# capstone_boston_neighborhoods

January 31, 2021

# 1 Applied Data Science Capstone, Week 4, Capstone Project

# 2 The Battle of Neighborhoods

## 2.1 Exploring Boston Neighborhoods

> *In a city of your choice, if someone is looking to open a restaurant, where would you recommend that they open it?*

### 2.1.1 Table of Contents



### 2.1.2 Introduction

In this project I would like to explore the neighborhoods of the city of Boston, MA, USA. Boston is among the most influential and wealthy cities in the United States. The city itself is home to 645 thousand people, but with the suburbs it forms Greater Boston and the population exceeds 4.5 million people. High conccentration of hi-tech companies, world's top universities, financial center, and center of medical and biotech developments positively affects the development of the region. The city looks especially attractive for small business owners knowing that it also attracts 20 millions of tourists annually (source: Wikipedia).

### 2.1.3 Problem Description

My main goal in this analysis is to narrow down potential locations to open a restaurant and give any other recomendations that can help to a write a business plan and increase future success. When choosing a location I will consider following: * income level of the neighborhood (who to target), * number of households (the size of the restaurant), * the number of other restaurants around (potential competition) * the number of other venues nearby that may drive traffic to the restaurant

### 2.1.4 About the data

To solve the above problem and answer the questions we will use household income data to assess wealth of the neighborhoods. The shapefiles of the neighborhoods to visually present the data on the map. Other data on nearby businesses will be extracted using Foursqare API. The neighborhoods households and income data has been extracted from the website of Analyze Boston, the City of Boston's open data hub at https://data.boston.gov/dataset/neighborhood-demographics, published by Department of Innovation and Technology. The Neighborhood boundaries data are also from the same source, published by Boston Maps, can be accessed at https://data.boston.gov/dataset/boston-neighborhoods. Both of the data sets are under Open Data Commons Public Domain Dedication and License (PDDL).

### 2.1.5 Methodology

Aiding in choosing a suitable location for a restaurant is the main purpose and goal of this analysis. Therefore, **Folium** package was used in order to visually display and annotate information on a geographic map.

Neighborhoods are assessed using median income and venues within 1 km of given neighborhood's coordinates. Foursquare API was used to extract data on venues.

Sequence of tasks to be performed: * [x] Identify the area to explore (city, region, etc.), data availability, sources, and goals * [x] Explore and study the data sets, decide what to import into python environment * [x] Import, clean, rename columns, correct data types * [x] Retrieve the coordinates of the neighborhoods * [x] Retrieve Boston coordinates and create a map of its neighborhoods * [x] Load neighborhoods shapefile and display median income choropleth map * [x] Get the most common venue types for each of the neighborhoods * [x] Clustering the neighborhoods by total households and median income * [x] Clustering the neighborhoods by venues

### 2.1.6 Results

This analysis is based on mainly 2 sources of data for city's neighborhoods - income with number of housholds and venues.

#### Median income and Total households in neighborhoods

**Clustering neighborhood by total households and median income** k-means algorithm ws used to cluster the neighborhood into 5 clusters.

| Labels | median_income | total_households | lat | lon |
|---|---|---|---|---|
| 0 | 99456.098000 | 6537.000000 | 42.363230 | -71.065706 |
| 1 | 38859.695000 | 8700.833333 | 42.327903 | -71.106263 |
| 2 | 49662.360000 | 44086.000000 | 42.297320 | -71.074495 |
| 3 | 75015.912222 | 14712.222222 | 42.321197 | -71.100258 |
| 4 | 150677.510000 | 1830.000000 | 42.333431 | -71.049495 |

Note that each row in our dataset represents a neighborhood, and therefore, each row is assigned a label (22 labels in total). Now we can easily check the centroid values by averaging the features in each cluster. k-means will partition neighborhoods into five groups since we specified the algorithm to generate 5 clusters. The neighborhoods in each cluster are similar to each other in terms of the features included in the dataset, i.e. median income and total households. Next we can create a profile for each group, considering the common characteristics of each cluster. For example, the 5 clusters can be:

0: "Higher tier" 1: "Lower tier" 2: "Inbetweeners" 3: "Mid tier" 4: "The rich few"

### Venues in neighborhoods

**Number and types of venues** Overall, there are 184 uniques categories retrieved within 1000 meters radius from the coordinates of each of the neighborhoods. For example, within 1000 meters radius in Dorchester there are 21 venues accross 19 uniques categories. In total, for 22 neighborhoods in our dataframe Foursquare returned 882 venues.

**Clustering neighborhoods** k-means algorithm was used to cluster the neighborhood into 5 clusters. Below are the neighborhoods with its' clusters and top 5 most common venue types.

| neighborhoods | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|
| Mattapan | 0 | Mobile Phone Shop | Bakery | Food & Drink Shop | Shoe Store | Caribbean Restaurant |
| Dorchester | 0 | Plaza | Gym | Shoe Store | Food | Southern / Soul Food Restaurant |
| West Roxbury | 1 | Home Service | Discount Store | Yoga Studio | Opera House | Moroccan Restaurant |
| South End | 2 | Italian Restaurant | Coffee Shop | French Restaurant | Wine Shop | Wine Bar |
| Longwood | 2 | Park | Coffee Shop | Donut Shop | Falafel Restaurant | Fast Food Restaurant |
| Fenway | 2 | American Restaurant | Bakery | Café | Chinese Restaurant | Thai Restaurant |
| Allston | 2 | Korean Restaurant | Pizza Place | Chinese Restaurant | Bakery | Thai Restaurant |

| neighborhoods | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue |
|---|---|---|---|---|---|---|
| Brighton | 2 | Bus Station | Bakery | Pizza Place | Deli / Bodega | Bank |
| Beacon Hill | 2 | Italian Restaurant | Hotel Bar | Pizza Place | French Restaurant | Hotel |
| Back Bay | 2 | Coffee Shop | Hotel | Italian Restaurant | Seafood Restaurant | Cosmetics Shop |
| Downtown | 2 | Coffee Shop | Bakery | Asian Restaurant | Sandwich Place | Chinese Restaurant |
| Jamaica Plain | 2 | Bakery | Art Gallery | Coffee Shop | Park | Liquor Store |
| East Boston | 3 | Pizza Place | Pharmacy | Convenience Store | Art Gallery | Sandwich Place |
| Charlestown | 3 | Pizza Place | Coffee Shop | Pub | Donut Shop | Gastropub |
| Hyde Park | 3 | Pizza Place | American Restaurant | Pharmacy | Ice Cream Shop | Donut Shop |
| West End | 3 | Pizza Place | Hotel | Donut Shop | Bar | Convenience Store |
| Mission Hill | 3 | Pizza Place | Sandwich Place | Café | Sushi Restaurant | Grocery Store |
| North End | 3 | Italian Restaurant | Pizza Place | Seafood Restaurant | Bakery | Coffee Shop |
| Roxbury | 3 | Plaza | Gym | Rental Car Location | Park | Metro Station |
| South Boston | 3 | Pizza Place | Liquor Store | Sports Bar | Italian Restaurant | Bar |
| South Boston Waterfront | 3 | Pizza Place | Liquor Store | Sports Bar | Italian Restaurant | Bar |
| Roslindale | 4 | Yoga Studio | Big Box Store | Cuban Restaurant | Pool | Rental Car Location |

#### 2.1.7 Discussion

**Median income and Total households in neighborhoods** Since we are interested in both population and income level the plot below demonstrates if there are any tradeoffs when choosing a suitable location for a restaurant. For example, while Dorchester has the highest number of households among Boston's neighborhoods, the median income is relatively low. South Boston Waterfront on the other hand is the opposite.

Also, visually we may notice three distinctive clusters with Dorchester and South Boston Waterfront as outliers. If we are to take these two variables (`total_households` and `median_income`) as deciding factors narrowing down our choice to one of these clusters is a possible option. Applying clustering algerith returned following cluster.

However, for our recommendation to be complete, we need to look at data about competition, substitutes, and other venues around the neighborhood that could drive traffic to a restaurant. To retrieve these and other information we will utilize Folium package and Foursqaure API.

**Venues in neighborhoods** According to the list of top 5 venue categories, **Cluster 0** and **Cluster 1** are not rich for restaurants. Restaurants in these neighborhoods are 5th most common categories, but diverse (Caribbean, Moroccan, and Souther/Soul).

**Cluster 2** is the most diverse in its restaurants. Among top 5 venue categories there are American, Italian, Korean, French, Chinese, Asian, Thai cousines as well as Seafood, Falafel, and fast food restaurants.

**Cluster 3** seems to be popular with italian cousine judging by the number of pizza places these neighborhoods have. This may induce competitors or, on the contrary, drive crowd of pizza lovers to these neighborhoods. Therefore, further analysis needed if one would like to open here a restaurant with Italian cousine.

**Cluster 4** has only neighborhood. Roslindale neighborhood is distinct from others with its Yoga Studios as the most common categorie, followed by Big box stores and Cuban restaurants.

**Traffic drivers** Competition and substitutes are not the only factors. When making decision abouth the location it is also important to consider venues that attract crowd and drives traffic to restaurants. These could be shopping malls, sports venues, business centers, parks, or maybe some tourist attractions.

For example, Cluster 4 is common for yoga studios and pools. A reastaurant healthy options may be a good idea. Also in this cluster there are many Big Box stores which drives traffic of shoppers, especially over weekend. Locating near this zones can be an option if we want to target more traffic of customers.

### 2.1.8 Conclusion

The original purpose of this analysis was to get any meaningful information and insights for someone is looking to open a restaurant. Since we have not specified the type of a restaurant or targeted customer segment we limited this analysis only to describe the nieghborhoods and their similarity by applying clustering algortihms.

The following list gives some additional ideas on how one can continue to refine recommendations: * Add population density data * Add demopgraphics data such as age and occupation * Distance between venues * Reviews of competitor venues in the chosen location

[ go back to top ]

### 2.1.9 Code Section

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

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

### 2.1.10 Download and explore the dataset

[2]:
```
#boston_data_url = "https://data.boston.gov/dataset/
↪8202abf2-8434-4934-959b-94643c7dac18/resource/
↪e684798f-e175-4ab1-8f70-ed80e4e260cc/download/
↪boston_neighborhood_demographics_2013-2017.xlsx"
boston_data_url = "boston_neighborhood_demographics_2013-2017.xlsx"

df = pd.read_excel(boston_data_url,
                   sheet_name = "HH Income",
                   skiprows = [0, 1, 3, 4, 5, 29, 31],
                   usecols="A:C"
                  )
```

[3]:
```
df
```

[3]:
```
                                         Unnamed: 0  Median Income  \
0                                           Allston       46982.76
1                                          Back Bay      102070.55
2                                       Beacon Hill       98069.24
3                                          Brighton       62041.20
4                                       Charlestown      103243.16
5                                        Dorchester       49662.36
6                                          Downtown       67367.09
7                                       East Boston       52935.36
8                                            Fenway       39549.84
9                                   Harbor Islands             NaN
10                                        Hyde Park       70810.13
11                                    Jamaica Plain       84445.90
12                                         Longwood       35000.00
13                                         Mattapan       48196.90
14                                     Mission Hill       35707.32
15                                        North End       97110.39
16                                       Roslindale       76666.67
17                                          Roxbury       27721.35
18                                     South Boston       93077.60
19                        South Boston Waterfront      150677.51
20                                        South End       86994.80
21                                         West End       96787.15
22                                     West Roxbury       80804.46
23  Source: U.S. Census, 2013-2017 American Commun…            NaN
```

```
    Total Households
0             6457.0
1             9824.0
2             5458.0
3            21605.0
4             8931.0
5            44086.0
6             7552.0
7            16286.0
8            10926.0
9                0.0
10           12891.0
11           16092.0
12             280.0
13            8866.0
14            6270.0
15            5338.0
16           11406.0
17           19406.0
18           16628.0
19            1830.0
20           16193.0
21            3134.0
22           13757.0
23               NaN
```

[4]:

```
# rename the columns
df.rename(columns={'Unnamed: 0' : 'neighborhoods',
                   'Median Income' : 'median_income',
                   'Total Households' : 'total_households'
                  }, inplace=True)
#df.drop(df.tail(1).index,inplace=True) # drop last n rows
df.dropna(inplace=True)
df.reset_index(drop=True, inplace=True)
df
```

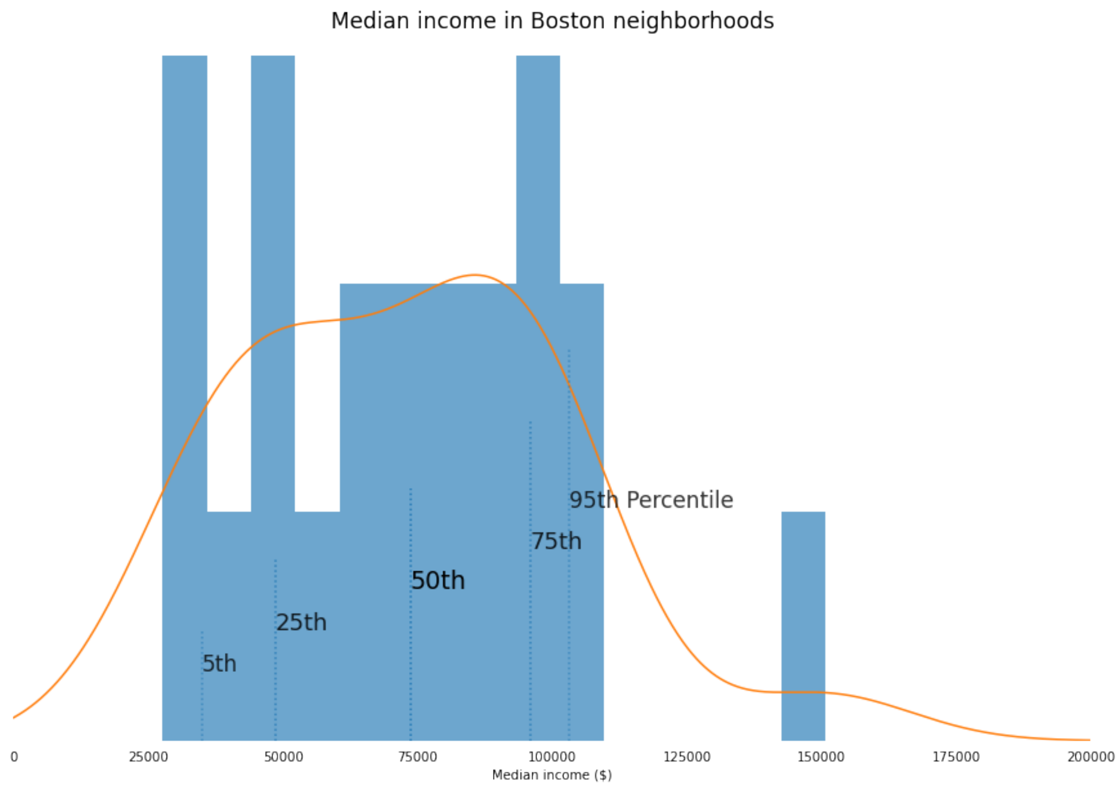
[4]:

|  | neighborhoods | median_income | total_households |
|---|---|---|---|
| 0 | Allston | 46982.76 | 6457.0 |
| 1 | Back Bay | 102070.55 | 9824.0 |
| 2 | Beacon Hill | 98069.24 | 5458.0 |
| 3 | Brighton | 62041.20 | 21605.0 |
| 4 | Charlestown | 103243.16 | 8931.0 |
| 5 | Dorchester | 49662.36 | 44086.0 |
| 6 | Downtown | 67367.09 | 7552.0 |
| 7 | East Boston | 52935.36 | 16286.0 |
| 8 | Fenway | 39549.84 | 10926.0 |
| 9 | Hyde Park | 70810.13 | 12891.0 |

```
10            Jamaica Plain      84445.90          16092.0
11                 Longwood      35000.00            280.0
12                 Mattapan      48196.90           8866.0
13             Mission Hill      35707.32           6270.0
14                North End      97110.39           5338.0
15               Roslindale      76666.67          11406.0
16                  Roxbury      27721.35          19406.0
17             South Boston      93077.60          16628.0
18  South Boston Waterfront     150677.51           1830.0
19                South End      86994.80          16193.0
20                 West End      96787.15           3134.0
21             West Roxbury      80804.46          13757.0
```

[5]:
```
# code borrowed from Max Hilsdorf's article on Towards Data Science
# https://towardsdatascience.com/
 ↪take-your-histograms-to-the-next-level-using-matplotlib-5f093ad7b9d3
fig1, ax1 = plt.subplots(figsize = (15,10))

# Plot
# Plot histogram
df.median_income.plot(kind = "hist", density = True, alpha = 0.65, bins = 15) #␣
 ↪change density to true, because KDE uses density

# Plot KDE
df.median_income.plot(kind = "kde")

# Quantile lines
quant_5, quant_25, quant_50, quant_75, quant_95 = df.median_income.quantile(0.
 ↪05),df.median_income.quantile(0.25),df.median_income.quantile(0.5),df.
 ↪median_income.quantile(0.75),df.median_income.quantile(0.95)
quants = [[quant_5, 0.6, 0.16], [quant_25, 0.8, 0.26], [quant_50, 1, 0.36], ␣
 ↪[quant_75, 0.8, 0.46], [quant_95, 0.6, 0.56]]
for i in quants:
    ax1.axvline(i[0], alpha = i[1], ymax = i[2], linestyle = ":")

# X
ax1.set_xlabel("Median income ($)")

# Limit x range to 0-200,000
x_start, x_end = 0, 200_000
ax1.set_xlim(x_start, x_end)

# Y
ax1.set_ylim(0, 0.000017)
ax1.set_yticklabels([])
ax1.set_ylabel("")
```

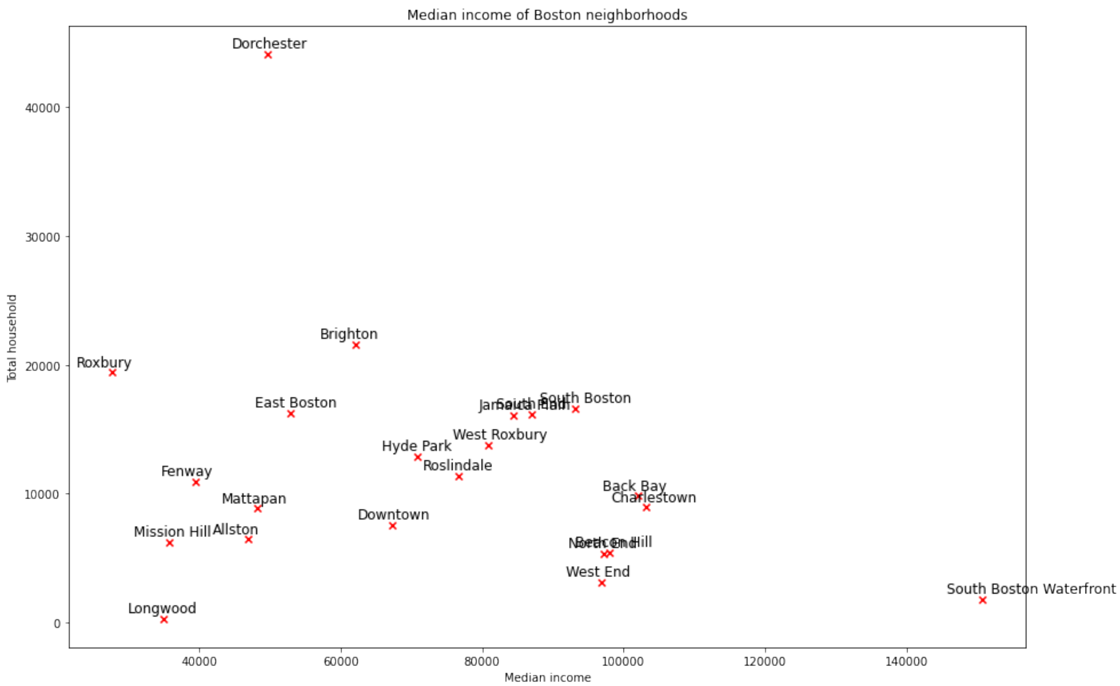
```
# Annotations
ax1.text(quant_5-.1, 0.0000017, "5th", size = 17, alpha = 0.8)
ax1.text(quant_25-.13, 0.0000027, "25th", size = 18, alpha = 0.85)
ax1.text(quant_50-.13, 0.0000037, "50th", size = 19, alpha = 1)
ax1.text(quant_75-.13, 0.0000047, "75th", size = 18, alpha = 0.85)
ax1.text(quant_95-.25, 0.0000057, "95th Percentile", size = 17, alpha =.8)

# Overall
ax1.grid(False)
ax1.set_title("Median income in Boston neighborhoods", size = 17, pad = 10)

# Remove ticks and spines
ax1.tick_params(left = False, bottom = False)
for ax1, spine in ax1.spines.items():
    spine.set_visible(False)

#plt.show()
```




[6]:
```
fig, ax = plt.subplots(figsize=(15, 10))
ax.scatter(x = df.median_income, y = df.total_households, marker="x", c="red")

for i,neighborhood in enumerate(df.neighborhoods):
```

```python
    x = df.median_income[i]
    y = df.total_households[i]
    ax.text(x-5000, y+500, neighborhood, fontsize=12)

plt.xlabel("Median income")
plt.ylabel("Total household")
plt.title("Median income of Boston neighborhoods")
```

[6]: Text(0.5, 1.0, 'Median income of Boston neighborhoods')




[ go back to top ]

### 2.1.11 Retrieve the coordinates and write to previous dataframe

[7]:
```python
from geopy.geocoders import Nominatim
import folium # map rendering library
```

[8]:
```python
for val, neighborhood in enumerate(df.neighborhoods):
    geolocator = Nominatim(user_agent="boston_explorer")
    loc = geolocator.geocode(neighborhood +', Boston, MA, USA')
    df.loc[val, 'lat'] = loc.latitude
    df.loc[val, 'lon'] = loc.longitude
df
```

[8]:
```
              neighborhoods  median_income  total_households        lat  \
0                   Allston       46982.76            6457.0  42.355434
1                  Back Bay      102070.55            9824.0  42.350549
2               Beacon Hill       98069.24            5458.0  42.358708
3                  Brighton       62041.20           21605.0  42.350097
4               Charlestown      103243.16            8931.0  42.377875
5                Dorchester       49662.36           44086.0  42.297320
6                  Downtown       67367.09            7552.0  42.354886
7               East Boston       52935.36           16286.0  42.375097
8                    Fenway       39549.84           10926.0  42.345187
9                 Hyde Park       70810.13           12891.0  42.255654
10            Jamaica Plain       84445.90           16092.0  42.309820
11                 Longwood       35000.00             280.0  42.341826
12                 Mattapan       48196.90            8866.0  42.267566
13             Mission Hill       35707.32            6270.0  42.332560
14                North End       97110.39            5338.0  42.365097
15               Roslindale       76666.67           11406.0  42.291209
16                  Roxbury       27721.35           19406.0  42.324843
17             South Boston       93077.60           16628.0  42.333431
18  South Boston Waterfront      150677.51            1830.0  42.333431
19                South End       86994.80           16193.0  42.341310
20                 West End       96787.15            3134.0  42.363919
21             West Roxbury       80804.46           13757.0  42.279265

          lon
0  -71.132127
1  -71.080311
2  -71.067829
3  -71.156442
4  -71.061996
5  -71.074495
6  -71.061118
7  -71.039217
8  -71.104599
9  -71.124496
10 -71.120330
11 -71.109798
12 -71.092427
13 -71.103608
14 -71.054495
15 -71.124497
16 -71.095016
17 -71.049495
18 -71.049495
19 -71.077230
20 -71.063899
21 -71.149497
```

[ go back to top ]

#### 2.1.12 Retrieve Boston coordinates and create a map of its neighborhoods

[9]:
```
#Let's get the geographical coordinates of Boston
address = 'Boston, MA'

geolocator = Nominatim(user_agent="boston_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Boston are {}, {}.'.format(latitude,␣
 ↪longitude))
```

```
The geograpical coordinate of Boston are 42.3602534, -71.0582912.
```

[10]:
```
# create map of Boston using latitude and longitude values
map_boston = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lon, neighborhood in zip(df['lat'], df['lon'], df['neighborhoods']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_boston)

map_boston
```

[10]:
```
<folium.folium.Map at 0x22de0c80dc0>
```

#### 2.1.13 Load the shape file of neighborhood and display choropleth map of median income in Boston neighborhoods

[11]:
```
#https://data.boston.gov/dataset/boston-neighborhoods
url = (
    "http://bostonopendata-boston.opendata.arcgis.com/datasets/
 ↪3525b0ee6e6b427f9aab5d0a1d0a1a28_0.geojson?outSR={%22latestWkid%22:
 ↪2249,%22wkid%22:102686}"
)
```

[12]:
```
#response = requests.get(f"{url}/limadmin.geojson")
#neighborhood_geo = response.json()
neighborhood_geo = f"{url}"
neighborhood_data = df.loc[:, ('neighborhoods','median_income')]
bins = list(neighborhood_data["median_income"].quantile([0, 0.25, 0.5, 0.75,␣
↪1]))

map_boston = folium.Map(location=[42.310320, -71.074495], zoom_start=11.5)

folium.Choropleth(
    geo_data=neighborhood_geo,
    name="choropleth",
    data=neighborhood_data,
    columns=['neighborhoods','median_income'],
    key_on="feature.properties.Name",
    fill_color="BuPu",
    fill_opacity=0.8,
    line_opacity=0.2,
    legend_name="Median income",
    bins=bins,
    reset=True,
    label=True
).add_to(map_boston)

folium.LayerControl().add_to(map_boston)

map_boston
```

[12]:
```
<folium.folium.Map at 0x22de0c6fa30>
```

[ go back to top ]

### 2.1.14 Get nearby restaurants to the chosen location and assess competition

**Getting data from Foursquare**

[13]:
```
#Define Foursquare Credentials and Version
CLIENT_ID = '2NOBVJ3XJE2WC0Q35EPX3JELGHNB42IX5LYOAFTVM4RWCRMO' # your␣
↪Foursquare ID
CLIENT_SECRET = 'PAOPLFNOIQ42WGH21EXCPVKGCL5HOVBLWYCHTDGCFBV0K4RO' # your␣
↪Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

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

```
Your credentails:
```

```
CLIENT_ID: 2NOBVJ3XJE2WC0Q35EPX3JELGHNB42IX5LYOAFTVM4RWCRM0
CLIENT_SECRET:PAOPLFN0IQ42WGH21EXCPVKGCL5H0VBLWYCHTDGCFBV0K4RO
```

**Explore one of the neighborhoods** Let's pick *Dorchester*. According to the charts above, this neighborhood was distinctive from other with high number of households and relatively low median income.

[14]:
```
df.loc[5, "neighborhoods"]
```

[14]:
```
'Dorchester'
```

[15]:
```
neighborhood_latitude = df.loc[5, 'lat'] # neighborhood latitude value
neighborhood_longitude = df.loc[5, 'lon'] # neighborhood longitude value

neighborhood_name = df.loc[5, 'neighborhoods'] # neighborhood name

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

```
Latitude and longitude values of Dorchester are 42.2973205, -71.0744952.
```

Now, let's get the top 100 venues that are in *Dorchester* within a radius of 500 meters. First, we need to create the GET request URL with Foursquare API credentials.

[16]:
```
LIMIT = 100 # limit of number of venues returned by Foursquare API

radius = 1000 # define radius

# create URL
url = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url # display URL
```

[16]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=2NOBVJ3XJE2WC0Q35EPX3JE
LGHNB42IX5LYOAFTVM4RWCRM0&client_secret=PAOPLFN0IQ42WGH21EXCPVKGCL5H0VBLWYCHTDGC
FBV0K4RO&v=20180605&ll=42.2973205,-71.0744952&radius=1000&limit=100'
```

[17]:
```
results = requests.get(url).json()
#results
```

[18]:
```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

[19]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
↪'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type,␣
↪axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

```
<ipython-input-19-561c05f0fdd1>:3: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  nearby_venues = json_normalize(venues) # flatten JSON
```

[19]:
```
                                          name  \
0                                  Daily Table
1                           Down Home Delivery
2  William J Devine Franklin Park Golf Course
3                                 Stash's Pizza
4                                     Walgreens

                     categories        lat        lng
0                        Market  42.295689 -71.071979
1  Southern / Soul Food Restaurant  42.299496 -71.073426
2                   Golf Course  42.300395 -71.080642
3                   Pizza Place  42.300391 -71.080660
```

```
4                          Pharmacy  42.291078 -71.072110
```

```
[20]: print(nearby_venues.shape)
      print(len(nearby_venues.categories.unique()))
```

```
(21, 4)
19
```

```
[21]: print('Within {} meters radius in {} \
      there are {} venues \
      accross {} uniques categories.'.format(radius, neighborhood_name,
                                             nearby_venues.shape[0],
                                             len(nearby_venues.categories.unique())))
```

```
Within 1000 meters radius in Dorchester there are 21 venues accross 19 uniques
categories.
```

[ go back to top ]

#### 2.1.15 Analyze Each Neighborhood

Now let's see how other neighborhoods stand.

The function below will repeat the same process to all the neighborhoods in Boston.

```
[22]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

          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={}'.format(
                  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)
```

Let's run the above function on each neighborhood and create a new dataframe called *boston_venues*.

[23]:
```
boston_venues = getNearbyVenues(names=df['neighborhoods'],
                                latitudes=df['lat'],
                                longitudes=df['lon']
                               )
```

```
Allston
Back Bay
Beacon Hill
Brighton
Charlestown
Dorchester
Downtown
East Boston
Fenway
Hyde Park
Jamaica Plain
Longwood
Mattapan
Mission Hill
North End
Roslindale
Roxbury
South Boston
South Boston Waterfront
South End
West End
West Roxbury
```

**Now let's explore the resulting dataframe**

[24]:
```
print(boston_venues.shape)
boston_venues.head()
```

```
(882, 7)
```

[24]:
```
  Neighborhood  Neighborhood Latitude  Neighborhood Longitude  \
0      Allston              42.355434              -71.132127
1      Allston              42.355434              -71.132127
2      Allston              42.355434              -71.132127
3      Allston              42.355434              -71.132127
4      Allston              42.355434              -71.132127

                  Venue  Venue Latitude  Venue Longitude  \
0         Lulu's Allston       42.355068       -71.134107
1          Allston Diner       42.354979       -71.134295
2         Kaju Tofu House       42.354329       -71.132374
3             Azama Grill       42.354422       -71.132358
4  Fish Market Sushi Bar       42.353039       -71.132975

             Venue Category
0  Comfort Food Restaurant
1                    Diner
2        Korean Restaurant
3       Falafel Restaurant
4         Sushi Restaurant
```

[25]:
```
boston_venues.groupby('Neighborhood').count()

print('There are {} uniques categories.'.format(len(boston_venues['Venue␣
↪Category'].unique())))
```

```
There are 184 uniques categories.
```

Following sequence of actions: * One hot encode the categories column (183 categories) into a separate dataframe * Add neighborhood data to this column * Examine the new dataframe size * Group rows by neighborhood and by taking the mean of the frequency of occurrence of each category * Examine the new dataframe and its size * Print each neighborhood along with the top 5 most common venues

[26]:
```
# one hot encoding
boston_onehot = pd.get_dummies(boston_venues[['Venue Category']], prefix="",␣
↪prefix_sep="")

# add neighborhood column back to dataframe
boston_onehot['Neighborhood'] = boston_venues['Neighborhood']

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

```
boston_onehot = boston_onehot[fixed_columns]

boston_onehot.head()
```

[26]:

```
   Yoga Studio  ATM  Accessories Store  American Restaurant  Arepa Restaurant  \
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

   Art Gallery  Asian Restaurant  Athletics & Sports  Automotive Shop  \
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

   BBQ Joint  …  Tourist Information Center  Trail  Train Station  \
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

   Udon Restaurant  Vegetarian / Vegan Restaurant  Video Game Store  \
0                0                              0                 0
1                0                              0                 0
2                0                              0                 0
3                0                              0                 0
4                0                              0                 0

   Vietnamese Restaurant  Wine Bar  Wine Shop  Women's 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

[5 rows x 184 columns]
```

[27]:

```
boston_onehot.shape
#boston_onehot.columns
```

[27]: (882, 184)

[28]:
```
boston_grouped = boston_onehot.groupby('Neighborhood').mean().reset_index()
boston_grouped
```

[28]:
```
                 Neighborhood  Yoga Studio       ATM  Accessories Store  \
0                     Allston     0.000000  0.000000           0.000000
1                    Back Bay     0.010000  0.000000           0.000000
2                 Beacon Hill     0.024390  0.000000           0.000000
3                    Brighton     0.000000  0.000000           0.000000
4                 Charlestown     0.027778  0.000000           0.000000
5                  Dorchester     0.000000  0.000000           0.000000
6                    Downtown     0.000000  0.000000           0.000000
7                 East Boston     0.000000  0.027778           0.000000
8                      Fenway     0.027778  0.000000           0.000000
9                   Hyde Park     0.000000  0.052632           0.000000
10              Jamaica Plain     0.000000  0.000000           0.050000
11                   Longwood     0.025641  0.000000           0.000000
12                   Mattapan     0.000000  0.000000           0.000000
13               Mission Hill     0.000000  0.000000           0.000000
14                  North End     0.014286  0.000000           0.000000
15                 Roslindale     0.125000  0.000000           0.000000
16                    Roxbury     0.000000  0.000000           0.000000
17               South Boston     0.000000  0.000000           0.000000
18    South Boston Waterfront     0.000000  0.000000           0.000000
19                  South End     0.023256  0.000000           0.023256
20                   West End     0.011236  0.000000           0.000000
21               West Roxbury     0.000000  0.000000           0.000000

    American Restaurant  Arepa Restaurant  Art Gallery  Asian Restaurant  \
0              0.000000          0.000000     0.013158          0.026316
1              0.040000          0.000000     0.000000          0.010000
2              0.024390          0.000000     0.000000          0.000000
3              0.021739          0.000000     0.000000          0.000000
4              0.027778          0.000000     0.000000          0.000000
5              0.000000          0.000000     0.000000          0.000000
6              0.020000          0.000000     0.000000          0.050000
7              0.027778          0.000000     0.055556          0.000000
8              0.083333          0.000000     0.000000          0.000000
9              0.105263          0.000000     0.000000          0.000000
10             0.050000          0.000000     0.100000          0.000000
11             0.025641          0.000000     0.000000          0.000000
12             0.000000          0.000000     0.000000          0.000000
13             0.043478          0.000000     0.000000          0.000000
14             0.000000          0.000000     0.000000          0.000000
15             0.000000          0.000000     0.000000          0.000000
16             0.000000          0.000000     0.125000          0.000000
17             0.028571          0.000000     0.000000          0.000000
18             0.028571          0.000000     0.000000          0.000000
```

| | | | | |
|---|---|---|---|---|
| 19 | 0.023256 | 0.023256 | 0.000000 | 0.000000 |
| 20 | 0.033708 | 0.000000 | 0.000000 | 0.000000 |
| 21 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

| | Athletics & Sports | Automotive Shop | … | Tourist Information Center | \ |
|---|---|---|---|---|---|
| 0 | 0.000000 | 0.013158 | … | 0.00 | |
| 1 | 0.010000 | 0.000000 | … | 0.00 | |
| 2 | 0.000000 | 0.000000 | … | 0.00 | |
| 3 | 0.000000 | 0.000000 | … | 0.00 | |
| 4 | 0.027778 | 0.000000 | … | 0.00 | |
| 5 | 0.000000 | 0.000000 | … | 0.00 | |
| 6 | 0.010000 | 0.000000 | … | 0.01 | |
| 7 | 0.000000 | 0.000000 | … | 0.00 | |
| 8 | 0.000000 | 0.000000 | … | 0.00 | |
| 9 | 0.000000 | 0.000000 | … | 0.00 | |
| 10 | 0.000000 | 0.000000 | … | 0.00 | |
| 11 | 0.000000 | 0.000000 | … | 0.00 | |
| 12 | 0.000000 | 0.000000 | … | 0.00 | |
| 13 | 0.000000 | 0.000000 | … | 0.00 | |
| 14 | 0.000000 | 0.000000 | … | 0.00 | |
| 15 | 0.000000 | 0.000000 | … | 0.00 | |
| 16 | 0.000000 | 0.000000 | … | 0.00 | |
| 17 | 0.000000 | 0.000000 | … | 0.00 | |
| 18 | 0.000000 | 0.000000 | … | 0.00 | |
| 19 | 0.000000 | 0.000000 | … | 0.00 | |
| 20 | 0.000000 | 0.000000 | … | 0.00 | |
| 21 | 0.000000 | 0.000000 | … | 0.00 | |

| | Trail | Train Station | Udon Restaurant | Vegetarian / Vegan Restaurant | \ |
|---|---|---|---|---|---|
| 0 | 0.00 | 0.000000 | 0.000000 | 0.026316 | |
| 1 | 0.01 | 0.000000 | 0.000000 | 0.000000 | |
| 2 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 3 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 4 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 5 | 0.00 | 0.000000 | 0.000000 | 0.090909 | |
| 6 | 0.00 | 0.000000 | 0.000000 | 0.010000 | |
| 7 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 8 | 0.00 | 0.000000 | 0.027778 | 0.000000 | |
| 9 | 0.00 | 0.052632 | 0.000000 | 0.000000 | |
| 10 | 0.05 | 0.000000 | 0.000000 | 0.000000 | |
| 11 | 0.00 | 0.000000 | 0.000000 | 0.025641 | |
| 12 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 13 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 14 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 15 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 16 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |
| 17 | 0.00 | 0.000000 | 0.000000 | 0.000000 | |

| | | | | |
|---|---|---|---|---|
| 18 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 19 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 20 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 21 | 0.00 | 0.000000 | 0.000000 | 0.000000 |

| | Video Game Store | Vietnamese Restaurant | Wine Bar | Wine Shop \ |
|---|---|---|---|---|
| 0 | 0.00 | 0.013158 | 0.000000 | 0.000000 |
| 1 | 0.00 | 0.010000 | 0.000000 | 0.000000 |
| 2 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.00 | 0.000000 | 0.000000 | 0.021739 |
| 4 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 6 | 0.01 | 0.010000 | 0.000000 | 0.010000 |
| 7 | 0.00 | 0.027778 | 0.000000 | 0.000000 |
| 8 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 11 | 0.00 | 0.000000 | 0.000000 | 0.025641 |
| 12 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 13 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 14 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 15 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 16 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 17 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 18 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 19 | 0.00 | 0.000000 | 0.046512 | 0.046512 |
| 20 | 0.00 | 0.000000 | 0.000000 | 0.000000 |
| 21 | 0.00 | 0.000000 | 0.000000 | 0.000000 |

| | Women's Store |
|---|---|
| 0 | 0.00 |
| 1 | 0.02 |
| 2 | 0.00 |
| 3 | 0.00 |
| 4 | 0.00 |
| 5 | 0.00 |
| 6 | 0.00 |
| 7 | 0.00 |
| 8 | 0.00 |
| 9 | 0.00 |
| 10 | 0.00 |
| 11 | 0.00 |
| 12 | 0.00 |
| 13 | 0.00 |
| 14 | 0.00 |
| 15 | 0.00 |
| 16 | 0.00 |

```
17           0.00
18           0.00
19           0.00
20           0.00
21           0.00

[22 rows x 184 columns]
```

[29]:
```
boston_grouped.shape
```

[29]:
```
(22, 184)
```

**Neighborhoods along with the top 5 most common venues**

[30]:
```
num_top_venues = 5

for hood in boston_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = boston_grouped[boston_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_venues))
    print('\n')
```

```
----Allston----
                venue  freq
0   Korean Restaurant  0.08
1         Pizza Place  0.04
2  Chinese Restaurant  0.04
3              Bakery  0.04
4     Thai Restaurant  0.04


----Back Bay----
                venue  freq
0         Coffee Shop  0.08
1               Hotel  0.05
2  Italian Restaurant  0.05
3  Seafood Restaurant  0.05
4       Cosmetics Shop  0.05


----Beacon Hill----
                venue  freq
```

```
0  Italian Restaurant  0.07
1           Hotel Bar  0.05
2         Pizza Place  0.05
3   French Restaurant  0.05
4               Hotel  0.05


----Brighton----
           venue  freq
0    Bus Station  0.07
1         Bakery  0.07
2    Pizza Place  0.07
3  Deli / Bodega  0.07
4           Bank  0.07


----Charlestown----
         venue  freq
0  Pizza Place  0.11
1  Coffee Shop  0.08
2          Pub  0.06
3   Donut Shop  0.06
4    Gastropub  0.06


----Dorchester----
                           venue  freq
0                          Plaza  0.09
1                            Gym  0.09
2                     Shoe Store  0.09
3                           Food  0.09
4  Southern / Soul Food Restaurant  0.09


----Downtown----
                venue  freq
0         Coffee Shop  0.07
1              Bakery  0.06
2    Asian Restaurant  0.05
3      Sandwich Place  0.04
4  Chinese Restaurant  0.04


----East Boston----
               venue  freq
0        Pizza Place  0.06
1           Pharmacy  0.06
2  Convenience Store  0.06
```

```
3        Art Gallery  0.06
4     Sandwich Place  0.06


----Fenway----
                 venue  freq
0  American Restaurant  0.08
1               Bakery  0.06
2                 Café  0.06
3   Chinese Restaurant  0.06
4      Thai Restaurant  0.06


----Hyde Park----
                 venue  freq
0          Pizza Place  0.11
1  American Restaurant  0.11
2             Pharmacy  0.05
3       Ice Cream Shop  0.05
4           Donut Shop  0.05


----Jamaica Plain----
          venue  freq
0        Bakery  0.10
1   Art Gallery  0.10
2   Coffee Shop  0.10
3          Park  0.05
4  Liquor Store  0.05


----Longwood----
                  venue  freq
0                  Park  0.08
1           Coffee Shop  0.08
2            Donut Shop  0.08
3    Falafel Restaurant  0.05
4  Fast Food Restaurant  0.05


----Mattapan----
                  venue  freq
0     Mobile Phone Shop  0.11
1                Bakery  0.11
2     Food & Drink Shop  0.11
3            Shoe Store  0.11
4  Caribbean Restaurant  0.11
```

```
----Mission Hill----
              venue  freq
0       Pizza Place  0.09
1    Sandwich Place  0.09
2              Café  0.09
3  Sushi Restaurant  0.09
4     Grocery Store  0.04


----North End----
                venue  freq
0  Italian Restaurant  0.20
1         Pizza Place  0.09
2  Seafood Restaurant  0.07
3              Bakery  0.07
4         Coffee Shop  0.06


----Roslindale----
                 venue  freq
0          Yoga Studio  0.12
1        Big Box Store  0.12
2     Cuban Restaurant  0.12
3                 Pool  0.12
4  Rental Car Location  0.12


----Roxbury----
                 venue  freq
0                Plaza  0.12
1                  Gym  0.12
2  Rental Car Location  0.12
3                 Park  0.12
4        Metro Station  0.12


----South Boston----
                venue  freq
0         Pizza Place  0.11
1        Liquor Store  0.06
2          Sports Bar  0.06
3  Italian Restaurant  0.06
4                 Bar  0.06


----South Boston Waterfront----
                venue  freq
```

```
0          Pizza Place  0.11
1         Liquor Store  0.06
2           Sports Bar  0.06
3   Italian Restaurant  0.06
4                  Bar  0.06


----South End----
                 venue  freq
0   Italian Restaurant  0.07
1          Coffee Shop  0.07
2    French Restaurant  0.05
3            Wine Shop  0.05
4             Wine Bar  0.05


----West End----
                venue  freq
0         Pizza Place  0.07
1               Hotel  0.06
2          Donut Shop  0.04
3                 Bar  0.04
4   Convenience Store  0.04


----West Roxbury----
                  venue  freq
0          Home Service   0.5
1        Discount Store   0.5
2          Yoga Studio    0.0
3           Opera House   0.0
4   Moroccan Restaurant   0.0
```

- Function to sort the venues in descending order.
- Create the new dataframe and display the top 10 venues for each neighborhood.

[31]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

[32]:
```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

```

```
# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = boston_grouped['Neighborhood']

for ind in np.arange(boston_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
 ↪return_most_common_venues(boston_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

[32]:

```
   Neighborhood 1st Most Common Venue 2nd Most Common Venue  \
0       Allston     Korean Restaurant           Pizza Place
1      Back Bay           Coffee Shop                 Hotel
2   Beacon Hill    Italian Restaurant             Hotel Bar
3      Brighton           Bus Station                Bakery
4   Charlestown           Pizza Place           Coffee Shop

  3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0    Chinese Restaurant                Bakery       Thai Restaurant
1    Italian Restaurant    Seafood Restaurant         Cosmetics Shop
2           Pizza Place     French Restaurant                 Hotel
3           Pizza Place          Deli / Bodega                  Bank
4                   Pub            Donut Shop             Gastropub

  6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
0    Fried Chicken Joint    Mexican Restaurant      Asian Restaurant
1        Clothing Store   American Restaurant      Department Store
2      Sushi Restaurant    Hotpot Restaurant                  Lake
3                   Pub           Coffee Shop    Chinese Restaurant
4              Pharmacy                  Café         Grocery Store

           9th Most Common Venue 10th Most Common Venue
0  Vegetarian / Vegan Restaurant         Bubble Tea Shop
1                   Dessert Shop           Shopping Mall
2                    Coffee Shop              Restaurant
3                  Grocery Store             Dry Cleaner
4                 Discount Store      Monument / Landmark
```

**Interactive and visual map with top 5 the most common venue types of a neighborhood**
Particulary we are interested where the restaurants are clustered.

[33]:
```
# map of boston with its neighborhoods
map_boston

tooltip = "Click me!"

# I can add marker one by one on the map
for i in range(0,len(neighborhoods_venues_sorted)):

    html = '''<b>{}</b><br>1: {}<br>2: {}<br>3: {}<br>4: {}<br>5: {}'''.
 ↪format(neighborhoods_venues_sorted.Neighborhood[i],
            neighborhoods_venues_sorted["1st Most Common Venue"][i],
            neighborhoods_venues_sorted["2nd Most Common Venue"][i],
            neighborhoods_venues_sorted["3rd Most Common Venue"][i],
            neighborhoods_venues_sorted["4th Most Common Venue"][i],
            neighborhoods_venues_sorted["5th Most Common Venue"][i])

    iframe = folium.IFrame(html,
                       width=230,
                       height=140)

    popup = folium.Popup(iframe,
                     max_width=400)

    folium.Marker([df.lat[i],
                  df.lon[i]],
                 popup=popup,
                 tooltip=tooltip
                ).add_to(map_boston)

map_boston
```

[33]:
```
<folium.folium.Map at 0x22de0c6fa30>
```

**Clustering neighborhood by total households and median income**

[34]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

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

[35]:
```
X = df.values[:,1:3]
X = np.nan_to_num(X)
cluster_dataset = StandardScaler().fit_transform(X)
```

```
#cluster_dataset
```

[36]:
```
num_clusters = 5

k_means = KMeans(init="k-means++", n_clusters=num_clusters, n_init=12)
k_means.fit(cluster_dataset)
labels = k_means.labels_

print(labels)
```

```
[2 0 0 1 0 4 2 1 2 3 3 2 2 2 0 3 1 3 0 3 0 3]
```

[37]:
```
df["Labels"] = labels
df.head(3)
```

[37]:

| | neighborhoods | median_income | total_households | lat | lon | Labels |
|---|---|---|---|---|---|---|
| 0 | Allston | 46982.76 | 6457.0 | 42.355434 | -71.132127 | 2 |
| 1 | Back Bay | 102070.55 | 9824.0 | 42.350549 | -71.080311 | 0 |
| 2 | Beacon Hill | 98069.24 | 5458.0 | 42.358708 | -71.067829 | 0 |

[38]:
```
centroids = df.groupby('Labels').mean()
centroids
```

[38]:

| | median_income | total_households | lat | lon |
|---|---|---|---|---|
| Labels | | | | |
| 0 | 107993.000000 | 5752.500000 | 42.358263 | -71.063004 |
| 1 | 47565.970000 | 19099.000000 | 42.350012 | -71.096892 |
| 2 | 45467.318333 | 6725.166667 | 42.332910 | -71.100613 |
| 3 | 82133.260000 | 14494.500000 | 42.301782 | -71.107591 |
| 4 | 49662.360000 | 44086.000000 | 42.297320 | -71.074495 |

Note that each row in our dataset represents a neighborhood, and therefore, each row is assigned a label (22 labels in total). Now we can easily check the centroid values by averaging the features in each cluster. k-means will partition neighborhoods into five groups since we specified the algorithm to generate 5 clusters. The neighborhoods in each cluster are similar to each other in terms of the features included in the dataset, i.e. median income and total households. Next we can create a profile for each group, considering the common characteristics of each cluster. For example, the 5 clusters can be: * 0: "Higher tier" * 1: "Lower tier" * 2: "Inbetweeners" * 3: "Mid tier" * 4: "The rich few"

[39]:
```
df
```

[39]:

| | neighborhoods | median_income | total_households | lat | \ |
|---|---|---|---|---|---|
| 0 | Allston | 46982.76 | 6457.0 | 42.355434 | |
| 1 | Back Bay | 102070.55 | 9824.0 | 42.350549 | |
| 2 | Beacon Hill | 98069.24 | 5458.0 | 42.358708 | |
| 3 | Brighton | 62041.20 | 21605.0 | 42.350097 | |
| 4 | Charlestown | 103243.16 | 8931.0 | 42.377875 | |

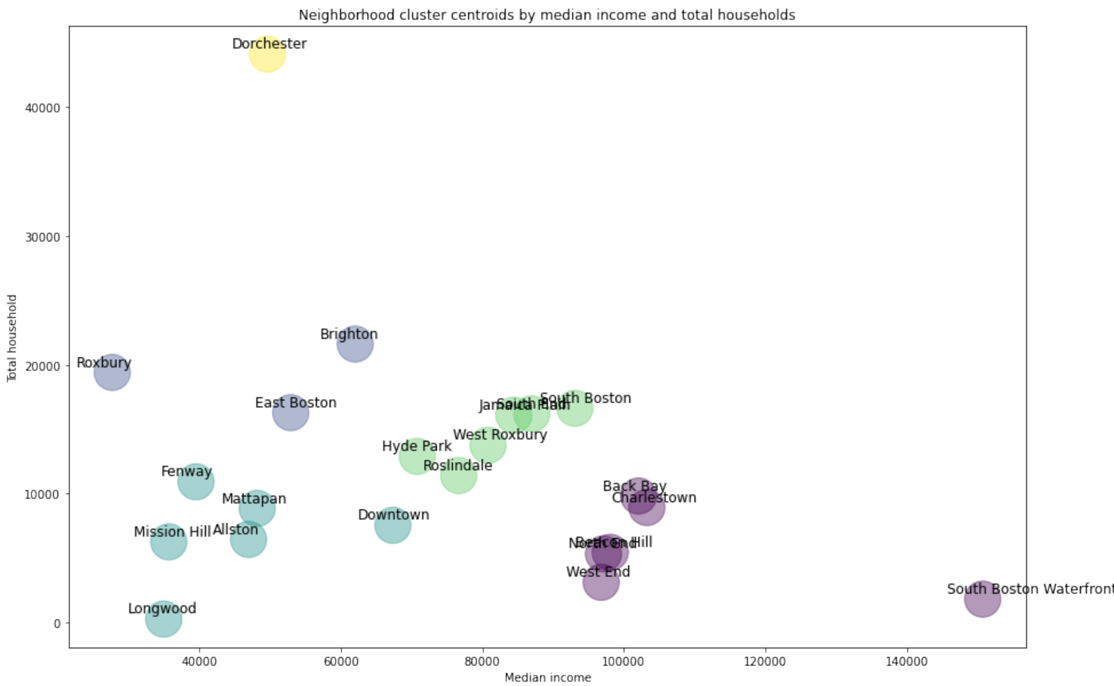
```
5                  Dorchester       49662.36           44086.0  42.297320
6                    Downtown       67367.09            7552.0  42.354886
7                 East Boston       52935.36           16286.0  42.375097
8                      Fenway       39549.84           10926.0  42.345187
9                   Hyde Park       70810.13           12891.0  42.255654
10              Jamaica Plain       84445.90           16092.0  42.309820
11                   Longwood       35000.00             280.0  42.341826
12                   Mattapan       48196.90            8866.0  42.267566
13               Mission Hill       35707.32            6270.0  42.332560
14                  North End       97110.39            5338.0  42.365097
15                 Roslindale       76666.67           11406.0  42.291209
16                    Roxbury       27721.35           19406.0  42.324843
17               South Boston       93077.60           16628.0  42.333431
18  South Boston Waterfront      150677.51            1830.0  42.333431
19                  South End       86994.80           16193.0  42.341310
20                   West End       96787.15            3134.0  42.363919
21               West Roxbury       80804.46           13757.0  42.279265

          lon  Labels
0  -71.132127       2
1  -71.080311       0
2  -71.067829       0
3  -71.156442       1
4  -71.061996       0
5  -71.074495       4
6  -71.061118       2
7  -71.039217       1
8  -71.104599       2
9  -71.124496       3
10 -71.120330       3
11 -71.109798       2
12 -71.092427       2
13 -71.103608       2
14 -71.054495       0
15 -71.124497       3
16 -71.095016       1
17 -71.049495       3
18 -71.049495       0
19 -71.077230       3
20 -71.063899       0
21 -71.149497       3
```

[40]:

```
fig, ax = plt.subplots(figsize=(15, 10))
ax.scatter(x = df.median_income, y = df.total_households, s=1000, alpha=0.4,␣
 ↪c=df.Labels)

for i,neighborhood in enumerate(df.neighborhoods):
```

```
    x = df.median_income[i]
    y = df.total_households[i]
    ax.text(x-5000, y+500, neighborhood, fontsize=12)

plt.xlabel("Median income")
plt.ylabel("Total household")
plt.title("Neighborhood cluster centroids by median income and total␣
 ↪households")
```

[40]: Text(0.5, 1.0, 'Neighborhood cluster centroids by median income and total households')




**Clustering neighborhoods by venues** Run k-means to cluster the neighborhood into 5 clusters.

[41]:
```
# set number of clusters
kclusters = 5

boston_grouped_clustering = boston_grouped.drop('Neighborhood', 1)

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

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

[41]:
```
array([2, 2, 2, 2, 3, 0, 2, 3, 2, 3])
```

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

[42]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

boston_merged = df

# merge boston_grouped with boston_data to add latitude/longitude for each␣
↪neighborhood
boston_merged = boston_merged.join(neighborhoods_venues_sorted.
↪set_index('Neighborhood'), on='neighborhoods')

boston_merged.head() # check the last columns!
```

[42]:
```
  neighborhoods  median_income  total_households        lat        lon  \
0       Allston       46982.76            6457.0  42.355434 -71.132127
1      Back Bay      102070.55            9824.0  42.350549 -71.080311
2   Beacon Hill       98069.24            5458.0  42.358708 -71.067829
3      Brighton       62041.20           21605.0  42.350097 -71.156442
4   Charlestown      103243.16            8931.0  42.377875 -71.061996

   Labels  Cluster Labels 1st Most Common Venue 2nd Most Common Venue  \
0       2               2     Korean Restaurant           Pizza Place
1       0               2           Coffee Shop                 Hotel
2       0               2    Italian Restaurant             Hotel Bar
3       1               2           Bus Station                Bakery
4       0               3           Pizza Place           Coffee Shop

  3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0    Chinese Restaurant                Bakery       Thai Restaurant
1    Italian Restaurant    Seafood Restaurant        Cosmetics Shop
2           Pizza Place     French Restaurant                 Hotel
3           Pizza Place          Deli / Bodega                  Bank
4                   Pub            Donut Shop             Gastropub

  6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
0    Fried Chicken Joint   Mexican Restaurant      Asian Restaurant
1         Clothing Store  American Restaurant      Department Store
2       Sushi Restaurant   Hotpot Restaurant                  Lake
3                    Pub          Coffee Shop    Chinese Restaurant
4               Pharmacy                 Café         Grocery Store
```

```
           9th Most Common Venue 10th Most Common Venue
0  Vegetarian / Vegan Restaurant        Bubble Tea Shop
1                   Dessert Shop          Shopping Mall
2                    Coffee Shop             Restaurant
3                  Grocery Store            Dry Cleaner
4                 Discount Store    Monument / Landmark
```

Finally, let's visualize the resulting clusters

[43]:
```python
# 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(boston_merged['lat'], boston_merged['lon'],␣
 ↪boston_merged['neighborhoods'], boston_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
```

[43]: <folium.folium.Map at 0x22de1ca2070>

**Summary of nieghborhood clusters with top 5 venue categories**

[44]:
```python
boston_merged[["neighborhoods",
               "Cluster Labels",
               "1st Most Common Venue",
               "2nd Most Common Venue",
               "3rd Most Common Venue",
               "4th Most Common Venue",
               "5th Most Common Venue"]].sort_values("Cluster Labels")
```

[44]:
```
                neighborhoods  Cluster Labels 1st Most Common Venue  \
12                   Mattapan               0     Mobile Phone Shop
```

| | | | |
|---|---|---|---|
| 5 | Dorchester | 0 | Plaza |
| 21 | West Roxbury | 1 | Home Service |
| 19 | South End | 2 | Italian Restaurant |
| 11 | Longwood | 2 | Park |
| 8 | Fenway | 2 | American Restaurant |
| 0 | Allston | 2 | Korean Restaurant |
| 3 | Brighton | 2 | Bus Station |
| 2 | Beacon Hill | 2 | Italian Restaurant |
| 1 | Back Bay | 2 | Coffee Shop |
| 6 | Downtown | 2 | Coffee Shop |
| 10 | Jamaica Plain | 2 | Bakery |
| 7 | East Boston | 3 | Pizza Place |
| 4 | Charlestown | 3 | Pizza Place |
| 9 | Hyde Park | 3 | Pizza Place |
| 20 | West End | 3 | Pizza Place |
| 13 | Mission Hill | 3 | Pizza Place |
| 14 | North End | 3 | Italian Restaurant |
| 16 | Roxbury | 3 | Plaza |
| 17 | South Boston | 3 | Pizza Place |
| 18 | South Boston Waterfront | 3 | Pizza Place |
| 15 | Roslindale | 4 | Yoga Studio |

| | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue \ |
|---|---|---|---|
| 12 | Bakery | Food & Drink Shop | Shoe Store |
| 5 | Gym | Shoe Store | Food |
| 21 | Discount Store | Yoga Studio | Opera House |
| 19 | Coffee Shop | French Restaurant | Wine Shop |
| 11 | Coffee Shop | Donut Shop | Falafel Restaurant |
| 8 | Bakery | Café | Chinese Restaurant |
| 0 | Pizza Place | Chinese Restaurant | Bakery |
| 3 | Bakery | Pizza Place | Deli / Bodega |
| 2 | Hotel Bar | Pizza Place | French Restaurant |
| 1 | Hotel | Italian Restaurant | Seafood Restaurant |
| 6 | Bakery | Asian Restaurant | Sandwich Place |
| 10 | Art Gallery | Coffee Shop | Park |
| 7 | Pharmacy | Convenience Store | Art Gallery |
| 4 | Coffee Shop | Pub | Donut Shop |
| 9 | American Restaurant | Pharmacy | Ice Cream Shop |
| 20 | Hotel | Donut Shop | Bar |
| 13 | Sandwich Place | Café | Sushi Restaurant |
| 14 | Pizza Place | Seafood Restaurant | Bakery |
| 16 | Gym | Rental Car Location | Park |
| 17 | Liquor Store | Sports Bar | Italian Restaurant |
| 18 | Liquor Store | Sports Bar | Italian Restaurant |
| 15 | Big Box Store | Cuban Restaurant | Pool |

5th Most Common Venue

```
12              Caribbean Restaurant
5    Southern / Soul Food Restaurant
21               Moroccan Restaurant
19                          Wine Bar
11              Fast Food Restaurant
8                    Thai Restaurant
0                    Thai Restaurant
3                               Bank
2                              Hotel
1                     Cosmetics Shop
6                 Chinese Restaurant
10                      Liquor Store
7                     Sandwich Place
4                          Gastropub
9                         Donut Shop
20                 Convenience Store
13                     Grocery Store
14                       Coffee Shop
16                     Metro Station
17                               Bar
18                               Bar
15               Rental Car Location
```

According to the list of top 5 venue categories, **Cluster 0** and **Cluster 1** are not rich for restaurants. Restaurants in these neighborhoods are 5th most common categories, but diverse (Caribbean, Moroccan, and Souther/Soul).

**Cluster 2** is the most diverse in its restaurants. Among top 5 venue categories there are American, Italian, Korean, French, Chinese, Asian, Thai cousines as well as Seafood, Falafel, and fast food restaurants.

**Cluster 3** seems to be popular with italian cousine judging by the number of pizza places these neighborhoods have. This may induce competitors or, on the contrary, drive crowd of pizza lovers to these neighborhoods. Therefore, further analysis needed if one would like to open here a restaurant with Italian cousine.

**Cluster 4** has only neighborhood. Roslindale neighborhood is distinct from others with its Yoga Studios as the most common categorie, followed by Big box stores and Cuban restaurants.

**Cluster 0**

[45]:
```
boston_merged.loc[boston_merged['Cluster Labels'] == 0, boston_merged.
 ↪columns[[1] + list(range(5, boston_merged.shape[1]))]]
```

[45]:
```
    median_income  Labels  Cluster Labels 1st Most Common Venue  \
5        49662.36       4               0                 Plaza
12       48196.90       2               0     Mobile Phone Shop

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
5                    Gym            Shoe Store                  Food
```

```
12                 Bakery     Food & Drink Shop            Shoe Store

                5th Most Common Venue 6th Most Common Venue  \
5   Southern / Soul Food Restaurant           Pizza Place
12             Caribbean Restaurant  Fast Food Restaurant

   7th Most Common Venue            8th Most Common Venue  \
5    Fried Chicken Joint                           Market
12                  Bank  Southern / Soul Food Restaurant

   9th Most Common Venue 10th Most Common Venue
5         Breakfast Spot               Pharmacy
12              Pharmacy      Other Repair Shop
```

### Cluster 1

[46]:
```python
boston_merged.loc[boston_merged['Cluster Labels'] == 1, boston_merged.
 ↪columns[[1] + list(range(5, boston_merged.shape[1]))]]
```

[46]:
```
    median_income  Labels  Cluster Labels 1st Most Common Venue  \
21       80804.46       3               1          Home Service

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
21        Discount Store           Yoga Studio           Opera House

   5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
21   Moroccan Restaurant         Movie Theater                Museum

   8th Most Common Venue 9th Most Common Venue   10th Most Common Venue
21           Music Venue         National Park  New American Restaurant
```

### Cluster 2

[47]:
```python
boston_merged.loc[boston_merged['Cluster Labels'] == 2, boston_merged.
 ↪columns[[1] + list(range(5, boston_merged.shape[1]))]]
```

[47]:
```
    median_income  Labels  Cluster Labels 1st Most Common Venue  \
0        46982.76       2               2     Korean Restaurant
1       102070.55       0               2           Coffee Shop
2        98069.24       0               2    Italian Restaurant
3        62041.20       1               2           Bus Station
6        67367.09       2               2           Coffee Shop
8        39549.84       2               2   American Restaurant
10       84445.90       3               2                Bakery
11       35000.00       2               2                  Park
19       86994.80       3               2    Italian Restaurant

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
0            Pizza Place    Chinese Restaurant                Bakery
```

```
1                  Hotel    Italian Restaurant    Seafood Restaurant
2              Hotel Bar           Pizza Place     French Restaurant
3                 Bakery           Pizza Place         Deli / Bodega
6                 Bakery     Asian Restaurant        Sandwich Place
8                 Bakery                  Café    Chinese Restaurant
10           Art Gallery           Coffee Shop                  Park
11           Coffee Shop           Donut Shop    Falafel Restaurant
19           Coffee Shop     French Restaurant             Wine Shop

   5th Most Common Venue 6th Most Common Venue   7th Most Common Venue  \
0        Thai Restaurant   Fried Chicken Joint      Mexican Restaurant
1          Cosmetics Shop       Clothing Store     American Restaurant
2                  Hotel     Sushi Restaurant      Hotpot Restaurant
3                   Bank                   Pub             Coffee Shop
6     Chinese Restaurant           Burger Joint       Sushi Restaurant
8        Thai Restaurant    Mexican Restaurant  Furniture / Home Store
10          Liquor Store              Theater      Seafood Restaurant
11  Fast Food Restaurant                 Hotel                Pharmacy
19              Wine Bar  Gym / Fitness Center                    Park

   8th Most Common Venue         9th Most Common Venue 10th Most Common Venue
0       Asian Restaurant  Vegetarian / Vegan Restaurant         Bubble Tea Shop
1       Department Store                  Dessert Shop          Shopping Mall
2                   Lake                   Coffee Shop             Restaurant
3     Chinese Restaurant                 Grocery Store            Dry Cleaner
6             Restaurant            Falafel Restaurant                  Hotel
8             Donut Shop                   Coffee Shop     Seafood Restaurant
10                 Trail                           Bar           Noodle House
11         Metro Station              College Cafeteria                 Diner
19                Bakery                           Bar              Gift Shop
```

**Cluster 3**

[48]:
```python
boston_merged.loc[boston_merged['Cluster Labels'] == 3, boston_merged.
 ↪columns[[1] + list(range(5, boston_merged.shape[1]))]]
```

[48]:
```
    median_income  Labels  Cluster Labels 1st Most Common Venue  \
4       103243.16       0               3           Pizza Place
7        52935.36       1               3           Pizza Place
9        70810.13       3               3           Pizza Place
13       35707.32       2               3           Pizza Place
14       97110.39       0               3    Italian Restaurant
16       27721.35       1               3                 Plaza
17       93077.60       3               3           Pizza Place
18      150677.51       0               3           Pizza Place
20       96787.15       0               3           Pizza Place

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
```

```
4           Coffee Shop                 Pub          Donut Shop
7              Pharmacy   Convenience Store         Art Gallery
9   American Restaurant            Pharmacy      Ice Cream Shop
13       Sandwich Place                Café   Sushi Restaurant
14          Pizza Place   Seafood Restaurant             Bakery
16                  Gym  Rental Car Location               Park
17         Liquor Store           Sports Bar  Italian Restaurant
18         Liquor Store           Sports Bar  Italian Restaurant
20                Hotel           Donut Shop                 Bar

   5th Most Common Venue      6th Most Common Venue 7th Most Common Venue  \
4              Gastropub                   Pharmacy                  Café
7         Sandwich Place  Latin American Restaurant            Donut Shop
9             Donut Shop              Grocery Store                   Gym
13         Grocery Store                        Pub            Donut Shop
14           Coffee Shop                       Park                   Pub
16         Metro Station     Furniture / Home Store           Pizza Place
17                   Bar                Coffee Shop        Sandwich Place
18                   Bar                Coffee Shop        Sandwich Place
20     Convenience Store         Italian Restaurant            Sports Bar

   8th Most Common Venue 9th Most Common Venue   10th Most Common Venue
4          Grocery Store        Discount Store      Monument / Landmark
7             Restaurant         Grocery Store         Community Center
9       Business Service               Theater                      Bar
13                  Park           Coffee Shop  New American Restaurant
14         Historic Site        Sandwich Place           Ice Cream Shop
16           Art Gallery    Israeli Restaurant      Moroccan Restaurant
17      Sushi Restaurant    Mexican Restaurant       Chinese Restaurant
18      Sushi Restaurant    Mexican Restaurant       Chinese Restaurant
20                  Café        Sandwich Place              Coffee Shop
```

### Cluster 4

[49]:
```
boston_merged.loc[boston_merged['Cluster Labels'] == 4, boston_merged.
 ↪columns[[1] + list(range(5, boston_merged.shape[1]))]]
```

[49]:
```
    median_income  Labels  Cluster Labels 1st Most Common Venue  \
15       76666.67       3               4          Yoga Studio

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
15         Big Box Store      Cuban Restaurant                  Pool

   5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
15   Rental Car Location            Donut Shop         Scenic Lookout

   8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
15           Pizza Place   Japanese Restaurant            Music Venue
```

```
[ ]:
```