The Battle of Neighborhoods: Coursera Capstone Project

Food taste in Music City Nashville, TN

Anuj Tripathi

Table of Contents

Introduction:

Nashville is the capital of the U.S. state of Tennessee and called as Music City due to the epicenter of country and western music in the United States. Today, Nashville-Tn not only music city, it is a leader in health care industry with more than 500 health care companies, operated from Nashville.

The Nashville Convention & Visitors Bureau have reported that the tourism industry broke another record, with 16.1 million visitors to the city in 2019. Throughout the whole year, tourism spending in Nashville topped more than $7 billion. Due to these specialties, now Nashville was recently named one of 8 cities with booming entrepreneur communities, mostly due to music, food scene and healthcare industry ties.

Different taste cuisine and fast food center have targeted whole year by music lovers, travelers and foodies. So, currently a lots of diverse culture foods are available and every year more than 100 new restaurants and bars opened in Nashville, TN.

Target Audience:

This information provided by this report would be useful for people who are :-

- ✓ In Nashville and Pizza Lover
- ✓ Business personnel who wants to invest or open a Pizza place / restaurant.
- ✓ First time visitor of Nashville, TN, finding the best location for different taste of foods in Music city, Nashville -TN.

Problem Statement:

1. Where and What no of pizza place in Nashville and its neighbors?
2. In what Neighborhood what taste of restaurant is good?

Data Description:

To consider the objective stated above, we can list the below data sources used for the analysis.

A. Nashville, TN PostalCode list with Boroughs, Neighborhoods with their Postal code.

Data source: **PostalCode_Nashville_Tn.xlsx**

Description: This data set contains the required information. And we will use this data set to explore various neighborhoods of Nashville, TN. Previously scraped Wikipedia and make .xlsx file.

B. GeoSpace data for latitude and longitude of US Zip code.

Data source: **us-zip-code-latitude-and-longitude.xlsx**

Description: US Zip Code Latitude and Longitude open data (https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/export/)

C. Fousquare API for search foods in each neighborhood of Nashville, TN.

Data source: **Fousquare API**

Description: Fousquare API for search food venues in each neighborhood of Nashville, TN (https://foursquare.com/).

Methodology

(I) Data Preparation

The data for this project will be extracted, processed and analyzed by integrating the borough information for Nashville imported from the .xlsx file from public data domain and venue related information acquired through Foursquare API. Using the pandas library, the data will be cleaned and processed to prepare a final data frame for analysis. In order to render my data onto a map, I will be using the Folium library. Also, to create clusters of similar regions of interest, I will be using k-means clustering technique. For this analytical method, I will be utilizing the sklearn library of python, to create a clustering model for my project.

(A) Environment Preparation

```
]: import numpy as np # library to handle data in a vectorized manner
   from bs4 import BeautifulSoup
   import pandas as pd # library for data analsysis
   pd.set_option('display.max_columns', None)
   pd.set_option('display.max_rows', None)

   import json # library to handle JSON files

   !conda install -c conda-forge geopy --yes
   from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

   import requests # library to handle requests
   from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

   # Matplotlib and associated plotting modules
   import matplotlib.cm as cm
   import matplotlib.colors as colors

   # import k-means from clustering stage
   from sklearn.cluster import KMeans

   !conda install -c conda-forge folium=0.5.0 --yes
   import folium # map rendering library

   print('Libraries imported.')
```

(B) Data import from data .xlsx file and convert in to a Data Frame

```
3]: df_Nashville = pd.read_excel (r'C:\Users\Dr Anuj Tripathi\Desktop\PostalCode_Nashville_Tn.xlsx')## from nashville postal site
    print (df_Nashville)
```

	PostalCode	Borough	Neighborhood
0	37013	Nashville	Antioch
1	37205	Belle Meade	Belle Meade Plantation
2	37221	Nashville	Bellevue
3	37207	Williamson	Brentwood
4	37206	Nashville	East Nashville
5	37064	Williamson	Franklin
6	37214	Donelson	Gaylord Opryland
7	37130	Murfreesboro	Geographic Center of Tennessee

(II) Data Processing

(A) Cleaning the Data

In this step, the data frame will be cleaned with respect to missing values, error data values and shall be transformed into a more workable framework for the analytical and machine learning logarithms.

In [4]:
```
not_assigned_boroughs = df_Nashville.index[df_Nashville['Borough'] == 'Not assigned']
not_assigned_neighborhoods = df_Nashville.index[df_Nashville['Neighborhood'] == 'Not assigned']
not_assigned_neighborhoods_and_borough = not_assigned_boroughs & not_assigned_neighborhoods

print('The DataFrame shape is {}'.format(df_Nashville.shape),'\n')
print('There are:')
print('  {} PostalCodes'.format(df_Nashville['PostalCode'].unique().shape[0]))
print('  {} Boroughs'.format(df_Nashville['Borough'].unique().shape[0] - 1)) # substract one because "not assigned" doesn't coun
t
print('  {} Neighborhoods'.format(df_Nashville['Neighborhood'].unique().shape[0] - 1)) # substract one because "not assigned" do
esn't count
```

```
The DataFrame shape is (27, 3)

There are:
  26 PostalCodes
  7 Boroughs
  26 Neighborhoods
```

In [6]:
```
group = df_Nashville.groupby('PostalCode')
grouped_neighborhoods = group['Neighborhood'].apply(lambda x: "%s" % ', '.join(x))
grouped_boroughs = group['Borough'].apply(lambda x: set(x).pop())
grouped_df = pd.DataFrame(list(zip(grouped_boroughs.index, grouped_boroughs, grouped_neighborhoods)))
grouped_df.columns = ['PostalCode', 'Borough', 'Neighborhood']

grouped_df.head(5)
```

Out[6]:

(B) Merging Geographical Coordinates through another data file

In [8]:
```
# US Zip Code Latitude and Longitude data excel(https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitu
de/export/)

coordinates_df = pd.read_excel (r'C:\Users\Dr Anuj Tripathi\Desktop\us-zip-code-latitude-and-longitude.xlsx')

coordinates_df.head()
```

Out[8]:

	PostalCode	City	State	Latitude	Longitude	Timezone	Daylight savings time flag	geopoint
0	71937	Cove	AR	34.398483	-94.39398	-6	1	34.398483, -94.39398
1	72044	Edgemont	AR	35.624351	-92.16056	-6	1	35.624351, -92.16056
2	56171	Sherburn	MN	43.660847	-94.74357	-6	1	43.660847, -94.74357
3	49430	Lamont	MI	43.010337	-85.89754	-5	1	43.010337, -85.89754
4	52585	Richland	IA	41.194129	-91.98027	-6	1	41.194129, -91.98027

In [9]:
```
coordinates_df = coordinates_df.drop(["Timezone","geopoint","State","Daylight savings time flag","City"],axis=1)
coordinates_df.head()
```

Out[9]:

```
df_Nashville_Tn = pd.merge(grouped_df, coordinates_df, how='left', left_on = 'PostalCode', right_on = 'PostalCode')
# remove the "PostalCode" column
df_Nashville_Tn.head(26)
```

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	37013	Nashville	Antioch	36.055115	-86.647820
1	37064	Williamson	Franklin	35.893823	-86.899190
2	37072	Sumner	Goodlettsville	36.354650	-86.718790
3	37075	Sumner	Hendersonville	36.311047	-86.611730
4	37076	Nashville	The Hermitage	36.180507	-86.601110

(III) Data Exploration

(A) Rendering the Map and Adding the Boroughs

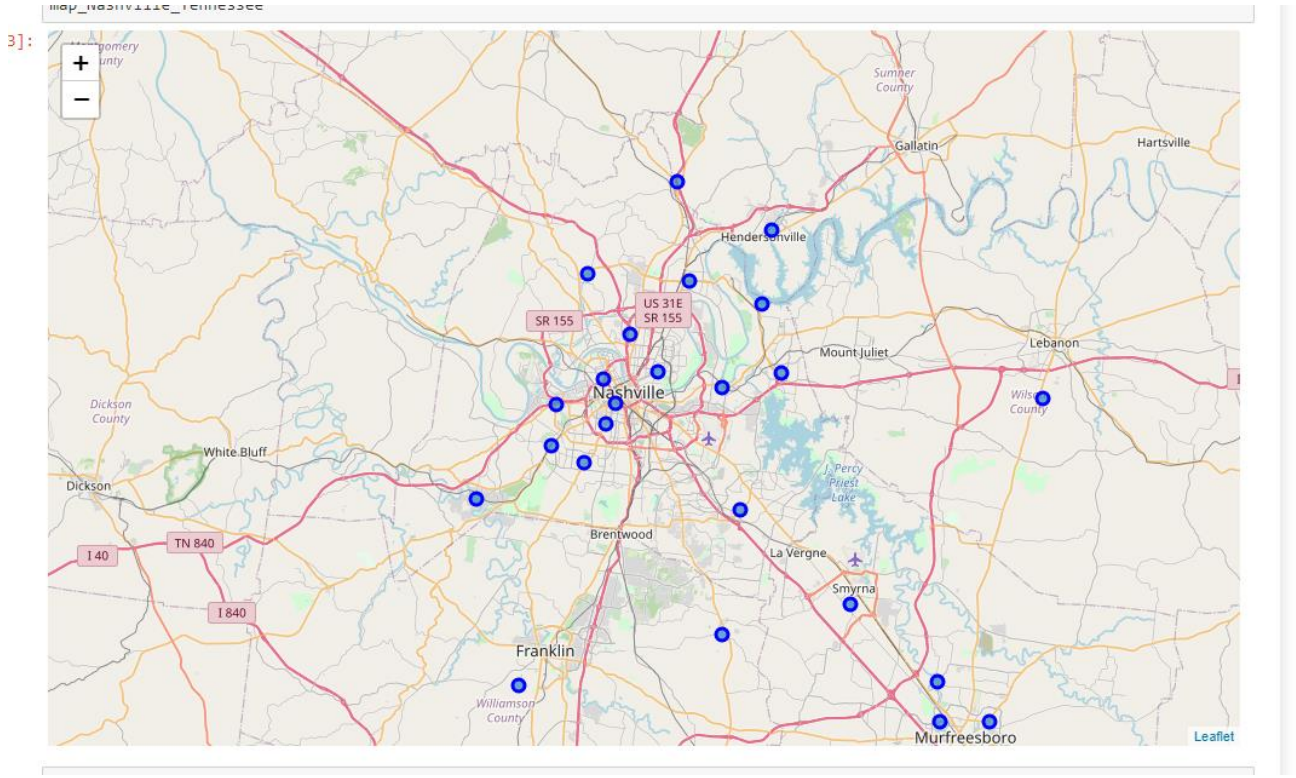
We then use the python **folium** library to visualize geographic details of Nashville, TN and its Neighbor boroughs. I created a map of Nashville, TN with boroughs superimposed on top using the latitude and longitude values to get the visual as below:

```
address = "Nashville Tennessee, ON"

geolocator = Nominatim(user_agent="Nashville Tennessee_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Nashville Tennessee are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Nashville Tennessee are 36.17405235, -86.76364612210365.
```

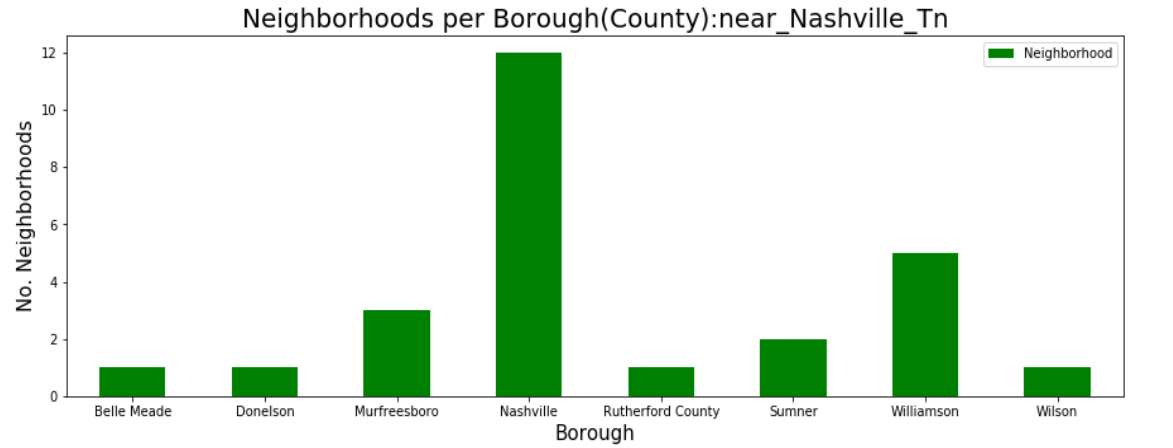
```
# create map of Nashville using latitude and longitude values
map_Nashville_Tennessee = folium.Map(location=[latitude, longitude], zoom_start=10)
map_Nashville_Tennessee
```

```
for lat, lng, borough, neighborhood in zip(
        df_Nashville_Tn['Latitude'],
        df_Nashville_Tn['Longitude'],
        df_Nashville_Tn['Borough'],
        df_Nashville_Tn['Neighborhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_Nashville_Tennessee)

map_Nashville_Tennessee
```



(B) plot bar diagram to visualize Neighborhood per Borough in Nashville city.

```
]: import matplotlib.pyplot as plt
   clr = "green"
   df_Nashville_Tn.groupby('Borough')['Neighborhood'].count().plot.bar(figsize=(15,5), color=clr)
   plt.title('Neighborhoods per Borough(County):near_Nashville_Tn', fontsize = 20)
   plt.xlabel('Borough', fontsize = 15)
   plt.ylabel('No. Neighborhoods',fontsize = 15)
   plt.xticks(rotation = 'horizontal')
   plt.legend()
   plt.show()
```



(C) Explore Pizza Place in the Borough of Nashville, TN using Foursqaure API

[16]:
```
CLIENT_ID = '[redacted]' # your Foursquare ID
CLIENT_SECRET = '[redacted]' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
```

```
Your credentails:
CLIENT_ID: H[redacted]
```

[17]:
```
LIMIT = 10 # Maximum is 100
cities = ["Nashville, TN", 'Belle Meade, TN', 'Brentwood, TN', 'Franklin, TN', 'Goodlettsville, TN', 'Mount Juliet, TN', 'Murfre
esboro, TN', 'Spring Hill, TN', 'Smyrna, TN']
results = {}
for city in cities:
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&near={}&limit={}&categoryId={}'.form
at(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        city,
        LIMIT,
        "4bf58dd8d48988d1ca941735") # PIZZA PLACE CATEGORY ID
    results[city] = requests.get(url).json()
```

[18]:
```
df_venues={}
for city in cities:
    venues = json_normalize(results[city]['response']['groups'][0]['items'])
    df_venues[city] = venues[['venue.name', 'venue.location.address', 'venue.location.lat', 'venue.location.lng']]
    df_venues[city].columns = ['Name', 'Address', 'Lat', 'Lng']
```

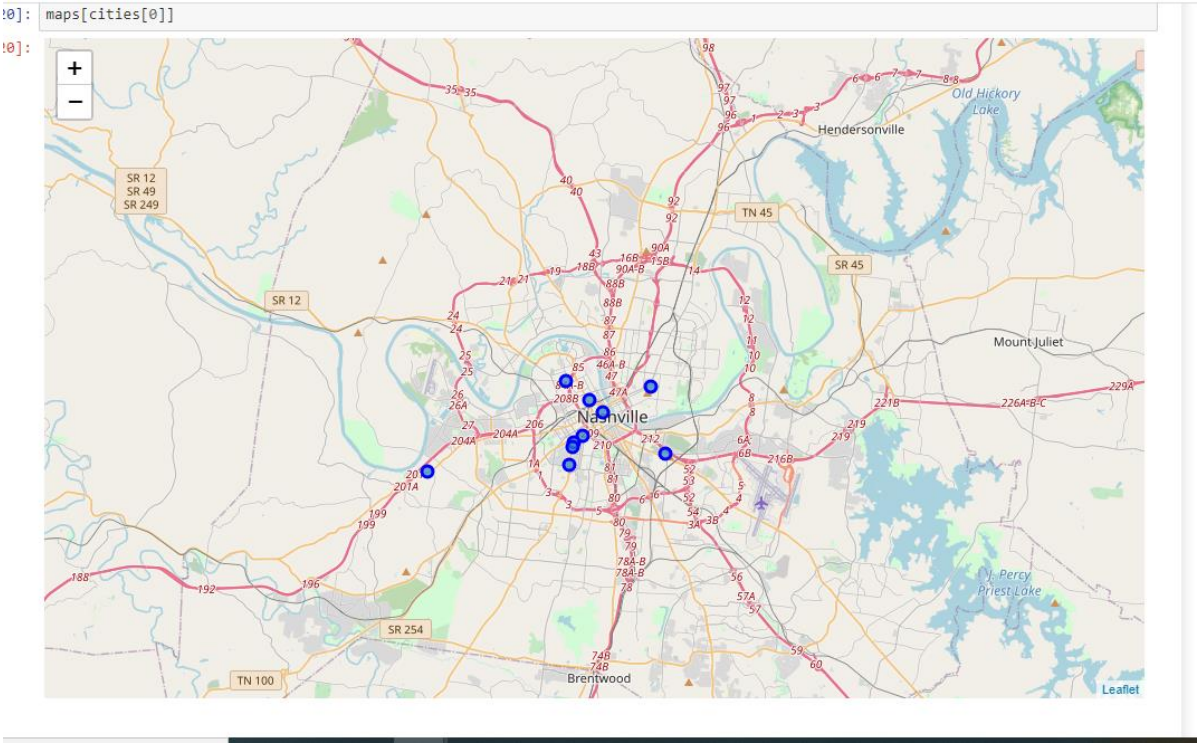
]:
```
maps = {}
for city in cities:
    city_lat = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lat'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lat']])
    city_lng = np.mean([results[city]['response']['geocode']['geometry']['bounds']['ne']['lng'],
                        results[city]['response']['geocode']['geometry']['bounds']['sw']['lng']])
    maps[city] = folium.Map(location=[city_lat, city_lng], zoom_start=11)

    # add markers to map
    for lat, lng, label in zip(df_venues[city]['Lat'], df_venues[city]['Lng'], df_venues[city]['Name']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='#3186cc',
            fill_opacity=0.7,
            parse_html=False).add_to(maps[city])
    print(f"Total number of pizza places in {city} = ", results[city]['response']['totalResults'])
    print("Showing Top 10")
```

```
Total number of pizza places in Nashville, TN =  123
Showing Top 10
Total number of pizza places in Belle Meade, TN =  18
Showing Top 10
Total number of pizza places in Brentwood, TN =  13
Showing Top 10
Total number of pizza places in Franklin, TN =  46
Showing Top 10
Total number of pizza places in Goodlettsville, TN =  34
Showing Top 10
Total number of pizza places in Mount Juliet, TN =  17
Showing Top 10
Total number of pizza places in Murfreesboro, TN =  64
Showing Top 10
Total number of pizza places in Spring Hill, TN =  13
Showing Top 10
Total number of pizza places in Smyrna, TN =  14
Showing Top 10
```

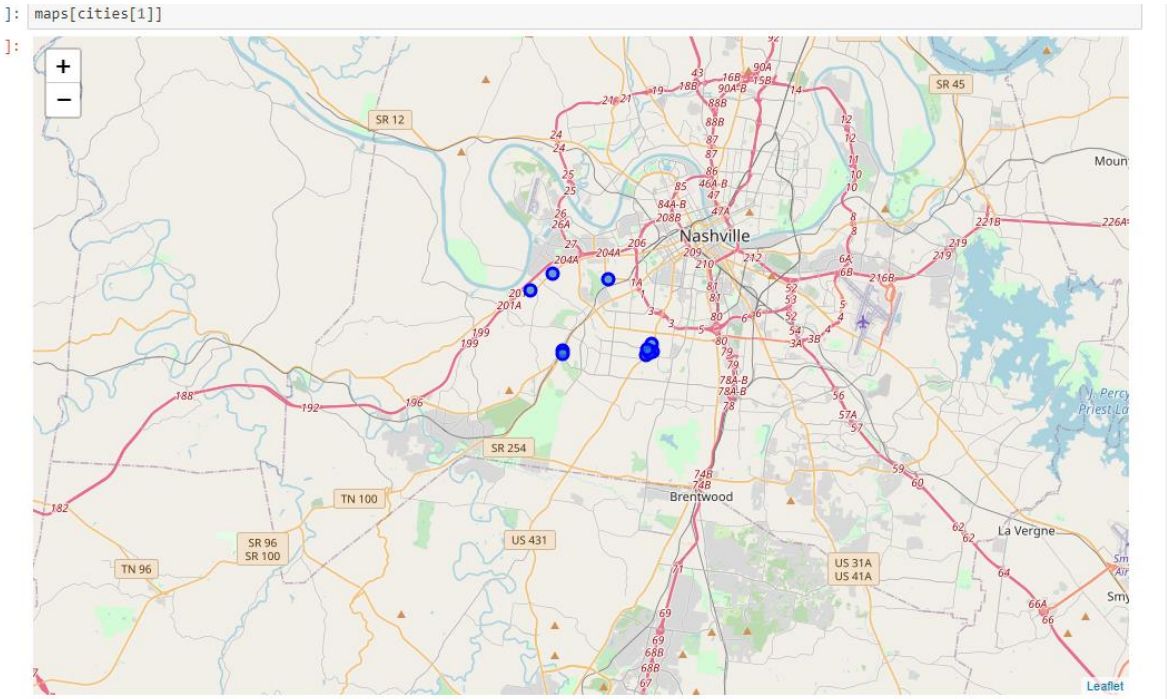
Visualized Top 10 Pizza Place in map of each Borough of Nashville, TN

In Foursquare API, we set the LIMIT parameter to **10**, which would limit the top10 pizza place as returned values. Using the python **folium** library to visualize the Top 10 Pizza Place of each Borough of Nashville, TN and Neighbors.

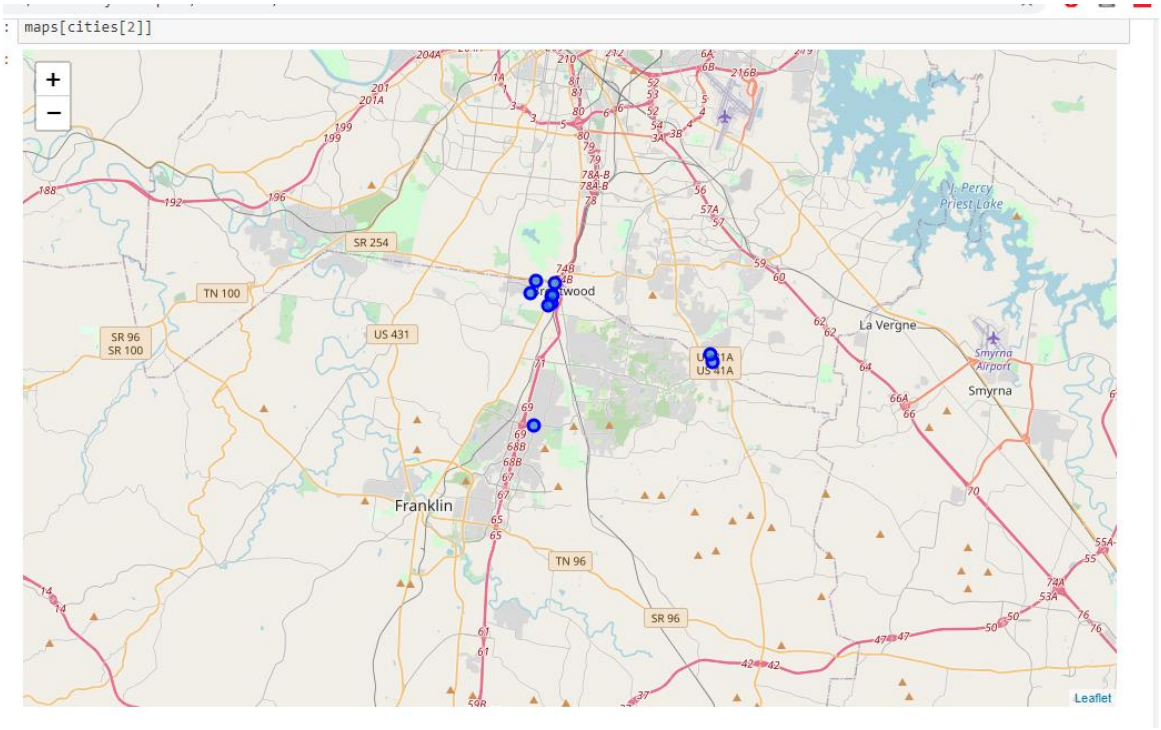
Top 10 Pizza Place in city Map of Nashville, TN



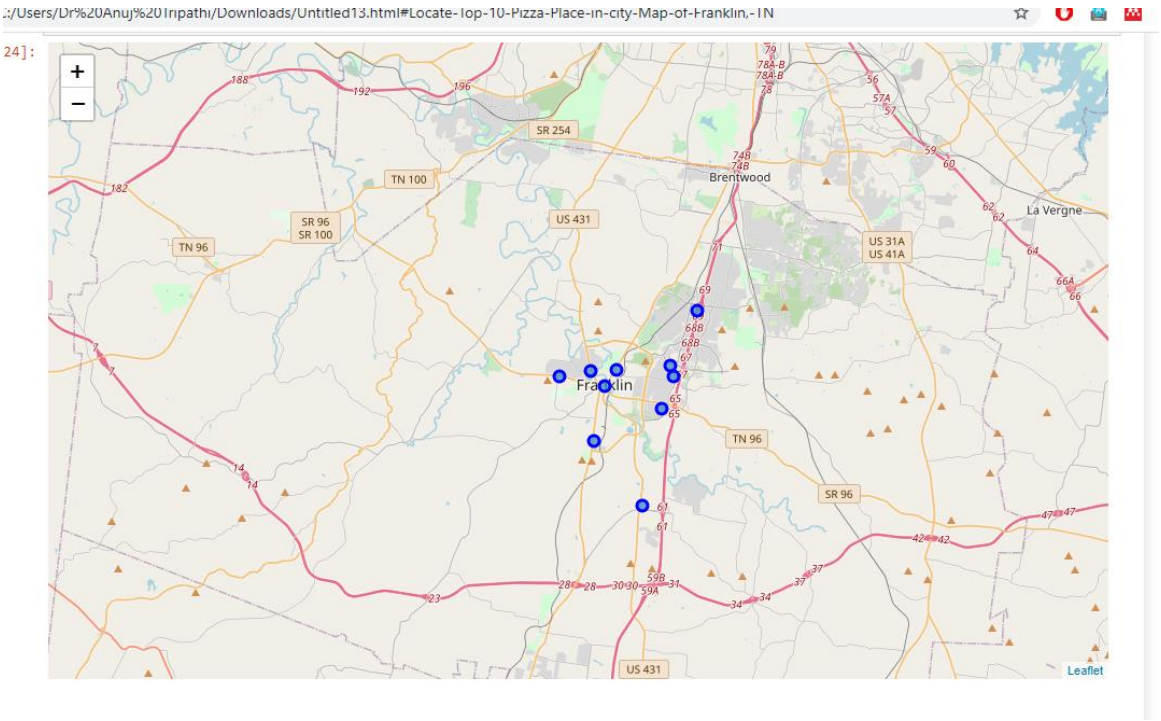
Top 10 Pizza Place in city Map of Belle Meade, TN



Top 10 Pizza Place in city Map of Brentwood, TN



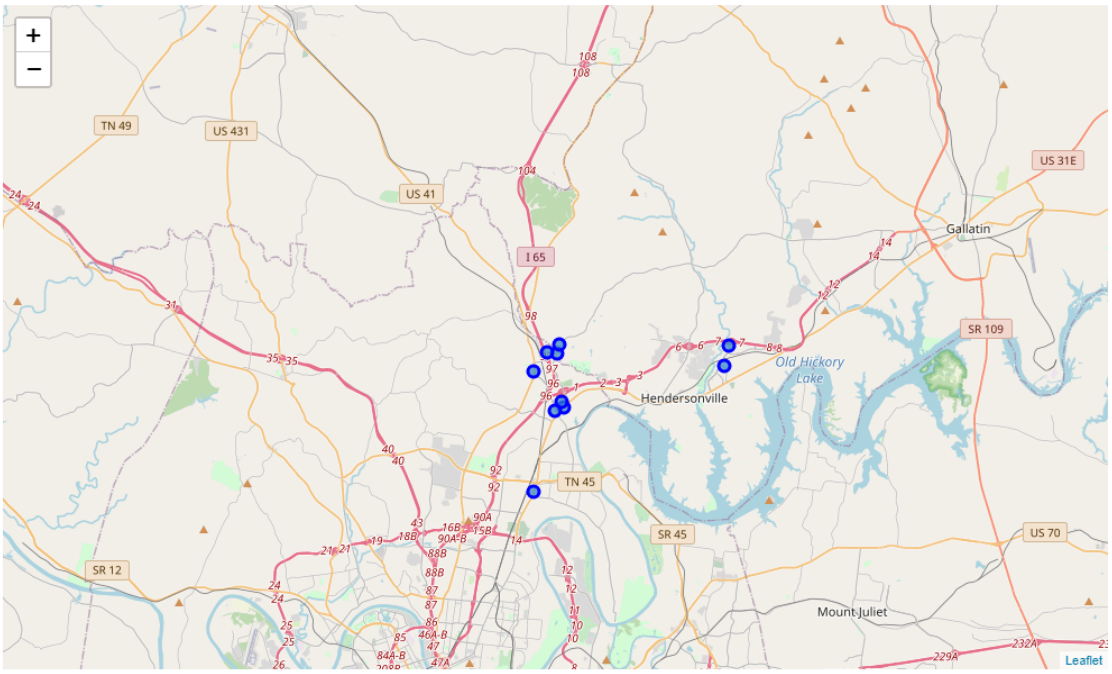
Top 10 Pizza Place in city Map of Franklin, TN



Top 10 Pizza Place in city Map of Goodlettsville, TN

```
5]: maps[cities[4]]
```

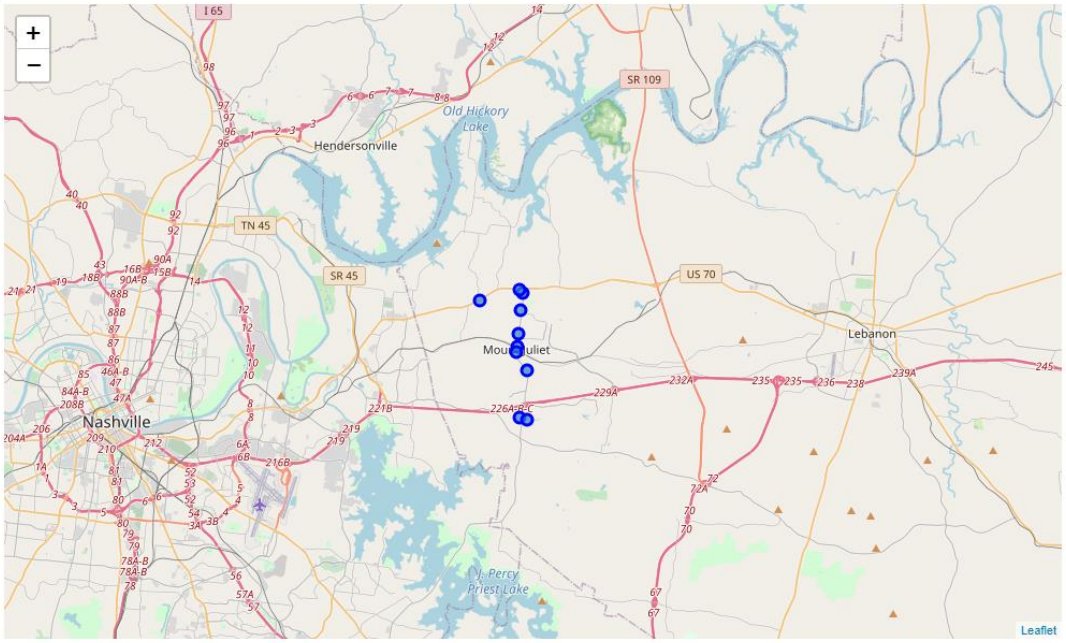
5]:



Top 10 Pizza Place in city Map of Mount Juliet, TN

```
26]: maps[cities[5]]
```

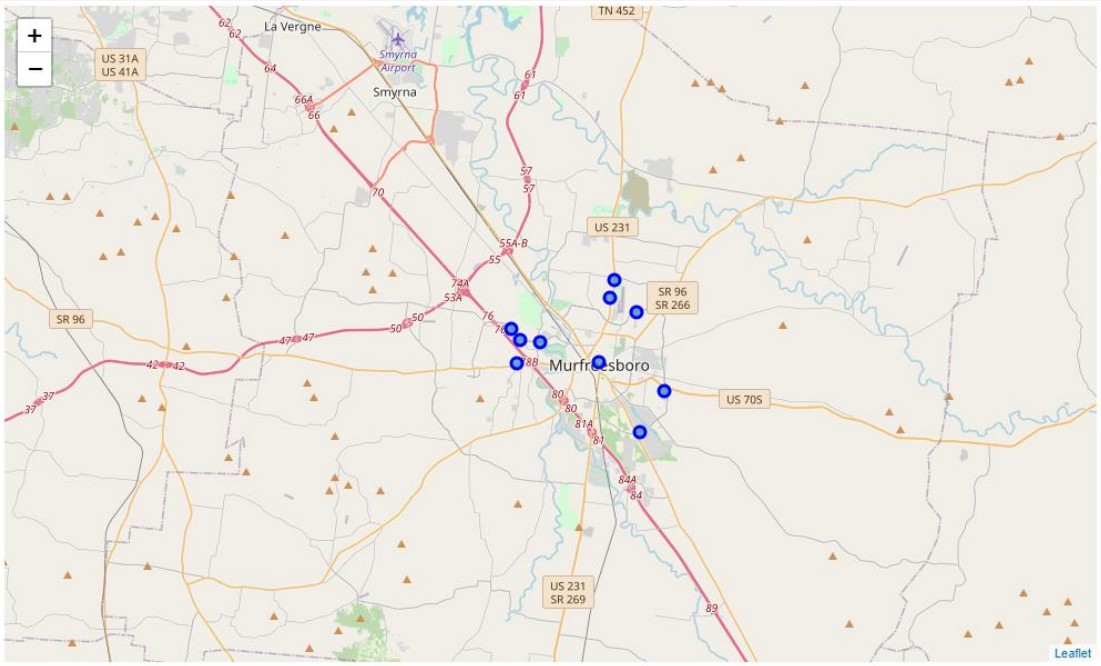
26]:



Top 10 Pizza Place in city Map of Murfreesboro, TN

```
[27]: maps[cities[6]]
```

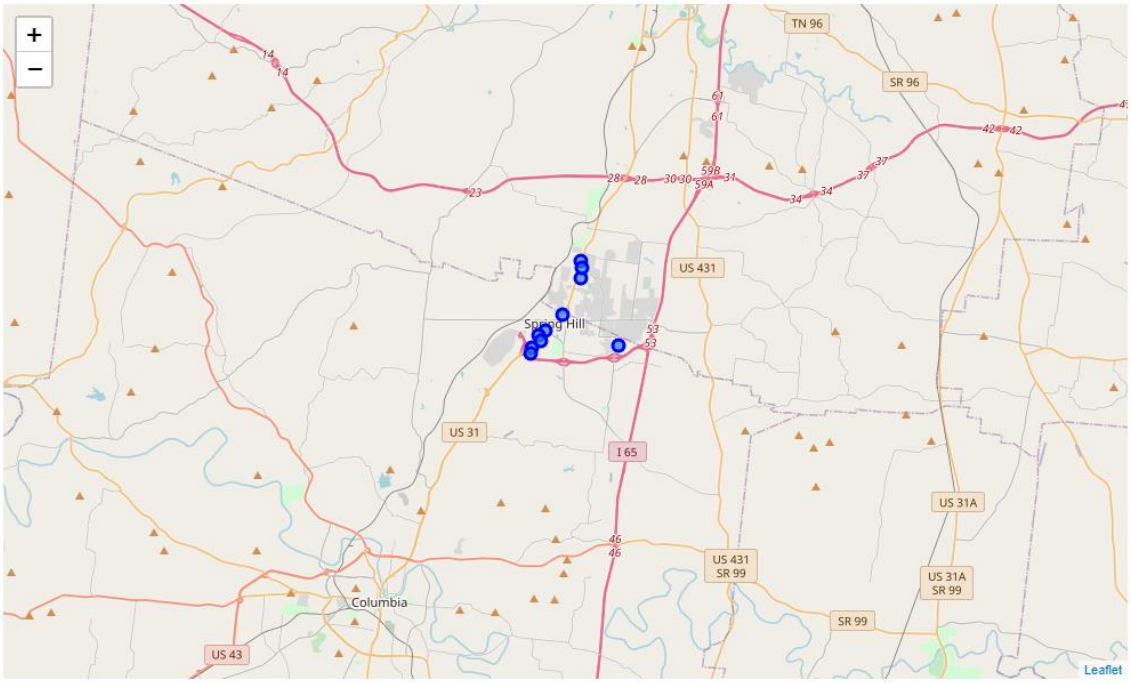
[27]:



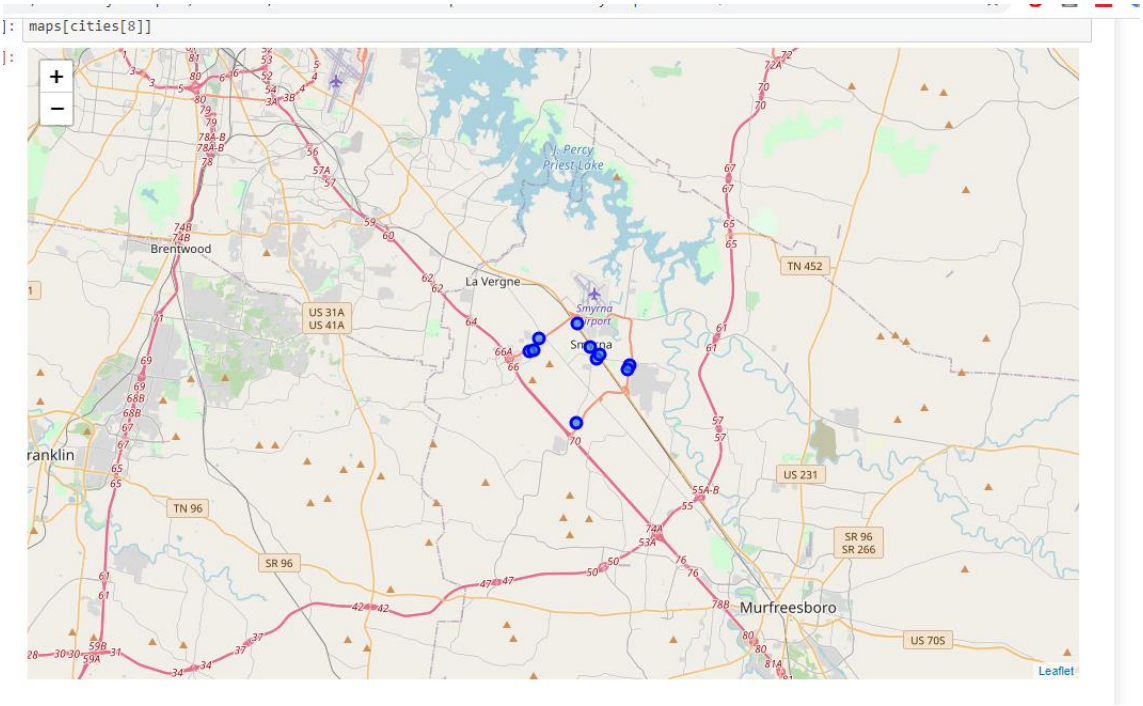
Top 10 Pizza Place in city Map of Spring Hill, TN

```
[28]: maps[cities[7]]
```

[28]:



Top 10 Pizza Place in city Map of Smyrna, TN



mean distance of top 10 Pizza Place from mean coordinate of each Borough of Nashville, TN

Here, we show how much top 10 Pizza of each Borough of Nashville is located. For this I will use some basic statistics. I will get the mean location of the pizza places which should be near to most of them if they are dense or far if not. Next I will take the average of the distance of the venues to the mean coordinates.

```
Nashville, TN
Mean Distance from Mean coordinates
0.029552093139911924
Belle Meade, TN
Mean Distance from Mean coordinates
0.0320715042526388
Brentwood, TN
Mean Distance from Mean coordinates
0.03955409706907507
Franklin, TN
Mean Distance from Mean coordinates
0.033816588198426784
Goodlettsville, TN
Mean Distance from Mean coordinates
0.0405341941817435
```

```
Mount Juliet, TN
Mean Distance from Mean coordinates
0.02084485389917041
Murfreesboro, TN
Mean Distance from Mean coordinates
0.038090735959114205
Spring Hill, TN
Mean Distance from Mean coordinates
0.021085356499334216
Smyrna, TN
Mean Distance from Mean coordinates
0.023821369467637125
```

(D) Exploring Music City Nashville, TN venues using Foursqaure API

```
9]: def getNearbyVenues(names, latitudes, longitudes, radius=1000):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.for
mat(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

```
Nashville_venues.head()
```

```
(710, 7)
```

```
]:
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Antioch	36.055115	-86.64782	Ford Ice Center	36.053014	-86.656621	Skating Rink
1	Antioch	36.055115	-86.64782	Zaxby's Chicken Fingers & Buffalo Wings	36.049856	-86.651551	Fried Chicken Joint
2	Antioch	36.055115	-86.64782	Waffle House	36.051700	-86.645400	Breakfast Spot
3	Antioch	36.055115	-86.64782	Thorntons	36.051673	-86.644662	Convenience Store
4	Antioch	36.055115	-86.64782	AMC Theater	36.050990	-86.651680	Movie Theater

Extracting only restaurants from venue category list and creating a dataframe "Nashville_restaurants"

We create a new data-frame "Nashville_restaurants that will contain only 24 unique categories ta ste restaurants for Nashville city. Here is a head () value of this dataframe.

```
75]: # one hot encoding
Nashville_onehot = pd.get_dummies(Nashville_restaurants[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
Nashville_onehot['Neighborhood'] = Nashville_restaurants['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [Nashville_onehot.columns[-1]] + list(Nashville_onehot.columns[:-1])
Nashville_onehot = Nashville_onehot[fixed_columns]

Nashville_onehot.head()
```

75]:

	Neighborhood	American Restaurant	Asian Restaurant	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	French Restaurant	Greek Restaurant	Indian Restaurant	Indonesian Restaurant	Italian Restaurant	Japanese Restaurant	Ame Resta
7	Antioch	0	1	0	0	0	0	0	0	0	0	0	
8	Antioch	0	0	0	0	0	0	0	0	0	0	0	
9	Antioch	0	0	0	0	0	0	0	0	0	0	0	
25	Antioch	1	0	0	0	0	0	0	0	0	0	0	
28	Antioch	0	0	0	0	1	0	0	0	0	0	0	

```
76]: print('There are {} restaurants in Nashville with {} different style of cuisines.'.format(Nashville_onehot.shape[0],(Nashville_onehot.shape[1]-1)))
```

```
There are 123 restaurants in Nashville with 24 different style of cuisines.
```

Grouping rows by neighborhood and taking the mean of the frequency of occurrence of each category/restaurants

```
]: Nashville_grouped = Nashville_onehot.groupby('Neighborhood').mean().reset_index()
Nashville_grouped
```

]:

	Neighborhood	American Restaurant	Asian Restaurant	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	French Restaurant	Greek Restaurant	Indian Restaurant	Indonesian Restaurant	Italian Restaurant	Japanese Restaurant	Ame Resta
0	Antioch	0.142857	0.142857	0.0000	0.000000	0.428571	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.1
1	Brentwood	0.000000	0.000000	0.0000	0.000000	1.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.0
2	East Nashville	0.312500	0.062500	0.0625	0.000000	0.000000	0.062500	0.0625	0.000000	0.000000	0.062500	0.0625	0.0
3	Gaylord Opryland	0.000000	0.000000	0.0000	0.000000	0.200000	0.000000	0.2000	0.000000	0.000000	0.000000	0.0000	0.0
4	Geographic Center of Tennessee	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.0
5	Germantown, North Nashville	0.166667	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.166667	0.0000	0.0
6	Green Hills	0.250000	0.250000	0.0000	0.000000	0.250000	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.0
7	Hendersonville	0.000000	0.000000	0.0000	0.000000	0.250000	0.000000	0.0000	0.000000	0.000000	0.000000	0.0000	0.0
	Hillsboro												

(D). Find Top 3 restaurants taste for each Neighborhood based on cuisine

```
: num_top_rest = 3

for hood in Nashville_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = Nashville_grouped[Nashville_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_rest))
    print('\n')
```

```
----Antioch----
                venue  freq
0  Fast Food Restaurant  0.43
1   American Restaurant  0.14
2            Restaurant  0.14
```

(E). Machine Learning Algorithm: Clustering and Segmentation of Neighborhoods with Similar Venues

Custering Neighborhoods using K-means

Finding the best k

9]:
```
# set number of clusters
kclusters = 7

Nashville_grouped_clustering = Nashville_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(Nashville_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

9]:
```
array([1, 2, 1, 1, 4, 1, 1, 1, 1, 3, 6, 1, 5, 1, 1, 0, 1])
```

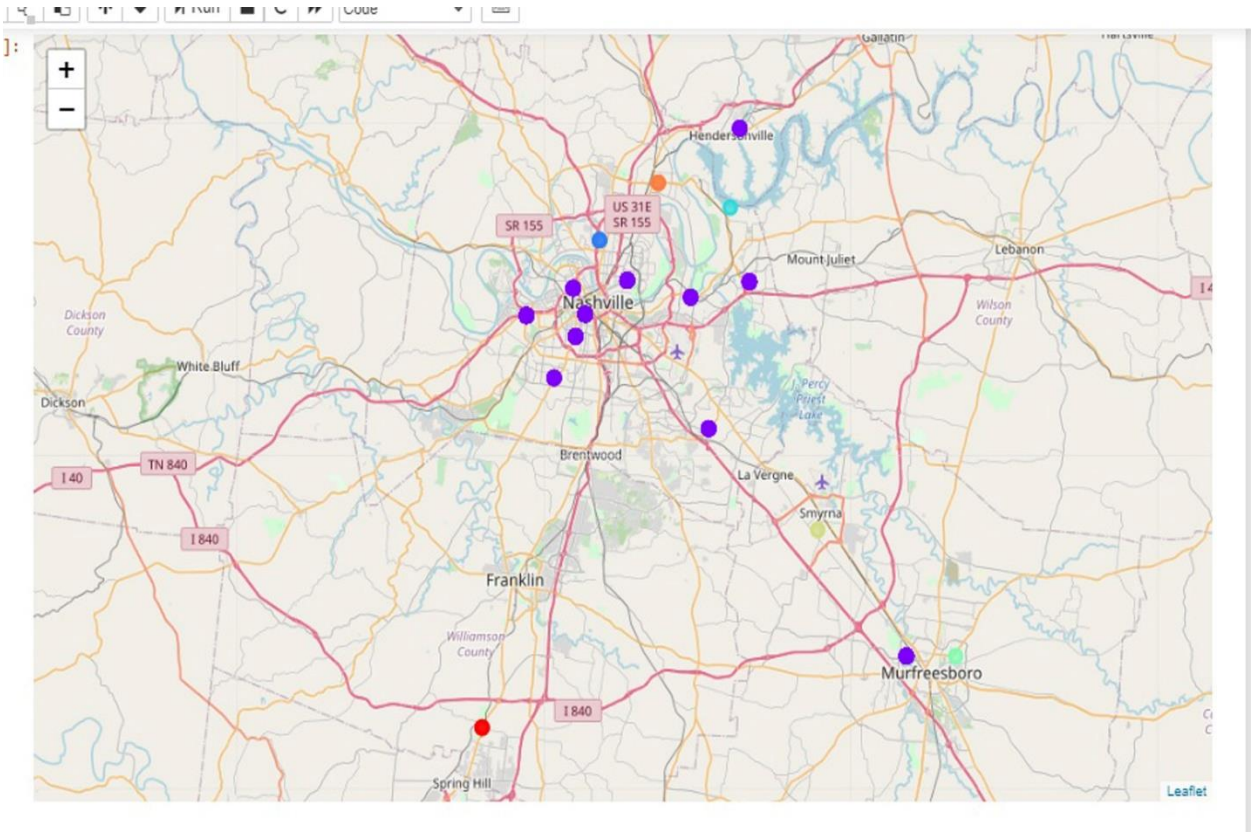
0]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

Nashville_merged = Nashville_restaurants

# merge Nashville_grouped with Nashville_data to add latitude/longitude for each neighborhood
Nashville_merged = Nashville_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')
Nashville_merged.fillna(0)
Nashville_merged.head() # check the last columns!
```

0]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo Commo Venu
7	Antioch	36.055115	-86.64782	Kirin Sushi	36.051159	-86.649784	Asian Restaurant	1	Fast Food Restaurant	Latin American Restaurant	Restaurant	Asian Restaurant	America Restaura
8	Antioch	36.055115	-86.64782	Chuck E. Cheese	36.047788	-86.651820	Restaurant	1	Fast Food Restaurant	Latin American Restaurant	Restaurant	Asian Restaurant	America Restaura
9	Antioch	36.055115	-86.64782	Joselito's	36.050453	-86.651222	Latin American Restaurant	1	Fast Food Restaurant	Latin American Restaurant	Restaurant	Asian Restaurant	America Restaura

(F). Visualize the resulting clusters

41]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(Nashville_merged['Neighborhood Latitude'], Nashville_merged['Neighborhood Longitude'], Nashvil
le_merged['Neighborhood'], Nashville_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



Examine Each Clusters

Cluster 1

```
Nashville_merged.loc[Nashville_merged['Cluster Labels'] == 0, Nashville_merged.columns[[1] + list(range(5, Nashville_merged.shape[1]))]]
```

	Neighborhood Latitude	Venue Longitude	Venue Category	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10 C
257	35.79867	-86.910504	New American Restaurant	0	Restaurant	New American Restaurant	Vietnamese Restaurant	Japanese Restaurant	Asian Restaurant	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	French Restaurant	Re
259	35.79867	-86.905258	Restaurant	0	Restaurant	New American Restaurant	Vietnamese Restaurant	Japanese Restaurant	Asian Restaurant	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	French Restaurant	Re

Cluster 2

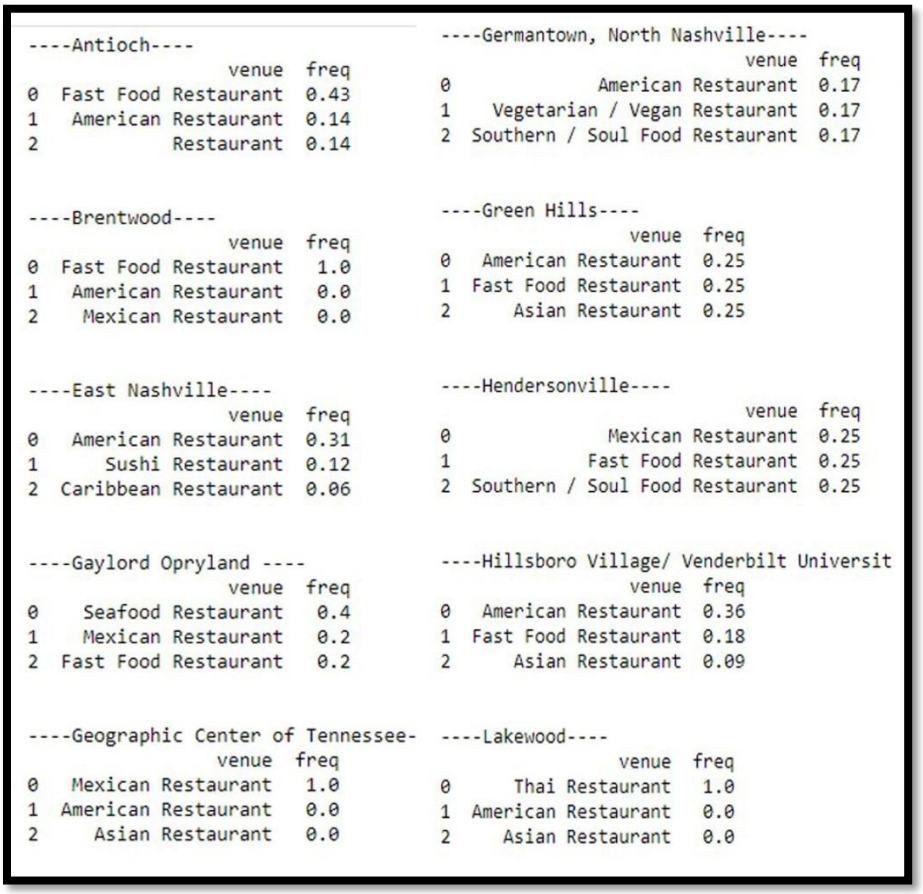
```
Nashville_merged.loc[Nashville_merged['Cluster Labels'] == 1, Nashville_merged.columns[[1] + list(range(5, Nashville_merged.shape[1]))]]
```

	Neighborhood Latitude	Venue Longitude	Venue Category	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9tl Co
7			Asian		Fast Food	Latin		Asian	American	Southern /	Indonesian	Thai	Car

Result:

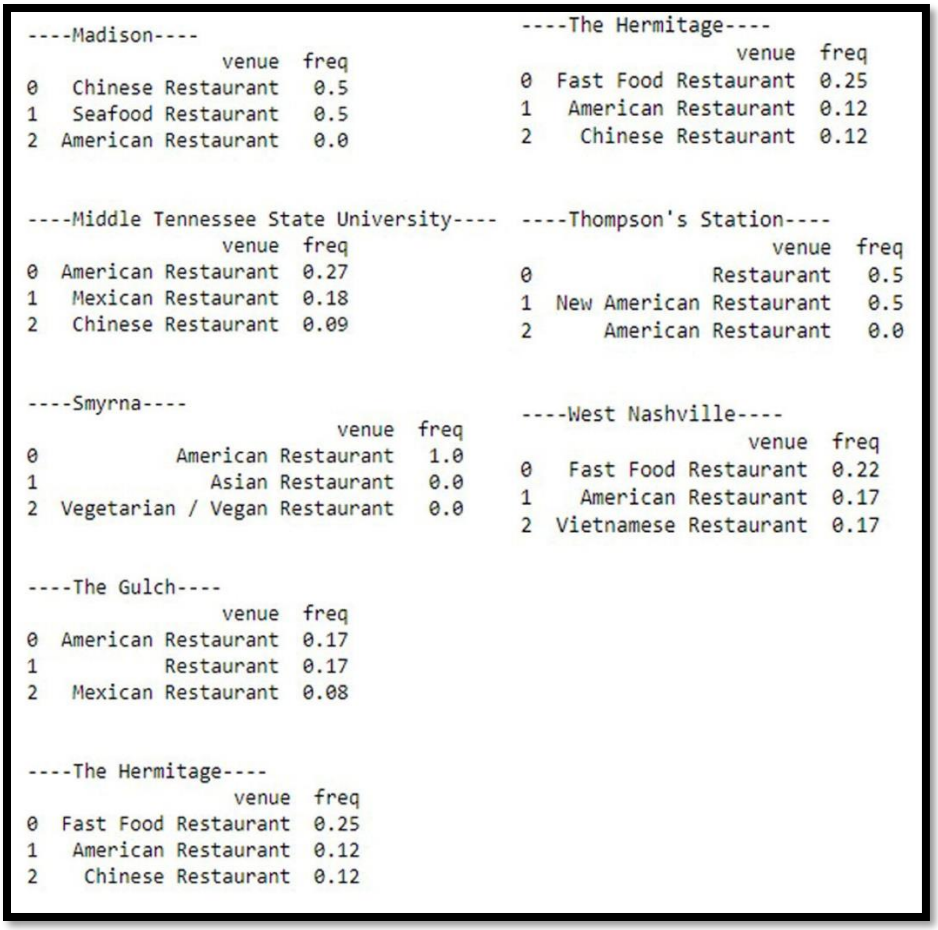
The results of the exploratory data analysis and clustering is summarized below:

1. Maximum visitors interest place in Nashville Borough in Nashville city.
2. In Total number of Pizza place Nashville Borough is NO-1 with total 123 Pizza place, follow with Murfreesboro (Total number; 64) and Franklin (Total number; 46) at Music City Nashville.
3. Top 10 Pizza places are dense placed in Mount Juliet, TN (with mean distance of 0.02084) an d scattered placed in Goodlettsville, TN (with mean distance of 0.04053).
4. Using Foursquare API, found initially 185 unique categories Music city- Nashville, in -which 24 related with unique categories cuisines / restaurants.
5. Cluster 4, 5 and 6 neighborhoods have the least number of unique taste restaurants.
6. Top 3 taste restaurants for each Neighborhood as following:



```
----Antioch----
                venue  freq
0  Fast Food Restaurant  0.43
1   American Restaurant  0.14
2            Restaurant  0.14


----Brentwood----
                venue  freq
0  Fast Food Restaurant   1.0
1   American Restaurant   0.0
2    Mexican Restaurant   0.0


----East Nashville----
                venue  freq
0   American Restaurant  0.31
1      Sushi Restaurant  0.12
2  Caribbean Restaurant  0.06


----Gaylord Opryland ----
                venue  freq
0    Seafood Restaurant   0.4
1    Mexican Restaurant   0.2
2  Fast Food Restaurant   0.2


----Geographic Center of Tennessee-
                venue  freq
0   Mexican Restaurant   1.0
1  American Restaurant   0.0
2     Asian Restaurant   0.0
```

```
----Germantown, North Nashville----
                              venue  freq
0              American Restaurant  0.17
1    Vegetarian / Vegan Restaurant  0.17
2  Southern / Soul Food Restaurant  0.17


----Green Hills----
                venue  freq
0   American Restaurant  0.25
1  Fast Food Restaurant  0.25
2      Asian Restaurant  0.25


----Hendersonville----
                              venue  freq
0              Mexican Restaurant  0.25
1            Fast Food Restaurant  0.25
2  Southern / Soul Food Restaurant  0.25


----Hillsboro Village/ Venderbilt Universit
                venue  freq
0   American Restaurant  0.36
1  Fast Food Restaurant  0.18
2      Asian Restaurant  0.09


----Lakewood----
                venue  freq
0      Thai Restaurant   1.0
1  American Restaurant   0.0
2     Asian Restaurant   0.0
```



```
----Madison----
                venue  freq
0   Chinese Restaurant   0.5
1   Seafood Restaurant   0.5
2  American Restaurant   0.0


----Middle Tennessee State University----
                venue  freq
0  American Restaurant  0.27
1   Mexican Restaurant  0.18
2   Chinese Restaurant  0.09


----Smyrna----
                         venue  freq
0           American Restaurant   1.0
1             Asian Restaurant   0.0
2  Vegetarian / Vegan Restaurant   0.0


----The Gulch----
                venue  freq
0  American Restaurant  0.17
1          Restaurant  0.17
2   Mexican Restaurant  0.08


----The Hermitage----
                 venue  freq
0  Fast Food Restaurant  0.25
1   American Restaurant  0.12
2    Chinese Restaurant  0.12


----The Hermitage----
                 venue  freq
0  Fast Food Restaurant  0.25
1   American Restaurant  0.12
2    Chinese Restaurant  0.12


----Thompson's Station----
                      venue  freq
0                Restaurant   0.5
1  New American Restaurant   0.5
2      American Restaurant   0.0


----West Nashville----
                      venue  freq
0   Fast Food Restaurant  0.22
1    American Restaurant  0.17
2  Vietnamese Restaurant  0.17
```

Discussion:

According to the analysis, Nashville is best for visitors of music city for different taste of Pizza as well taste of different categories cuisines / restaurants. As Business point of view Nashville have a lots of Pizza place so tuff competition for new pizza shop but in compare to other Boroughs Nashville city have more attraction for visitors. Other better option is Goodlettsville, TN, that have less no of Pizza places with high scattered manner.

Some drawbacks of analysis are, the clustering is completely based on the data provided by Fours quare API. Since the distance of venues from the closest station, the number of potential customer s, could all play a major role and thus, this analysis is definitely far from being conclusory. Howe ver, it definitely gives some very important preliminary information food habit of Music City-Nas hville, TN. Furthermore, these results also could potentially vary if we use some other clustering t echniques like DBSCAN.

Conclusion:

In a fast-moving world, there are many real life problems or scenarios where data can be used to find solutions to those problems. Like seen in the example above data used to benefited for food lovers, who explore new city for different taste.

Finally, to conclude this project, we have got a small glimpse of how real-life Data science project looks like. I have used some frequently used python libraries to handle JSON file, plotting graphs, and other exploratory data analysis. Use Foursquare API to major boroughs of Music city-Nashville, TN and their neighborhoods. Potential for this kind of analysis in a real-life business problem is discussed in great detail. Also, some of the drawbacks and chances for improvements to represent even more realistic pictures are mentioned. As a final note, all of the above analyses is depended on the only and only accuracy of Four Square data. A more comprehensive analysis and future work would need to incorporate data from other external databases.

Link of my code file: https://github.com/anuj1tripathi/The-Battle-of-Neighborhoods_Capstone-Project-/blob/master/The_Battle_of_Neighborhoods_Capstone_Project_full.ipynb

References:

- Wikipedia, Nashville, Tennessee; https://en.wikipedia.org/wiki/Nashville,_Tennessee.
- https://en.wikipedia.org/wiki/Nashville,_Tennessee#Neighborhoods.
- US Zip code data; https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/export/.
- Foursquare Developer Documentation: https://developer.foursquare.com/docs.