": [
    {
      "rating": 4,
      "price": "$",
      "phone": "+14152520800",
      "id": "E8RJkjfdcwgtyoPMjQ_Olg",
      "alias": "four-barrel-coffee-san-francisco",
      "is_closed": false,
      "categories": [
        {
          "alias": "coffee",
          "title": "Coffee & Tea"
        }
      ],
      "review_count": 1738,
      "name": "Four Barrel Coffee",
      "url": "https://www.yelp.com/biz/four-barrel-coffee-san-francisco",
      "coordinates": {
        "latitude": 37.7670169511878,
        "longitude": -122.42184275
      },
      "image_url": "http://s3-media2.fl.yelpcdn.com/bphoto/MmgtASP3l_t4tPCL1iAsCg/
      "location": {
        "city": "San Francisco",
        "country": "US",
        "address2": "",
        "address3": "",
        "state": "CA",
        "address1": "375 Valencia St",
        "zip_code": "94103"
      },
      "distance": 1604.23,
      "transactions": ["pickup", "delivery"]
    },
    // ...
  ],
  "region": {
    "center": {
      "latitude": 37.767413217936834,
      "longitude": -122.42820739746094
    }
  }
}
```



- City:

2020 rank	City	State[c]	2020 census	2010 census	Change	2020 land area		2020 population density		Location
1	*New York*[d]	New York	8,804,190	8,175,133	+7.69%	300.5 sq mi	778.3 km²	29,298/sq mi	11,312/km²	40.66°N 73.93°W
2	*Los Angeles*	California	3,898,747	3,792,621	+2.80%	469.5 sq mi	1,216.0 km²	8,304/sq mi	3,206/km²	34.01°N 118.41°W
3	*Chicago*	Illinois	2,746,388	2,695,598	+1.88%	227.7 sq mi	589.7 km²	12,061/sq mi	4,657/km²	41.83°N 87.68°W
4	*Houston*	Texas	2,304,580	2,099,451	+9.77%	640.4 sq mi	1,658.6 km²	3,599/sq mi	1,390/km²	29.78°N 95.39°W
5	*Phoenix*	Arizona	1,608,139	1,445,632	+11.24%	518.0 sq mi	1,341.6 km²	3,105/sq mi	1,199/km²	33.57°N 112.09°W
6	*Philadelphia*[e]	Pennsylvania	1,603,797	1,526,006	+5.10%	134.4 sq mi	348.1 km²	11,933/sq mi	4,607/km²	40.00°N 75.13°W
7	San Antonio	Texas	1,434,625	1,327,407	+8.08%	498.8 sq mi	1,291.9 km²	2,876/sq mi	1,110/km²	29.47°N 98.52°W
8	San Diego	California	1,386,932	1,307,402	+6.08%	325.9 sq mi	844.1 km²	4,256/sq mi	1,643/km²	32.81°N 117.13°W
9	Dallas	Texas	1,304,379	1,197,816	+8.90%	339.6 sq mi	879.6 km²	3,841/sq mi	1,483/km²	32.79°N 96.76°W
10	San Jose	California	1,013,240	945,942	+7.11%	178.3 sq mi	461.8 km²	5,683/sq mi	2,194/km²	37.29°N 121.81°W
11	Austin	Texas	961,855	790,390	+21.69%	319.9 sq mi	828.5 km²	3,007/sq mi	1,161/km²	30.30°N 97.75°W
12	*Jacksonville*[f]	Florida	949,611	821,784	+15.55%	747.3 sq mi	1,935.5 km²	1,271/sq mi	491/km²	30.33°N 81.66°W
13	Fort Worth	Texas	918,915	741,206	+23.98%	342.9 sq mi	888.1 km²	2,646/sq mi	1,022/km²	32.78°N 97.34°W
14	*Columbus*	Ohio	905,748	787,033	+15.08%	220.0 sq mi	569.8 km²	4,117/sq mi	1,590/km²	39.98°N 82.98°W
15	*Indianapolis*[g]	Indiana	887,642	820,445	+8.19%	361.6 sq mi	936.5 km²	2,455/sq mi	948/km²	39.77°N 86.14°W
16	*Charlotte*	North Carolina	874,579	731,424	+19.57%	308.3 sq mi	798.5 km²	2,837/sq mi	1,095/km²	35.20°N 80.83°W
17	San Francisco[h]	California	873,965	805,235	+8.54%	46.9 sq mi	121.5 km²	18,635/sq mi	7,195/km²	37.72°N 123.03°W
18	*Seattle*	Washington	737,015	608,660	+21.09%	83.8 sq mi	217.0 km²	8,795/sq mi	3,396/km²	47.62°N 122.35°W
19	*Denver*[i]	Colorado	715,522	600,158	+19.22%	153.1 sq mi	396.5 km²	4,674/sq mi	1,805/km²	39.76°N 104.88°W
20	*Washington*[j]	District of Columbia	689,545	601,723	+14.60%	61.1 sq mi	158.2 km²	11,286/sq mi	4,358/km²	38.90°N 77.01°W
21	*Nashville*[k]	Tennessee	689,447	601,222	+14.67%	475.8 sq mi	1,232.3 km²	1,449/sq mi	559/km²	36.17°N 86.78°W
22	*Oklahoma City*	Oklahoma	681,054	579,999	+17.42%	606.2 sq mi	1,570.1 km²	1,123/sq mi	434/km²	35.46°N 97.51°W
23	El Paso	Texas	678,815	649,121	+4.57%	258.4 sq mi	669.3 km²	2,627/sq mi	1,014/km²	31.84°N 106.42°W
24	*Boston*	Massachusetts	675,647	617,594	+9.40%	48.3 sq mi	125.1 km²	13,989/sq mi	5,401/km²	42.33°N 71.02°W
25	*Portland*	Oregon	652,503	583,776	+11.77%	133.5 sq mi	345.8 km²	4,888/sq mi	1,887/km²	45.53°N 122.65°W
26	*Las Vegas*	Nevada	641,903	583,756	+9.96%	141.8 sq mi	367.3 km²	4,527/sq mi	1,748/km²	36.22°N 115.26°W
27	*Detroit*	Michigan	639,111	713,777	−10.46%	138.7 sq mi	359.2 km²	4,608/sq mi	1,779/km²	42.38°N 83.10°W

Interaction and Presentation Options:

- User choice and interactive instruction:
  - The program will ask the user whether or not to show Wikipedia page of a city. “Do you want to open the Wikipedia of the city?”
    - If Yes:
      - The user will input a name of the city. “Which city? (Format: Ann_Arbor)”

        - Then the user will input the state of the city. “Which state? (Format: Michigan)”
        - Then the webpage of the city will open in the browser
        - Meanwhile, the program will ask the user again whether or not to select another city. If yes, repeat the steps above. If no, Go to the next step.
    - If No: the program will ask the user to enter a name of city to get the restaurant data. “Please enter a city to get the restaurant data.”
- The information of restaurants in selected city will be stored as a JSON file, and the command line will show “JSON File yelp_Ann arbor.json Has been stored”
- Then the tree structure of the restaurants will show in the command line after “The structure of the tree is blow”
- A list of restaurant name in the city will show after “The restaurant name of the city”
- The review of restaurants in the city will be stored as a JSON file. “The reviews are saved as a JSON file”
- The category of restaurant will show with names and categories. “The category of the restaurant”
- A unique list of categories of restaurant will show. “The unique restaurant list of the city”
- A pie char of category will show. After you close the chart, the program will continue
- The information of cities will be scrapped from the website and show in the command line. “The information of the city is below.”
- A bar chart of population of the city will be shown in the browser
- A bar chart of number of restaurants in different rating will be shown. After you close the chart, the program will continue
- The program will ask the user to select the price level to search a resuaurant. “What price level are you looking for? (e.g. $,$$,$$$,$$$$)”
- The program will list the restaurant with selected price level. “Here is the name of the restaurant:”
- The program will ask whether to open the website of the restaurant on Yelp. “Do you want to open the website of a restaurant?”
    - If yes: the program will ask the user to enter the name of the restaurant. “Please enter the name of the restaurant”

        - A website of the restaurant on Yelp will be open in the browser. The user can see all the relevant information of the restaurant and reserve a table. The program stops.
    - If no: the program will show “Bye” and stop.

- Interactive and presentation technologies used: Plotly, Matplotlib, webbrowser, command line prompts

Demo Link:

- Link to demo video: https://youtu.be/7ShAyLzmwzs