Moscow neighborhoods vs. NYC neighborhoods

Guretskiy Evgeny

May 29, 2020

1. Introduction

1.1 Background

Moscow and NYC are among the biggest capitals in the world. Moscow has area of 2511 sq. km., while NYC has 784. They are both densely populated cosmopolitan metropolises (In 2012 Moscow population was almost 12 mln, NYC 8.4 mln), but they also have their differences. Assume that we have been asked to find the better city for launching a new branch of fitness clubs. We have conducted an experiment where we compare the two cities in terms of their public venues, including coffee shops, cafes, restaurants, and etc. to get a good measure of each city's residents towards their choices of venue.

1.2 Business

As mentioned above, sport has a certain degree of importance for nowadays people. Moreover, people globally more and more tend to follow healthy lifestyle and become more aware of their state. Therefore, we will model a situation in which we have decided to open a new fitness center full of areas for different kinds of sports, spa-facilities etc.

1.3 Target audience

A target audience are those of 12 mln people in Moscow or 8.4 mln in NYC who keen on sport.

2. Data

2.2 Moscow Data

Link: 'https://en.wikipedia.org/wiki/Administrative_divisions_of_Moscow'

As there were no .csv file with ready data, it was necessary to scrape the data. For this purpose, we gathered data of Moscow administrative division from Wikipedia webpage. We got a list of neighborhoods names, but the problem was that the name consisted of Russian name and name in English in brackets, so we had to separate languages using “re” library.

11]: url

11]:

	Name
8	Arbat (Арбат)
9	Basmanny (Басманный)
10	Khamovniki (Хамовники)
11	Krasnoselsky (Красносельский)
12	Meshchansky (Мещанский)
...	...
182	Novofyodorovskoye Settlement (Поселение Новофё...
183	Pervomayskoye Settlement (Поселение Первомайское)
184	Rogovskoye Settlement (Поселение Роговское)
185	Troitsky Settlement (Поселение Троицк)
186	Shchapovskoye Settlement (Поселение Щаповское)

147 rows × 1 columns

Data after scrapping and cleaning

[17]:

	Name_R	Name_E
0	Арбат	Arbat
1	Басманный	Basmanny
2	Хамовники	Khamovniki
3	Красносельский	Krasnoselsky
4	Мещанский	Meshchansky
...	...	...
142	Поселение Новофёдоровское	Novofyodorovskoye Settlement
143	Поселение Первомайское	Pervomayskoye Settlement
144	Поселение Роговское	Rogovskoye Settlement
145	Поселение Троицк	Troitsky Settlement
146	Поселение Щаповское	Shchapovskoye Settlement

147 rows × 2 columns

Data after separating languages

The next problem related to coordinates absence. Using “geocode” and “nominatim” libraries we easily fulfilled our table.

```
#get coords by address in dfc
locator = Nominatim(user_agent="Moscow")
# 1 - conveneint function to delay between geocoding calls
geocode = RateLimiter(locator.geocode, min_delay_seconds=1)
# 2- - create location column
df3['location'] = dfc['Address'].apply(geocode)
# 3 - create longitude, laatitude and altitude from location column (returns tuple)
df3['point'] = df3['location'].apply(lambda loc: tuple(loc.point) if loc else None)
# 4 - split point column into latitude, longitude and altitude columns
df3[['latitude', 'longitude', 'altitude']] = pd.DataFrame(df3['point'].tolist(), index=df3.index)
```

```
df4=df3.dropna()

df4
```

	Name_R	location	point	latitude	longitude	altitude
0	Арбат	(район Арбат, Центральный административный окр...	(55.751199, 37.5898715, 0.0)	55.751199	37.589872	0.0
1	Басманный	(Басманный район, Центральный административный...	(55.7672809, 37.6697732, 0.0)	55.767281	37.669773	0.0
2	Хамовники	(район Хамовники, Центральный административный...	(55.729229, 37.5729484, 0.0)	55.729229	37.572948	0.0
3	Красносельский	(Красносельский район, Центральный администрат...	(55.7774472, 37.6541598, 0.0)	55.777447	37.654160	0.0
4	Мещанский	(Мещанский район, Центральный административный...	(55.779172, 37.6279886, 0.0)	55.779172	37.627989	0.0
...	...	...	...	...	...	...
142	Поселение Новофёдоровское	(поселение Новофёдоровское, Троицкий администр...	(55.43792585, 36.99971817597381, 0.0)	55.437926	36.999718	0.0
143	Поселение Первомайское	(поселение Первомайское, Троицкий администрати...	(55.50190525, 37.21279504750347, 0.0)	55.501905	37.212795	0.0
144	Поселение Роговское	(поселение Роговское, Троицкий административны...	(55.2290933, 37.05519465465914, 0.0)	55.229093	37.055195	0.0
145	Поселение Троицк	(городской округ Троицк, Троицкий администрати...	(55.479412249999996, 37.29803052194669, 0.0)	55.479412	37.298031	0.0
146	Поселение Щаповское	(поселение Щаповское, Троицкий административны...	(55.384577699999994, 37.38527174994181, 0.0)	55.384578	37.385272	0.0

147 rows × 6 columns

ENG 12:38 29.05.2020

Code for cords gathering and data after

2.3 NYC data

Link: https://cocl.us/new_york_dataset

NYC data was structured in .csv file that was used in previous labs of course. So there were no any problems downloading it.

2.4 Foursquare API

We actively used this data provider during tutorials in previous labs. Therefore, this service is our powerful tool for getting the list of venues for each neighborhood in Moscow and NYC lists.

3. Methodology

3.2 Moscow data preparation

As it was mentioned, first we parse the data from Wikipedia webpage, then clean it and separate names of neighborhoods. The next step was to get coordinates using “geocode” library (picture above).

3.3 NYC data preparation

Downloading .json file.

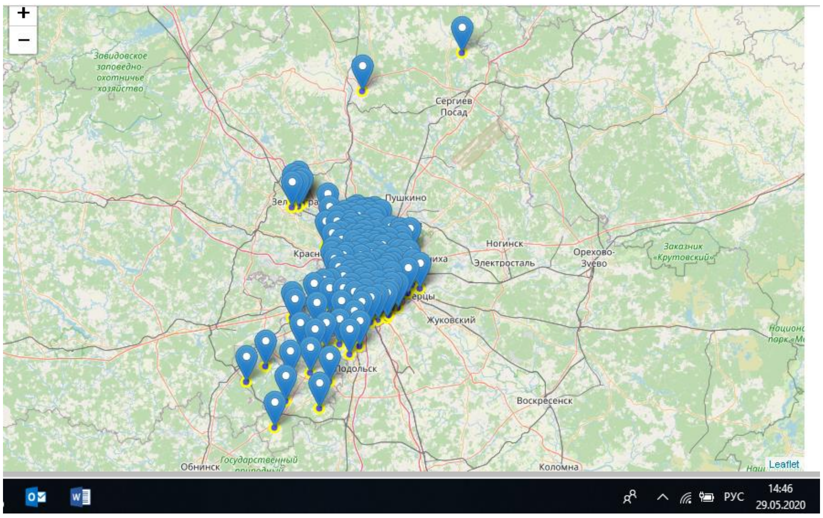
49]:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

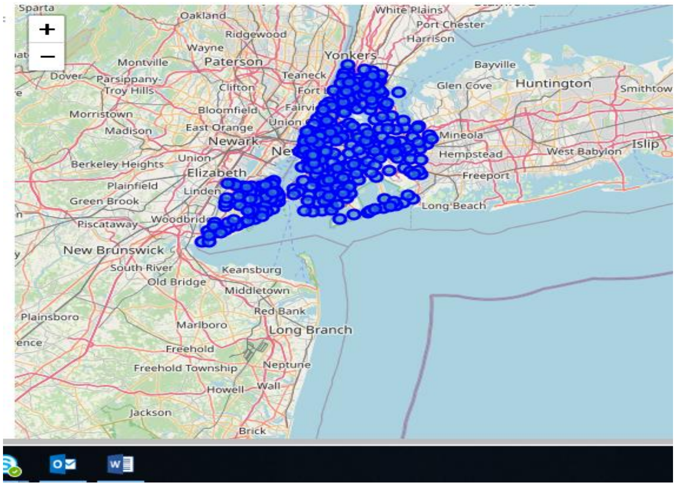
NYC data after loading

3.4 Data visualization

Both prepared Moscow data and NYC data (latitude and longitude) were used to put point on the map using “folium” library, so we could better percept the data.



Moscow map



NYC map

3.5 API Calls to Foursquare

The Foursquare API is used to explore the neighborhoods and segment them. To access the API, ‘CLIENT_ID’, ‘CLIENT_SECRET’ and ‘VERSION’ are set. To overcome the redundancy of the process followed above, a function ‘getNearbyVenues’ is created. This functions loop through all the neighborhoods of Moscow and New York City and creates an API request URL with radius = 500, LIMIT = 100. By limit, it is defined that the maximum of 100 nearby venues should be returned. Further, the GET request is made to Foursquare API and only relevant information for each nearby venue is extracted from it. The data is then appended to a python ‘list’. Lastly, the python ‘list’ is unfolded or flattened to append it to the data frame being returned by the function.

```python
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 = ['Name',
                  'Name_Latitude',
                  'Name_Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

The returned data frames are as follows:

	Name	Name_Latitude	Name_Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Арбат	55.751199	37.589872	Шоколадница	55.751886	37.588327	Coffee Shop
1	Арбат	55.751199	37.589872	Corner Café & Kitchen	55.751496	37.586757	Japanese Restaurant
2	Арбат	55.751199	37.589872	Театр им. Вахтангова	55.749569	37.591638	Theater
3	Арбат	55.751199	37.589872	Спортивный комплекс «Арбат»	55.750928	37.587130	Pool
4	Арбат	55.751199	37.589872	Obedbufet (Обедбуфет)	55.752268	37.592275	Buffet
...	...	...	...	...	...	...	...
2748	Поселение Троицк	55.479412	37.298031	Т-клуб	55.475971	37.296819	Gym
2749	Поселение Троицк	55.479412	37.298031	Дворец спорта «Квант»	55.477341	37.298947	Stadium
2750	Поселение Троицк	55.479412	37.298031	Физическая Кунсткамера	55.475638	37.295264	Science Museum
2751	Поселение Троицк	55.479412	37.298031	Городской Рынок	55.476763	37.297657	Farmers Market
2752	Поселение Троицк	55.479412	37.298031	Центральный стадион	55.475311	37.297908	Soccer Field

2753 rows × 7 columns

Moscow venues

	Name	Name_Latitude	Name_Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Lollipops Gelato	40.894123	-73.845892	Dessert Shop
1	Wakefield	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
2	Wakefield	40.894705	-73.847201	Walgreens	40.896528	-73.844700	Pharmacy
3	Wakefield	40.894705	-73.847201	Rite Aid	40.896649	-73.844846	Pharmacy
4	Wakefield	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop

NYC venues

3.6 Data converting

As we have all necessary data, now we can use so called ‘one-hot encoding’ function of python ‘pandas’ library. One hot encoding converts the categorical variables (which are ‘Venue Category’) into a form that could be provided to ML algorithms to do a better job in prediction. We have already come across it in previous labs.

```
# one hot encoding
nyc_onehot = pd.get_dummies(nyc_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
nyc_onehot['Neighbourhood'] = nyc_venues['Neighbourhood']

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

nyc_onehot.head()
```

	Neighbourhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport Terminal	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery
0	Wakefield	0	0	0	0	0	0	0	0	0	0	0
1	Wakefield	0	0	0	0	0	0	0	0	0	0	0
2	Wakefield	0	0	0	0	0	0	0	0	0	0	0
3	Wakefield	0	0	0	0	0	0	0	0	0	0	0
4	Wakefield	0	0	0	0	0	0	0	0	0	0	0

Now, data allows us to count top 5 venues both for Moscow and NYC, and conduct high-level description.

	count	mean	std	min	25%	50%	75%	max
Supermarket	137.0	0.861314	1.237778	0.0	0.0	1.0	1.0	9.0
Coffee Shop	137.0	0.795620	1.591054	0.0	0.0	0.0	1.0	9.0
Clothing Store	137.0	0.350365	0.895998	0.0	0.0	0.0	0.0	7.0
Cosmetics Shop	137.0	0.569343	1.069604	0.0	0.0	0.0	1.0	6.0
Hotel	137.0	0.262774	0.788428	0.0	0.0	0.0	0.0	6.0
Theater	137.0	0.145985	0.612854	0.0	0.0	0.0	0.0	6.0
Gym / Fitness Center	137.0	0.474453	0.832080	0.0	0.0	0.0	1.0	5.0
Arcade	137.0	0.153285	0.554268	0.0	0.0	0.0	0.0	5.0
Café	137.0	0.532847	0.970530	0.0	0.0	0.0	1.0	5.0
Bar	137.0	0.175182	0.554365	0.0	0.0	0.0	0.0	4.0

Moscow data description

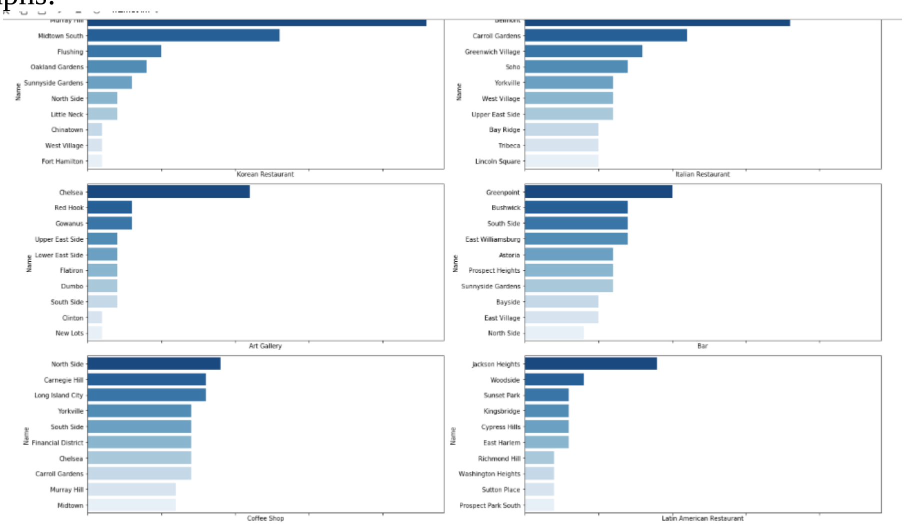
	count	mean	std	min	25%	50%	75%	max
Korean Restaurant	301.0	0.232558	1.589257	0.0	0.0	0.0	0.0	23.0
Italian Restaurant	301.0	0.916944	1.810086	0.0	0.0	0.0	1.0	18.0
Art Gallery	301.0	0.169435	0.762362	0.0	0.0	0.0	0.0	11.0
Bar	301.0	0.710963	1.402205	0.0	0.0	0.0	1.0	10.0
Coffee Shop	301.0	1.003322	1.774821	0.0	0.0	0.0	1.0	9.0
Latin American Restaurant	301.0	0.275748	0.783396	0.0	0.0	0.0	0.0	9.0
Beach	301.0	0.146179	0.786486	0.0	0.0	0.0	0.0	9.0
Deli / Bodega	301.0	0.930233	1.210420	0.0	0.0	1.0	1.0	8.0
Park	301.0	0.598007	1.163842	0.0	0.0	0.0	1.0	8.0
Gym / Fitness Center	301.0	0.385382	0.947445	0.0	0.0	0.0	0.0	8.0

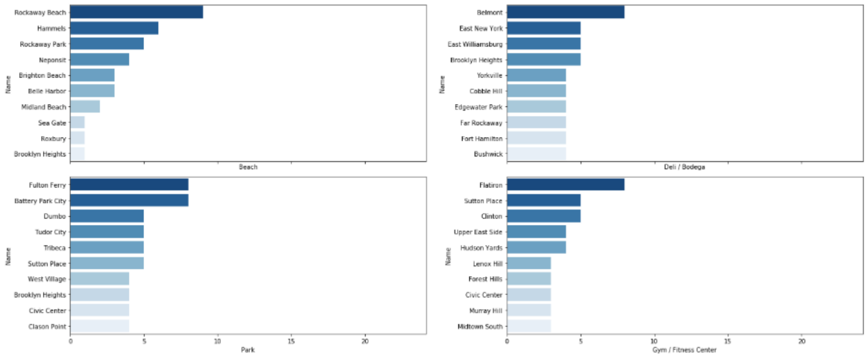
NYC data description

4. Data Visualization

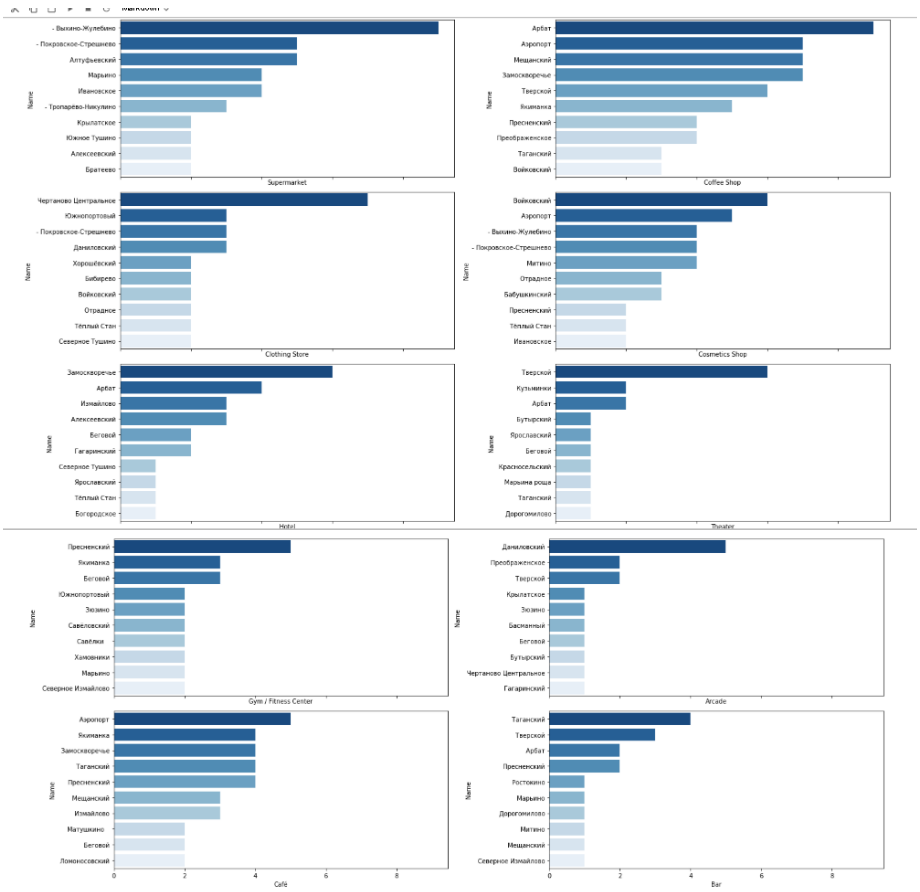
These top 10 categories are further plotted individually on bar graph using python 'seaborn' library. A code block is created which loops and plots the graph of top 10 neighborhoods for a category.

NYC graphs:





Moscow graphs:



Next, the rows of the neighborhoods are grouped together and the frequency of occurrence of each category is calculated by taking the mean. As the limit is set to be 100, there will be many venues being returned by the Foursquare API. The created data frame is then fed with the top 5 most common venues categories in the respective neighborhoods.

```python
num_top_venues = 5

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

# create columns according to number of top venues
columns = ['Neighbourhood']
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
nyc_venues_sorted = pd.DataFrame(columns=columns)
nyc_venues_sorted['Neighbourhood'] = nyc_grouped['Neighbourhood']

for ind in np.arange(nyc_grouped.shape[0]):
    nyc_venues_sorted.iloc[ind, 1:] = return_most_common_venues(nyc_grouped.iloc[ind, :], num_top_venues)

_venues_sorted.head()
```

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Allerton	Pizza Place	Bakery	Chinese Restaurant	Deli / Bodega	Department Store
1	Annadale	Pizza Place	American Restaurant	Dance Studio	Sports Bar	Train Station
2	Arden Heights	Pizza Place	Deli / Bodega	Pharmacy	Bus Stop	Business Service
3	Arlington	Grocery Store	Deli / Bodega	American Restaurant	Intersection	Bus Stop
4	Arrochar	Italian Restaurant	Bus Stop	Deli / Bodega	Pizza Place	Cosmetics Shop

NYC

5. ML

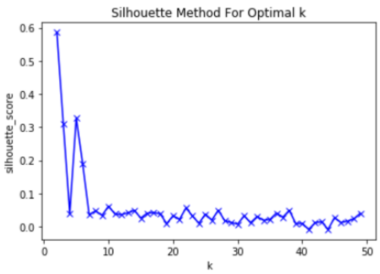
We used The Silhouette Method to determine number of cluster, as the elbow method was inefficient.

```python
[72]: from sklearn.metrics import silhouette_score

sil = []
K_sil = range(2,50)
# minimum 2 clusters required, to define dissimilarity
for k in K_sil:
    print(k, end=' ')
    kmeans = KMeans(n_clusters = k).fit(nyc_grouped_clustering)
    labels = kmeans.labels_
    sil.append(silhouette_score(nyc_grouped_clustering, labels, metric = 'euclidean'))
```

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
```

```python
[73]: plt.plot(K_sil, sil, 'bx-')
plt.xlabel('k')
plt.ylabel('silhouette_score')
plt.title('Silhouette Method For Optimal k')
plt.show()
```



NYC silhouette score

[102]:

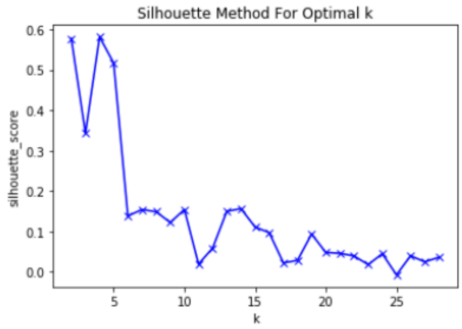
```
from sklearn.metrics import silhouette_score

sil = []
K_sil = range(2,29)
# minimum 2 clusters required, to define dissimilarity
for k in K_sil:
    print(k, end=' ')
    kmeans = KMeans(n_clusters = k).fit(moscow_grouped_clustering)
    labels = kmeans.labels_
    sil.append(silhouette_score(moscow_grouped_clustering, labels,metric = 'euclidean'))
```

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
```

[103]:

```
plt.plot(K_sil, sil, 'bx-')
plt.xlabel('k')
plt.ylabel('silhouette_score')
plt.title('Silhouette Method For Optimal k')
plt.show()
```



Moscow silhouette score

For NYC There is a peak at k = 2, k =5 and k = 6. Two clusters will give a very broad classification of the venues. Therefore, the number of New York clusters (i.e. ‘k’) is chosen to be 5.

For Moscow there is a peak at k=4.

As we are going to find diversities of clusters then it will be more reasonable to use k=4 for both NYC and Moscow.

.]:

```
#Run k-means to cluster the neighborhood into 4 clusters.
# set number of clusters
kclusters = 4

moscow_grouped_clustering = moscow_grouped.drop('Name', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_
# to change use .astype()
```

Further, the cluster labels curated are added to the data frame to get the desired results of segmenting the neighborhood based upon the most common venues in its vicinity. Moscow and New York City’s neighborhoods are visualized utilizing the python ‘folium’ library as we did it before.

6. Results

6.1 NYC

Cluster 1: Top 1 – Carrebian restaurant, Top 2 – cosmetic shops

```
nyc_cluster_1=nyc_merged.loc[nyc_merged['Cluster Labels1'] == 0, nyc_merged.columns[[0]+[1]+ list(range(5, nyc_merged.shape[1]))]]
nyc_cluster_1
```

	Borough	Neighborhood	Cluster Labels1	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
9	Bronx	Williamsbridge	0.0	Caribbean Restaurant	Bar	Nightclub	Convenience Store	Soup Place
41	Bronx	Olinville	0.0	Caribbean Restaurant	Fried Chicken Joint	Laundromat	Basketball Court	Supermarket
165	Queens	St. Albans	0.0	Caribbean Restaurant	Dance Studio	Southern / Soul Food Restaurant	Shopping Mall	Grocery Store
168	Queens	Cambria Heights	0.0	Caribbean Restaurant	Cosmetics Shop	Restaurant	Liquor Store	Nightclub
188	Queens	Laurelton	0.0	Caribbean Restaurant	Cosmetics Shop	Sculpture Garden	Train Station	Yoga Studio
257	Staten Island	Howland Hook	0.0	0	0	0	0	0
300	Brooklyn	Erasmus	0.0	Caribbean Restaurant	Grocery Store	Yoga Studio	Convenience Store	Bank

```
for col in nyc_required_column:
    print(nyc_cluster_1[col].value_counts(ascending=False))
    print('-------------------------------------')
```

```
Queens           3
Bronx            2
Brooklyn         1
Staten Island    1
Name: Borough, dtype: int64
-------------------------------------
Caribbean Restaurant    6
0                       1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Cosmetics Shop         2
Grocery Store          1
Fried Chicken Joint    1
0                      1
Bar                    1
Dance Studio           1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------
```

Cluster 2: Top 1- Bus stops, Top – 2 parks

	Borough	Neighborhood	Cluster Labels1	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
27	Bronx	Clason Point	1.0	Park	American Restaurant	Grocery Store	Boat or Ferry	Pool
77	Brooklyn	Manhattan Beach	1.0	Bus Stop	Café	Ice Cream Shop	Harbor / Marina	Playground
172	Queens	Breezy Point	1.0	Trail	Beach	Monument / Landmark	Yoga Studio	Filipino Restaurant
179	Queens	Neponsit	1.0	Beach	Yoga Studio	Ethiopian Restaurant	Event Space	Exhibit
198	Staten Island	New Brighton	1.0	Bus Stop	Park	Deli / Bodega	Bowling Alley	Discount Store
202	Staten Island	Grymes Hill	1.0	Dog Run	Bus Stop	Yoga Studio	Event Space	Exhibit
212	Staten Island	Oakwood	1.0	Bar	Bus Stop	Lawyer	Financial or Legal Service	Event Space
224	Staten Island	Park Hill	1.0	Bus Stop	Athletics & Sports	Hotel	Coffee Shop	Gym / Fitness Center
228	Staten Island	Arrochar	1.0	Bus Stop	Deli / Bodega	Italian Restaurant	Food Truck	Supermarket
229	Staten Island	Grasmere	1.0	Bus Stop	Bagel Shop	Bank	Grocery Store	Deli / Bodega
238	Staten Island	Butler Manor	1.0	Baseball Field	Pool	Bus Stop	Convenience Store	Yoga Studio
241	Staten Island	Arden Heights	1.0	Pizza Place	Pharmacy	Coffee Shop	Bus Stop	Lawyer
245	Staten Island	Bloomfield	1.0	Recreation Center	Park	Theme Park	Bus Stop	Discount Store
256	Staten Island	Randall Manor	1.0	Bus Stop	Pizza Place	Bagel Shop	Carpet Store	Playground
285	Staten Island	Willowbrook	1.0	Bus Stop	Chinese Restaurant	Pizza Place	Yoga Studio	Filipino Restaurant
286	Staten Island	Sandy Ground	1.0	Bus Stop	Food Truck	Art Gallery	Intersection	Racetrack
302	Queens	Hammels	1.0	Beach	Neighborhood	Diner	Building	Gym / Fitness Center
303	Queens	Bayswater	1.0	Playground	Park	Yoga Studio	Event Service	Event Space
305	Staten Island	Fox Hills	1.0	Bus Stop	Print Shop	Sandwich Place	Yoga Studio	Field

```
for col in nyc_required_column:
    print(nyc_cluster_2[col].value_counts(ascending=False))
    print('-------------------------------------')
```

```
Staten Island    13
Queens            4
Brooklyn          1
Bronx             1
Name: Borough, dtype: int64
-------------------------------------
Bus Stop             9
Beach                2
Playground           1
Park                 1
Baseball Field       1
Trail                1
Bar                  1
Dog Run              1
Pizza Place          1
Recreation Center    1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Park                   3
Bus Stop               2
Bagel Shop             1
Beach                  1
```

Cluster 3: Top 1 – Pizza place, Top 2 – coffee shop

cluster 2:Bus stops (9) and parks (3)

112]:
```
nyc_cluster_3=nyc_merged.loc[nyc_merged['Cluster Labels1'] == 2, nyc_merged.columns[[0]+[1] + list(range(5, nyc_merged.shape[1]))]]
nyc_cluster_3
```

112]:

	Borough	Neighborhood	Cluster Labels1	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Bronx	Wakefield	2.0	Pharmacy	Ice Cream Shop	Dessert Shop	Sandwich Place	Gas Station
1	Bronx	Co-op City	2.0	Bus Station	Baseball Field	Park	Gift Shop	Pharmacy
2	Bronx	Eastchester	2.0	Bus Station	Caribbean Restaurant	Deli / Bodega	Bus Stop	Diner
3	Bronx	Fieldston	2.0	Plaza	River	Bus Station	Filipino Restaurant	Event Service
4	Bronx	Riverdale	2.0	Park	Bus Station	Home Service	Food Truck	Bank
...	...	...	...	...	...	...	...	...
297	Bronx	Bronxdale	2.0	Performing Arts Venue	Pizza Place	Paper / Office Supplies Store	Italian Restaurant	Gym
298	Bronx	Allerton	2.0	Pizza Place	Supermarket	Deli / Bodega	Bus Station	Chinese Restaurant
299	Bronx	Kingsbridge Heights	2.0	Pizza Place	Coffee Shop	Chinese Restaurant	Grocery Store	Mexican Restaurant
301	Manhattan	Hudson Yards	2.0	Gym / Fitness Center	Hotel	Italian Restaurant	American Restaurant	Dog Run
304	Queens	Queensbridge	2.0	Hotel	Sandwich Place	Gym / Fitness Center	Scenic Lookout	Performing Arts Venue

278 rows × 8 columns

```
Queens           73
Brooklyn         68
Bronx            49
Staten Island    48
Manhattan        40
Name: Borough, dtype: int64
-------------------------------------
Pizza Place                     27
Deli / Bodega                   23
Italian Restaurant              19
Coffee Shop                     17
Chinese Restaurant              15
                                ..
Plaza                            1
Wine Bar                         1
Liquor Store                     1
Ice Cream Shop                   1
Vegetarian / Vegan Restaurant    1
Name: 1st Most Common Venue, Length: 78, dtype: int64
-------------------------------------
Pizza Place           28
Coffee Shop           19
Deli / Bodega         17
Italian Restaurant    14
Chinese Restaurant    11
```

Cluster 4: Top 1 – parks, Top 2- yoga

Cluster 3: definetely pizza and coffee

]:
```
nyc_cluster_4=nyc_merged.loc[nyc_merged['Cluster Labels1'] ==3, nyc_merged.columns[[0]+[1] + list(range(5, nyc_merged.shape[1]))]]
nyc_cluster_4
```

]:

	Borough	Neighborhood	Cluster Labels1	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
192	Queens	Somerville	3.0	Park	Yoga Studio	Ethiopian Restaurant	Event Space	Exhibit
203	Staten Island	Todt Hill	3.0	Park	Yoga Studio	Ethiopian Restaurant	Event Space	Exhibit

]:
```
for col in nyc_required_column:
    print(nyc_cluster_4[col].value_counts(ascending=False))
    print('-------------------------------------')
```

```
Queens           1
Staten Island    1
Name: Borough, dtype: int64
-------------------------------------
Park    2
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Yoga Studio    2
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------
```

6.2 Moscow

Cluster 1: Top 1-lakes, Top 2 - zoo

	location	altitude	Cluster_Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
111	(Внуково, Дмитровский городской округ, Московс...	0.0	0	Lake	Zoo Exhibit	Fabric Shop	Farmers Market	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop	Food
124	(район Силино, Зеленоградский административный...	0.0	0	Lake	Zoo Exhibit	Fabric Shop	Farmers Market	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop	Food

```
for col in moscow_required_column:
    print(moscow_cluster_1[col].value_counts(ascending=False))
    print('-----------------------------------------')
```

```
(Внуково, Дмитровский городской округ, Московская область, Центральный федеральный округ, Россия, (56.3651669, 37.58425))           1
(район Силино, Зеленоградский административный округ, Москва, Центральный федеральный округ, Россия, (56.0017567, 37.1688978))    1
Name: location, dtype: int64
-------------------------------------
Lake    2
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Zoo Exhibit    2
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------
```

Cluster 2: Top 1- coffee shops, top 2 - parks

```
moscow_cluster_2=moscow_merged.loc[moscow_merged['Cluster_Labels'] == 1, moscow_merged.columns[[1] + list(range(5, moscow_merged.shape[1]))]]
moscow_cluster_2
```

	location	altitude	Cluster_Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	(район Арбат, Центральный административный окр...	0.0	1	Coffee Shop	Hotel	Yoga Studio	Hostel	Museum	Caucasian Restaurant	Burger Joint	Italian Restaurant	Concert Hall	Bar
1	(Басманный район, Центральный административный...	0.0	1	History Museum	Gun Range	Vegetarian / Vegan Restaurant	Flower Shop	Dance Studio	Supermarket	Park	Gift Shop	Massage Studio	Pet Café
2	(район Хамовники, Центральный административный...	0.0	1	Gym / Fitness Center	Park	Bath House	Coffee Shop	Bakery	Sushi Restaurant	BBQ Joint	Seafood Restaurant	Supermarket	Farmers Market
3	(Красносельский район, Центральный администрат...	0.0	1	Gym / Fitness Center	Hotel Bar	Coffee Shop	Varenyky restaurant	Gaming Cafe	Bus Stop	Flower Shop	Mobile Phone Shop	Café	Fast Food Restaurant
4	(Мещанский район, Центральный административный...	0.0	1	Coffee Shop	Café	Middle Eastern Restaurant	Bakery	Restaurant	Cosmetics Shop	Spa	Caucasian Restaurant	Museum	Health Food Store
...	...	...	...	...	...	...	...	...	...	...	...	...	...
129	(поселение Кокошкино, Новомосковский администр...	0.0	1	Farmers Market	Zoo Exhibit	Food & Drink Shop	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop	Food	Food Court
131	(поселение Московский, Новомосковский админист...	0.0	1	Café	Resort	Wine Bar	Middle Eastern Restaurant	Bus Stop	Big Box Store	Zoo Exhibit	Food	Film Studio	Fish Market

Mode: Command Ln 1, Col 1 Capstone.ipynb

Code Python

```
..
(Зеленоград, Зеленоградский административный округ, Москва, Центральный федеральный округ, Россия, (55.9964233, 37.198534))
1
(Измайлово, Окружной проезд, Соколиная Гора, район Соколиная Гора, Восточный административный округ, Москва, Центральный федеральный округ, 105613, Р
оссия, (55.7897928, 37.7431771))          1
(район Южное Бутово, Юго-Западный административный округ, Москва, Центральный федеральный округ, Россия, (55.5412721, 37.5325392))
1
(Матушкино, район Матушкино, Зеленоградский административный округ, Москва, Центральный федеральный округ, 103489, Россия, (56.0095351, 37.1973782))
1
(Северное Бутово, район Северное Бутово, Юго-Западный административный округ, Москва, Центральный федеральный округ, 117628, Россия, (55.5714094, 37.
5776237))          1
Name: location, Length: 110, dtype: int64
-------------------------------------
Coffee Shop              11
Park                      8
Gym / Fitness Center      6
Café                      6
Cosmetics Shop            5
Pharmacy                  5
Fast Food Restaurant      4
Supermarket               4
Mobile Phone Shop         4
Bus Stop                  4
Soccer Field              3
Auto Workshop             3
Pizza Place               3
Restaurant                2
Sushi Restaurant          2
Music Venue               2
Electronics Store         2
Clothing Store            2
Flower Shop               2
Sporting Goods Shop       2
```

Cluster 3: Top 1 – supermarkets, **top 2 – gyms. That is what we have been looking for.**

```
4]: moscow_cluster_3=moscow_merged.loc[moscow_merged['Cluster_Labels'] == 2, moscow_merged.columns[[1] + list(range(5, moscow_merged.shape[1]))]]
moscow_cluster_3
```

4]:

	location	altitude	Cluster_Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
13	(Дмитровский, Москва, Северный административны...	0.0	2	Supermarket	Gym	Pizza Place	Shipping Store	Eastern European Restaurant	Bus Stop	Modern European Restaurant	Bath House	Park	Restaurant
16	(Ховрино, улица Дыбенко, район Ховрино, Северн...	0.0	2	Supermarket	Auto Workshop	Park	Campground	French Restaurant	Fountain	Forest	Food Truck	Food Court	Food & Drink Shop
23	(Восточное Дегунино, район Восточное Дегунино,...	0.0	2	Supermarket	Bus Stop	Park	Pedestrian Plaza	Toy / Game Store	Business Service	Pet Service	Soccer Field	Health Food Store	Film Studio
27	(Алтуфьевский, Москва, Северо-Восточный админи...	0.0	2	Supermarket	Dry Cleaner	Gym Pool	Grocery Store	Café	Health Food Store	Pharmacy	Salon / Barbershop	Bus Stop	Food & Drink Shop
42	(район Южное Медведково, Северо-Восточный адми...	0.0	2	Supermarket	Flower Shop	Pharmacy	Gym / Fitness Center	Cafeteria	Bakery	BBQ Joint	Bus Stop	Gym	Sushi Restaurant
45	(район Ивановское, Восточный административный ...	0.0	2	Supermarket	Cosmetics Shop	Shopping Mall	Bus Stop	Event Space	Pizza Place	Fast Food Restaurant	Electronics Store	Pet Store	Auto Workshop

```
1
(Алтуфьевский, Москва, Северо-Восточный административный округ, Москва, Центральн
1
(поселение Сосенское, Новомосковский административный округ, Москва, Центральный
1
(Орехово-Борисово Южное, Москва, Южный административный округ, Москва, Центральнь
1
Name: location, dtype: int64
-------------------------------------
Supermarket     18
Pizza Place      1
Bus Stop         1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Gym                     4
Cosmetics Shop          2
Bus Stop                2
Convenience Store       2
Auto Workshop           1
Korean Restaurant       1
Sushi Restaurant        1
Dry Cleaner             1
Supermarket             1
Flower Shop             1
Grocery Store           1
Bakery                  1
Park                    1
Gym / Fitness Center    1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------
```

Cluster 4: Top 1 – conveniences store, top 2 – bus stop.

Code Python

```
]: moscow_cluster_4=moscow_merged.loc[moscow_merged['Cluster_Labels'] == 3, moscow_merged.columns[[1] + list(range(5, moscow_merged.shape[1]))]]
moscow_cluster_4
```

]:

	location	altitude	Cluster_Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
39	(Северный, Талдомский городской округ, Московс...	0.0	3	Convenience Store	Motel	Zoo Exhibit	Food Court	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop	Food
83	(Орехово-Борисово Северное, Москва, Южный адми...	0.0	3	Convenience Store	Bus Stop	Dairy Store	Grocery Store	Pharmacy	Playground	Dance Studio	Discount Store	Fast Food Restaurant	Fried Chicken Joint
105	(Ново-Переделкино, район Ново-Переделкино, Зап...	0.0	3	Convenience Store	Bus Stop	Tennis Court	Pool	Hockey Rink	Zoo Exhibit	Food	Fast Food Restaurant	Film Studio	Fish Market
137	(поселение Вороновское, Троицкий административ...	0.0	3	Convenience Store	Zoo Exhibit	Food & Drink Shop	Farmers Market	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop	Food
143	(поселение Первомайское, Троицкий администрати...	0.0	3	Recreation Center	Beach	Convenience Store	Zoo Exhibit	Food & Drink Shop	Fast Food Restaurant	Film Studio	Fish Market	Flea Market	Flower Shop

```
9]: for col in moscow_required_column:
        print(moscow_cluster_4[col].value_counts(ascending=False))
        print('-------------------------------------')
```

```
(Ново-Переделкино, район Ново-Переделкино, Западный административный округ, Москва, Центральный федеральный округ, 119634, Россия, (55.6461043, 37.35
2684))    1
(Орехово-Борисово Северное, Москва, Южный административный округ, Москва, Центральный федеральный округ, 115569, Россия, (55.619223, 37.7011025))
1
(Северный, Талдомский городской округ, Московская область, Центральный федеральный округ, 141912, Россия, (56.725784, 37.6219781))
1
(поселение Вороновское, Троицкий административный округ, Москва, Центральный федеральный округ, 108830, Россия, (55.321132399999996, 37.1219867578463
4))       1
(поселение Первомайское, Троицкий административный округ, Москва, Центральный федеральный округ, Россия, (55.50190525, 37.21279504750347))
1
Name: location, dtype: int64
-------------------------------------
Convenience Store    4
Recreation Center    1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------
Bus Stop       2
Beach          1
Zoo Exhibit    1
Motel          1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------
```

Conclusion

We have done a lot: loading data, formatting and wrangling it, getting coordinates, putting points on the map, preparation of data for ML, building graphs, conducting k-mean algorithm in order to define better place for launching a new branch of sport gyms. Therefore, the winner is Moscow, as the only cluster that had “gym” in top 2 attendance places was cluster 3 in Moscow.