

CourseraCapstoneCCalameraFinalCopy

January 28, 2020

1 IBM Applied Data Science Capstone Course by Coursera (CCalamera)

1.1 Week 5 Final Report

1.1.1 Opening a New Music Venue in Staten Island, New York

Staten Island, New York used to be home to a bunch of fantastic music venues when I was younger and in a fun cover band. The borough does not quite have the same musical impact as it used to, so it would be interesting to see if a music venue could be opened and survive on Staten Island today.

I believe it would be best if we look for areas with colleges, coffee shops and restaurants that could attract music lovers to a small coffeehouse-type venue on weekday nights and weekends.

Some thoughts on putting together this type of project:

1. Build a dataframe of neighborhoods in Staten Island, New York by web scraping the data from Wikipedia page
2. Get the geographical coordinates of the neighborhoods
3. Obtain the venue data for the neighborhoods from Foursquare API
4. Explore and cluster the neighborhoods
5. Select the best cluster/neighborhoods to open a new coffeehouse-type music venue!

```
[40]: !pip install beautifulsoup4
      !pip install lxml
      !pip install geocoder
      import requests # library to handle requests
      import pandas as pd # library for data analysis
      import numpy as np # library to handle data in a vectorized manner
      import random # library for random number generation

      !conda install -c conda-forge geopy --yes
      from geopy.geocoders import Nominatim # module to convert an address into
      ↪ latitude and longitude values
      import geocoder # to get coordinates

      # libraries for displaying images
      from IPython.display import Image
      from IPython.core.display import HTML
```

```

from IPython.display import display_html
import pandas as pd
import numpy as np

# transforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library
from bs4 import BeautifulSoup
from sklearn.cluster import KMeans
import matplotlib.cm as cm
import matplotlib.colors as colors

print('Folium installed')
print('Libraries imported.')

```

```

Requirement already satisfied: beautifulsoup4 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (4.8.2)
Requirement already satisfied: soupsieve>=1.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
beautifulsoup4) (1.9.5)
Requirement already satisfied: lxml in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (4.4.2)
Requirement already satisfied: geocoder in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.38.1)
Requirement already satisfied: ratelim in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(0.1.6)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(2.22.0)
Requirement already satisfied: future in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(0.18.2)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(1.13.0)
Requirement already satisfied: click in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geocoder)
(7.0)
Requirement already satisfied: decorator in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ratelim->geocoder) (4.4.1)
Requirement already satisfied: urllib3!=1.25.0,!<1.25.1,<1.26,>=1.21.1 in

```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (1.25.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests->geocoder) (2019.11.28)
Solving environment: done
```

```
==> WARNING: A newer version of conda exists. <==
current version: 4.5.11
latest version: 4.8.1
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

```
# All requested packages already installed.
```

```
Solving environment: done
```

```
==> WARNING: A newer version of conda exists. <==
current version: 4.5.11
latest version: 4.8.1
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

```
# All requested packages already installed.
```

```
Folium installed
Libraries imported.
```

1.2 Let's scrape the Wikipedia page for Staten Island into a dataframe format for our work

```
[41]: # send the GET request
source = requests.get('https://en.wikipedia.org/wiki/
↳List_of_Staten_Island_neighborhoods').text

[42]: # parse data from the html into a BeautifulSoup object
soup = BeautifulSoup(source, 'html.parser')

[43]: # create a list to store neighborhood data
neighborhoodList = []

[44]: # append the data into the list
for row in soup.find_all("div", class_="mw-parser-output")[0].findAll("li"):
    neighborhoodList.append(row.text)

[45]: # create a new DataFrame from the list
si_df = pd.DataFrame({"Neighborhood": neighborhoodList})

si_df.head()
```

```
[45]:      Neighborhood
0      Annadale
1  Arden Heights
2    Arlington
3    Arrochar
4    Bay Terrace
```

```
[46]: # print the number of rows of the dataframe
si_df.shape
```

```
[46]: (151, 1)
```

2 Let's look at getting some of the geographical coordinates for our neighborhoods

```
[47]: # define a function to get coordinates:
def get_latlng(neighborhood):
    # initialize your variable to None:
    lat_lng_coors = None
    # loop until you get the coordinates:
    while(lat_lng_coors is None):
        g = geocoder.arcgis('{}, Staten Island, New York'.format(neighborhood))
        lat_lng_coors = g.latlng
    return lat_lng_coors
```

```
[48]: # call the function to get the coordinates and store to a new list
      coords = [ get_latlng(neighborhood) for neighborhood in si_df["Neighborhood"].
      ↪tolist() ]
```

Status code Unknown from

<https://geocode.arcgis.com/arcgis/rest/services/World/GeocodeServer/find>: ERROR
- HTTPConnectionPool(host='geocode.arcgis.com', port=443): Read timed out.
(read timeout=5.0)

```
[49]: coords
```

```
[49]: [[40.54920585567783, -74.17471027206285],
      [40.55988912771604, -74.1987876596362],
      [40.63718816631124, -74.16746062705108],
      [40.64242000000007, -74.07526999999999],
      [40.55452574359662, -74.13585209231593],
      [40.61059356999272, -74.17965179803514],
      [40.504025650927595, -74.2432936296758],
      [40.64242000000007, -74.07526999999999],
      [40.62122100000001, -74.12915204536496],
      [40.54941463169402, -74.21683765550642],
      [40.50430584927211, -74.24434442608066],
      [40.61238824039109, -74.0720961742667],
      [40.59861039506885, -74.10096738887606],
      [40.58799912088246, -74.10066030168086],
      [40.64242000000007, -74.07526999999999],
      [40.61501987643392, -74.10035265223087],
      [40.55725144521767, -74.16715049999999],
      [40.60403256961371, -74.1003419567669],
      [40.60659000000004, -74.06063999999998],
      [40.62531994777578, -74.15070002943034],
      [40.64242000000007, -74.07526999999999],
      [40.60223746946969, -74.08402151178691],
      [40.54790743994775, -74.14344329329849],
      [40.62931200000001, -74.11077],
      [40.62332700000002, -74.0909593114627],
      [40.61494350000001, -74.08664999999999],
      [40.59140000000008, -74.16400999999996],
      [40.53754461591505, -74.19461926435893],
      [40.642819808079125, -74.07201003377993],
      [40.60213723661052, -74.12852507641475],
      [40.611044410609495, -74.12175429906439],
      [40.63548641727852, -74.15247284007366],
      [40.61253999942312, -74.1304100183218],
      [40.56956000000008, -74.09045999999995],
      [40.644738000000004, -74.08896525],
      [40.56936093374798, -74.10768607858978],
```

[40.63731701452034, -74.10354975135184],
[40.629748500000005, -74.10125925],
[40.600732500000001, -74.10550275],
[40.595959298086285, -74.08706348086278],
[40.52370797168411, -74.21940063040172],
[40.57147999942316, -74.11167001110844],
[40.634250000000065, -74.18464999999998],
[40.629276252169866, -74.15006617623028],
[40.632763032954, -74.09912601894658],
[40.52026483872216, -74.22997242030695],
[40.57052810189513, -74.14641167298372],
[40.614124500000002, -74.0673945],
[40.548675228274675, -74.21073124449644],
[40.59299372288795, -74.1054560817534],
[40.50357307729661, -74.23839260922273],
[40.60900180969129, -74.061054691979],
[40.63222261774508, -74.09370801000613],
[40.590160000000026, -74.06625999999994],
[40.627250000000006, -74.07523999999995],
[40.57364656674685, -74.09112363423853],
[40.614456381115964, -74.0996198597422],
[40.58867510935208, -74.11170580170767],
[40.637570000000004, -74.07601999999997],
[40.5069165, -74.25378],
[40.5069165, -74.25378],
[40.593645926434924, -74.16573341888974],
[40.63287845307819, -74.08295353753743],
[40.63348368423633, -74.11829911263045],
[40.6180000510705, -74.13277003664369],
[40.60320434334497, -74.14125454505232],
[40.55041981815081, -74.18738472369448],
[40.642420000000007, -74.07526999999999],
[40.642420000000007, -74.07526999999999],
[40.50252699930734, -74.24180689364445],
[40.60290222942214, -74.12608214798281],
[40.714550000000003, -74.00713999999994],
[40.642420000000007, -74.07526999999999],
[40.642420000000007, -74.07526999999999],
[40.642420000000007, -74.07526999999999],
[40.642420000000007, -74.07526999999999],
[40.642420000000007, -74.07526999999999],
[40.54920585567783, -74.17471027206285],
[40.55988912771604, -74.1987876596362],
[40.63718816631124, -74.16746062705108],
[40.642420000000007, -74.07526999999999],
[40.55452574359662, -74.13585209231593],
[40.61059356999272, -74.17965179803514],
[40.504025650927595, -74.2432936296758],

[40.64242000000007, -74.07526999999999],
[40.62122100000001, -74.12915204536496],
[40.54941463169402, -74.21683765550642],
[40.50430584927211, -74.24434442608066],
[40.61238824039109, -74.0720961742667],
[40.59861039506885, -74.10096738887606],
[40.58799912088246, -74.10066030168086],
[40.64242000000007, -74.07526999999999],
[40.61501987643392, -74.10035265223087],
[40.55725144521767, -74.16715049999999],
[40.60403256961371, -74.1003419567669],
[40.60659000000004, -74.06063999999998],
[40.62531994777578, -74.15070002943034],
[40.64242000000007, -74.07526999999999],
[40.60223746946969, -74.08402151178691],
[40.54790743994775, -74.14344329329849],
[40.62931200000001, -74.11077],
[40.62332700000002, -74.0909593114627],
[40.59140000000008, -74.16400999999996],
[40.53754461591505, -74.19461926435893],
[40.642819808079125, -74.07201003377993],
[40.60213723661052, -74.12852507641475],
[40.611044410609495, -74.12175429906439],
[40.63548641727852, -74.15247284007366],
[40.61253999942312, -74.1304100183218],
[40.56956000000008, -74.09045999999995],
[40.644738000000004, -74.08896525],
[40.56936093374798, -74.10768607858978],
[40.63731701452034, -74.10354975135184],
[40.629748500000005, -74.10125925],
[40.60073250000001, -74.10550275],
[40.595959298086285, -74.08706348086278],
[40.615969459182075, -74.0813938023901],
[40.52370797168411, -74.21940063040172],
[40.57147999942316, -74.11167001110844],
[40.634250000000065, -74.18464999999998],
[40.629276252169866, -74.15006617623028],
[40.632763032954, -74.09912601894658],
[40.57052810189513, -74.14641167298372],
[40.52026483872216, -74.22997242030695],
[40.61412450000002, -74.0673945],
[40.548675228274675, -74.21073124449644],
[40.59299372288795, -74.1054560817534],
[40.50357307729661, -74.23839260922273],
[40.60900180969129, -74.061054691979],
[40.63222261774508, -74.09370801000613],
[40.590160000000026, -74.06625999999994],

```
[40.62725000000006, -74.07523999999995],
[40.57364656674685, -74.09112363423853],
[40.614456381115964, -74.0996198597422],
[40.60593209212038, -74.17721590550927],
[40.58867510935208, -74.11170580170767],
[40.63757000000004, -74.07601999999997],
[40.5069165, -74.25378],
[40.593645926434924, -74.16573341888974],
[40.63287845307819, -74.08295353753743],
[40.6180000510705, -74.13277003664369],
[40.63348368423633, -74.11829911263045],
[40.60320434334497, -74.14125454505232],
[40.55041981815081, -74.18738472369448],
[40.64242000000007, -74.07526999999999],
[40.64242000000007, -74.07526999999999],
[40.62230808552039, -74.13430683730975],
[40.51486001226237, -74.24837005554244],
[40.557607536581294, -74.20436869757242],
[40.60758313565781, -74.10051647426945],
[40.52917164040448, -74.22377806029924],
[40.62735, -74.18070900000001]]
```

```
[50]: # create a temporary dataframe to populate the coordinates into Latitude and
      ↪Longitude
df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
```

```
[51]: # merge the coordinates into the original dataframe
si_df['Latitude'] = df_coords['Latitude']
si_df['Longitude'] = df_coords['Longitude']
```

```
[52]: # check out the neighborhoods and the coordinates
pd.set_option('display.max_rows', None)
print(si_df.shape)
si_df
```

(151, 3)

```
[52]:
```

	Neighborhood	Latitude	Longitude
0	Annadale	40.549206	-74.174710
1	Arden Heights	40.559889	-74.198788
2	Arlington	40.637188	-74.167461
3	Arrochar	40.642420	-74.075270
4	Bay Terrace	40.554526	-74.135852
5	Bloomfield	40.610594	-74.179652
6	Brighton Heights	40.504026	-74.243294
7	Bulls Head	40.642420	-74.075270
8	Castleton Corners	40.621221	-74.129152

9	Charleston	40.549415	-74.216838
10	Chelsea	40.504306	-74.244344
11	Clifton	40.612388	-74.072096
12	Concord	40.598610	-74.100967
13	Dongan Hills	40.587999	-74.100660
14	Egbertville	40.642420	-74.075270
15	Elm Park	40.615020	-74.100353
16	Eltingville	40.557251	-74.167150
17	Emerson Hill	40.604033	-74.100342
18	Fort Wadsworth	40.606590	-74.060640
19	Graniteville	40.625320	-74.150700
20	Grant City	40.642420	-74.075270
21	Grasmere	40.602237	-74.084022
22	Great Kills	40.547907	-74.143443
23	Greenridge	40.629312	-74.110770
24	Grymes Hill	40.623327	-74.090959
25	Hamilton Park	40.614944	-74.086650
26	Heartland Village	40.591400	-74.164010
27	Huguenot	40.537545	-74.194619
28	Lighthouse Hill	40.642820	-74.072010
29	Livingston	40.602137	-74.128525
30	Manor Heights	40.611044	-74.121754
31	Mariners Harbor	40.635486	-74.152473
32	Meiers Corners	40.612540	-74.130410
33	Midland Beach	40.569560	-74.090460
34	New Brighton	40.644738	-74.088965
35	New Dorp	40.569361	-74.107686
36	New Springville	40.637317	-74.103550
37	Oakwood	40.629749	-74.101259
38	Old Place	40.600733	-74.105503
39	Old Town	40.595959	-74.087063
40	Pleasant Plains	40.523708	-74.219401
41	Port Ivory	40.571480	-74.111670
42	Port Richmond	40.634250	-74.184650
43	Prince's Bay	40.629276	-74.150066
44	Randall Manor	40.632763	-74.099126
45	Richmond Valley	40.520265	-74.229972
46	Richmondtown	40.570528	-74.146412
47	Rosebank	40.614125	-74.067395
48	Rossville	40.548675	-74.210731
49	Saint George	40.592994	-74.105456
50	Sandy Ground	40.503573	-74.238393
51	Shore Acres	40.609002	-74.061055
52	Silver Lake	40.632223	-74.093708
53	South Beach	40.590160	-74.066260
54	Stapleton	40.627250	-74.075240
55	Stapleton Heights	40.573647	-74.091124

56	Sunnyside	40.614456	-74.099620
57	Todt Hill	40.588675	-74.111706
58	Tompkinsville	40.637570	-74.076020
59	Tottenville	40.506917	-74.253780
60	Tottenville Beach	40.506917	-74.253780
61	Travis	40.593646	-74.165733
62	Ward Hill	40.632878	-74.082954
63	West New Brighton	40.633484	-74.118299
64	Westerleigh	40.618000	-74.132770
65	Willowbrook	40.603204	-74.141255
66	Woodrow	40.550420	-74.187385
67	List of Bronx neighborhoods	40.642420	-74.075270
68	List of Brooklyn neighborhoods	40.642420	-74.075270
69	List of Manhattan neighborhoods	40.502527	-74.241807
70	List of Queens neighborhoods	40.602902	-74.126082
71	"NYC Neighborhoods Map", NYC Department of Cit...	40.714550	-74.007140
72	"Staten Island neighborhoods: What you need to...	40.642420	-74.075270
73	v	40.642420	-74.075270
74	t	40.642420	-74.075270
75	e	40.642420	-74.075270
76	Annadale	40.549206	-74.174710
77	Arden Heights	40.559889	-74.198788
78	Arlington	40.637188	-74.167461
79	Arrochar	40.642420	-74.075270
80	Bay Terrace	40.554526	-74.135852
81	Bloomfield	40.610594	-74.179652
82	Brighton Heights	40.504026	-74.243294
83	Bulls Head	40.642420	-74.075270
84	Castleton Corners	40.621221	-74.129152
85	Charleston	40.549415	-74.216838
86	Chelsea	40.504306	-74.244344
87	Clifton	40.612388	-74.072096
88	Concord	40.598610	-74.100967
89	Dongan Hills	40.587999	-74.100660
90	Egbertville	40.642420	-74.075270
91	Elm Park	40.615020	-74.100353
92	Eltingville	40.557251	-74.167150
93	Emerson Hill	40.604033	-74.100342
94	Fort Wadsworth	40.606590	-74.060640
95	Graniteville	40.625320	-74.150700
96	Grant City	40.642420	-74.075270
97	Grasmere	40.602237	-74.084022
98	Great Kills	40.547907	-74.143443
99	Greenridge	40.629312	-74.110770
100	Grymes Hill	40.623327	-74.090959
101	Heartland Village	40.591400	-74.164010
102	Huguenot	40.537545	-74.194619

103	Lighthouse Hill	40.642820	-74.072010
104	Livingston	40.602137	-74.128525
105	Manor Heights	40.611044	-74.121754
106	Mariners Harbor	40.635486	-74.152473
107	Meiers Corners	40.612540	-74.130410
108	Midland Beach	40.569560	-74.090460
109	New Brighton	40.644738	-74.088965
110	New Dorp	40.569361	-74.107686
111	New Springville	40.637317	-74.103550
112	Oakwood	40.629749	-74.101259
113	Old Place	40.600733	-74.105503
114	Old Town	40.595959	-74.087063
115	Park Hill	40.615969	-74.081394
116	Pleasant Plains	40.523708	-74.219401
117	Port Ivory	40.571480	-74.111670
118	Port Richmond	40.634250	-74.184650
119	Prince's Bay	40.629276	-74.150066
120	Randall Manor	40.632763	-74.099126
121	Richmondtown	40.570528	-74.146412
122	Richmond Valley	40.520265	-74.229972
123	Rosebank	40.614125	-74.067395
124	Rossville	40.548675	-74.210731
125	Saint George	40.592994	-74.105456
126	Sandy Ground	40.503573	-74.238393
127	Shore Acres	40.609002	-74.061055
128	Silver Lake	40.632223	-74.093708
129	South Beach	40.590160	-74.066260
130	Stapleton	40.627250	-74.075240
131	Stapleton Heights	40.573647	-74.091124
132	Sunnyside	40.614456	-74.099620
133	Teleport	40.605932	-74.177216
134	Todt Hill	40.588675	-74.111706
135	Tompkinsville	40.637570	-74.076020
136	Tottenville	40.506917	-74.253780
137	Travis	40.593646	-74.165733
138	Ward Hill	40.632878	-74.082954
139	Westerleigh	40.618000	-74.132770
140	West New Brighton	40.633484	-74.118299
141	Willowbrook	40.603204	-74.141255
142	Woodrow	40.550420	-74.187385
143	Geographic sections: East Shore	40.642420	-74.075270
144	Mid-Island	40.642420	-74.075270
145	North Shore	40.622308	-74.134307
146	South Shore	40.514860	-74.248370
147	West Shore	40.557608	-74.204369
148	Community boards: 1	40.607583	-74.100516
149	2	40.529172	-74.223778

3 Some neighborhoods need additional parsing out, as the code picked up some errant data from the scraping

```
[53]: si_df = si_df.drop(si_df.index[67:151])
```

```
[54]: si_df
```

```
[54]:
```

	Neighborhood	Latitude	Longitude
0	Annadale	40.549206	-74.174710
1	Arden Heights	40.559889	-74.198788
2	Arlington	40.637188	-74.167461
3	Arrochar	40.642420	-74.075270
4	Bay Terrace	40.554526	-74.135852
5	Bloomfield	40.610594	-74.179652
6	Brighton Heights	40.504026	-74.243294
7	Bulls Head	40.642420	-74.075270
8	Castleton Corners	40.621221	-74.129152
9	Charleston	40.549415	-74.216838
10	Chelsea	40.504306	-74.244344
11	Clifton	40.612388	-74.072096
12	Concord	40.598610	-74.100967
13	Dongan Hills	40.587999	-74.100660
14	Egbertville	40.642420	-74.075270
15	Elm Park	40.615020	-74.100353
16	Eltingville	40.557251	-74.167150
17	Emerson Hill	40.604033	-74.100342
18	Fort Wadsworth	40.606590	-74.060640
19	Graniteville	40.625320	-74.150700
20	Grant City	40.642420	-74.075270
21	Grasmere	40.602237	-74.084022
22	Great Kills	40.547907	-74.143443
23	Greenridge	40.629312	-74.110770
24	Grymes Hill	40.623327	-74.090959
25	Hamilton Park	40.614944	-74.086650
26	Heartland Village	40.591400	-74.164010
27	Huguenot	40.537545	-74.194619
28	Lighthouse Hill	40.642820	-74.072010
29	Livingston	40.602137	-74.128525
30	Manor Heights	40.611044	-74.121754
31	Mariners Harbor	40.635486	-74.152473
32	Meiers Corners	40.612540	-74.130410
33	Midland Beach	40.569560	-74.090460
34	New Brighton	40.644738	-74.088965
35	New Dorp	40.569361	-74.107686

36	New Springville	40.637317	-74.103550
37	Oakwood	40.629749	-74.101259
38	Old Place	40.600733	-74.105503
39	Old Town	40.595959	-74.087063
40	Pleasant Plains	40.523708	-74.219401
41	Port Ivory	40.571480	-74.111670
42	Port Richmond	40.634250	-74.184650
43	Prince's Bay	40.629276	-74.150066
44	Randall Manor	40.632763	-74.099126
45	Richmond Valley	40.520265	-74.229972
46	Richmondtown	40.570528	-74.146412
47	Rosebank	40.614125	-74.067395
48	Rossville	40.548675	-74.210731
49	Saint George	40.592994	-74.105456
50	Sandy Ground	40.503573	-74.238393
51	Shore Acres	40.609002	-74.061055
52	Silver Lake	40.632223	-74.093708
53	South Beach	40.590160	-74.066260
54	Stapleton	40.627250	-74.075240
55	Stapleton Heights	40.573647	-74.091124
56	Sunnyside	40.614456	-74.099620
57	Todt Hill	40.588675	-74.111706
58	Tompkinsville	40.637570	-74.076020
59	Tottenville	40.506917	-74.253780
60	Tottenville Beach	40.506917	-74.253780
61	Travis	40.593646	-74.165733
62	Ward Hill	40.632878	-74.082954
63	West New Brighton	40.633484	-74.118299
64	Westerleigh	40.618000	-74.132770
65	Willowbrook	40.603204	-74.141255
66	Woodrow	40.550420	-74.187385

```
[55]: # check out the neighborhoods and the coordinates
print(si_df.shape)
si_df
```

(67, 3)

```
[55]:
```

	Neighborhood	Latitude	Longitude
0	Annadale	40.549206	-74.174710
1	Arden Heights	40.559889	-74.198788
2	Arlington	40.637188	-74.167461
3	Arrochar	40.642420	-74.075270
4	Bay Terrace	40.554526	-74.135852
5	Bloomfield	40.610594	-74.179652
6	Brighton Heights	40.504026	-74.243294
7	Bulls Head	40.642420	-74.075270

8	Castleton Corners	40.621221	-74.129152
9	Charleston	40.549415	-74.216838
10	Chelsea	40.504306	-74.244344
11	Clifton	40.612388	-74.072096
12	Concord	40.598610	-74.100967
13	Dongan Hills	40.587999	-74.100660
14	Egbertville	40.642420	-74.075270
15	Elm Park	40.615020	-74.100353
16	Eltingville	40.557251	-74.167150
17	Emerson Hill	40.604033	-74.100342
18	Fort Wadsworth	40.606590	-74.060640
19	Graniteville	40.625320	-74.150700
20	Grant City	40.642420	-74.075270
21	Grasmere	40.602237	-74.084022
22	Great Kills	40.547907	-74.143443
23	Greenridge	40.629312	-74.110770
24	Grymes Hill	40.623327	-74.090959
25	Hamilton Park	40.614944	-74.086650
26	Heartland Village	40.591400	-74.164010
27	Huguenot	40.537545	-74.194619
28	Lighthouse Hill	40.642820	-74.072010
29	Livingston	40.602137	-74.128525
30	Manor Heights	40.611044	-74.121754
31	Mariners Harbor	40.635486	-74.152473
32	Meiers Corners	40.612540	-74.130410
33	Midland Beach	40.569560	-74.090460
34	New Brighton	40.644738	-74.088965
35	New Dorp	40.569361	-74.107686
36	New Springville	40.637317	-74.103550
37	Oakwood	40.629749	-74.101259
38	Old Place	40.600733	-74.105503
39	Old Town	40.595959	-74.087063
40	Pleasant Plains	40.523708	-74.219401
41	Port Ivory	40.571480	-74.111670
42	Port Richmond	40.634250	-74.184650
43	Prince's Bay	40.629276	-74.150066
44	Randall Manor	40.632763	-74.099126
45	Richmond Valley	40.520265	-74.229972
46	Richmondtown	40.570528	-74.146412
47	Rosebank	40.614125	-74.067395
48	Rossville	40.548675	-74.210731
49	Saint George	40.592994	-74.105456
50	Sandy Ground	40.503573	-74.238393
51	Shore Acres	40.609002	-74.061055
52	Silver Lake	40.632223	-74.093708
53	South Beach	40.590160	-74.066260
54	Stapleton	40.627250	-74.075240

```

55 Stapleton Heights 40.573647 -74.091124
56      Sunnyside 40.614456 -74.099620
57      Todt Hill 40.588675 -74.111706
58      Tompkinsville 40.637570 -74.076020
59      Tottenville 40.506917 -74.253780
60 Tottenville Beach 40.506917 -74.253780
61      Travis 40.593646 -74.165733
62      Ward Hill 40.632878 -74.082954
63 West New Brighton 40.633484 -74.118299
64      Westerleigh 40.618000 -74.132770
65      Willowbrook 40.603204 -74.141255
66      Woodrow 40.550420 -74.187385

```

```

[56]: # save the DataFrame as CSV file
si_df.to_csv("si_df.csv", index=False)

```

4 Create a map of Staten Island with Neighborhoods Superimposed on top

```

[57]: # get the coordinates of Staten Island
address = 'Staten Island, New York'

geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Staten Island, New York {}, {}.'.
      ↪format(latitude, longitude))

```

The geograpical coordinate of Staten Island, New York 40.5834557, -74.1496048.

```

[58]: # create map of Staten Island using latitude and longitude values
map_si = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, neighborhood in zip(si_df['Latitude'], si_df['Longitude'],
    ↪si_df['Neighborhood']):
    label = '{}'.format(neighborhood)
    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).add_to(map_si)

map_si

```

[58]: <folium.folium.Map at 0x7f9e9fb2b630>

```

[59]: # save the map as HTML file
map_si.save('map_si.html')

```

5 Let's use the FourSquare API to start exploring the neighborhoods in Staten Island, NY

```

[60]: CLIENT_ID = 'LQERMGGYJALP5SOUVTKRBXGKP3RU33ILLXOKWJLUG52MFKRY' # your ↵
      ↪Foursquare ID
CLIENT_SECRET = '3P1VIPQZ3GZ3HI1QFCOXOYHZLMOGAFWMC415F4JCWRRHCVME' # your ↵
      ↪Foursquare Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)

```

Your credentails:

CLIENT_ID: LQERMGGYJALP5SOUVTKRBXGKP3RU33ILLXOKWJLUG52MFKRY

CLIENT_SECRET:3P1VIPQZ3GZ3HI1QFCOXOYHZLMOGAFWMC415F4JCWRRHCVME

```

[61]: si_df.loc[0, 'Neighborhood']

```

[61]: 'Annadale'

```

[62]: neighborhood_latitude = si_df.loc[0, 'Latitude'] # neighborhood latitude value
      neighborhood_longitude = si_df.loc[0, 'Longitude'] # neighborhood longitude ↵
      ↪value

neighborhood_name = si_df.loc[0, 'Neighborhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.
      ↪format(neighborhood_name,
      ↪neighborhood_latitude,
      ↪neighborhood_longitude))

```

Latitude and longitude values of Annadale are 40.54920585567783,
-74.17471027206285.

6 Let's look at some of the venues that are in the area of Staten Island, NY

```
[63]: address = 'Staten Island, NY'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

40.5834557 -74.1496048

```
[64]: radius = 5000
LIMIT = 100

venues = []

for lat, long, neighborhood in zip(si_df['Latitude'], si_df['Longitude'],
    ↪ si_df['Neighborhood']):

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

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

    # return only relevant information for each nearby venue
    for venue in results:
        venues.append((
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
            venue['venue']['categories'][0]['name']))
```

```
[65]: # convert the venues list into a new DataFrame
venues_df = pd.DataFrame(venues)

# define the column names
venues_df.columns = ['Neighborhood', 'Latitude', 'Longitude', 'VenueName', 'VenueLatitude', 'VenueLongitude', 'VenueCategory']

print(venues_df.shape)
venues_df.head()
```

(6700, 7)

```
[65]:
```

	Neighborhood	Latitude	Longitude	VenueName \
0	Annadale	40.549206	-74.17471	Pastosa Ravioli
1	Annadale	40.549206	-74.17471	Campania Coal Fired Pizza
2	Annadale	40.549206	-74.17471	Ralph's Ices
3	Annadale	40.549206	-74.17471	Annadale Diner
4	Annadale	40.549206	-74.17471	Holiday Beverage

	VenueLatitude	VenueLongitude	VenueCategory
0	40.545310	-74.165364	Gourmet Shop
1	40.543206	-74.164033	Pizza Place
2	40.559805	-74.169273	Ice Cream Shop
3	40.542079	-74.177325	Diner
4	40.542539	-74.165401	Liquor Store

```
[66]: venues_df.groupby(["Neighborhood"]).count()
```

```
[66]:
```

	Latitude	Longitude	VenueName	VenueLatitude \
Neighborhood				
Annadale	100	100	100	100
Arden Heights	100	100	100	100
Arlington	100	100	100	100
Arrochar	100	100	100	100
Bay Terrace	100	100	100	100
Bloomfield	100	100	100	100
Brighton Heights	100	100	100	100
Bulls Head	100	100	100	100
Castleton Corners	100	100	100	100
Charleston	100	100	100	100
Chelsea	100	100	100	100
Clifton	100	100	100	100
Concord	100	100	100	100
Dongan Hills	100	100	100	100
Egberville	100	100	100	100
Elm Park	100	100	100	100
Eltingville	100	100	100	100

Emerson Hill	100	100	100	100
Fort Wadsworth	100	100	100	100
Graniteville	100	100	100	100
Grant City	100	100	100	100
Grasmere	100	100	100	100
Great Kills	100	100	100	100
Greenridge	100	100	100	100
Grymes Hill	100	100	100	100
Hamilton Park	100	100	100	100
Heartland Village	100	100	100	100
Huguenot	100	100	100	100
Lighthouse Hill	100	100	100	100
Livingston	100	100	100	100
Manor Heights	100	100	100	100
Mariners Harbor	100	100	100	100
Meiers Corners	100	100	100	100
Midland Beach	100	100	100	100
New Brighton	100	100	100	100
New Dorp	100	100	100	100
New Springville	100	100	100	100
Oakwood	100	100	100	100
Old Place	100	100	100	100
Old Town	100	100	100	100
Pleasant Plains	100	100	100	100
Port Ivory	100	100	100	100
Port Richmond	100	100	100	100
Prince's Bay	100	100	100	100
Randall Manor	100	100	100	100
Richmond Valley	100	100	100	100
Richmondtown	100	100	100	100
Rosebank	100	100	100	100
Rossville	100	100	100	100
Saint George	100	100	100	100
Sandy Ground	100	100	100	100
Shore Acres	100	100	100	100
Silver Lake	100	100	100	100
South Beach	100	100	100	100
Stapleton	100	100	100	100
Stapleton Heights	100	100	100	100
Sunnyside	100	100	100	100
Todt Hill	100	100	100	100
Tompkinsville	100	100	100	100
Tottenville	100	100	100	100
Tottenville Beach	100	100	100	100
Travis	100	100	100	100
Ward Hill	100	100	100	100
West New Brighton	100	100	100	100

Westerleigh	100	100	100	100
Willowbrook	100	100	100	100
Woodrow	100	100	100	100

	VenueLongitude	VenueCategory
Neighborhood		
Annadale	100	100
Arden Heights	100	100
Arlington	100	100
Arrochar	100	100
Bay Terrace	100	100
Bloomfield	100	100
Brighton Heights	100	100
Bulls Head	100	100
Castleton Corners	100	100
Charleston	100	100
Chelsea	100	100
Clifton	100	100
Concord	100	100
Dongan Hills	100	100
Egbertville	100	100
Elm Park	100	100
Eltingville	100	100
Emerson Hill	100	100
Fort Wadsworth	100	100
Graniteville	100	100
Grant City	100	100
Grasmere	100	100
Great Kills	100	100
Greenridge	100	100
Grymes Hill	100	100
Hamilton Park	100	100
Heartland Village	100	100
Huguenot	100	100
Lighthouse Hill	100	100
Livingston	100	100
Manor Heights	100	100
Mariners Harbor	100	100
Meiers Corners	100	100
Midland Beach	100	100
New Brighton	100	100
New Dorp	100	100
New Springville	100	100
Oakwood	100	100
Old Place	100	100
Old Town	100	100
Pleasant Plains	100	100

Port Ivory	100	100
Port Richmond	100	100
Prince's Bay	100	100
Randall Manor	100	100
Richmond Valley	100	100
Richmondtown	100	100
Rosebank	100	100
Rossville	100	100
Saint George	100	100
Sandy Ground	100	100
Shore Acres	100	100
Silver Lake	100	100
South Beach	100	100
Stapleton	100	100
Stapleton Heights	100	100
Sunnyside	100	100
Todt Hill	100	100
Tompkinsville	100	100
Tottenville	100	100
Tottenville Beach	100	100
Travis	100	100
Ward Hill	100	100
West New Brighton	100	100
Westerleigh	100	100
Willowbrook	100	100
Woodrow	100	100

```
[67]: print('There are {} uniques categories.'.format(len(venues_df['VenueCategory'].
        ↳unique()))

# print out the list of categories
venues_df['VenueCategory'].unique()[:50]
```

There are 169 uniques categories.

```
[67]: array(['Gourmet Shop', 'Pizza Place', 'Ice Cream Shop', 'Diner',
        'Liquor Store', 'Gym / Fitness Center', 'Sushi Restaurant',
        'Wine Shop', 'Italian Restaurant', 'Restaurant', 'Pharmacy',
        'Bakery', 'Coffee Shop', 'Sports Bar', 'Bagel Shop', 'Gym',
        'Japanese Restaurant', 'Gastropub', 'Clothing Store', 'Park',
        'Toy / Game Store', 'Beach', 'Spa', 'Bar', 'Cosmetics Shop',
        'Fruit & Vegetable Store', 'Furniture / Home Store',
        'Warehouse Store', 'Grocery Store', 'American Restaurant',
        'Burger Joint', 'Seafood Restaurant', 'Electronics Store',
        'Food Service', 'Bookstore', 'History Museum', 'Big Box Store',
        'Trail', 'Steakhouse', 'Department Store', 'Shoe Store',
        'Spanish Restaurant', 'Donut Shop', 'Gift Shop',
```

```
'Mexican Restaurant', 'Frozen Yogurt Shop', 'Discount Store',
'Health & Beauty Service', 'Movie Theater', 'Golf Course'],
dtype=object)
```

7 Analyze some neighborhoods for Coffee Shops

```
[68]: # one hot encoding
si_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="",
    ↪ prefix_sep="")

# add neighborhood column back to dataframe
si_onehot['Neighborhoods'] = venues_df['Neighborhood']

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

print(si_onehot.shape)
si_onehot.head()
```

(6700, 170)

```
[68]: Neighborhoods  Accessories Store  American Restaurant  Arcade  Art Gallery \
0      Annadale          0          0          0          0
1      Annadale          0          0          0          0
2      Annadale          0          0          0          0
3      Annadale          0          0          0          0
4      Annadale          0          0          0          0

      Art Museum  Arts & Crafts Store  Asian Restaurant  Athletics & Sports \
0              0              0          0              0
1              0              0          0              0
2              0              0          0              0
3              0              0          0              0
4              0              0          0              0

      Automotive Shop  ...  Turkish Restaurant  Video Game Store  Video Store \
0              0  ...          0          0          0
1              0  ...          0          0          0
2              0  ...          0          0          0
3              0  ...          0          0          0
4              0  ...          0          0          0

      Vietnamese Restaurant  Warehouse Store  Wine Bar  Wine Shop  Wings Joint \
0              0          0          0          0          0
1              0          0          0          0          0
```

2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Women's Store	Zoo
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

[5 rows x 170 columns]

```
[69]: si_grouped = si_onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(si_grouped.shape)
si_grouped
```

(67, 170)

```
[69]:
```

	Neighborhoods	Accessories	Store	American Restaurant	Arcade	\
0	Annadale		0.00	0.01	0.00	
1	Arden Heights		0.00	0.00	0.00	
2	Arlington		0.02	0.02	0.01	
3	Arrochar		0.00	0.03	0.00	
4	Bay Terrace		0.00	0.01	0.00	
5	Bloomfield		0.00	0.00	0.02	
6	Brighton Heights		0.00	0.02	0.00	
7	Bulls Head		0.00	0.03	0.00	
8	Castleton Corners		0.00	0.02	0.01	
9	Charleston		0.00	0.01	0.00	
10	Chelsea		0.00	0.02	0.00	
11	Clifton		0.00	0.02	0.00	
12	Concord		0.00	0.01	0.00	
13	Dongan Hills		0.00	0.01	0.00	
14	Egbertville		0.00	0.03	0.00	
15	Elm Park		0.00	0.01	0.00	
16	Eltingville		0.00	0.01	0.00	
17	Emerson Hill		0.00	0.01	0.00	
18	Fort Wadsworth		0.00	0.02	0.00	
19	Graniteville		0.01	0.02	0.01	
20	Grant City		0.00	0.03	0.00	
21	Grasmere		0.00	0.01	0.00	
22	Great Kills		0.00	0.01	0.00	
23	Greenridge		0.00	0.02	0.00	
24	Grymes Hill		0.00	0.01	0.00	
25	Hamilton Park		0.00	0.01	0.00	

26	Heartland Village	0.00	0.00	0.00
27	Huguenot	0.00	0.03	0.00
28	Lighthouse Hill	0.00	0.03	0.00
29	Livingston	0.00	0.01	0.01
30	Manor Heights	0.00	0.01	0.01
31	Mariners Harbor	0.02	0.02	0.01
32	Meiers Corners	0.00	0.01	0.01
33	Midland Beach	0.00	0.01	0.00
34	New Brighton	0.00	0.03	0.00
35	New Dorp	0.00	0.01	0.00
36	New Springville	0.00	0.04	0.00
37	Oakwood	0.00	0.01	0.00
38	Old Place	0.00	0.01	0.00
39	Old Town	0.00	0.01	0.00
40	Pleasant Plains	0.00	0.02	0.00
41	Port Ivory	0.00	0.01	0.00
42	Port Richmond	0.02	0.01	0.01
43	Prince's Bay	0.02	0.02	0.01
44	Randall Manor	0.00	0.02	0.00
45	Richmond Valley	0.00	0.02	0.00
46	Richmondtown	0.00	0.01	0.00
47	Rosebank	0.00	0.04	0.00
48	Rossville	0.00	0.02	0.00
49	Saint George	0.00	0.01	0.00
50	Sandy Ground	0.00	0.02	0.00
51	Shore Acres	0.00	0.02	0.00
52	Silver Lake	0.00	0.01	0.00
53	South Beach	0.00	0.01	0.00
54	Stapleton	0.00	0.03	0.00
55	Stapleton Heights	0.00	0.01	0.00
56	Sunnyside	0.00	0.01	0.00
57	Todt Hill	0.00	0.01	0.00
58	Tompkinsville	0.00	0.02	0.00
59	Tottenville	0.00	0.02	0.00
60	Tottenville Beach	0.00	0.02	0.00
61	Travis	0.00	0.00	0.00
62	Ward Hill	0.00	0.01	0.00
63	West New Brighton	0.00	0.02	0.01
64	Westerleigh	0.00	0.02	0.01
65	Willowbrook	0.00	0.00	0.01
66	Woodrow	0.00	0.01	0.00

	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	\
0	0.00	0.00	0.00	0.00	
1	0.00	0.00	0.00	0.00	
2	0.00	0.00	0.00	0.00	
3	0.01	0.00	0.00	0.01	

4	0.00	0.00	0.00	0.01
5	0.00	0.00	0.01	0.01
6	0.00	0.00	0.00	0.01
7	0.01	0.00	0.00	0.01
8	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.01
11	0.00	0.00	0.00	0.01
12	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.01
14	0.01	0.00	0.00	0.01
15	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.01
17	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.01
19	0.00	0.00	0.00	0.00
20	0.01	0.00	0.00	0.01
21	0.00	0.00	0.00	0.01
22	0.00	0.00	0.00	0.01
23	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.01
25	0.00	0.00	0.00	0.01
26	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00
28	0.01	0.00	0.00	0.01
29	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.01
34	0.01	0.00	0.01	0.01
35	0.00	0.00	0.00	0.01
36	0.01	0.00	0.00	0.00
37	0.01	0.00	0.00	0.01
38	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.01
42	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00
44	0.01	0.00	0.00	0.01
45	0.00	0.00	0.00	0.01
46	0.00	0.00	0.00	0.01
47	0.00	0.00	0.00	0.01
48	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.01

51	0.00	0.00	0.00	0.01
52	0.01	0.00	0.00	0.01
53	0.00	0.00	0.00	0.00
54	0.00	0.00	0.00	0.01
55	0.00	0.00	0.00	0.01
56	0.00	0.00	0.00	0.01
57	0.00	0.00	0.00	0.01
58	0.01	0.00	0.00	0.01
59	0.00	0.00	0.00	0.01
60	0.00	0.00	0.00	0.01
61	0.00	0.00	0.00	0.00
62	0.01	0.00	0.00	0.01
63	0.00	0.00	0.00	0.00
64	0.00	0.00	0.00	0.00
65	0.00	0.01	0.00	0.00
66	0.00	0.00	0.00	0.00

	Athletics & Sports	Automotive	Shop	...	Turkish Restaurant	\
0	0.00		0.00	...		0.00
1	0.00		0.00	...		0.00
2	0.00		0.00	...		0.00
3	0.00		0.00	...		0.00
4	0.00		0.00	...		0.00
5	0.00		0.00	...		0.00
6	0.00		0.00	...		0.01
7	0.00		0.00	...		0.00
8	0.00		0.00	...		0.00
9	0.00		0.00	...		0.00
10	0.00		0.00	...		0.01
11	0.01		0.00	...		0.00
12	0.01		0.01	...		0.00
13	0.01		0.01	...		0.00
14	0.00		0.00	...		0.00
15	0.00		0.01	...		0.00
16	0.00		0.00	...		0.00
17	0.01		0.00	...		0.00
18	0.01		0.00	...		0.00
19	0.00		0.00	...		0.00
20	0.00		0.00	...		0.00
21	0.01		0.00	...		0.00
22	0.00		0.00	...		0.00
23	0.00		0.00	...		0.00
24	0.00		0.00	...		0.00
25	0.01		0.00	...		0.00
26	0.00		0.00	...		0.00
27	0.00		0.00	...		0.00
28	0.00		0.00	...		0.00

29	0.00	0.00	...	0.00
30	0.00	0.01	...	0.00
31	0.00	0.00	...	0.00
32	0.00	0.00	...	0.00
33	0.01	0.01	...	0.00
34	0.00	0.00	...	0.00
35	0.01	0.00	...	0.00
36	0.00	0.00	...	0.00
37	0.00	0.01	...	0.00
38	0.01	0.01	...	0.00
39	0.01	0.00	...	0.00
40	0.00	0.00	...	0.00
41	0.01	0.00	...	0.00
42	0.00	0.00	...	0.00
43	0.00	0.00	...	0.00
44	0.00	0.00	...	0.00
45	0.00	0.00	...	0.01
46	0.00	0.00	...	0.00
47	0.01	0.00	...	0.00
48	0.00	0.00	...	0.00
49	0.01	0.01	...	0.00
50	0.00	0.00	...	0.01
51	0.01	0.00	...	0.01
52	0.00	0.01	...	0.00
53	0.01	0.01	...	0.00
54	0.00	0.00	...	0.00
55	0.01	0.00	...	0.00
56	0.00	0.00	...	0.00
57	0.01	0.01	...	0.00
58	0.00	0.00	...	0.00
59	0.00	0.00	...	0.01
60	0.00	0.00	...	0.01
61	0.00	0.00	...	0.00
62	0.00	0.01	...	0.00
63	0.00	0.00	...	0.00
64	0.00	0.00	...	0.00
65	0.00	0.00	...	0.00
66	0.00	0.00	...	0.00

	Video Game Store	Video Store	Vietnamese Restaurant	Warehouse Store \
0	0.00	0.00	0.00	0.01
1	0.00	0.00	0.00	0.01
2	0.01	0.01	0.01	0.00
3	0.00	0.00	0.00	0.00
4	0.01	0.00	0.00	0.01
5	0.00	0.01	0.01	0.01
6	0.01	0.00	0.00	0.00

7	0.00	0.00	0.00	0.00
8	0.00	0.01	0.01	0.00
9	0.00	0.00	0.00	0.00
10	0.01	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00
13	0.01	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.01
17	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00
19	0.01	0.01	0.01	0.00
20	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00
22	0.01	0.00	0.00	0.01
23	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00
26	0.00	0.01	0.01	0.01
27	0.01	0.00	0.00	0.01
28	0.00	0.00	0.00	0.00
29	0.00	0.00	0.01	0.00
30	0.00	0.00	0.01	0.00
31	0.01	0.01	0.01	0.00
32	0.00	0.01	0.01	0.00
33	0.01	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00
35	0.01	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00
40	0.01	0.00	0.00	0.00
41	0.01	0.00	0.00	0.00
42	0.01	0.01	0.01	0.01
43	0.01	0.01	0.01	0.00
44	0.00	0.00	0.00	0.00
45	0.01	0.00	0.00	0.00
46	0.00	0.00	0.00	0.01
47	0.00	0.00	0.00	0.00
48	0.01	0.00	0.00	0.01
49	0.00	0.00	0.00	0.00
50	0.01	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00
53	0.00	0.00	0.00	0.00

54	0.00	0.00	0.00	0.00
55	0.01	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00
57	0.01	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00
61	0.00	0.01	0.01	0.01
62	0.00	0.00	0.00	0.00
63	0.00	0.00	0.00	0.00
64	0.00	0.01	0.01	0.00
65	0.00	0.01	0.01	0.00
66	0.00	0.00	0.00	0.01

	Wine Bar	Wine Shop	Wings Joint	Women's Store	Zoo
0	0.00	0.01	0.00	0.00	0.00
1	0.00	0.01	0.00	0.00	0.00
2	0.00	0.00	0.01	0.01	0.01
3	0.01	0.00	0.01	0.00	0.01
4	0.00	0.01	0.00	0.00	0.00
5	0.00	0.00	0.01	0.01	0.00
6	0.00	0.00	0.00	0.00	0.00
7	0.01	0.00	0.01	0.00	0.01
8	0.00	0.00	0.02	0.00	0.01
9	0.00	0.01	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00
11	0.01	0.00	0.01	0.00	0.01
12	0.00	0.00	0.01	0.00	0.01
13	0.00	0.00	0.01	0.00	0.01
14	0.01	0.00	0.01	0.00	0.01
15	0.00	0.00	0.01	0.00	0.01
16	0.00	0.01	0.00	0.00	0.00
17	0.00	0.00	0.01	0.00	0.01
18	0.01	0.00	0.00	0.00	0.00
19	0.00	0.00	0.02	0.00	0.01
20	0.01	0.00	0.01	0.00	0.01
21	0.00	0.00	0.01	0.00	0.01
22	0.00	0.01	0.00	0.00	0.00
23	0.00	0.00	0.01	0.00	0.01
24	0.00	0.00	0.01	0.00	0.01
25	0.00	0.00	0.01	0.00	0.01
26	0.00	0.01	0.01	0.00	0.00
27	0.00	0.01	0.00	0.00	0.00
28	0.01	0.00	0.01	0.00	0.01
29	0.00	0.00	0.00	0.00	0.01
30	0.00	0.00	0.01	0.00	0.01
31	0.00	0.00	0.02	0.00	0.01

32	0.00	0.00	0.02	0.00	0.01
33	0.00	0.00	0.01	0.00	0.00
34	0.00	0.00	0.01	0.00	0.01
35	0.00	0.00	0.01	0.00	0.00
36	0.00	0.00	0.01	0.00	0.01
37	0.00	0.00	0.01	0.00	0.01
38	0.00	0.00	0.01	0.00	0.01
39	0.00	0.00	0.01	0.00	0.01
40	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.01	0.00	0.00
42	0.00	0.00	0.01	0.01	0.00
43	0.00	0.00	0.02	0.00	0.01
44	0.00	0.00	0.01	0.00	0.01
45	0.00	0.00	0.00	0.00	0.00
46	0.00	0.01	0.00	0.00	0.00
47	0.01	0.00	0.00	0.00	0.00
48	0.00	0.01	0.00	0.00	0.00
49	0.00	0.00	0.01	0.00	0.01
50	0.00	0.00	0.00	0.00	0.00
51	0.01	0.00	0.00	0.00	0.00
52	0.00	0.00	0.01	0.00	0.01
53	0.00	0.00	0.00	0.00	0.00
54	0.01	0.00	0.01	0.00	0.01
55	0.00	0.00	0.01	0.00	0.00
56	0.00	0.00	0.01	0.00	0.01
57	0.00	0.00	0.01	0.00	0.01
58	0.01	0.00	0.01	0.00	0.01
59	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00
61	0.00	0.01	0.01	0.00	0.00
62	0.00	0.00	0.01	0.00	0.01
63	0.00	0.00	0.01	0.00	0.01
64	0.00	0.00	0.02	0.00	0.01
65	0.00	0.00	0.01	0.00	0.01
66	0.00	0.01	0.00	0.00	0.00

[67 rows x 170 columns]

```
[70]: len(si_grouped[si_grouped["Coffee Shop"] > 0])
```

```
[70]: 66
```

```
[71]: si_coffeeshop = si_grouped[["Neighborhoods", "Coffee Shop"]]
      si_coffeeshop.head()
```

```
[71]:   Neighborhoods  Coffee Shop
      0      Annadale         0.04
```

1	Arden Heights	0.04
2	Arlington	0.03
3	Arrochar	0.05
4	Bay Terrace	0.05

8 Cluster Neighborhoods

```
[72]: # set number of clusters
kclusters = 4

si_clustering = si_coffeeshop.drop(["Neighborhoods"], 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
[72]: array([2, 2, 0, 1, 1, 0, 3, 1, 2, 2], dtype=int32)
```

```
[73]: # create a new dataframe that includes the cluster as well as the top venues
      ↪ for each neighborhood.
si_merged = si_coffeeshop.copy()

# add clustering labels
si_merged["Cluster Labels"] = kmeans.labels_
```

```
[74]: si_merged.rename(columns={"Neighborhoods": "Neighborhood"}, inplace=True)
si_merged.head()
```

```
[74]:
```

	Neighborhood	Coffee Shop	Cluster Labels
0	Annadale	0.04	2
1	Arden Heights	0.04	2
2	Arlington	0.03	0
3	Arrochar	0.05	1
4	Bay Terrace	0.05	1

```
[75]: # merge si_grouped with si_data to add latitude/longitude for each neighborhood
si_merged = si_merged.join(si_df.set_index("Neighborhood"), on="Neighborhood")

print(si_merged.shape)
si_merged.head() # check the last columns!
```

```
(67, 5)
```

```
[75]:
```

	Neighborhood	Coffee Shop	Cluster Labels	Latitude	Longitude
0	Annadale	0.04	2	40.549206	-74.174710
1	Arden Heights	0.04	2	40.559889	-74.198788
2	Arlington	0.03	0	40.637188	-74.167461
3	Arrochar	0.05	1	40.642420	-74.075270
4	Bay Terrace	0.05	1	40.554526	-74.135852

```
[76]: # sort the results by Cluster Labels
print(si_merged.shape)
si_merged.sort_values(["Cluster Labels"], inplace=True)
si_merged
```

(67, 5)

```
[76]:
```

	Neighborhood	Coffee Shop	Cluster Labels	Latitude	Longitude
33	Midland Beach	0.03	0	40.569560	-74.090460
40	Pleasant Plains	0.03	0	40.523708	-74.219401
44	Randall Manor	0.03	0	40.632763	-74.099126
38	Old Place	0.03	0	40.600733	-74.105503
37	Oakwood	0.03	0	40.629749	-74.101259
36	New Springville	0.03	0	40.637317	-74.103550
35	New Dorp	0.03	0	40.569361	-74.107686
34	New Brighton	0.03	0	40.644738	-74.088965
65	Willowbrook	0.03	0	40.603204	-74.141255
45	Richmond Valley	0.03	0	40.520265	-74.229972
29	Livingston	0.03	0	40.602137	-74.128525
25	Hamilton Park	0.03	0	40.614944	-74.086650
24	Grymes Hill	0.03	0	40.623327	-74.090959
41	Port Ivory	0.03	0	40.571480	-74.111670
52	Silver Lake	0.03	0	40.632223	-74.093708
21	Grasmere	0.03	0	40.602237	-74.084022
2	Arlington	0.03	0	40.637188	-74.167461
62	Ward Hill	0.03	0	40.632878	-74.082954
55	Stapleton Heights	0.03	0	40.573647	-74.091124
12	Concord	0.03	0	40.598610	-74.100967
13	Dongan Hills	0.03	0	40.587999	-74.100660
5	Bloomfield	0.03	0	40.610594	-74.179652
22	Great Kills	0.06	1	40.547907	-74.143443
19	Graniteville	0.06	1	40.625320	-74.150700
7	Bulls Head	0.05	1	40.642420	-74.075270
3	Arrochar	0.05	1	40.642420	-74.075270
27	Huguenot	0.05	1	40.537545	-74.194619
28	Lighthouse Hill	0.05	1	40.642820	-74.072010
49	Saint George	0.05	1	40.592994	-74.105456
46	Richmondtown	0.05	1	40.570528	-74.146412
31	Mariners Harbor	0.05	1	40.635486	-74.152473
4	Bay Terrace	0.05	1	40.554526	-74.135852

14	Egbertville	0.05	1	40.642420	-74.075270
20	Grant City	0.05	1	40.642420	-74.075270
43	Prince's Bay	0.05	1	40.629276	-74.150066
64	Westerleigh	0.04	2	40.618000	-74.132770
54	Stapleton	0.04	2	40.627250	-74.075240
58	Tompkinsville	0.04	2	40.637570	-74.076020
47	Rosebank	0.04	2	40.614125	-74.067395
56	Sunnyside	0.04	2	40.614456	-74.099620
48	Rossville	0.04	2	40.548675	-74.210731
63	West New Brighton	0.04	2	40.633484	-74.118299
57	Todt Hill	0.04	2	40.588675	-74.111706
0	Annadale	0.04	2	40.549206	-74.174710
39	Old Town	0.04	2	40.595959	-74.087063
1	Arden Heights	0.04	2	40.559889	-74.198788
8	Castleton Corners	0.04	2	40.621221	-74.129152
9	Charleston	0.04	2	40.549415	-74.216838
16	Eltingville	0.04	2	40.557251	-74.167150
17	Emerson Hill	0.04	2	40.604033	-74.100342
15	Elm Park	0.04	2	40.615020	-74.100353
23	Greenridge	0.04	2	40.629312	-74.110770
30	Manor Heights	0.04	2	40.611044	-74.121754
32	Meiers Corners	0.04	2	40.612540	-74.130410
66	Woodrow	0.04	2	40.550420	-74.187385
51	Shore Acres	0.02	3	40.609002	-74.061055
18	Fort Wadsworth	0.02	3	40.606590	-74.060640
53	South Beach	0.00	3	40.590160	-74.066260
42	Port Richmond	0.01	3	40.634250	-74.184650
11	Clifton	0.02	3	40.612388	-74.072096
10	Chelsea	0.02	3	40.504306	-74.244344
26	Heartland Village	0.02	3	40.591400	-74.164010
59	Tottenville	0.02	3	40.506917	-74.253780
60	Tottenville Beach	0.02	3	40.506917	-74.253780
61	Travis	0.02	3	40.593646	-74.165733
6	Brighton Heights	0.02	3	40.504026	-74.243294
50	Sandy Ground	0.02	3	40.503573	-74.238393

```
[77]: # 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(si_merged['Latitude'],
↳si_merged['Longitude'], si_merged['Neighborhood'], si_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

```

[77]: <folium.folium.Map at 0x7f9e9f9fab0>

```

[78]: # save the map as HTML file
map_clusters.save('map_clusters.html')

```

9 Conclusion:

- 9.0.1 What we were trying to do in this exercise is to find areas on Staten Island with coffee shops in a nearby cluster.
- 9.0.2 This gives us an opportunity to explore opening a coffeehouse environment for music and using some of the nearby vendors as collateral for boosting/driving business.
- 9.0.3 Areas near Silver Lake and Meier's Corners represent potential landing spots for this business venture.

[]: