Project Documentation – SI 507 Final Project

Name: Xiaoyue Liu, Uniqname: liyayue, UMID: 41895539

1 Project code

1.1 Link to github

https://github.com/liyayue/SI507-Final-Project

1.2 README info (also on github)

REQUIRED PACKAGES

import requests

import json

import re

import html

from time import sleep

import os

from treelib import Tree

from pandas.core.frame import DataFrame

import plotly

import plotly.graph_objects as go

import plotly.offline as of

import plotly.express as px

from geopy.distance import geodesic

RUNNING INSTRUCTIONS

Firstly, make sure your python has the above package installed. Secondly, download all the files on the github in a single document. Unzip the **yelp_lv_dataset.zip**, since the json file is to large to upload, and make sure the json file in the same path as the other files. Then, open the **final programming code.py** and change the work path to the one where you just downloaded the files and run it. Since I have stored all the data needed in json files and uploaded them, so you can only run this file. Now you can start to interact with the programming.

If you want to run from getting the hotel data, open the **hotel_data_webscraping.py** file. In addition to changing the working path as mentioned in the previous paragraph, you also need to change the headers to your own device or the site will refuse access. Way to do so it to, taking Google Chrome for example, right click on the web page and click on Inspect, under Network click on Doc, press F5 to refresh the data if there is no response, find the user-agent in the displayed file and that is your header. Then, you can run this file and get the hotel data.

2 Data sources

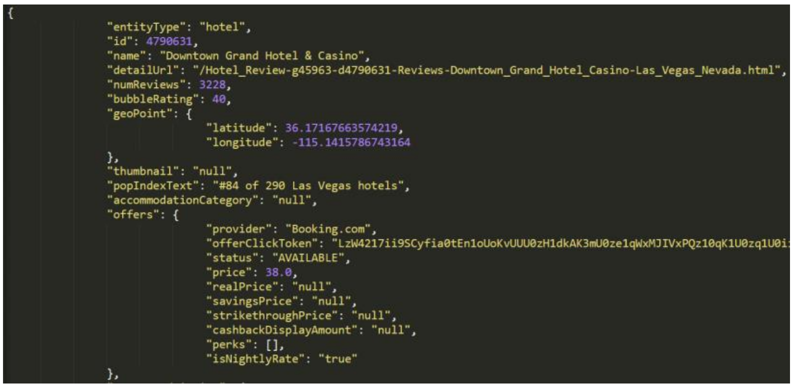
Hotel Data: https://www.tripadvisor.com/Hotels.This website contains information about the hotel, such as name, location, ratings, reviews, etc. I scraped the information of all available hotels in Las Vegas through url (13 pages and 380 hotels in total). To make it easier to query the data format and to prevent wasting time re-reading the url each time, I stored all the hotel information in a json file using caching. Since searching for hotels in a city will show several pages and the url of different hotels are different, it meets the requirement of “*Crawling [and scraping] multiple pages in a site you haven’t used before ✢ (8 score)*”.

Yelp Data: https://www.yelp.com/dataset. Yelp has open dataset with over 8 millions reviews and one hundred thousands businesses of 8 metropolitan areas (Montreal, Calgary, Toronto, Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland). The data is downloaded in JSON files with five in total. The one that will be used is business data including location data, attributes, and categories of restaurants (key: business_id). This time I only filtered the data of restaurants that did not close in Las Vegas and stored in a json file. There are 26,540 records and 99 keys in each dictionary in total. It meets the requirement of “*CSV or JSON file you haven’t used before with > 1000 records (2 score)*”.

3 Data Structure

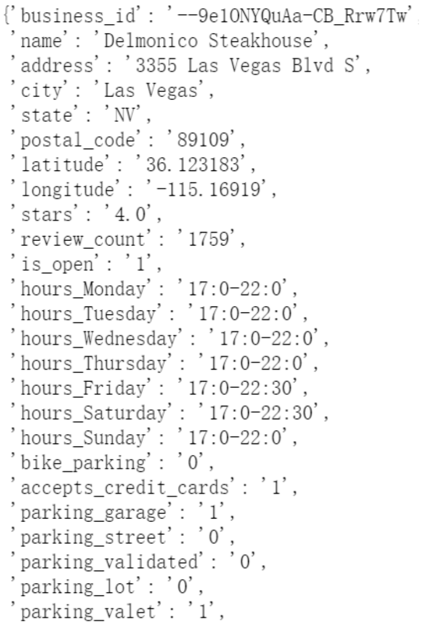
Only the important keys in the database are listed here.

3.1 Hotel Data



```
{
        "entityType": "hotel",
        "id": 4790631,
        "name": "Downtown Grand Hotel & Casino",
        "detailUrl": "/Hotel_Review-g45963-d4790631-Reviews-Downtown_Grand_Hotel_Casino-Las_Vegas_Nevada.html",
        "numReviews": 3228,
        "bubbleRating": 40,
        "geoPoint": {
                "latitude": 36.17167663574219,
                "longitude": -115.1415786743164
        },
        "thumbnail": "null",
        "popIndexText": "#84 of 290 Las Vegas hotels",
        "accommodationCategory": "null",
        "offers": {
                "provider": "Booking.com",
                "offerClickToken": "LzW4217ii9SCyfia0tEn1oUoKvUUU0zH1dkAK3mU0ze1qWxMJIVxPQz10qK1U0zq1U0i
                "status": "AVAILABLE",
                "price": 38.0,
                "realPrice": "null",
                "savingsPrice": "null",
                "strikethroughPrice": "null",
                "cashbackDisplayAmount": "null",
                "perks": [],
                "isNightlyRate": "true"
        },
```

3.2 Yelp Data - Las Vegas



```
{’business_id’: ’--9e1ONYQuAa-CB_Rrw7Tw’
 ’name’: ’Delmonico Steakhouse’,
 ’address’: ’3355 Las Vegas Blvd S’,
 ’city’: ’Las Vegas’,
 ’state’: ’NV’,
 ’postal_code’: ’89109’,
 ’latitude’: ’36.123183’,
 ’longitude’: ’-115.16919’,
 ’stars’: ’4.0’,
 ’review_count’: ’1759’,
 ’is_open’: ’1’,
 ’hours_Monday’: ’17:0-22:0’,
 ’hours_Tuesday’: ’17:0-22:0’,
 ’hours_Wednesday’: ’17:0-22:0’,
 ’hours_Thursday’: ’17:0-22:0’,
 ’hours_Friday’: ’17:0-22:30’,
 ’hours_Saturday’: ’17:0-22:30’,
 ’hours_Sunday’: ’17:0-22:0’,
 ’bike_parking’: ’0’,
 ’accepts_credit_cards’: ’1’,
 ’parking_garage’: ’1’,
 ’parking_street’: ’0’,
 ’parking_validated’: ’0’,
 ’parking_lot’: ’0’,
 ’parking_valet’: ’1’,
```

4 Interaction and Presentation Options

Step 1: Ask "Input the name of the hotel that you are considering to order in Las Vegas. Have no idea? Enter 0 to see the top 10 hotels in Las Vegas!"

-- Step 1.1: If a user enter 0, output the name of top 10 and then ask the user again to input a name of a hotel. (Tree)

-- Step 1.2: If a user enter a name of the hotel that is not in the dataset, it will ask him to try another one. (Command line)

Step 2: Ask "How many miles away from the hotel would you like to find the restaurant information about?" The program will give the range of the nearest and farthest restaurant distance. The user only needs to select a digital input in this range. (Based on the latitude and longitude of the restaurant and hotel)

-- Step 2.1: If the number is unavailable, it will ask him to try another one.

-- Step 2.2: If the number is available, it will print the total number of the restaurants that meet requirement. If the number is below 15, it will print all the restaurants' name, rating, No. review, and address. If not, it will give information about the top 15 rated restaurants. (Command line)

Step 3: Ask If the user wants to change the range or enter 9 for a comparison between two hotels

-- Step 3.1: If the user want to change a distance, it will loop the Step 2.

-- Step 3.2: If the user input 9, it will go to next step for a comparison.

Step 4: Ask the user to input two hotels.

-- Step 4.1: Same with Step 1.1

-- Step 4.2: Same with Step 1.2

Step 5: Ask the user "Enter 1 if you want to know the distance distribution of restaurants near the hotel, enter 2 if you want to know the scattered distribution between ratings and distances of restaurants near the hotel, enter 3 to exit".

-- Step 5.1: If the user enters 1, it will show the distance distribution of restaurants near the hotel. (Plotly-histogram)

-- Step 5.2: If the user enters 2, it will show the scattered distribution between ratings and distances of restaurants near the hotel. (Plotly- scatter diagram)

-- Step 5.3: If the user enters 3, exit.

5 Demo Link

Video Demo Link: https://youtu.be/FMovfmo6Ip0