Applied Data Science Capstone Project

What's the best location to set up a high-end coffee shop in Chicago?

Introduction/Business problem:

Chicago is a great city to start new business ventures.

It welcomed nearly 60 Million domestic and international visitors in 2018, making it the second most visited city in the nation after New York.

Chicago counts almost 3M residents, and numerous universities and work places.

When it comes to set up a new business, like a high-end coffee shop, in a city like Chicago, choosing the right location can be overwhelming. There are so many factors to take into account, like competition. Depending on who you want to cater to, you might want to set up your new venture in a touristic location, near a university or college, or nearby work places.

This project proposes to identify the best location / neighborhood to set up a new coffee shop in Chicago. This will be done calling the Foursquare api to retrieve existing venues in all Chicago neighborhood, and then by applying a scoring model to make the recommendations.

Data:

We will be using different sources of data to deliver the project:

- Chicago list of neighborhoods: Data will be collected by scraping the Wikipedia page *https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Chicago*

- OpenStreetMap Data: We'll use Nominatim, the search engine for OpenStreetMap data, to retrieve the GPS coordinates for the list of Chicago neighborhoods

```
df2.head()
```

	Neighborhood	Community area	Latitude	Longitude
0	Albany Park	Albany Park	41.971937	-87.716174
1	Altgeld Gardens	Riverdale	41.654864	-87.600439
2	Andersonville	Edgewater	41.977139	-87.669273
3	Archer Heights	Archer Heights	41.811422	-87.726165
4	Armour Square	Armour Square	41.840033	-87.633107

```
df2.shape
```

```
(219, 4)
```

There are 219 possible neighborhoods in Chicago to choose from.

- Foursquare data : We'll query the Foursquare APIs to extract venues of interest (ex: Coffee Shops, Work Places, Universities…) and their location.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Distance
0	Albany Park	41.971937	-87.716174	Nighthawk	41.967974	-87.713415	Cocktail Bar	496
1	Albany Park	41.971937	-87.716174	Tre Kronor	41.975842	-87.711037	Scandinavian Restaurant	608
2	Albany Park	41.971937	-87.716174	Cairo Nights Hookah Lounge	41.975776	-87.715547	Hookah Bar	430
3	Albany Park	41.971937	-87.716174	Great Sea Chinese Restaurant	41.968496	-87.710678	Chinese Restaurant	594
4	Albany Park	41.971937	-87.716174	Popeye's Louisiana Kitchen	41.968459	-87.713156	Fast Food Restaurant	460
5	Albany Park	41.971937	-87.716174	2 Asian Brothers	41.975832	-87.709655	Vietnamese Restaurant	692
6	Albany Park	41.971937	-87.716174	Noon O Kabab	41.966700	-87.708332	Middle Eastern Restaurant	872
7	Albany Park	41.971937	-87.716174	Chicago Kalbi Korean BBQ	41.968314	-87.722771	Korean Restaurant	678
8	Albany Park	41.971937	-87.716174	Lawrence Fish Market	41.968280	-87.726250	Seafood Restaurant	928
9	Albany Park	41.971937	-87.716174	L.D. Pho	41.967316	-87.708499	Vietnamese Restaurant	817
10	Albany Park	41.971937	-87.716174	la Michoacana Premium	41.968559	-87.706510	Ice Cream Shop	883
11	Albany Park	41.971937	-87.716174	Eugene Field Park	41.974506	-87.721473	Park	523
12	Albany Park	41.971937	-87.716174	Chicago Produce	41.970553	-87.716327	Grocery Store	154
13	Albany Park	41.971937	-87.716174	El Gallo Bravo #6	41.968324	-87.721338	Mexican Restaurant	586

As a result of merging and analyzing this data, we'll provide a neighborhood of preference as an output, to set up a new high-end coffee shop.

Methodology:

This project has been conducted using Jupyter notebooks with cognitiveclass.ai. Coding is performed in Python.

We've used a free membership account to access the Foursquare APIs, which explains why some of the venue categories are limited to 100.

1 – Import required packages:

The first thing to do is to import all required packages.

These includes:

- Pandas, to easily manipulate data
- Numpy, to handle data in a vectorized way
- Matplotlib, to plot results and maps
- Kmeans, to build a k-means clustering model
- Nominatim, to retrieve GPS coordinates

- Folium, to render maps

```
# First, let's install all required packages

import pandas as pd
import requests


import numpy as np # library to handle data in a vectorized manner

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

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

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

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

!conda install -c conda-forge folium=0.5.0 --yes
#=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library
```

```
Solving environment: done
```

2 – Extract GPS coordinates for Chicago neighborhoods

Next, we need to get hold of all the data we'll need.

- List of Chicago neighborhoods:
  We'll scrape the Wikipedia site

```python
wiki_page = requests.get('https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Chicago').text
chicago_neighborhoods = pd.read_html(wiki_page, header=0, attrs={"class":"wikitable sortable"})[0]

chicago_neighborhoods.head(10)
```

	Neighborhood	Community area
0	Albany Park	Albany Park
1	Altgeld Gardens	Riverdale
2	Andersonville	Edgewater
3	Archer Heights	Archer Heights
4	Armour Square	Armour Square
5	Ashburn	Ashburn
6	Ashburn Estates	Ashburn
7	Auburn Gresham	Auburn Gresham
8	Avalon Park	Avalon Park
9	Avondale	Avondale

- Then, we'll retrieve latitude and longitude for those neighborhoods, using Nominatim.

```python
df2.head()
```

	Neighborhood	Community area	Latitude	Longitude
0	Albany Park	Albany Park	41.971937	-87.716174
1	Altgeld Gardens	Riverdale	41.654864	-87.600439
2	Andersonville	Edgewater	41.977139	-87.669273
3	Archer Heights	Archer Heights	41.811422	-87.726165
4	Armour Square	Armour Square	41.840033	-87.633107

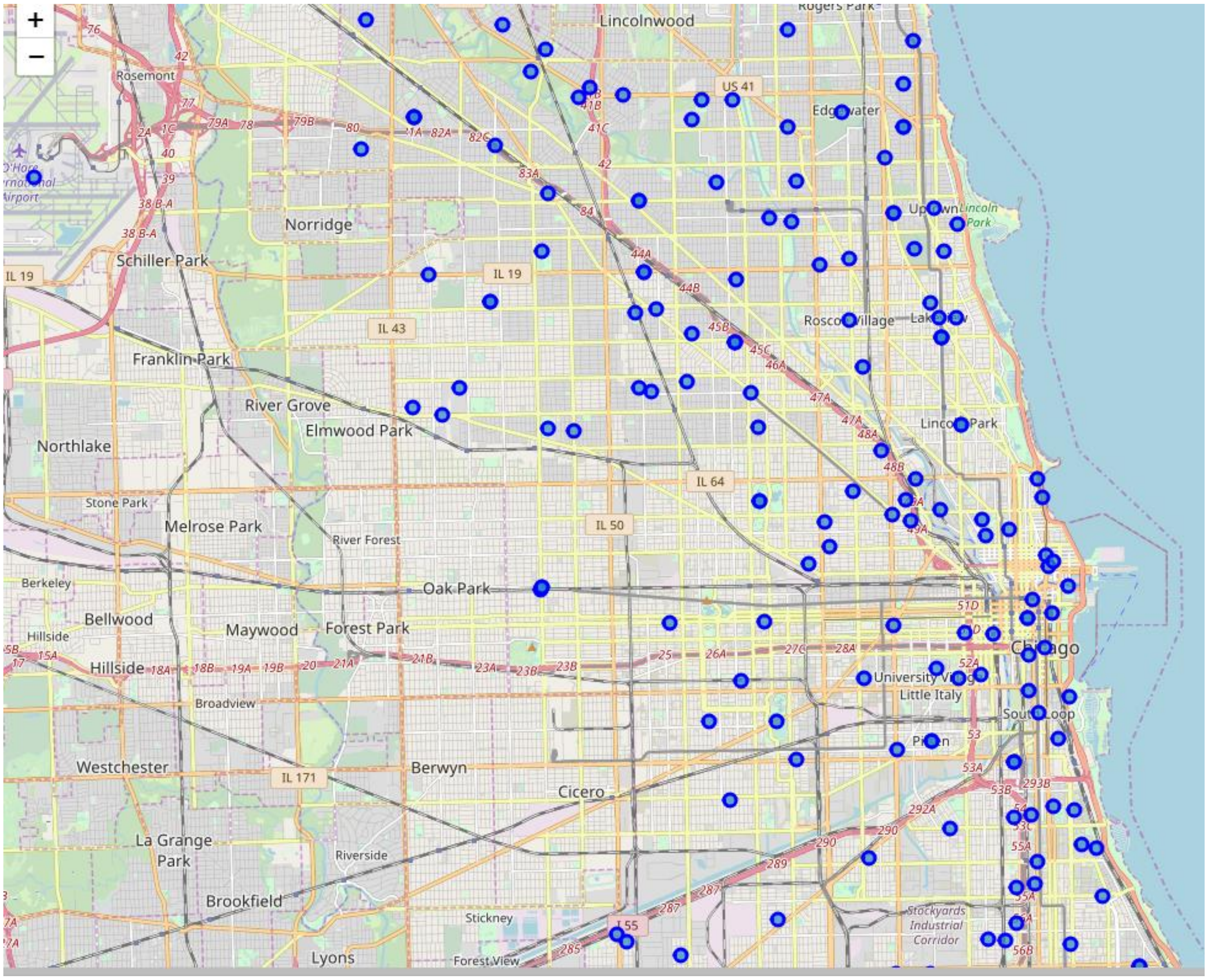
```python
df2.shape
```

```
(219, 4)
```

That's 219 neighborhoods to choose from

3 – Mapping the neighborhood

Plotting the neighborhoods on the map helps us figure out the stretch of Chicago



Not surprisingly, the closer we are to the city center, the more neighborhoods there are.

4 – Retrieve top venues categories across all neighborhoods

Using Foursquare, we extract venues categories for the Chicago neighborhoods.

```
[99]: # Let's take a look at the venue categories that we find the most in Chicago
chi = chicago_venues.groupby(['Venue Category'])['Venue Category'].count()
chi.nlargest(25)
```

```
[99]: Venue Category
Pizza Place               454
Mexican Restaurant        443
Bar                       386
Coffee Shop               386
Park                      376
Sandwich Place            373
Fast Food Restaurant      318
Grocery Store             258
American Restaurant       216
Italian Restaurant        212
Chinese Restaurant        204
Bakery                    193
Donut Shop                184
Café                      176
Discount Store            176
Gym                       156
Breakfast Spot            152
Pharmacy                  140
Ice Cream Shop            139
Gym / Fitness Center      138
Cosmetics Shop            135
Diner                     135
Sushi Restaurant          130
Hotel                     117
Liquor Store              117
Name: Venue Category, dtype: int64
```

Turns out the top venue category in Chicago is Pizza Place, while Coffee Shop if 3rd. We will need to take competition into account to make our decision as far as recommending a neighborhood for a high-end coffee shop.

5 – Extracting all venue categories of interest

We are going to create a score that will depend on a number of venue categories.

We've supposed that the main competition for a high-end coffee shop is other coffee shops, as well as dessert shops – like bakeries. So we need to know, for each neighborhood, if the competition is fierce or weak.

We also need to settle the coffee shop in a place where there are a lot of people. We chose to look a the venue categories Government offices and Universities / College for that.

We extract all venue categories of the 4 types in different dataframes:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Albany Park	41.971937	-87.716174	Consulate of Ecuador - Annex Office	41.968407	-87.710789	Embassy / Consulate
1	Altgeld Gardens	41.654864	-87.600439	MWRDGC - Calumet WRP	41.659293	-87.607727	Government Building
2	Andersonville	41.977139	-87.669273	http://www.andersonville.org/	41.976760	-87.668030	Government Building
3	Andersonville	41.977139	-87.669273	Fuller Counseling Group	41.975032	-87.675007	Office
4	Andersonville	41.977139	-87.669273	South-East Asia Center	41.975328	-87.659991	Government Building

Govt offices

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Albany Park	41.971937	-87.716174	Nyvall Hall, North Park Theological Seminary	41.974468	-87.711383	University
1	Albany Park	41.971937	-87.716174	North Park University	41.975154	-87.710029	University
2	Albany Park	41.971937	-87.716174	North Park Theological Seminary	41.974575	-87.711416	College Academic Building
3	Albany Park	41.971937	-87.716174	Caroline Hall - North Park University	41.974494	-87.710809	College Academic Building
4	Albany Park	41.971937	-87.716174	Magnuson Campus Center - North Park University	41.972823	-87.711839	General College & University

Universities

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Albany Park	41.971937	-87.716174	la Michoacana Premium	41.968559	-87.706510	Ice Cream Shop
1	Albany Park	41.971937	-87.716174	Jaafer Sweets	41.969251	-87.708161	Bakery
2	Albany Park	41.971937	-87.716174	Nazareth Sweets	41.965821	-87.708437	Bakery
3	Albany Park	41.971937	-87.716174	Markellos Baking Company	41.968602	-87.716607	Dessert Shop
4	Albany Park	41.971937	-87.716174	Baskin-Robbins	41.970937	-87.708739	Ice Cream Shop

Dessert shops

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Albany Park	41.971937	-87.716174	Starbucks	41.975922	-87.710019	Coffee Shop
1	Albany Park	41.971937	-87.716174	Cafe Chien	41.968060	-87.710754	Coffee Shop
2	Albany Park	41.971937	-87.716174	Viking Café	41.975091	-87.709615	Coffee Shop
3	Albany Park	41.971937	-87.716174	La Montana's Cafe	41.968433	-87.709750	Café
4	Albany Park	41.971937	-87.716174	Nancy's Rainbow	41.968433	-87.708520	Coffee Shop

Coffee shops

Next, we count the number of coffee shops, dessert shops, offices and universities in each neighborhood, and store that in a dataframe:

	Neighborhood	Community area	Latitude	Longitude	Universities	Offices	Bakeries	Coffee Shops
0	Albany Park	Albany Park	41.971937	-87.716174	64.0	1.0	15.0	6.0
1	Altgeld Gardens	Riverdale	41.654864	-87.600439	1.0	1.0	NaN	NaN
2	Andersonville	Edgewater	41.977139	-87.669273	15.0	6.0	25.0	12.0
3	Archer Heights	Archer Heights	41.811422	-87.726165	2.0	1.0	7.0	1.0
4	Armour Square	Armour Square	41.840033	-87.633107	54.0	6.0	7.0	5.0
5	Ashburn	Ashburn	41.747533	-87.711163	1.0	4.0	3.0	NaN
7	Auburn Gresham	Auburn Gresham	41.750474	-87.664304	NaN	4.0	4.0	1.0
8	Avalon Park	Avalon Park	41.745035	-87.588658	NaN	1.0	1.0	1.0
9	Avondale	Avondale	41.938921	-87.711168	6.0	3.0	18.0	6.0
10	Avondale Gardens	Irving Park	41.938921	-87.711168	6.0	3.0	18.0	6.0
11	Back of the Yards	New City	41.807533	-87.666163	3.0	5.0	5.0	2.0
12	Belmont Central	Belmont Cragin	41.939796	-87.653328	15.0	8.0	35.0	28.0
13	Belmont Gardens	Hermosa	41.939796	-87.653328	15.0	8.0	35.0	28.0
15	Belmont Terrace	Dunning	41.939796	-87.653328	15.0	8.0	35.0	28.0

6 – Computing the score:

The score, computed at for each Chicago neighborhood, will determine if the neighborhood is a good or bad prospect to welcome a new high-end coffee shop.

The score will depend on the 4 venue categories previously extracted. The higher the number of coffee and dessert shops in the neighborhood, the lower the score.

The higher the number of universities and government offices in the neighborhood, the higher the score.

Here are the weights chosen for each venue category:

Define weight of each venue category in score determination

```
weight_offices = 3 # Having offices around is good
```

```
weight_universities = 2 # Having universities around is good
```

```
weight_bakeries = -1 # Having bakeries around is bad (competition)
```

```
weight_coffee_shops = -3 # Having coffee shops around is bad (competition)
```

When we apply this logic to the dataframe previously created, we obtain the following:

Compute score

```
df_data = df_data.fillna(0)
chicago_score = df_data[['Neighborhood']].copy() # Create new df with just the neighborhood
chicago_score['Score'] = df_data['Offices'] * weight_offices + df_data['Universities'] * weight_universities + df_data['Bakeries'] * weight_bakeries + df_data['Coffee Shops'] * weight_coffee_shops
chicago_score = chicago_score.sort_values(by=['Score'], ascending=False)
```

```
chicago_score.head(5) # Top 5 recommendations
```

[42]:

	Neighborhood	Score
96	The Island	235.0
173	Printer's Row	204.0
212	Tri-Taylor	193.0
214	Union Ridge	191.0
215	University Village	184.0

Results:

The neighborhood ‘The Island’ emerges as the clear winner.

Let’s take a look at the map to see how the previously chosen venue categories are scattered today in this neighborhood:

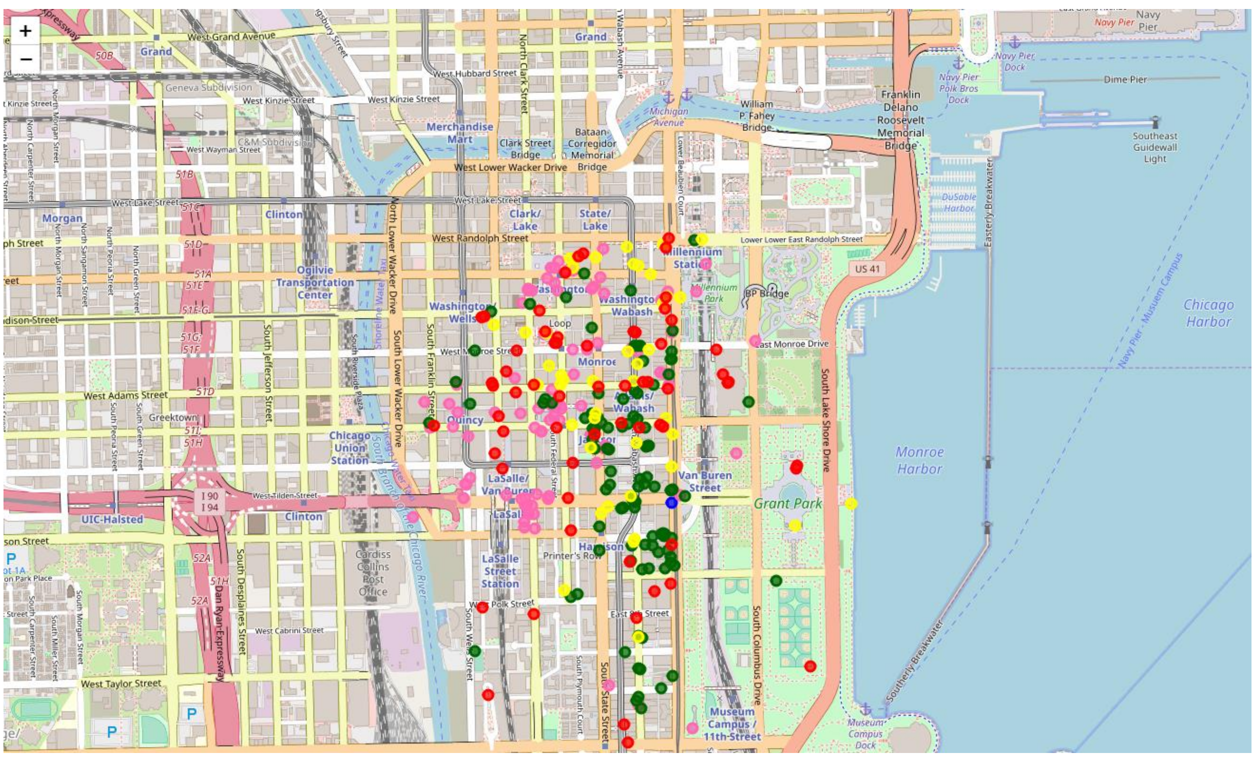
```python
#Get coordinates for Winning Neighborhood
address = 'The Island, Chicago, Illinois'

geolocator = Nominatim(user_agent="capstoneProject")
location = geolocator.geocode(address, timeout=60, exactly_one=True)
latitude = location.latitude
longitude = location.longitude
print("The winning neighbordhood Latidude is %s and Longitude is %s" % (latitude, longitude))
```

```
The winning neighbordhood Latidude is 41.8755616 and Longitude is -87.6244212
```

```python
newVenueMap(chicago_offices[chicago_offices['Neighborhood'] == 'The Island'], '#ff69b4', map_chicago_winner)  # offices in Pink
newVenueMap(chicago_universities[chicago_universities['Neighborhood'] == 'The Island'], '#006400', map_chicago_winner) # Universisties in Green
newVenueMap(chicago_bakeries[chicago_bakeries['Neighborhood'] == 'The Island'], '#ffff00', map_chicago_winner) # Dessert shops in Yellow
newVenueMap(chicago_coffee_shops[chicago_coffee_shops['Neighborhood'] == 'The Island'], '#ff0000', map_chicago_winner) # Coffee shops in Red

map_chicago_winner
```



So as a result, we could simply provide our recommendation as ‘The Island’. Or we could go a step further and look for neighborhoods that look similar to our winning one.

With K-means, we can cluster/segment neighborhoods according to the count of coffee/dessert shops, govt offices and universities.

Build a K-Means clustering model to see what other neighborhoods look like our winning neighborhood for more choices

```
#Copy dataframe
df_final = df_data.copy()

# Set number of clusters
clusters = 5

chicago_clustering = df_final.drop(['Neighborhood','Community area','Latitude','Longitude'], 1)

# run k-means
kmeans = KMeans(n_clusters=clusters, random_state=0).fit(chicago_clustering)

# check cluster labels
kmeans.labels_[0:10]
```

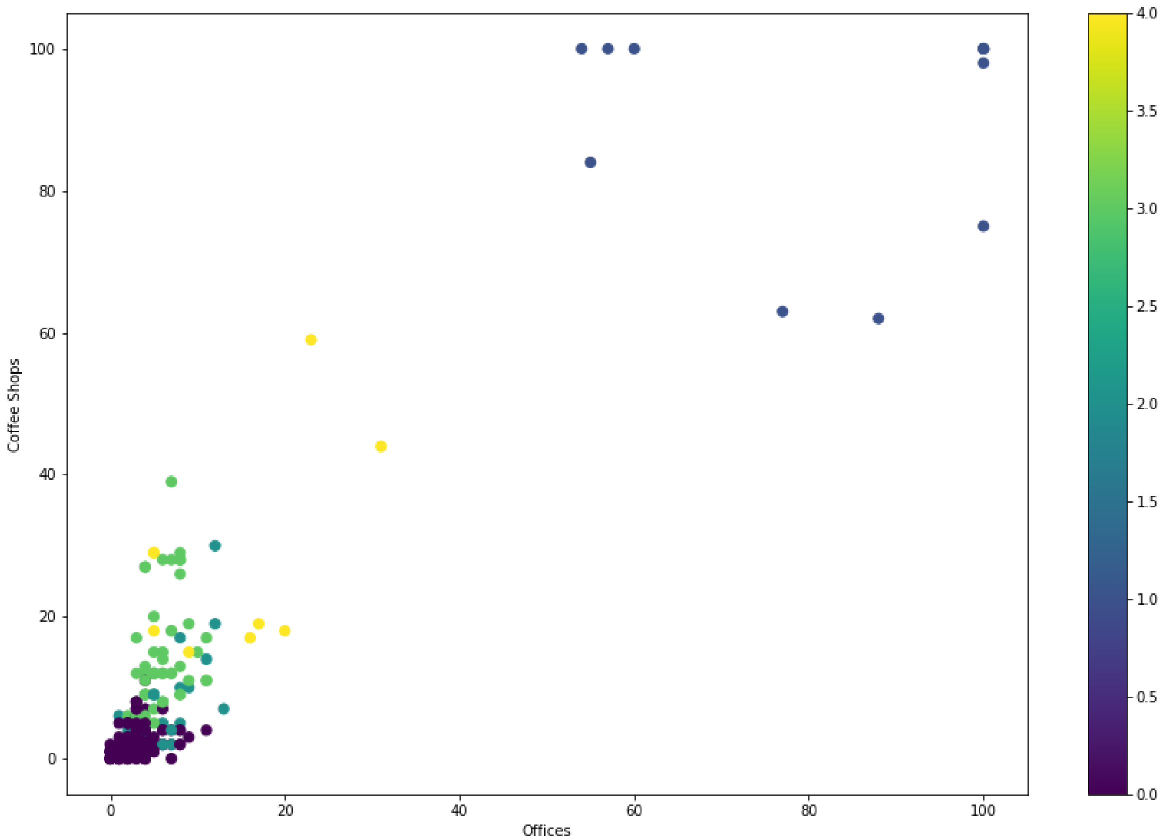
```
array([2, 0, 3, 0, 2, 0, 0, 0, 3, 3], dtype=int32)
```

```
# Add Cluster label column to main DF
df_final.insert(0, 'Cluster Labels', kmeans.labels_)
df_final.head()
```

	Cluster Labels	Neighborhood	Community area	Latitude	Longitude	Universities	Offices	Bakeries	Coffee Shops
0	2	Albany Park	Albany Park	41.971937	-87.716174	64.0	1.0	15.0	6.0
1	0	Altgeld Gardens	Riverdale	41.654864	-87.600439	1.0	1.0	0.0	0.0
2	3	Andersonville	Edgewater	41.977139	-87.669273	15.0	6.0	25.0	12.0
3	0	Archer Heights	Archer Heights	41.811422	-87.726165	2.0	1.0	7.0	1.0
4	2	Armour Square	Armour Square	41.840033	-87.633107	54.0	6.0	7.0	5.0

If we plot the generated clusters against number of coffee shops and offices, we get the following:

```
fig.show()
```



Not surprisingly, our segment of interest is the one comprised of neighborhoods where coffee shops and offices are numerous (above 60).

And here is our final list of neighborhood recommendations:

[83]:

	Cluster Labels	Neighborhood	Community area	Latitude	Longitude	Universities	Offices	Bakeries	Coffee Shops
71	1	The Gap	Douglas	41.892357	-87.623588	100.0	60.0	80.0	100.0
96	1	The Island	Austin	41.875562	-87.624421	100.0	88.0	43.0	62.0
108	1	Lake View	Lake View	41.885382	-87.627908	100.0	100.0	46.0	100.0
121	1	The Loop	The Loop	41.881609	-87.629457	100.0	100.0	55.0	100.0
124	1	Magnificent Mile	Near North Side	41.894523	-87.624228	100.0	54.0	65.0	100.0
162	1	Park West	Lincoln Park	41.882557	-87.622500	100.0	100.0	60.0	98.0
173	1	Printer's Row	The Loop	41.873787	-87.628900	100.0	77.0	38.0	63.0
180	1	River North	Near North Side	41.888341	-87.617903	86.0	55.0	62.0	84.0
210	1	Streeterville	Near North Side	41.893365	-87.621997	88.0	57.0	68.0	100.0
214	1	Union Ridge	Norwood Park	41.878295	-87.638949	84.0	100.0	52.0	75.0
233	1	West Loop	Near West Side	41.881609	-87.629457	100.0	100.0	55.0	100.0

Discussion:

This methodology is limited, but still provides valuable results.

Being able to retrieve at most 100 venues per neighborhood is limiting in that we reach that number of 100 quite often.

The method used to compute the score is also very simple. It presents the advantage of being very easy to understand and has a high level of explainability. We could have used more variables to compute the score, based on other venue categories or on demographic information, for example (mean income, dwelling,…)

Conclusion:

We've demonstrated here how to build a simple Neighborhood recommendation engine for someone who wishes to set up a new high-end coffee shop in Chicago.

We end up with a neighborhood of choice (The Island), as well as an extended list, should the entrepreneur wish to look at further options.