# Coursera - IBM Professional Certificate  - Applied Data Science Capstone Project:

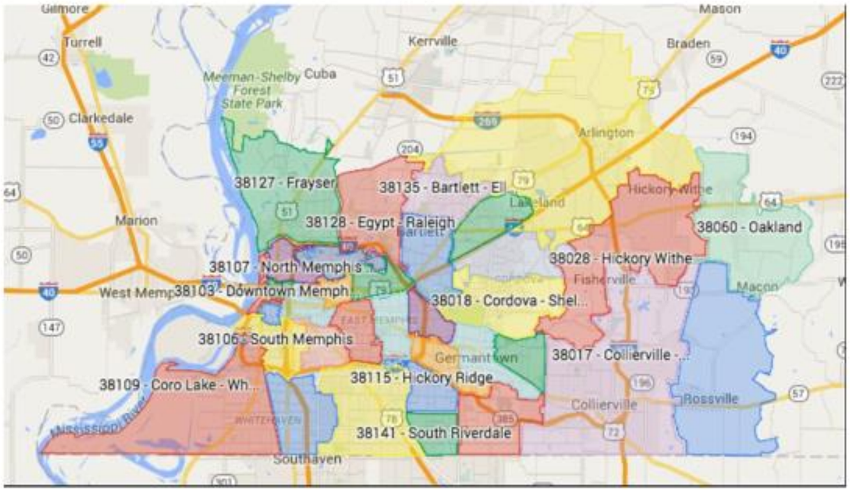
Week 5 : Peer-graded Assignment: Capstone Project - The Battle of Neighborhoods (Week 2)

Title of the project : Aerial view - venues mapping support system

**Introduction / Background :**Catalyst LLC is a new start-up company in Memphis, Tennessee, USA which provides support services to tourists & visitors as well as aiding to conduct personal/professional  events and functions. Inc.X has a plan to have a support system for selected option by the customer who likes to explore the city. As a small-win project, residential City of Memphis taken as a geo-location and through the coding in Python with the help of Coursera-IBM Skill lab and GitHub working out a proposal.

About the city selected : MEMPHIS is a city located in Tennessee, United States of America. The city population was 650,618 in 2018.Memphis is the 26th largest city out of 314 US cities. Memphis is one of the leading commercial centers in transportation and logistics. Memphis international airport is the second busiest cargo airport in the world and the international port of Memphis is the fifth busiest inland water port in America. As on date, Memphis is a regional center for commerce, education, media, art, music and entertainment.   (Note: Two memorable places in Memphis are : 1. King of rock and roll – singer and actor Elvis Presley's   and 2. The Noble Peace Prize 1964 winner for non-violence and Leader - Martin Luther King Jr.'s.) +  (P.S) Interesting to also note that each and every FedEx air package in US goes through Memphis, though there are direct routes.

Memphis map – with Postal Codes



4/17/2020 IBM Data Science Capstone Project 4

**Objective**: The objective of this project is to provide management and users to view list of combined venues like food centers, cafeteria fun places, Nightlife in the selected city together with their geo-spatial data or through an aerial map to save their decision support and saving their time by avoiding one by one search.

**Data Description**: 1. The data used in the project is obtained from FOURSQUARE API. By using Explorer, data collected for café, restaurant, fun-'Gym', nightlife , my Top pick-'Museum',etc., for multiple searches.

Link: https://foursquare.com/

2. For the city data: Latitude: **35.122864,**Longitude: **-89.951860,**Population: **729,307,**Area: **391.55** sq. miles, Land: **386.69** sq. miles, Water: **4.86** sq. miles, Population Density: **1,886.02** people per sq. mile, Elevation: **273.0** feet above sea level . Time Zone: **Central (GMT -6)**

Link: https://en.wikipedia.org/wiki/Memphis,_Tennessee

https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Memphis,_Tennessee

https://en.wikipedia.org/wiki/Category:Neighborhoods_in_Memphis,_Tennessee

https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population

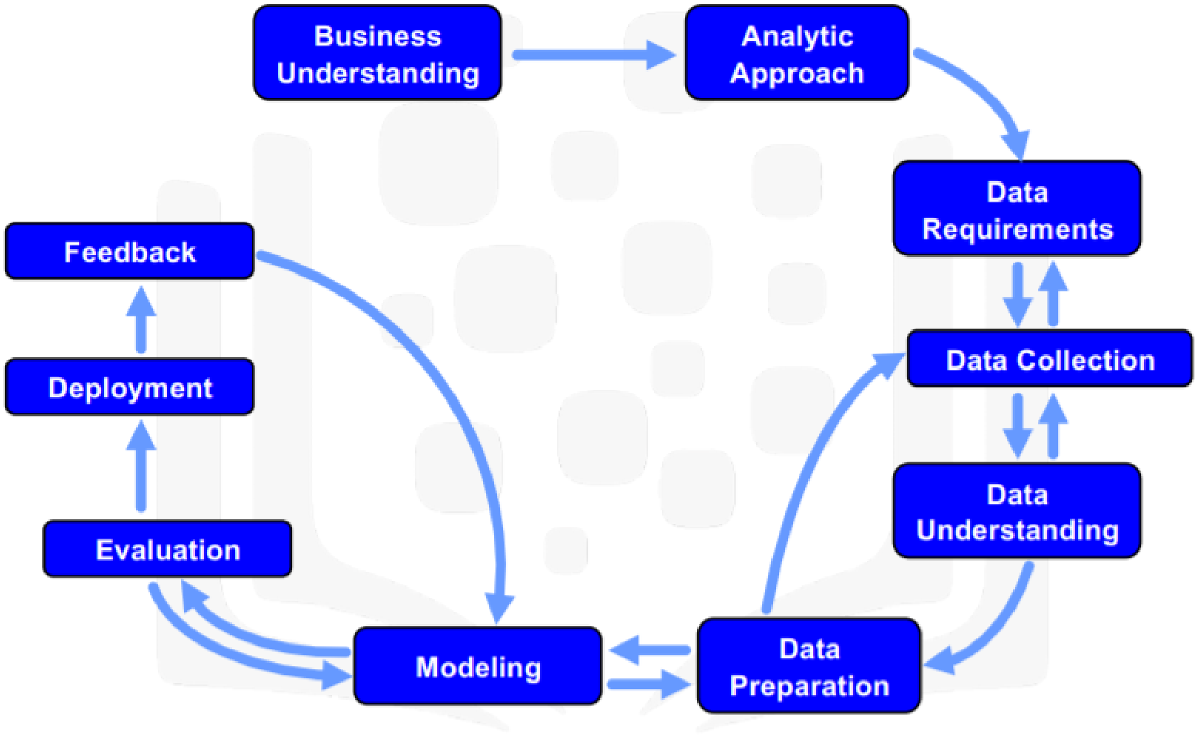
https://open311.memphistn.gov/#!/view-data

https://www.fedex.com/en-us/about/policy/aviation/why-memphis.html

3. For Choropleth maps , geojson file of Memphis used.

4.Python Folium library will be used to visualize the venues & allied clusters.

**Methodology proposed:** Learning from IBM Model – CRISP (Cross Industries Standard Process) through Coursera :



Remember: Above 10 steps are need not to be followed strictly in a sequence (i.e) a flexible and iterative approach.

**Table of Contents**: Project Overview -Data Overview -Data Wrangling -FourSquare API-Exploratory Data Analysis -Clustering- Cluster Mapping using Folium- Cluster Results - Conclusion

**Coverage planned:** Venues mapping – Food, Park, Cafeteria, Fun-'Gym', Breakfast, Restaurants, Nightlife & my Top pick-'Museum'

The following questions are to be attempted:

1.How many places or venues under each category?

2. How many similar venues exist and its' specialty ?

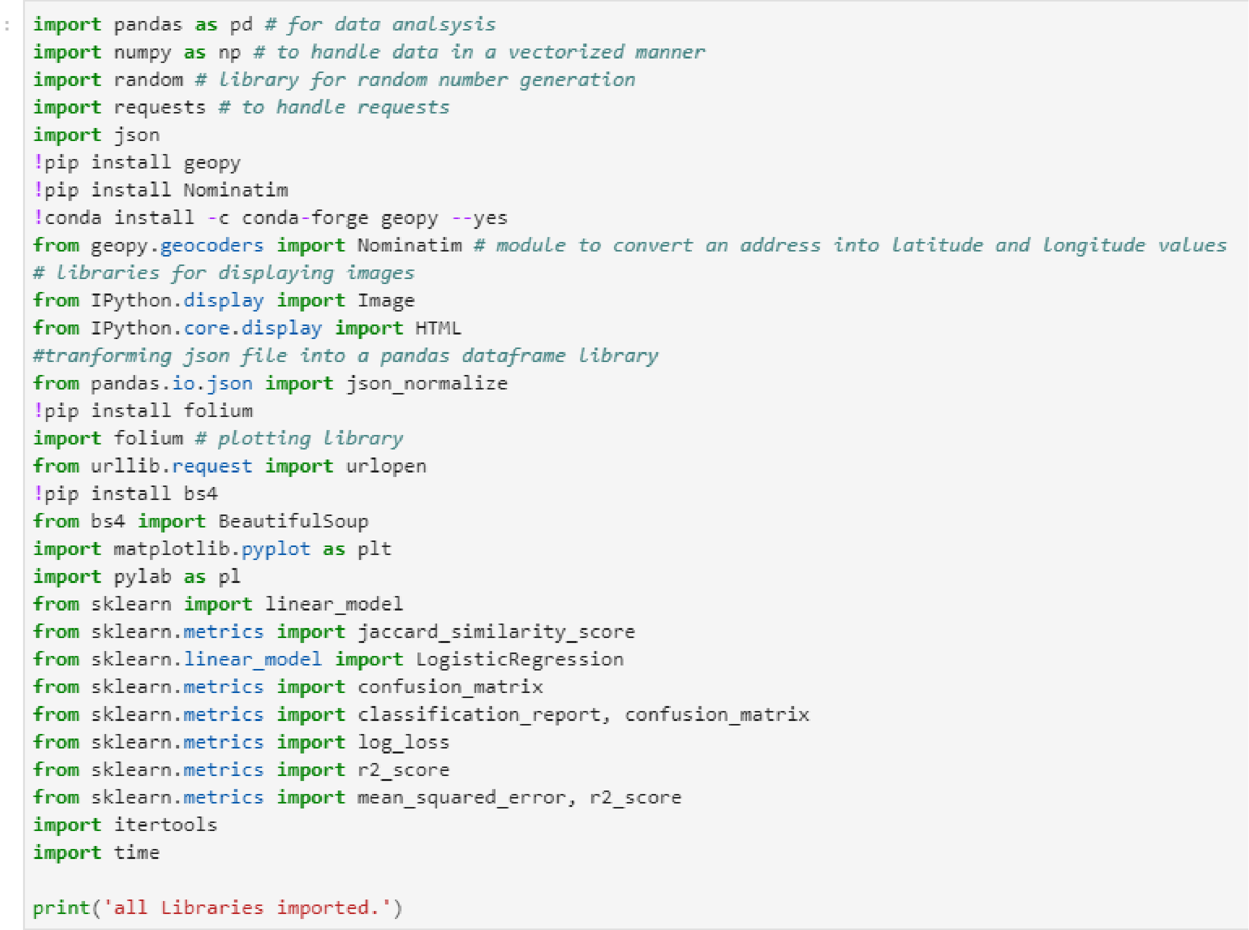
3. How to get 'Birds eye view' of their locations ?

**Scope**: Starting with one city – Memphis but can be deployed with multiple cities optional selection by clustering in next phase, after quick-win of the project. Other Tennessee cities (e,g) like Nashville (First largest) and Knoxville( Third largest ) may also be included by including their address 2 and 3 as shown in Python Jupyter note book.

**Target** : Users includes not only of our company management person but also mainly our customers and public on need (i.e) exploring different places or similar places.

## Steps followed:

### 1. Importing / installing Libraries:



```
import pandas as pd # for data analsysis
import numpy as np # to handle data in a vectorized manner
import random # library for random number generation
import requests # to handle requests
import json
!pip install geopy
!pip install Nominatim
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values
# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML
#tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize
!pip install folium
import folium # plotting library
from urllib.request import urlopen
!pip install bs4
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
import pylab as pl
from sklearn import linear_model
from sklearn.metrics import jaccard_similarity_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import log_loss
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error, r2_score
import itertools
import time

print('all Libraries imported.')
```

### 2.Data

### 2.1 Data Scraping and Cleaning :

(a) For the selected city (My choice: Memphis,Tennessee) ,obtaining its latitude & longitude :

In order to define an instance of the geocoder, we need to define a user_agent. I used our agent as foursquare_agent, as shown below.

Got the geographical coordinate of Memphis, Tennessee are 35.1490215, -90.0516285.

(b) To extend to another city (for an example -my choice: Nashville,TN),obtaining its latitude & longitude :

The geographical coordinate of Nashville, Tennessee are 36.1622296, -86.7743531.

© To enable multi cities, as another sample for adding additional city ( 3rd city - Knoxville, TN )

The geographical coordinate of Knoxville, Tennessee are 35.9603948, -83.9210261.

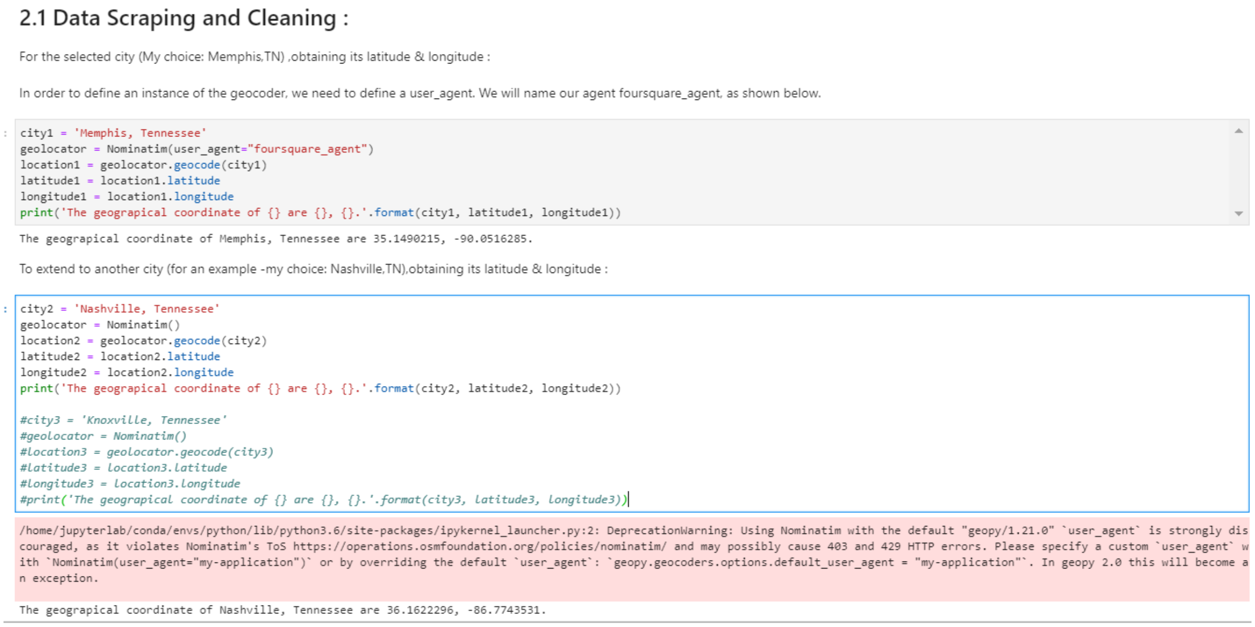
### *2.1.1 Defining Foursquare Credentials and Version:*

Got the credentials.

Neat process Segmentation based on selected venue search : with neighborhood data - postal code and their latitude and longitude details:-

- *created first city (Memphis) URL*
- *created second city (Nashville) URL*
- *Printed both URL for confirmation*
- *By using url -json-normalize-filtering-cleaning columns*

# 2. Data Wrangling:



## 2.1 Data Scraping and Cleaning :

For the selected city (My choice: Memphis,TN) ,obtaining its latitude & longitude :

In order to define an instance of the geocoder, we need to define a user_agent. We will name our agent foursquare_agent, as shown below.

```
city1 = 'Memphis, Tennessee'
geolocator = Nominatim(user_agent="foursquare_agent")
location1 = geolocator.geocode(city1)
latitude1 = location1.latitude
longitude1 = location1.longitude
print('The geograpical coordinate of {} are {}, {}.'.format(city1, latitude1, longitude1))
```

```
The geograpical coordinate of Memphis, Tennessee are 35.1490215, -90.0516285.
```

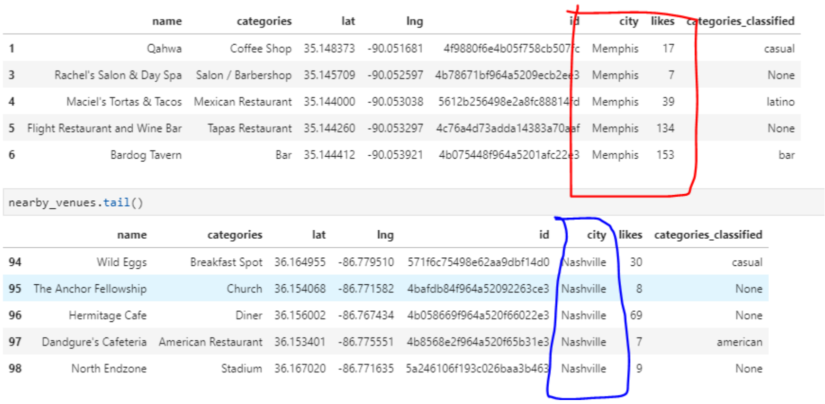
To extend to another city (for an example -my choice: Nashville,TN),obtaining its latitude & longitude :

```
city2 = 'Nashville, Tennessee'
geolocator = Nominatim()
location2 = geolocator.geocode(city2)
latitude2 = location2.latitude
longitude2 = location2.longitude
print('The geograpical coordinate of {} are {}, {}.'.format(city2, latitude2, longitude2))

#city3 = 'Knoxville, Tennessee'
#geolocator = Nominatim()
#location3 = geolocator.geocode(city3)
#latitude3 = location3.latitude
#longitude3 = location3.longitude
#print('The geograpical coordinate of {} are {}, {}.'.format(city3, latitude3, longitude3))
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/ipykernel_launcher.py:2: DeprecationWarning: Using Nominatim with the default "geopy/1.21.0" `user_agent` is strongly dis
couraged, as it violates Nominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ and may possibly cause 403 and 429 HTTP errors. Please specify a custom `user_agent` w
ith `Nominatim(user_agent="my-application")` or by overriding the default `user_agent`: `geopy.geocoders.options.default_user_agent = "my-application"`. In geopy 2.0 this will become a
n exception.
```

```
The geograpical coordinate of Nashville, Tennessee are 36.1622296, -86.7743531.
```

# Feasibility-2 cities data merge



| | name | categories | lat | lng | id | city | likes | categories_classified |
|---|---|---|---|---|---|---|---|---|
| 1 | Qahwa | Coffee Shop | 35.148373 | -90.051681 | 4f9880f6e4b05f758cb507fc | Memphis | 17 | casual |
| 3 | Rachel's Salon & Day Spa | Salon / Barbershop | 35.145709 | -90.052597 | 4b78671bf964a5209ecb2ee3 | Memphis | 7 | None |
| 4 | Maciel's Tortas & Tacos | Mexican Restaurant | 35.144000 | -90.053038 | 5612b256498e2a8fc88814fd | Memphis | 39 | latino |
| 5 | Flight Restaurant and Wine Bar | Tapas Restaurant | 35.144260 | -90.053297 | 4c76a4d73adda14383a70aaf | Memphis | 134 | None |
| 6 | Bardog Tavern | Bar | 35.144412 | -90.053921 | 4b075448f964a5201afc22e3 | Memphis | 153 | bar |

```
nearby_venues.tail()
```

| | name | categories | lat | lng | id | city | likes | categories_classified |
|---|---|---|---|---|---|---|---|---|
| 94 | Wild Eggs | Breakfast Spot | 36.164955 | -86.779510 | 571f6c75498e62aa9dbf14d0 | Nashville | 30 | casual |
| 95 | The Anchor Fellowship | Church | 36.154068 | -86.771582 | 4bafdb84f964a52092263ce3 | Nashville | 8 | None |
| 96 | Hermitage Cafe | Diner | 36.156002 | -86.767434 | 4b058669f964a520f66022e3 | Nashville | 69 | None |
| 97 | Dandgure's Cafeteria | American Restaurant | 36.153401 | -86.775551 | 4b8568e2f964a520f65b31e3 | Nashville | 7 | american |
| 98 | North Endzone | Stadium | 36.167020 | -86.771635 | 5a246106f193c026baa3b463 | Nashville | 9 | None |

For both cities ,obtained number of rows for both cities.

For Memphis -77 venues were returned by Foursquare.
For Nashville -100 venues were returned by Foursquare.

As a next decision , in line with scope mentioned, only one city taken for this proto project – ‘Memphis’ for further proceeding.

For better understanding , unique list of categories with respect of Memphis obtained.

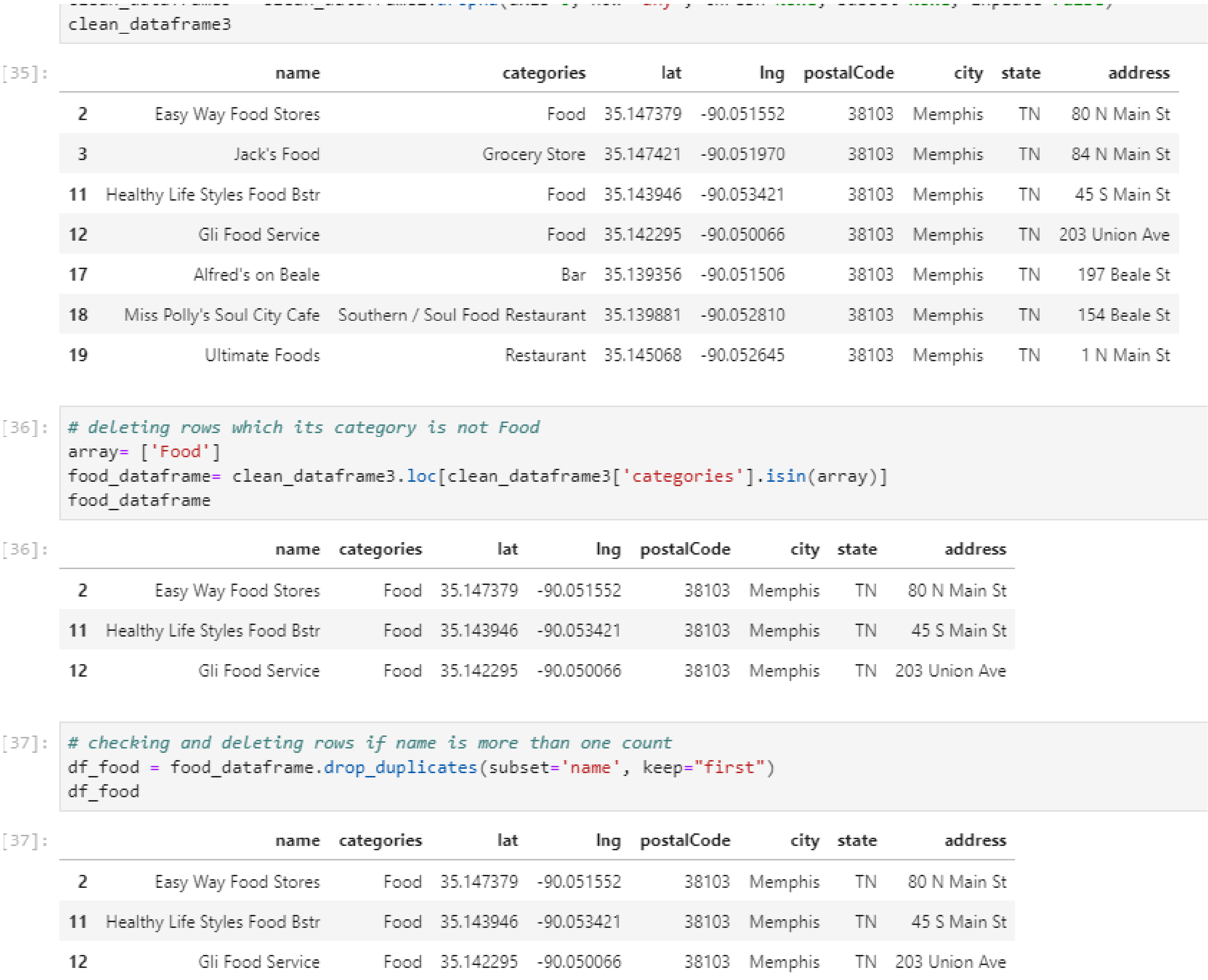
Memphis_unique_cat_list :

```
array(['Concert Hall', 'Coffee Shop', 'Park', 'Salon / Barbershop',
       'Mexican Restaurant', 'Tapas Restaurant', 'Bar', 'Pub', 'Trail',
       'American Restaurant', 'Cuban Restaurant', 'Hotel',
       'Gym / Fitness Center', 'Baseball Stadium',
       'Southern / Soul Food Restaurant', 'Sandwich Place',
       'Seafood Restaurant', 'Roof Deck', 'Pizza Place', 'Burger Joint',
       'Cocktail Bar', 'Rest Area', 'Sporting Goods Shop',
       'Brazilian Restaurant', 'Café', 'Gym', 'BBQ Joint', 'Smoke Shop',
       'Speakeasy', 'Lounge', 'Beer Garden', 'Event Space',
       'Sushi Restaurant', 'Food Court', 'Shopping Mall', 'Karaoke Bar',
       'Museum', 'Liquor Store', 'Theme Restaurant',
       'Fried Chicken Joint', 'Fast Food Restaurant', 'Hotel Bar',
       'Harbor / Marina', "Men's Store", 'Boat or Ferry',
       'Scenic Lookout', 'Gas Station', 'Music Venue', 'Intersection',
       'Department Store'], dtype=object)
```

Based on foursquare, decided to go for search data for following categories: 1 . Food, 2, Restaurant,3.Cafetaria , 4.Breakfast,5. Nightlife ( since Memphis is industry/employee belt area and Music loving city),6.Park, 7.Top pick- I selected Museum & 8.Fun – I selected Gym/Fitness center.

First Search - 'Food':

- Search query for selected category with radius in miles
- Defining the corresponding url ; I used url with latitude1 & longitude1 for Memphis.
- Sending the GET request and examining the results
- Assigning relevant part of JSON to venues
- Transforming venues into a dataframe
- Listing the dataframe
- Deleting the rows which not pertains to selected category
- Checking for duplicate records and deleting in that case.



```
clean_dataframe3= clean_dataframe3.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
clean_dataframe3
```

[35]:

| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 2 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 3 | Jack's Food | Grocery Store | 35.147421 | -90.051970 | 38103 | Memphis | TN | 84 N Main St |
| 11 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 12 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |
| 17 | Alfred's on Beale | Bar | 35.139356 | -90.051506 | 38103 | Memphis | TN | 197 Beale St |
| 18 | Miss Polly's Soul City Cafe | Southern / Soul Food Restaurant | 35.139881 | -90.052810 | 38103 | Memphis | TN | 154 Beale St |
| 19 | Ultimate Foods | Restaurant | 35.145068 | -90.052645 | 38103 | Memphis | TN | 1 N Main St |

[36]:

```
# deleting rows which its category is not Food
array= ['Food']
food_dataframe= clean_dataframe3.loc[clean_dataframe3['categories'].isin(array)]
food_dataframe
```

[36]:

| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 2 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 11 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 12 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |

[37]:

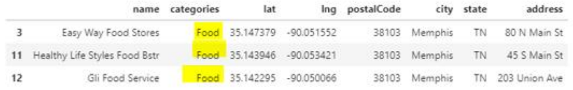
```
# checking and deleting rows if name is more than one count
df_food = food_dataframe.drop_duplicates(subset='name', keep="first")
df_food
```

[37]:

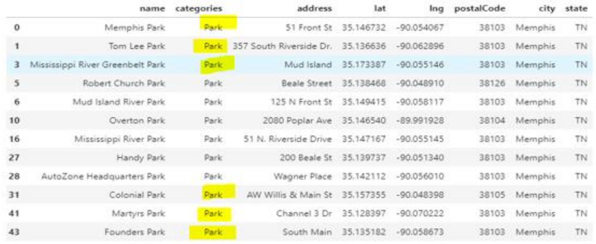
| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 2 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 11 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 12 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |

Similarly , in the same process steps, collected data for 2.Park, ,3.Cafetaria , 4.Breakfast,5. Nightlife, 6. Restaurant , 7.Museum & 8.Gym/Fitness center.

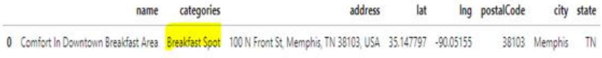
# Data Extraction – Various search inputs-after cleaning



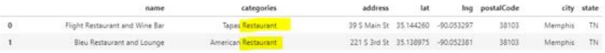
| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 3 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 11 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 12 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |



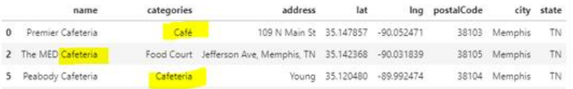
| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Memphis Park | Park | 51 Front St | 35.146732 | -90.054067 | 38103 | Memphis | TN |
| 1 | Tom Lee Park | Park | 357 South Riverside Dr. | 35.136636 | -90.062896 | 38103 | Memphis | TN |
| 3 | Mississippi River Greenbelt Park | Park | Mud Island | 35.173387 | -90.055146 | 38103 | Memphis | TN |
| 5 | Robert Church Park | Park | Beale Street | 35.138468 | -90.048910 | 38126 | Memphis | TN |
| 6 | Mud Island River Park | Park | 125 N Front St | 35.149415 | -90.058117 | 38103 | Memphis | TN |
| 10 | Overton Park | Park | 2080 Poplar Ave | 35.146540 | -89.991928 | 38104 | Memphis | TN |
| 16 | Mississippi River Park | Park | 51 N. Riverside Drive | 35.147167 | -90.055145 | 38103 | Memphis | TN |
| 27 | Handy Park | Park | 200 Beale St | 35.139737 | -90.051340 | 38103 | Memphis | TN |
| 28 | AutoZone Headquarters Park | Park | Wagner Place | 35.142112 | -90.056010 | 38103 | Memphis | TN |
| 31 | Colonial Park | Park | AW Willis & Main St | 35.157355 | -90.048398 | 38105 | Memphis | TN |
| 41 | Martyrs Park | Park | Channel 3 Dr | 35.128397 | -90.070222 | 38103 | Memphis | TN |
| 43 | Founders Park | Park | South Main | 35.135182 | -90.058673 | 38103 | Memphis | TN |



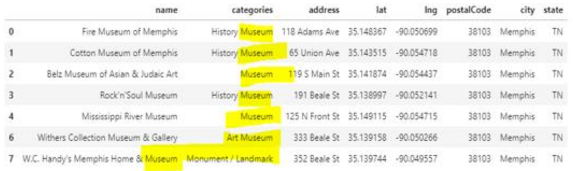
| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Comfort In Downtown Breakfast Area | Breakfast Spot | 100 N Front St, Memphis, TN 38103, USA | 35.147797 | -90.05155 | 38103 | Memphis | TN |



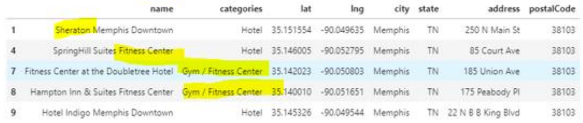
| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Flight Restaurant and Wine Bar | Tapas Restaurant | 39 S Main St | 35.144260 | -90.053297 | 38103 | Memphis | TN |
| 1 | Bleu Restaurant and Lounge | American Restaurant | 221 S 3rd St | 35.138975 | -90.052381 | 38103 | Memphis | TN |



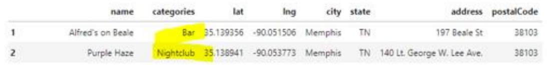
| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Premier Cafeteria | Café | 109 N Main St | 35.147857 | -90.052471 | 38103 | Memphis | TN |
| 2 | The MED Cafeteria | Food Court | Jefferson Ave, Memphis, TN | 35.142368 | -90.031839 | 38105 | Memphis | TN |
| 5 | Peabody Cafeteria | Cafeteria | Young | 35.120480 | -89.992474 | 38104 | Memphis | TN |



| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Fire Museum of Memphis | History Museum | 118 Adams Ave | 35.148367 | -90.050699 | 38103 | Memphis | TN |
| 1 | Cotton Museum of Memphis | History Museum | 65 Union Ave | 35.143515 | -90.054718 | 38103 | Memphis | TN |
| 2 | Belz Museum of Asian & Judaic Art | Museum | 119 S Main St | 35.141874 | -90.054437 | 38103 | Memphis | TN |
| 3 | Rock'n'Soul Museum | History Museum | 191 Beale St | 35.138997 | -90.052141 | 38103 | Memphis | TN |
| 4 | Mississippi River Museum | Museum | 125 N Front St | 35.149115 | -90.054715 | 38103 | Memphis | TN |
| 6 | Withers Collection Museum & Gallery | Art Museum | 333 Beale St | 35.139158 | -90.050266 | 38103 | Memphis | TN |
| 7 | W.C. Handy's Memphis Home & Museum | Monument / Landmark | 352 Beale St | 35.139744 | -90.049557 | 38103 | Memphis | TN |



| | name | categories | lat | lng | city | state | address | postalCode |
|---|---|---|---|---|---|---|---|---|
| 1 | Sheraton Memphis Downtown | Hotel | 35.151554 | -90.049635 | Memphis | TN | 250 N Main St | 38103 |
| 4 | SpringHill Suites Fitness Center | Hotel | 35.146005 | -90.052795 | Memphis | TN | 85 Court Ave | 38103 |
| 7 | Fitness Center at the Doubletree Hotel | Gym / Fitness Center | 35.142023 | -90.050803 | Memphis | TN | 185 Union Ave | 38103 |
| 8 | Hampton Inn & Suites Fitness Center | Gym / Fitness Center | 35.140010 | -90.051651 | Memphis | TN | 175 Peabody Pl | 38103 |
| 9 | Hotel Indigo Memphis Downtown | Hotel | 35.145326 | -90.049544 | Memphis | TN | 22 N B B King Blvd | 38103 |



| | name | categories | lat | lng | city | state | address | postalCode |
|---|---|---|---|---|---|---|---|---|
| 1 | Alfred's on Beale | Bar | 35.139356 | -90.051506 | Memphis | TN | 197 Beale St | 38103 |
| 2 | Purple Haze | Nightclub | 35.138941 | -90.053773 | Memphis | TN | 140 Lt. George W. Lee Ave. | 38103 |

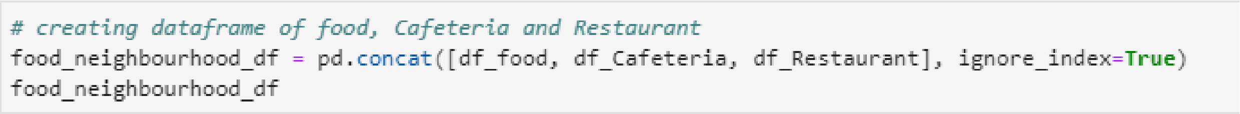
3. Clustering: Creating aerial / geographical location map view-for multi clusters combination options

After completing all search data and cleaned data, next step is to generate map to visualize

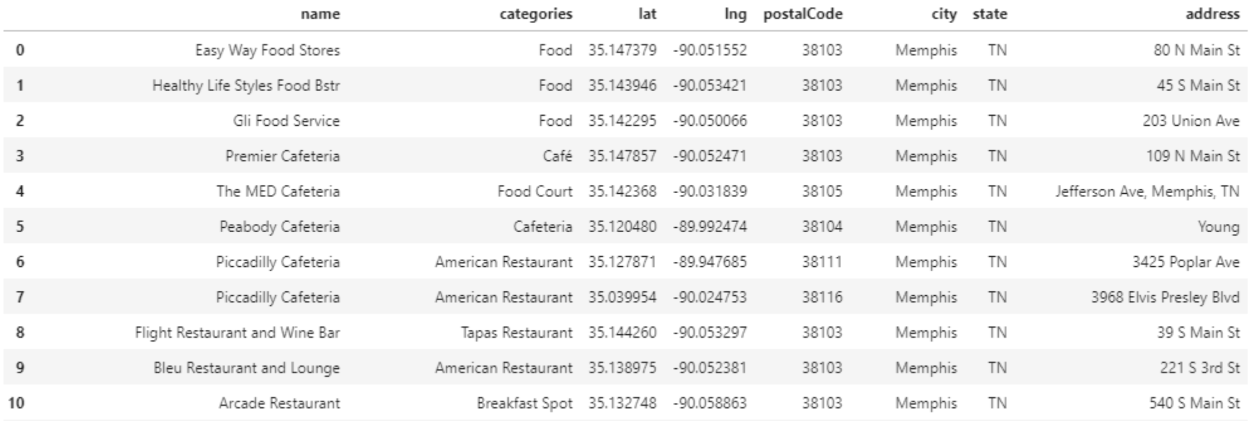
I grouped under 4 clusters as below:

*3.1 Cluster 1*: Combining Food,Cafeteria and Restaurant venues :

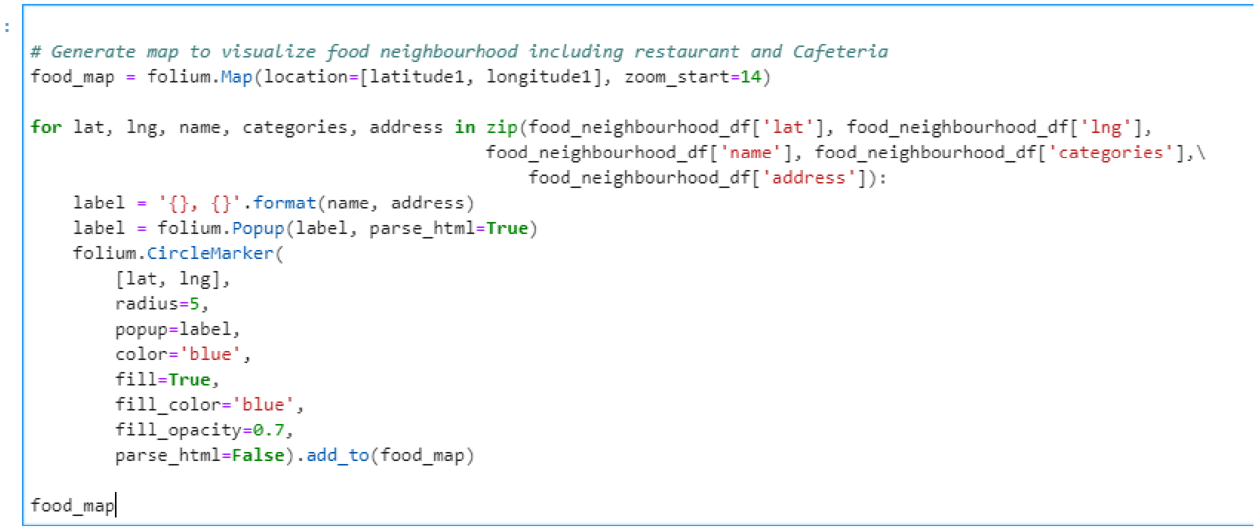
- Creating dataframe of food , Cafeteria and Restaurant by using 'concat' syntax
- Generating map to view the multi-category combined by using Folium



```
# creating dataframe of food, Cafeteria and Restaurant
food_neighbourhood_df = pd.concat([df_food, df_Cafeteria, df_Restaurant], ignore_index=True)
food_neighbourhood_df
```



| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 0 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 1 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 2 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |
| 3 | Premier Cafeteria | Café | 35.147857 | -90.052471 | 38103 | Memphis | TN | 109 N Main St |
| 4 | The MED Cafeteria | Food Court | 35.142368 | -90.031839 | 38105 | Memphis | TN | Jefferson Ave, Memphis, TN |
| 5 | Peabody Cafeteria | Cafeteria | 35.120480 | -89.992474 | 38104 | Memphis | TN | Young |
| 6 | Piccadilly Cafeteria | American Restaurant | 35.127871 | -89.947685 | 38111 | Memphis | TN | 3425 Poplar Ave |
| 7 | Piccadilly Cafeteria | American Restaurant | 35.039954 | -90.024753 | 38116 | Memphis | TN | 3968 Elvis Presley Blvd |
| 8 | Flight Restaurant and Wine Bar | Tapas Restaurant | 35.144260 | -90.053297 | 38103 | Memphis | TN | 39 S Main St |
| 9 | Bleu Restaurant and Lounge | American Restaurant | 35.138975 | -90.052381 | 38103 | Memphis | TN | 221 S 3rd St |
| 10 | Arcade Restaurant | Breakfast Spot | 35.132748 | -90.058863 | 38103 | Memphis | TN | 540 S Main St |



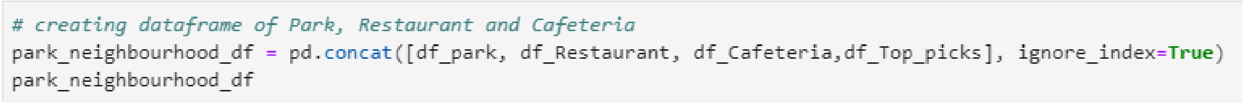
```
:
# Generate map to visualize food neighbourhood including restaurant and Cafeteria
food_map = folium.Map(location=[latitude1, longitude1], zoom_start=14)

for lat, lng, name, categories, address in zip(food_neighbourhood_df['lat'], food_neighbourhood_df['lng'],
                                           food_neighbourhood_df['name'], food_neighbourhood_df['categories'],\
                                               food_neighbourhood_df['address']):
    label = '{}, {}'.format(name, address)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7,
        parse_html=False).add_to(food_map)

food_map
```

just for cross checking , the number of rows obtained by using dataframe.shape:

50 rows x 8 columns

**<u>Similarly … for …</u>**

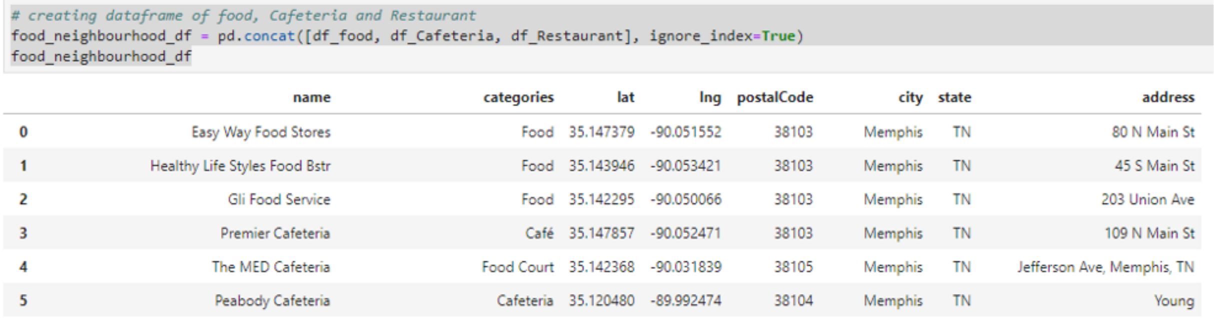
*<u>Cluster 2:</u>* Combining Park,Restaurant,Cafeteria and my Top Pick-Museum venues:



```
# creating dataframe of Park, Restaurant and Cafeteria
park_neighbourhood_df = pd.concat([df_park, df_Restaurant, df_Cafeteria,df_Top_picks], ignore_index=True)
park_neighbourhood_df
```

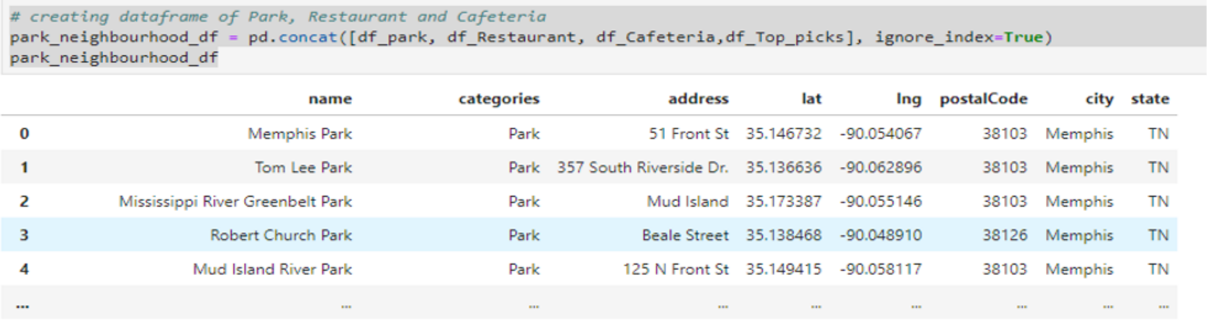
66 rows × 8 columns

# 1. Food cluster & 2. Park cluster



```
# creating dataframe of food, Cafeteria and Restaurant
food_neighbourhood_df = pd.concat([df_food, df_Cafeteria, df_Restaurant], ignore_index=True)
food_neighbourhood_df
```

| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 0 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 1 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 2 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |
| 3 | Premier Cafeteria | Café | 35.147857 | -90.052471 | 38103 | Memphis | TN | 109 N Main St |
| 4 | The MED Cafeteria | Food Court | 35.142368 | -90.031839 | 38105 | Memphis | TN | Jefferson Ave, Memphis, TN |
| 5 | Peabody Cafeteria | Cafeteria | 35.120480 | -89.992474 | 38104 | Memphis | TN | Young |

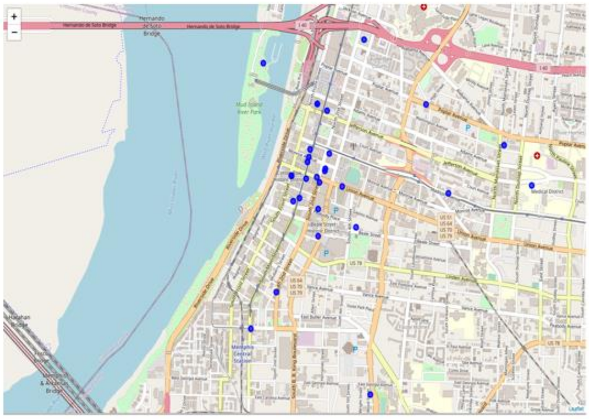


```
# creating dataframe of Park, Restaurant and Cafeteria
park_neighbourhood_df = pd.concat([df_park, df_Restaurant, df_Cafeteria,df_Top_picks], ignore_index=True)
park_neighbourhood_df
```

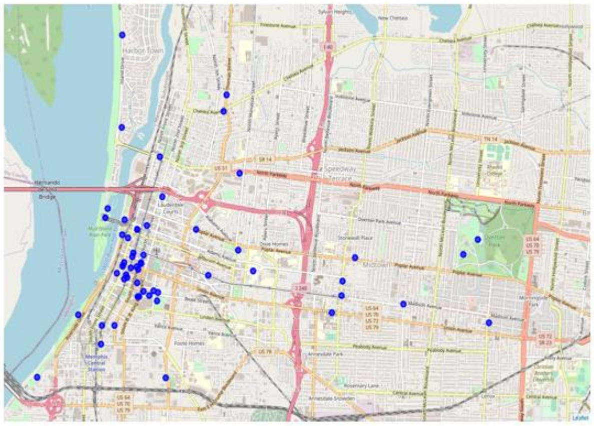
| | name | categories | address | lat | lng | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|
| 0 | Memphis Park | Park | 51 Front St | 35.146732 | -90.054067 | 38103 | Memphis | TN |
| 1 | Tom Lee Park | Park | 357 South Riverside Dr. | 35.136636 | -90.062896 | 38103 | Memphis | TN |
| 2 | Mississippi River Greenbelt Park | Park | Mud Island | 35.173387 | -90.055146 | 38103 | Memphis | TN |
| 3 | Robert Church Park | Park | Beale Street | 35.138468 | -90.048910 | 38126 | Memphis | TN |
| 4 | Mud Island River Park | Park | 125 N Front St | 35.149415 | -90.058117 | 38103 | Memphis | TN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

## 3.Clustering: Creating aerial / geographical location map view- for multi clusters combination

### 3.1 Cluster 1: Food-Combining Food, Cafeteria and Restaurant venues



### 3.2 Cluster 2: Park- Combining Park, Restaurant, Cafeteria and Top Pick venues



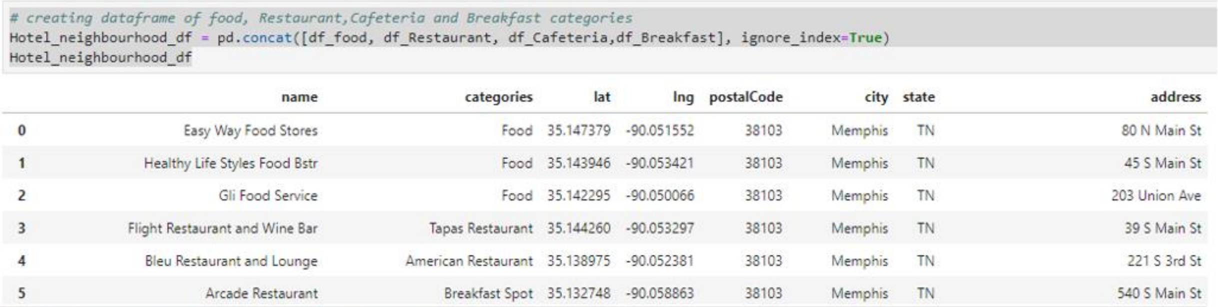
*Cluster 3*: Combining Food,Restaurant,Cafeteria and Breakfast venues:

51rows x 8 columns

*Cluster 4:* Combining Fun-Gym/Fitness Center , my Top Pick-Museum and Nightlife venues :

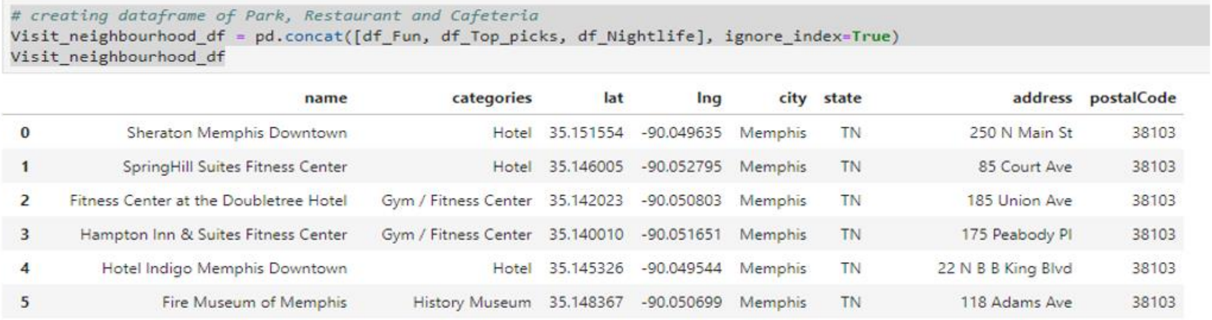
16 rows x 8 columns

# 3. Hotel cluster & 4. Visit cluster



```
# creating dataframe of food, Restaurant,Cafeteria and Breakfast categories
Hotel_neighbourhood_df = pd.concat([df_food, df_Restaurant, df_Cafeteria,df_Breakfast], ignore_index=True)
Hotel_neighbourhood_df
```

| | name | categories | lat | lng | postalCode | city | state | address |
|---|---|---|---|---|---|---|---|---|
| 0 | Easy Way Food Stores | Food | 35.147379 | -90.051552 | 38103 | Memphis | TN | 80 N Main St |
| 1 | Healthy Life Styles Food Bstr | Food | 35.143946 | -90.053421 | 38103 | Memphis | TN | 45 S Main St |
| 2 | Gli Food Service | Food | 35.142295 | -90.050066 | 38103 | Memphis | TN | 203 Union Ave |
| 3 | Flight Restaurant and Wine Bar | Tapas Restaurant | 35.144260 | -90.053297 | 38103 | Memphis | TN | 39 S Main St |
| 4 | Bleu Restaurant and Lounge | American Restaurant | 35.138975 | -90.052381 | 38103 | Memphis | TN | 221 S 3rd St |
| 5 | Arcade Restaurant | Breakfast Spot | 35.132748 | -90.058863 | 38103 | Memphis | TN | 540 S Main St |

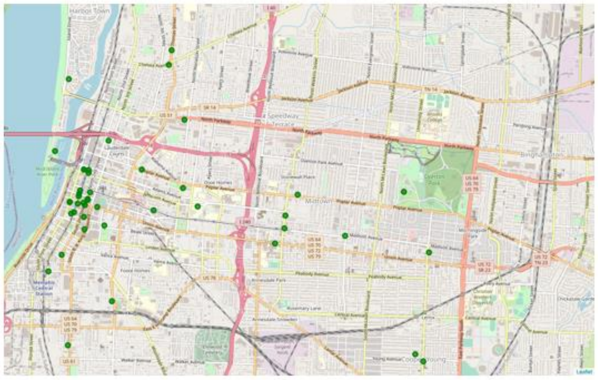


```
# creating dataframe of Park, Restaurant and Cafeteria
Visit_neighbourhood_df = pd.concat([df_Fun, df_Top_picks, df_Nightlife], ignore_index=True)
Visit_neighbourhood_df
```

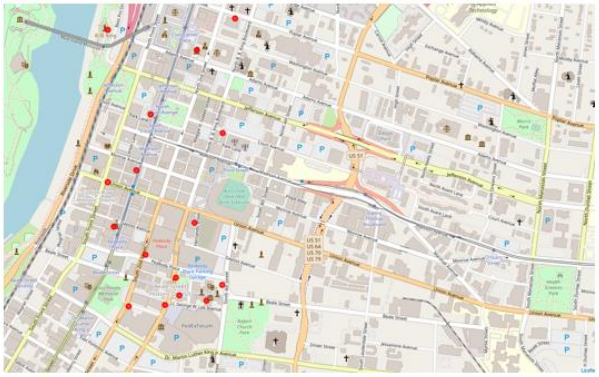
| | name | categories | lat | lng | city | state | address | postalCode |
|---|---|---|---|---|---|---|---|---|
| 0 | Sheraton Memphis Downtown | Hotel | 35.151554 | -90.049635 | Memphis | TN | 250 N Main St | 38103 |
| 1 | SpringHill Suites Fitness Center | Hotel | 35.146005 | -90.052795 | Memphis | TN | 85 Court Ave | 38103 |
| 2 | Fitness Center at the Doubletree Hotel | Gym / Fitness Center | 35.142023 | -90.050803 | Memphis | TN | 185 Union Ave | 38103 |
| 3 | Hampton Inn & Suites Fitness Center | Gym / Fitness Center | 35.140010 | -90.051651 | Memphis | TN | 175 Peabody Pl | 38103 |
| 4 | Hotel Indigo Memphis Downtown | Hotel | 35.145326 | -90.049544 | Memphis | TN | 22 N B B King Blvd | 38103 |
| 5 | Fire Museum of Memphis | History Museum | 35.148367 | -90.050699 | Memphis | TN | 118 Adams Ave | 38103 |

## 3.Clustering: Creating aerial / geographical location map view- for multi clusters combination

### 3.3 Cluster 3: Hotel- Combining Food, Restaurant, Cafeteria and Breakfast venues:



### 3.4 Cluster 4: Visit-Combining Fun-Gym/Fitness center, Top Pick-Museum and Nightlife venues :



## 4.Conclusion:

Above exercise carried out as a pilot clustering and view with combined option for one city, keeping in mind that to be extended with multi cities combination or for comparison for user need for quicker overall view and decision. ( In 2.1 - we had created for 2 cities as an example with url1 & url2 , but 2.2 onwards for easier working , url1 used as url only but with same latitude & longitude )

### *4.1 Assumption* :

Though above exercise limited with clustering concept with four square API and we will be noticing that initially all libraries are imported for k-means also, may be used for future deep learning.

May also be noted that -NaN ( Not a Number) data as fetched from url and got removed based on syntax and not equated or filled with assumed or filled or copied data.

Category names Food, Park, Cafeteria, Nightlife, Fun & Top Pick are based on Four Square. ( Shopping-excluded for this proto-type project )

Catalyst LLC is my dream organization and named for this project learning only.

**GitHub -my repositories ref**:

https://github.com/geethanarayanan/IBM-Datascience-Capstone-Project

https://github.com/geethanarayanan/IBM-Datascience-Capstone-Project/blob/master/wk%205-Capstone%20project-The%20Battle%20of%20Neighborhoods%20(Week%202)%20(1).ipynb

https://github.com/geethanarayanan/IBM-Datascience-Capstone-Project/blob/master/Coursera-IBM%20Applied%20Datascience-Capstone%20learning%20project.pps

https://github.com/geethanarayanan/IBM-Datascience-Capstone-Project/blob/master/Coursera-IBM%20Applied%20Datascience-Capstone%20learning%20project.pptx

## 5. Thanking:

The codes are most of part of Coursera -IBM data science lab exercises and learning and my sincere thanks to all *instructors and IBM for structured coaching and guidelines*.

Also thanking our learning *peers / data scientists* for timely evaluation, sharing and support.

Thanks to all peers and Instructors !

- Geetha Narayanan