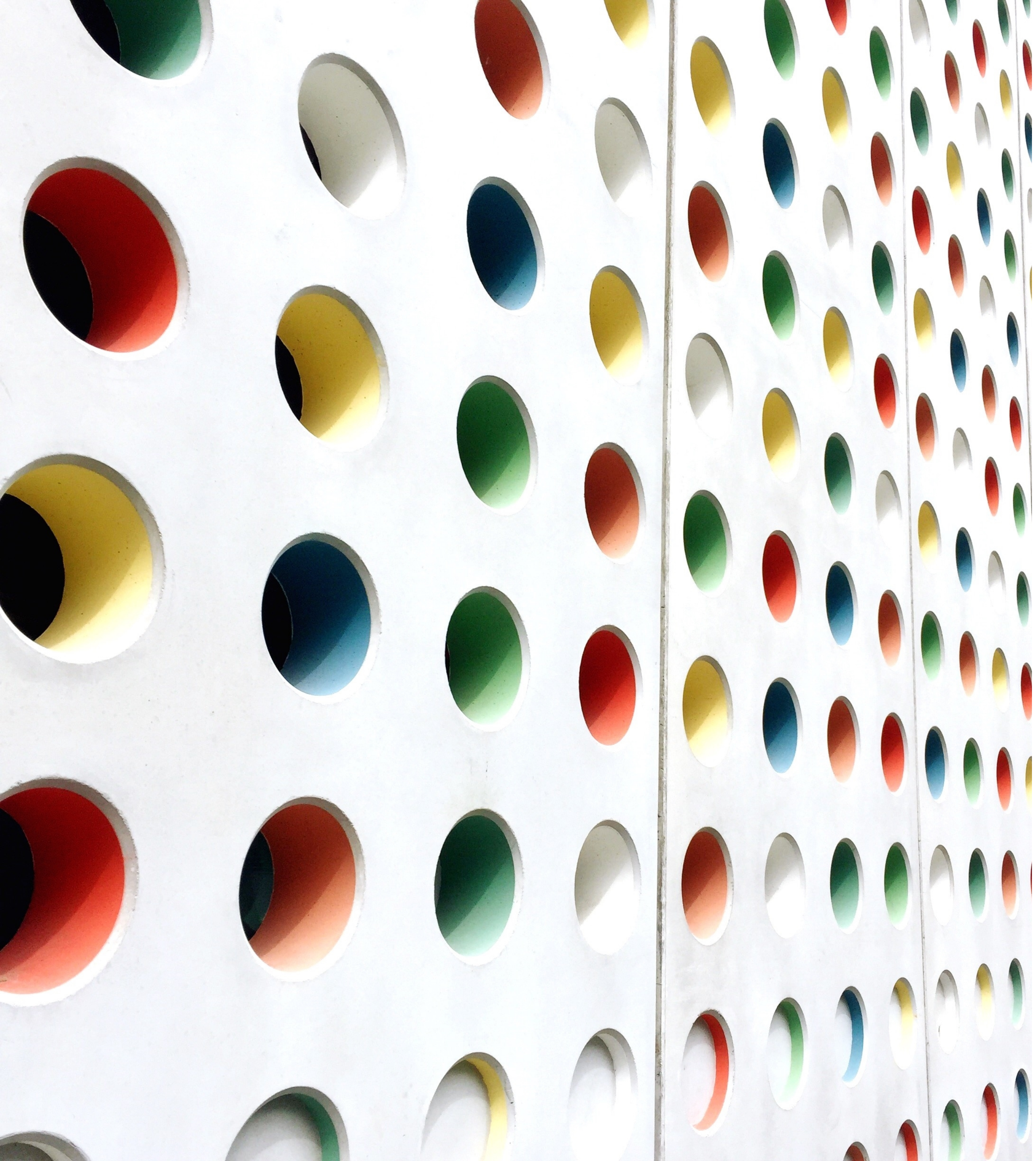


Capstone Project - The Battle of the Neighbourhoods Project: “Dress to Impress”

APPLIED DATA SCIENCE CAPSTONE BY IBM

SILVIA IOANA CARBUNAREA PRISECARU

SILVIA.PRISECARU@GMAIL.COM

Table of Contents

Business Problem

- Bucharest is the capital of Romania, 2.3M population, 6th largest in European Union
- Bucharest has a burgeoning Cafe culture and offers residents an array of venues catering to every budget and desire.
- Andreea is a fashion vlogger moving to Bucharest, Romania, to follow her dream of opening her own coffee shop.
- The target audience for this project should also be other self-employed people looking for fame and cash-flow generated by their presence in certain places, in the city of Bucharest.

Data: Wikipedia

```
df.head(10)
```

14]:

	Neighbourhood
0	Băneasa, Bucharest
1	Berceni, Bucharest
2	Bucureștii Noi
3	Centrul Civic
4	Colentina, Bucharest
5	Cotroceni
6	Crângași
7	Dămăroaia
8	Dealul Spirii
9	Dorobanți

```
df['Neighbourhood'] = df.Neighbourhood.str.replace(', Bucharest,?' , '')
```

```
df.head(10)
```

7]:

	Neighbourhood
0	Băneasa
1	Berceni
2	Bucureștii Noi
3	Centrul Civic
4	Colentina
5	Cotroceni
6	Crângași
7	Dămăroaia
8	Dealul Spirii
9	Dorobanți

https://en.wikipedia.org/wiki/Category:Districts_of_Bucharest

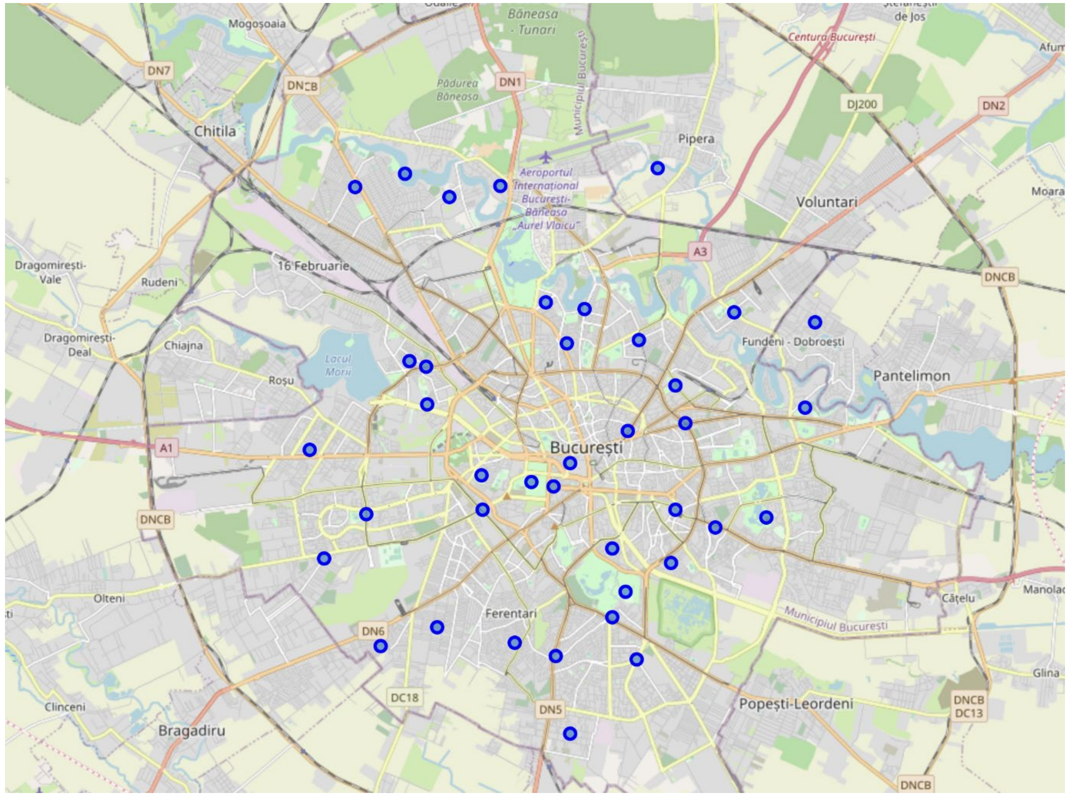
Data: Google Maps API Geocoding

```python
df['Latitude'] = latitudes
df['Longitude'] = longitudes
```

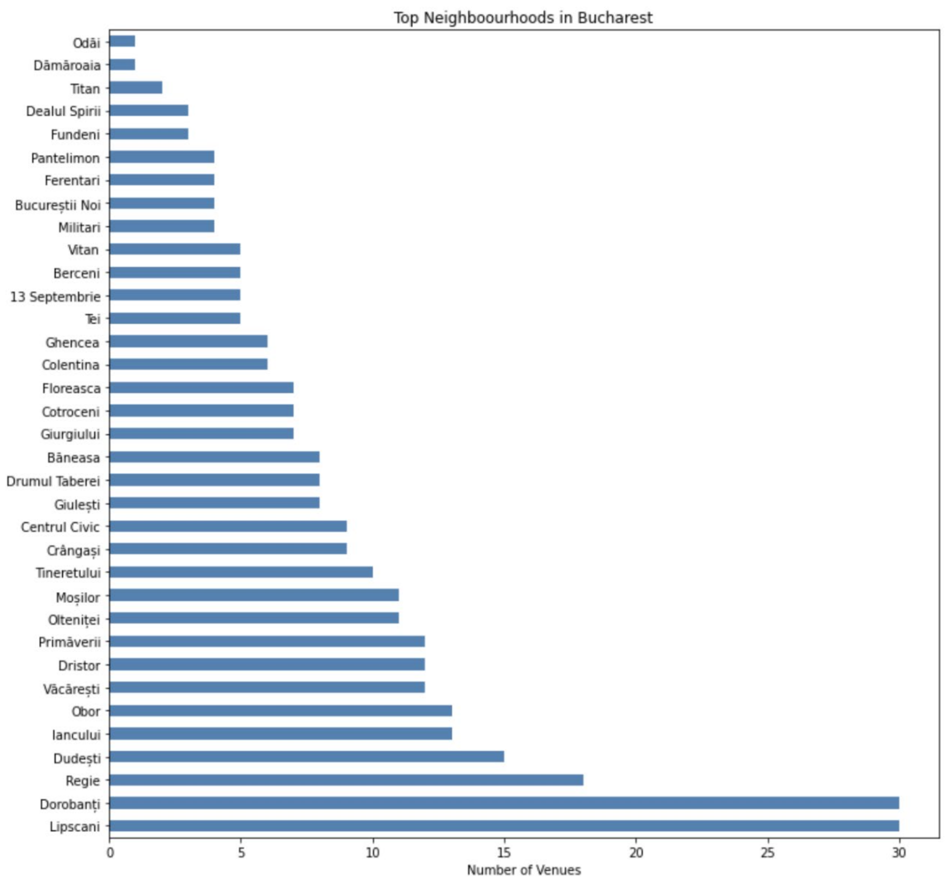
```python
df.head(10)
```

5]:

	Neighbourhood	Latitude	Longitude
0	Băneasa	44.493726	26.076048
1	Berceni	44.389221	26.118203
2	Bucureștii Noi	44.493619	26.031081
3	Centrul Civic	44.427285	26.092441
4	Colentina	44.465766	26.148647
5	Cotroceni	44.429874	26.070091
6	Crângași	44.455002	26.047913
7	Dămăroaia	44.491447	26.060160
8	Dealul Spirii	44.428385	26.085606
9	Dorobanți	44.459076	26.096738



Data: Foursquare and its most popular 3 neighbourhoods considering number of tips within 300m radious



Methodology: One-hot-Encoding to convert categorical to binary values and calculate frequency

Analyzing each Neighbourhood

In [345]:

```
bucharest_onehot = pd.concat([bucharest_venues['Neighbourhood'],pd.get_dummies(bucharest_venues['Venue Category'])], axis=1)
print(bucharest_onehot.shape)
bucharest_onehot.head()
```

(308, 111)

Out[345]:

	Neighbourhood	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auto Dealership	BBQ Joint	Bagel Shop	Bakery	Bar	...	Taco Place
0	Băneasa	0	0	0	0	0	0	0	0	0	...	0
1	Băneasa	0	0	0	0	0	0	0	0	0	...	0
2	Băneasa	0	0	0	0	0	0	0	0	0	...	0
3	Băneasa	0	0	0	0	0	0	0	0	0	...	0
4	Băneasa	0	0	0	0	0	0	0	0	0	...	0

5 rows × 111 columns

In [346]:

```
bucharest_grouped = bucharest_onehot.groupby('Neighbourhood').mean().reset_index()
bucharest_grouped
```

Out[346]:

	Neighbourhood	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auto Dealership	BBQ Joint	Bagel Shop	Bakery	Bar	...	T Pl
0	13 Septembrie	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	...	0.000
1	Berceni	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.200000	0.000000	...	0.000
2	Bucureștii Noi	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.250000	0.000000	...	0.000
3	Băneasa	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.125000	0.000000	...	0.000

Methodology: Top 5 most common venues

Putting into Pandas framework

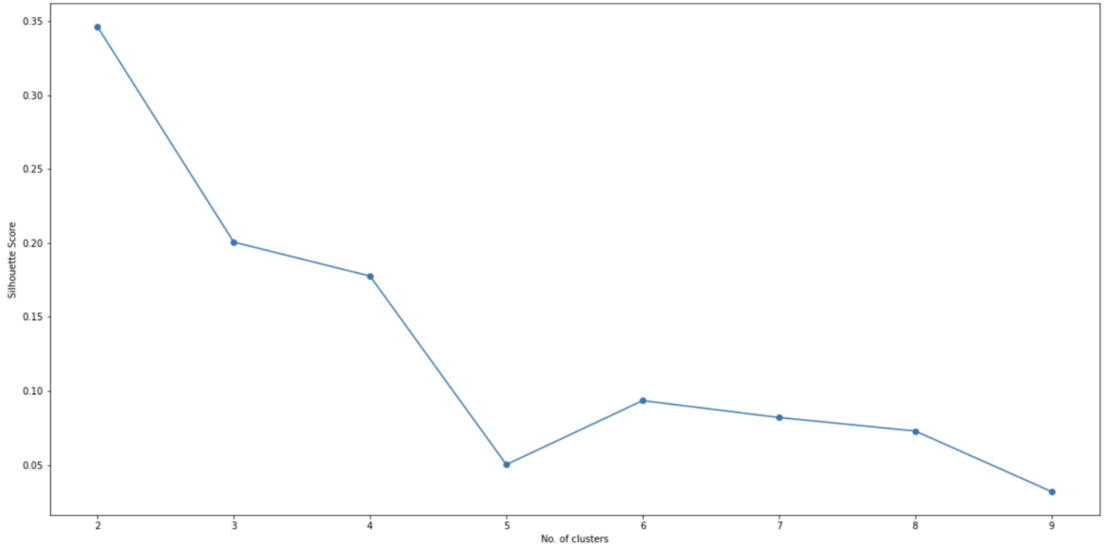
```
[48]: def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

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

```
[49]: num_top_venues = 5

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

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))

neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighbourhood'] = bucharest_grouped['Neighbourhood']

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

neighborhoods_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	13 Septembrie	Romanian Restaurant	Indian Restaurant	Plaza	Pizza Place	Department Store
1	Berceni	Pub	Lebanese Restaurant	Bakery	Cheese Shop	Fountain
2	București Noi	Dessert Shop	Gym	Bakery	Supermarket	Fried Chicken Joint
3	Băneasa	Café	Restaurant	Tunnel	Theme Restaurant	Pizza Place
4	Centrul Civic	Restaurant	Theater	Romanian Restaurant	Clothing Store	Chocolate Shop

Methodology: K-means Clustering

```
]: plot(max_range, scores, "No. of clusters", "Silhouette Score")
```




```
]: opt = np.argmax(scores) + 2
   print(opt)
```

2

K-means for optimal number of clustering

```
kclusters = opt
kmeans = KMeans(n_clusters = kclusters, init = 'k-means++', random_state = 0).fit(bucharest_grouped_clustering)
```

```
neighborhoods_venues_sorted.drop(['Cluster Labels'], axis=1, inplace=True)
```

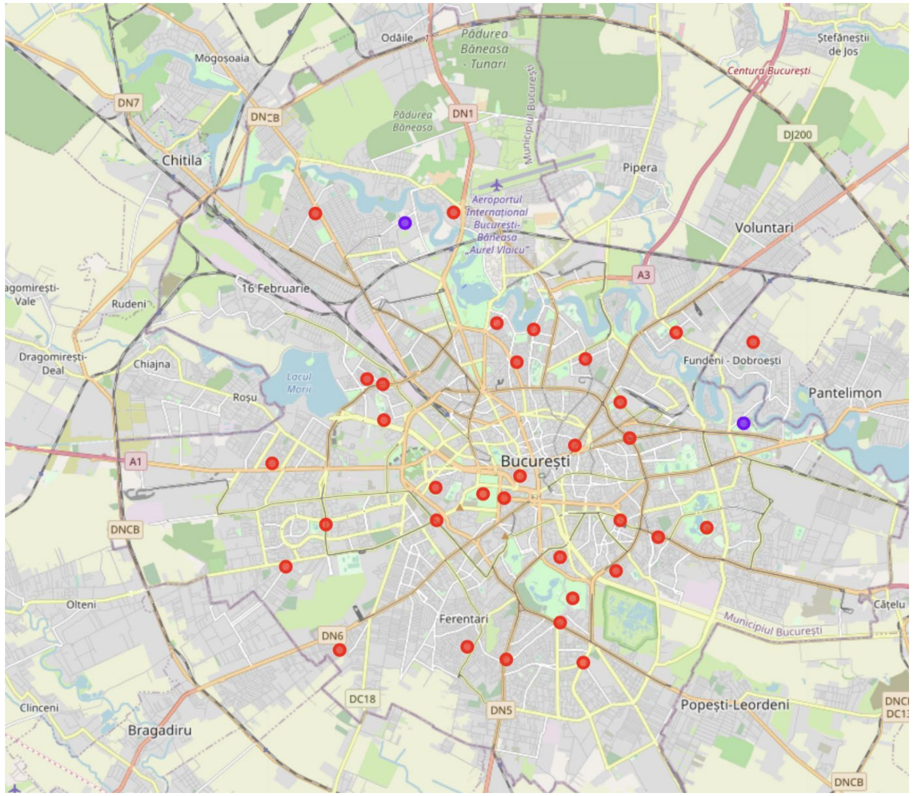
```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
neighborhoods_venues_sorted.head()
```

]:

	Cluster Labels	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	0	13 Septembrie	Romanian Restaurant	Indian Restaurant	Plaza	Pizza Place	Department Store
1	0	Berceni	Pub	Lebanese Restaurant	Bakery	Cheese Shop	Fountain
2	0	Bucureștii Noi	Dessert Shop	Gym	Bakery	Supermarket	Fried Chicken Joint
3	0	Băneasa	Café	Restaurant	Tunnel	Theme Restaurant	Pizza Place
4	0	Centrul Civic	Restaurant	Theater	Romanian Restaurant	Clothing Store	Chocolate Shop

Methodology: Cluster visualization to choose Cluster 0



Results: Filtering Cluster 0 without 'Cafe' -with 'Restaurant' listed as most common

```
In [376]: venues0a = venues0a[venues0a['4th Most Common Venue'] != 'Café']

In [377]: venues0a = venues0a[venues0a['5th Most Common Venue'] != 'Café']

In [378]: venues0a.shape
Out[378]: (27, 9)

In [379]: venues0a.head()

In [380]: venues0a['1st Most Common Venue'].value_counts()
```

Choosing Neighbourhood that has Restaurant as Most Common Venue

```
In [381]: venues=venues0a.loc[venues0a['1st Most Common Venue'] == 'Restaurant'].reset_index().drop(['index'],axis=1)
          venues
Out[381]:
```

	Neighbourhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Centrul Civic	44.427285	26.092441	0.0	Restaurant	Theater	Romanian Restaurant	Clothing Store	Chocolate Shop
1	Drumul Taberei	44.421340	26.034485	0.0	Restaurant	Grocery Store	Farmers Market	Park	Skating Rink
2	Fundeni	44.463675	26.173474	0.0	Restaurant	Bar	Wine Bar	Fried Chicken Joint	Dessert Shop
3	Tei	44.459806	26.118913	0.0	Restaurant	Doner Restaurant	Italian Restaurant	Electronics Store	Bar
4	Titan	44.420545	26.158415	0.0	Restaurant	Park	IT Services	Dessert Shop	Doner Restaurant

Results: lower end of the price/sqm

In [389]:
```
df_price.head(20)
```

Out[389]:

	Neighbourhood	Price/sqm	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Kiseleff	2580	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Aviatorilor	2580	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Herăstrau	2410	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Nordului	2410	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Dorobanți	1990	44.459076	26.096738	0.0	Sushi Restaurant	Café	Bakery	Restaurant	Vegetarian / Vegan Restaurant
5	Floreasca	1990	44.466539	26.102152	0.0	Pool	Hotel	French Restaurant	Eastern European Restaurant	Lounge
6	Aviației	1870	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Unirii	1720	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	Drumul Taberei	1050	44.421340	26.034485	0.0	Restaurant	Grocery Store	Farmers Market	Park	Skating Rink
9	Giurgiului	1040	44.389770	26.093142	0.0	Playground	Pizza Place	Sandwich Place	Electronics Store	Supermarket
10	Giulesti	950	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	Rahova	940	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12	Ghencea	940	44.411369	26.021560	0.0	Gym	Athletics & Sports	Pub	Supermarket	Bus Station

In [ ]:

Results

- Used a combination of APIs in order to generate similar clusters of neighbourhoods from Bucharest
- Based on the frequency of the venues located in these neighbourhoods, but also on the lack of presence of Cafes among the top 5 of the frequencies, a neighbourhood was to be selected that matches also Andreea's budget.
- Further possible research could be done, were we to get a hold of further information about other dissimilarity criteria, such as: average income, average spending in Cafes, average time spent in cafes, and closeness of office building from cafes.