What Neighborhood should I live in?
A recommendation system for new Expats

Using K-means Clustering and Content based recommendation to select suitable neighborhoods in a new city, basis user preferences

Dimitris Mertikas

April 10th 2019

1. Introductory Information

Where to live in a new city is one of the most daunting tasks and depends at a big part, a matter of one's preferences and lifestyle. This project tries to create a content based recommendation system for assisting expats in choosing a neighborhood basis the way they rank a number of lifestyle categories. For testing this system we will use Toronto/Canada as an example city.

Toronto, is a multicultural city that continues to attract a big number of expats from different countries in the world. Its neighborhoods are constantly evolving, and boundaries can become blurred and disputed, so the pure geographically defined Neighborhoods are not the best benchmark for suitability to a new expat that doesn't know the city and help him choose where to live.

A recommendation system suggesting suitable neighborhoods for new expats, could be a value added feature to existing recommendation platforms as Tripadvisor, Foursquare, Time Out etc where the user can be guided on a suitable Neighborhood for him and relative venues can be suggested around this area basis his profile. This system could also have an application in real estate agency services where the agent can provide a more personalized recommendation on property to a client at a new city. Business owners, marketers and city designers would also benefit in redesigning their models around "lifestyle clusters" that can be created from developing this system rather than focusing purely on the standardized center/suburb layout.

2. Data Sources

For the purpose of this project we used data from two sources: We scrapped the Neighborhood/area data for Toronto from Wikipedia to create a list of Neighborhoods with geographical coordinates and then using the Foursquare (online venue recommendation platform) API we explored the venues listed in Toronto, which helped us create the lifestyle categories we needed.

Links to the data:

List of Neighborhoods organized by Postal Codes:
https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

List of Coordinates: http://cocl.us/Geospatial_data

For the venue categories: **Foursquare API**
https://developer.foursquare.com/docs/api/venues/explore

3.Methodology

Data cleaning

The first thing we did was to organize our data: We used the Wikipedia page to gather the information on Neighborhoods along with Postal Codes and Boroughs. Some neighborhoods were sharing the same postal code whereas others had not assigned values. We replaced in the “not assigned” values with the name of the corresponding Borough and grouped the Neighborhoods that shared the same Postal Code under the same row. We then imported the coordinates and assigned them to each Neighborhood using the Postal Code as the “ID” for matching the values.

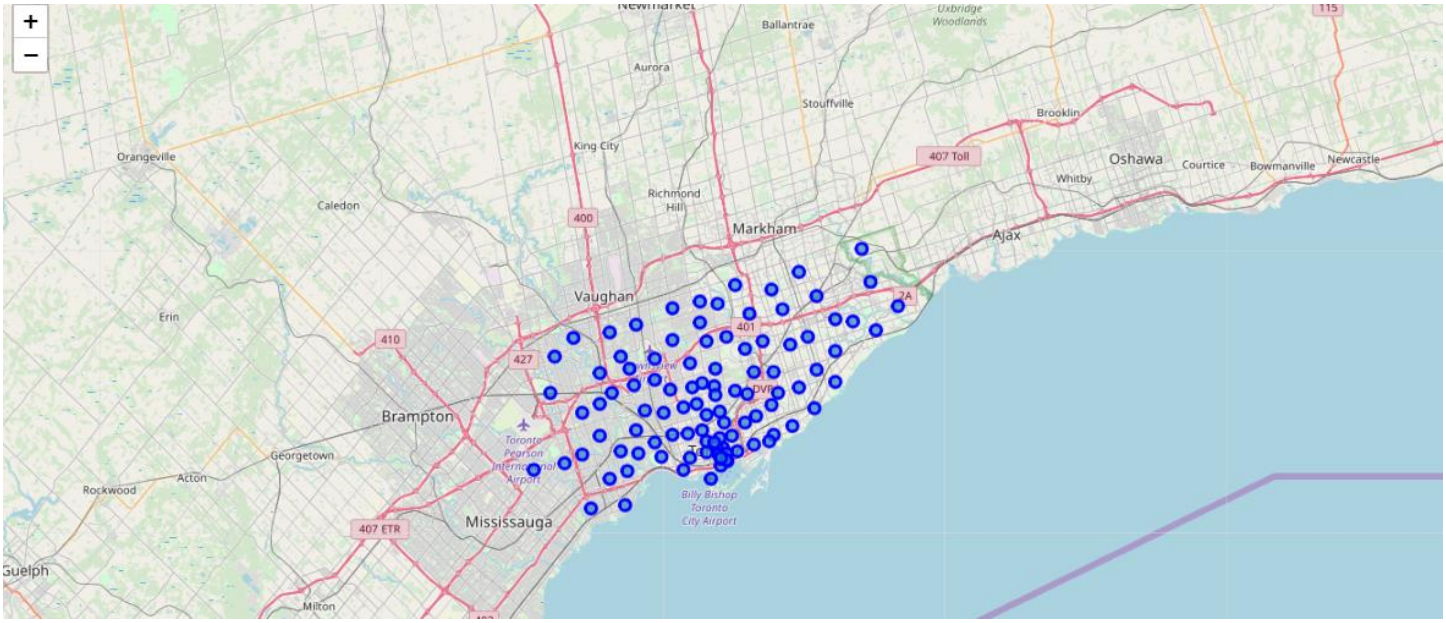
We then created our dataframe:

```
[14]: df_toronto.dropna()
      df_toronto.head()
```

[14]:

	Postcode	Borough	Neighborhood	Latitude	Longitude
0	M1B	Scarborough	Rouge,Malvern	43.806686	-79.194353
1	M1C	Scarborough	Highland Creek,Rouge Hill,Port Union	43.784535	-79.160497
2	M1E	Scarborough	Guildwood,Morningside,West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

Next we used Python Geocoding toolbox – *GeoPy / Geolocator* to establish the coordinates of our center (Toronto) and map the Neighborhoods using *Folium* library.



The next step was to collect the data on the venues and create a new dataframe with the venue categories for each neighborhood.

Foursquare has numerous 'Venue Categories' that are used to identify each type of venue. A 'GET' request was sent to the *'api.foursquare.com/v2/venues/explore?'* endpoint and results were appended to a list that was used to start the segmentation and explore Toronto Neigbhorhoods and venues.

After filtering our initial data we came up with a dataframe containing 100 Neighborhoods and 275 unique venue categories.

Before further processing we needed to detect any outliers in the dataset which would affect our further results. From a visual estimation (as the number of data is small) two venue categories stood out: "Coffee Shop" with 188 occurrences, and "Café" with 99. To verify this visual assumption, we utilized Z score with a threshold=3 to identify outliers using Python *SciPy* library.

```
    Venue Category  count
66     Coffee Shop    188
53            Café     99
219     Restaurant     60
5.728462643903424 11.340893019370231 3.2690830411707785
```

Z score is a measure of how many standard deviations below or above the population mean a raw score is. Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve).

From Z score method we found that actually 3 observations were above the threshold - "Coffee Shop" and "Café" at 11.3399 and 5.7278 followed by a lower score of 3.2686 for "Restaurants". For our furth er feature selection we removed the two top outliers and intentionally kept "Restaurants" as a categor y but we limited it through aggregation of selected features. The reason is that apart from the margina lly higher Z score value, Restaurants are distinct venue categories which are considered important pa rameters for lifestyle benchmarking. Coffee Shops and Café on the other side (which are practically u nder the same category) are rather generic descriptions of a category that is present at the majority of a city's locations and can be part of a further description of a venue that has a different core activity – i.e a Restaurant that serves coffee can also be tagged as a Coffee Shop in a venue recommendation engine.

To help us with organizing the venues we used Onehot encoding in our dataset to create a dataframe with occurence of each venue present (1) or not (0) under each Neighborhood and then we grouped t he rows by neighborhood and the mean of the frequency of occurrence of each category was calculat ed and included in a new dataframe.

[29]:

	Neighborhood	Yoga Studio	Accessories Store	Adult Boutique	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	...	Trail
0	Adelaide,King,Richmond	0.0	0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
1	Agincourt	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
2	Agincourt North,L'Amoreaux East,Milliken,Steel...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
3	Albion Gardens,Beaumond Heights,Humbergate,Jam...	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
4	Alderwood,Long Branch	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

5 rows × 275 columns

Feature Selection

As mentioned, there are 274 venue categories in the dataset with a lot of them belonging to the same general "group". i.e there is "Vietnamese Restaurant", "Greek Restaurant", "Vegetarian Restaurant" all belonging to the same logical group. The 274 venues will be filtered through a key words search so to create a number of higher level categories which will be used for both clustering as well as recommendation.

For this purpose the categories created with corresponding filters were:

1. Restaurants - keywords (*Restaurant*|*Steak*|*Bistro*)
2. Fast_Food - keywords (*Fast*|*Burger*|*Food Truck*|*Sandwich*|*Pizza Place*|*Place*|*Joint*)
3. Health_Fitness - keywords (*Yoga*|*Gym*|*Martial*|*Health*|*Fitness*|*Pool*|*Healthy*|*Field*|*Sports*|*Trail*|
4. *Stadium*|*Tennis*|*Climbing*)
5. Kids_Friendly - keywords (*Kid*|*Play*|*Park*|*Nursery*|*Playground*|*School*|*College*|*Nursery*|*Medical Center*|*Clinic*|*Basketball*|*Football*)

6. Nightlife' - keywords (*Night*|*Bar*|*Pub*|*Wine*|*Dance*|*Strip Club*|*Adult*|*Club*|*Speakeasy*|*Liquor*|*Brewery*)
7. Culture -keywords (*Gallery*|*Museum*|*Paintings*|*Monument*|*Church*|*Theater*|*Cinema*|
8. *Sculpture*|*Aquarium*|*Concert*|*Art*)
9. Shopping - keywords (*Mall*|*Shopping*|*Shop*|*Retail*|*Store*|*Boutique*)
10. Short_stay - keywords (*Motel*|*Hotel*|*Airport*|*Stay*)
11. Personal_care - keywords (*Salon*|*Barber*|*Spa*|*Massage*|*Hair*|*Nail*|*Tanning*)
12. Food_markets - keywords (*Supermarket*|*Market*|*Grocery*|*Bakery*|*Butcher*)
13. Transportation - keywords (*Train*|*Metro*|*Bus*|*Tram*|*Boat*)
14. Leisure - keywords (*Beach*|*Plaza*|*Skating*)

```
pd.options.mode.chained_assignment = None

Toronto_1=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Restaurant|Steak|Bistro')]
Toronto_1['total1']=Toronto_1.sum(axis=1)

Toronto_2=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Fast|Burger|Food Truck|Sandwich|Pizza Place|Place|Joint')]
Toronto_2['total2']=Toronto_2.sum(axis=1)

Toronto_3=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Yoga|Gym|Martial|Health|Fitness|Pool|Healthy|Field|Sports|Trail|Stadium|Tennis|Climbing')]
Toronto_3['total3']=Toronto_3.sum(axis=1)

Toronto_4=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Kid|Play|Park|Nursery|Playground|School|College|Nursery|Medical Center|Clinic|Basketball|Football')]
Toronto_4['total4']=Toronto_4.sum(axis=1)

Toronto_5=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Night|Bar|Pub|Wine|Dance|Strip Club|Adult|Club|Speakeasy|Liquor|Brewery')]
Toronto_5['total5']=Toronto_5.sum(axis=1)

Toronto_6=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Gallery|Museum|Paintings|Monument|Church|Theater|Cinema|Sculpture|Aquarium|Concert|Art')]
Toronto_6['total6']=Toronto_6.sum(axis=1)

Toronto_7=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Mall|Shopping|Shop|Retail|Store|Boutique')]
Toronto_7['total7']=Toronto_7.sum(axis=1)

Toronto_8=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Motel|Hotel|Airport|Stay')]
Toronto_8['total8']=Toronto_8.sum(axis=1)

Toronto_9=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Salon|Barber|Spa|Massage|Hair|Nail|Tanning')]
Toronto_9['total9']=Toronto_9.sum(axis=1)

Toronto_10=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Supermarket|Market|Grocery|Bakery|Butcher')]
Toronto_10['total10']=Toronto_10.sum(axis=1)

Toronto_11=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Train|Metro|Bus|Tram|Boat')]
Toronto_11['total11']=Toronto_11.sum(axis=1)

Toronto_12=Toronto_grouped.loc[:, Toronto_grouped.columns.str.contains('Beach|Plaza|Skating')]
Toronto_12['total12']=Toronto_12.sum(axis=1)
```

```
Toronto_grouped['Restaurants']=Toronto_1['total1']
Toronto_grouped['Fast_Food']=Toronto_2['total2']
Toronto_grouped['Health_Fitness']=Toronto_3['total3']
Toronto_grouped['Kids_Friendly']=Toronto_4['total4']
Toronto_grouped['Nightlife']=Toronto_5['total5']
Toronto_grouped['Culture']=Toronto_6['total6']
Toronto_grouped['Shopping']=Toronto_7['total7']
Toronto_grouped['Short_stay']=Toronto_8['total8']
Toronto_grouped['Personal_care']=Toronto_9['total9']
Toronto_grouped['Food_markets']=Toronto_10['total10']
Toronto_grouped['Transportation']=Toronto_11['total11']
Toronto_grouped['Leisure']=Toronto_12['total12']
Toronto_grouped.head(5)
```

```
Toronto_grouped_new= Toronto_grouped[['Restaurants',
                                      'Fast_Food',
                                      'Health_Fitness',
                                      'Kids_Friendly',
                                      'Nightlife',
                                      'Culture',
                                      'Shopping',
                                      'Short_stay',
                                      'Personal_care',
                                      'Food_markets',
                                      'Transportation',
                                      'Leisure']].copy()
Toronto_grouped_new.head()
```

The result was a new dataframe with 12 key categories as columns and the corresponding mean frequencies per Neighborhood as sums of the values we calculated earlier with Onehot encoding grouped dataframe.

Neighborhood	Restaurants	Fast_Food	Health_Fitness	Kids_Friendly	Nightlife	Culture	Shopping	Short_stay	Personal_care	Food_markets	Transportation	Leisure
Adelaide,King,Richmond	0.320000	0.090000	0.03	0.0	0.100000	0.06	0.15	0.03	0.01	0.030000	0.0	0.01
Agincourt	0.000000	0.250000	0.00	0.0	0.000000	0.00	0.25	0.00	0.00	0.000000	0.0	0.00
Agincourt North,L'Amoreaux East,Milliken,Steeles East	0.000000	0.000000	0.00	1.0	0.000000	0.00	0.00	0.00	0.00	0.000000	0.0	0.00
Albion Gardens,Beaumond Heights,Humbergate,Jamestown,Mount Olive,Silverstone,South Steeles,Thistletown	0.166667	0.333333	0.00	0.0	0.083333	0.00	0.50	0.00	0.00	0.166667	0.0	0.00
Alderwood,Long Branch	0.000000	0.300000	0.20	0.0	0.200000	0.00	0.10	0.00	0.00	0.000000	0.0	0.10

Next we created and applied a function that sorted our dataframe basis top 5 most common categories for each Neighborhood which would help us in the evaluation of our recommendation system results.

```
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]


num_top_venues = 5

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


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))


neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = Toronto_grouped_new['Neighborhood']

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

neighborhoods_venues_sorted.head(10)
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Adelaide,King,Richmond	Restaurants	Shopping	Nightlife	Fast_Food	Culture
1	Agincourt	Shopping	Fast_Food	Leisure	Transportation	Food_markets
2	Agincourt North,L'Amoreaux East,Milliken,Steel...	Kids_Friendly	Leisure	Transportation	Food_markets	Personal_care
3	Albion Gardens,Beaumond Heights,Humbergate,Jam...	Shopping	Fast_Food	Food_markets	Restaurants	Nightlife
4	Alderwood,Long Branch	Fast_Food	Nightlife	Health_Fitness	Leisure	Shopping
5	Bathurst Manor,Downsview North,Wilson Heights	Shopping	Fast_Food	Restaurants	Food_markets	Leisure
6	Bayview Village	Restaurants	Leisure	Transportation	Food_markets	Personal_care
7	Bedford Park,Lawrence Manor East	Restaurants	Shopping	Fast_Food	Nightlife	Food_markets
8	Berczy Park	Restaurants	Shopping	Nightlife	Food_markets	Culture
9	Birch Cliff,Cliffside West	Leisure	Kids_Friendly	Health_Fitness	Transportation	Food_markets

Modeling:

Our next step would be to identify similarities of Neighborhoods based on the Venues occurrences and see if we can create Neighborhood groups. For this purpose we utilized the K-means clustering algorithm. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K* . It works iteratively to assign each data point to one of K groups based on the features that are provided and ata points are clustered based on feature similarity.

Before running the algorithm we normalized our data using StandardScaler() from sklearn library.

```
from sklearn.preprocessing import StandardScaler
X = Toronto_grouped_new.values[:,1:].astype(float)
X = np.nan_to_num(X)
cluster_dataset = StandardScaler().fit_transform(X)
cluster_dataset
```

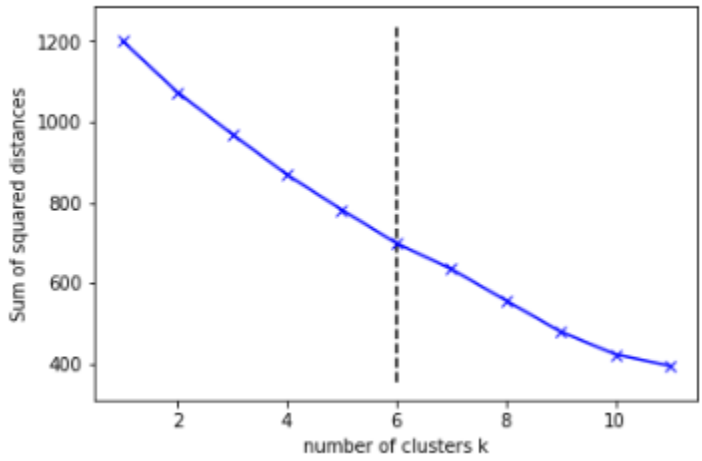
```
array([[ 0.81010254, -0.17712967, -0.35258293, ..., -0.21202455,
        -0.30773218,  0.08072822],
       [-1.02195508,  0.88648141, -0.56617468, ..., -0.57977124,
        -0.30773218, -0.19456288],
       [-1.02195508, -0.7754109 , -0.56617468, ..., -0.57977124,
        -0.30773218, -0.19456288],
       ...,
       [-0.14115815,  1.26999502,  1.07683881, ..., -0.57977124,
        -0.30773218, -0.19456288],
       [-1.02195508, -0.7754109 , -0.56617468, ..., -0.57977124,
        -0.30773218,  6.68771461],
       [-1.02195508, -0.7754109 , -0.56617468, ..., -0.57977124,
        -0.30773218, -0.19456288]])
```

To evaluate the optimum K we used the “Elbow method”. The theory behind this method is that we should choose a number of clusters so that adding another cluster wouldn’t improve much the performance and therefore find the optimal number of clusters where the curvature is maximized. We used KneeLocator (https://pypi.org/project/kneed/) to mark the best K in our graph rather than making a rough estimation.

```
!pip install kneed
y=Sum_of_squared_distances

x = range(1, len(y)+1)

from kneed import KneeLocator
kn = KneeLocator(x, y, curve='convex', direction='decreasing')
print(kn.knee)
plt.xlabel('number of clusters k')
plt.ylabel('Sum of squared distances')
plt.plot(x, y, 'bx-')
plt.vlines(kn.knee, plt.ylim()[0], plt.ylim()[1], linestyles='dashed')
```

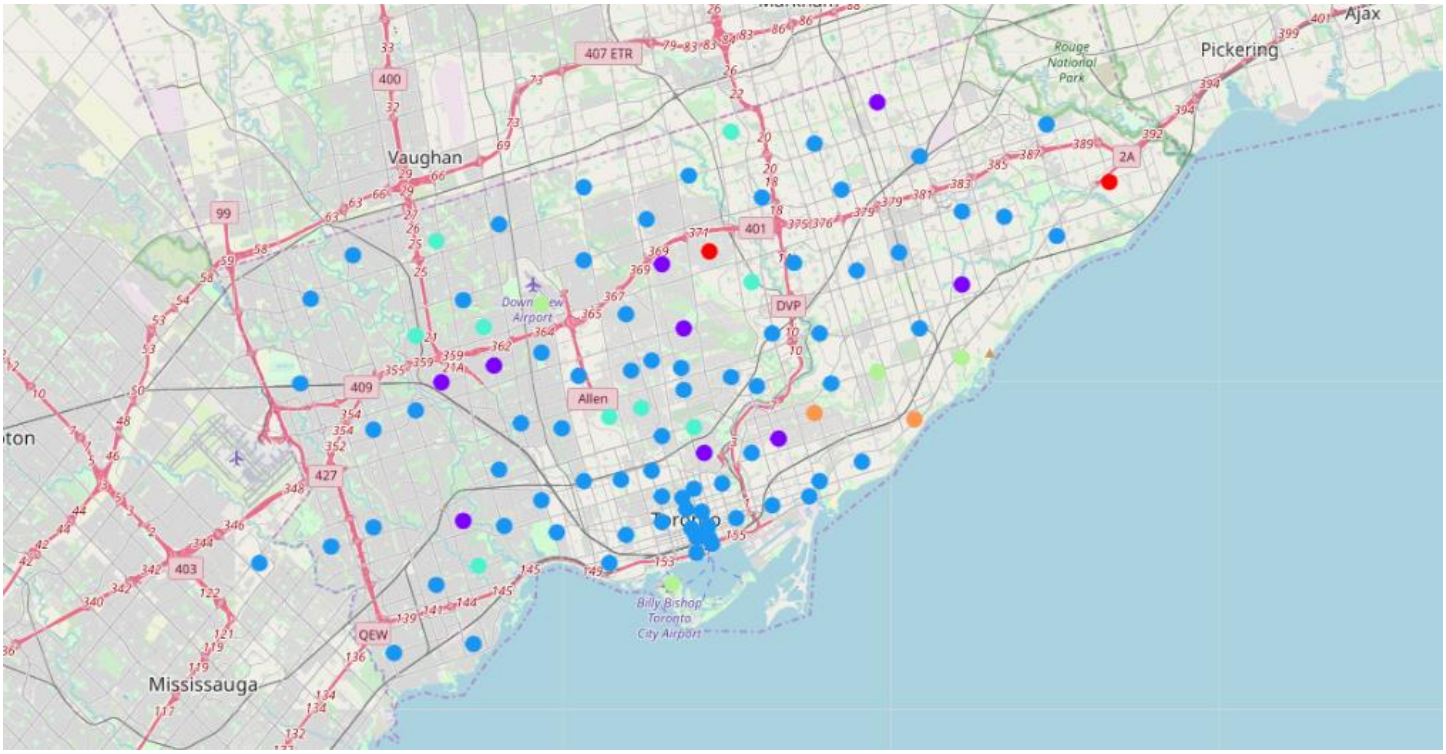
```
y=Sum_of_squared_distances

x = range(1, len(y)+1)

from kneed import KneeLocator
kn = KneeLocator(x, y, curve='convex', direction='decreasing')
print(kn.knee)
plt.xlabel('number of clusters k')
plt.ylabel('Sum of squared distances')
plt.plot(x, y, 'bx-')
plt.vlines(kn.knee, plt.ylim()[0], plt.ylim()[1], linestyles='dashed')
```

```
<matplotlib.collections.LineCollection at 0x7f48b093deb8>
```



We assigned back the labels to the rows of our dataframe and mapped the Neighborhoods accordingly:

Neighborhood	Restaurants	Fast_Food	Health_Fitness	Kids_Friendly	Nightlife	Culture	Shopping	Short_stay	Personal_care	Food_markets	Transportation	Leisure	Labels
Adelaide,King,Richmond	0.320000	0.090000	0.03	0.0	0.100000	0.06	0.15	0.03	0.01	0.030000	0.0	0.01	0
Agincourt	0.000000	0.250000	0.00	0.0	0.000000	0.00	0.25	0.00	0.00	0.000000	0.0	0.00	0
Agincourt North,L'Amoreaux East,Milliken,Steeles East	0.000000	0.000000	0.00	1.0	0.000000	0.00	0.00	0.00	0.00	0.000000	0.0	0.00	1
Albion Gardens,Beaumond Heights,Humbergate,Jamestown,Mount Olive,Silverstone,South Steeles,Thistletown	0.166667	0.333333	0.00	0.0	0.083333	0.00	0.50	0.00	0.00	0.166667	0.0	0.00	0
Alderwood,Long Branch	0.000000	0.300000	0.20	0.0	0.200000	0.00	0.10	0.00	0.00	0.000000	0.0	0.10	0



Next we proceeded with our recommendation system – We used the theory behind *Content based recommendation* systems. In content-based types of filtering, the similarity between different items or products is calculated on the basis of the attributes of the products. For instance, in a content-based movie recommender system as i.e *Neflix*, the similarity between the movies is calculated on the basis of genres, the actors in the movie, the director of the movie, etc and based on what we like, the algorithm will simply pick movies with similar content to recommend us.

To apply content based filtering in our case we had to consider that there is no pre-stored user profile with previous Neighborhood preferences as we consider that the user will be visiting the city for the first time so he has no knowledge of its Neighborhoods.

The first step was to feed the system with user ratings for each of the 12 categories we created depending on how important each was for the user in a scale of 1-10. For the testing purpose we used a “persona” of a user that is married with children and rated each category accordingly.

```
userInput_Family = [
            {'Venue':'Restaurants','Rating':7},
            {'Venue':'Fast_Food','Rating':5},
            {'Venue':'Health_Fitness','Rating':2},
            {'Venue':'Kids_Friendly','Rating':10},
            {'Venue':'Nightlife','Rating':1},
            {'Venue':'Culture','Rating':8},
            {'Venue':'Shopping','Rating':6},
            {'Venue':'Short_stay','Rating':0},
            {'Venue':'Personal_care','Rating':6},
            {'Venue':'Food_markets','Rating':10},
            {'Venue':'Transportation','Rating':5},
            {'Venue':'Leisure','Rating':8}
        ]
inputVenues_Family = pd.DataFrame(userInput_Family)

inputVenues_Family
```

	Rating	Venue
0	7	Restaurants
1	5	Fast_Food
2	2	Health_Fitness
3	10	Kids_Friendly
4	1	Nightlife
5	8	Culture
6	6	Shopping
7	0	Short_stay
8	6	Personal_care
9	10	Food_markets
10	5	Transportation
11	8	Leisure

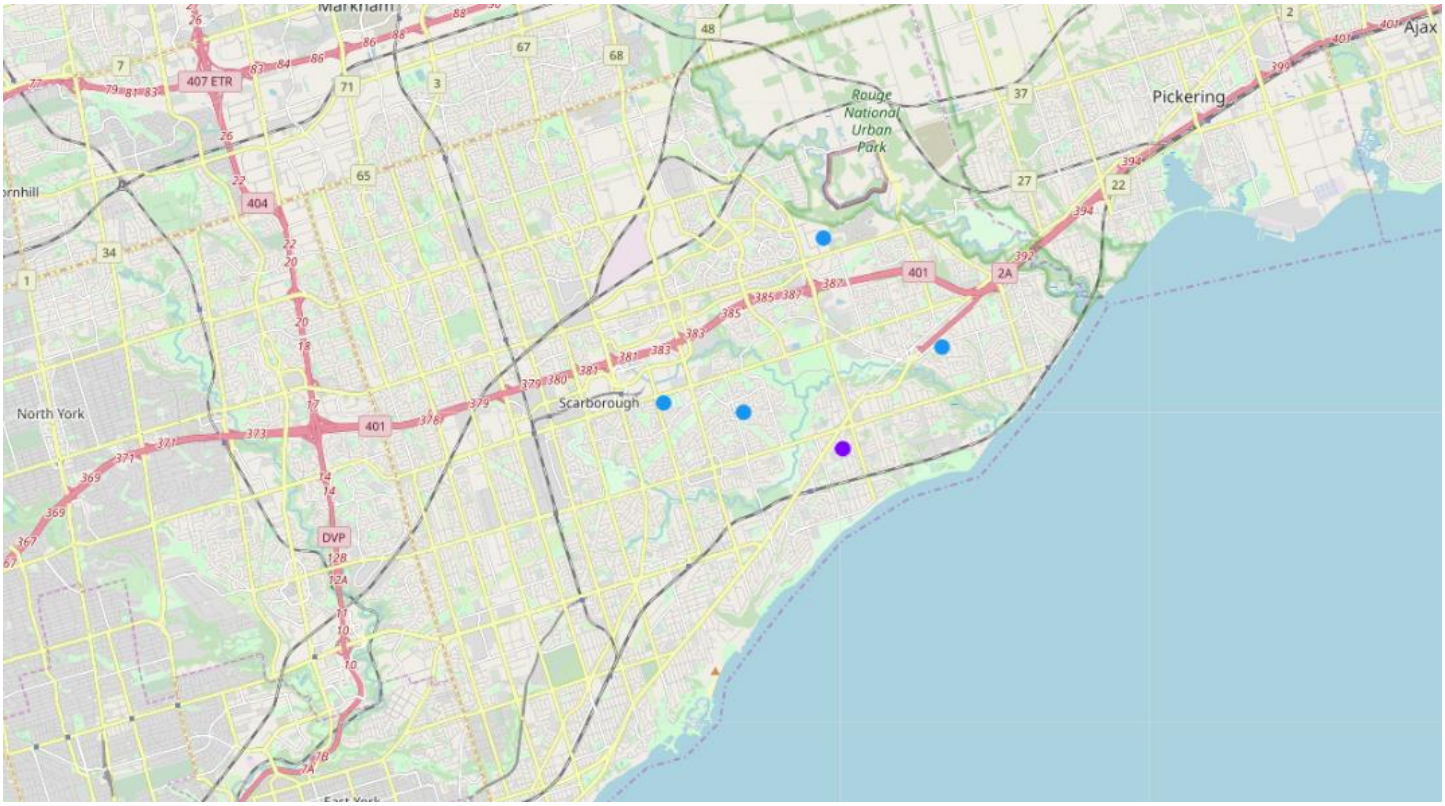
Next we created the user preferences matrix by calculating the weighted average of the user ratings against the frequency of each category and for all Toronto neighborhoods in our dataframe. The result was sorted to a top 5 list of recommended Neighborhoods basis user profile which were then plotted on the map of Toronto.

```
recommendationTable_family = ((Toronto_grouped_new*inputVenues_Family['Rating']).sum(axis=1))/(inputVenues_Family['Rating'].sum())
recommendationTable_family.head(5)
```

```
Neighborhood
Adelaide,King,Richmond                                                                  0.068676
Agincourt                                                                               0.040441
Agincourt North,L'Amoreaux East,Milliken,Steeles East                                   0.147059
Albion Gardens,Beaumond Heights,Humbergate,Jamestown,Mount Olive,Silverstone,South Steeles,Thistletown    0.111520
Alderwood,Long Branch                                                                   0.051471
dtype: float64
```

```
recommendationTable_family = recommendationTable_family.sort_values(ascending=False)
recommendationTable_family.head()
```

```
Neighborhood
Rouge,Malvern                                    0.176471
Scarborough Village                              0.147059
Agincourt North,L'Amoreaux East,Milliken,Steeles East    0.147059
Downsview West                                   0.139706
Weston                                           0.127451
dtype: float64
```



4. Results

K-Means Clustering: The algorithm gave us 6 clusters according to the 12 categories created. The cluster with the most Neighborhoods was the one that included Downtown Toronto which was expected considering that the venues in a city are more condensed around its center. Kmeans did help identify area clusters based on the venue concentration and give us a first distinction on the areas as i.e where is the busiest locations and which ones are more of residential ones (i.e downtown vs suburbs).

Recommendation System: As mentioned we do not have previous user history of preferences or knowledge of the city of Toronto and its Neighborhoods to verify the suitability of the suggestions, so we are dealing with cold start users where it is not possible to make a comparative evaluation. To measure our results we cross-referenced the recommendations given with the top 5 venue dataset we created earlier in our process.

```
compare=neighborhoods_venues_sorted.loc[neighborhoods_venues_sorted['Neighborhood'].isin(result['Neighborhood'])]
compare
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
18	Cedarbrae	Restaurants	Food_markets	Health_Fitness	Fast_Food	Leisure
48	Guildwood,Morningside,West Hill	Personal_care	Shopping	Kids_Friendly	Fast_Food	Restaurants
53	Highland Creek,Rouge Hill,Port Union	Culture	Nightlife	Leisure	Transportation	Food_markets
76	Rouge,Malvern	Fast_Food	Restaurants	Leisure	Transportation	Food_markets
96	Woburn	Shopping	Restaurants	Leisure	Transportation	Food_markets

By maping the recommended neighborhoods we can see that out of the 5 recommended neighborhoods, 4 belong to the same cluster as calculated by K-means and 1 in a different one.

A qualitative analysis of the recommended results is a more challenging task since to properly evaluate the suitability of the neighborhoods basis the user preferences there should be a sampling process where a number of Toronto residents would participate as users and evaluate the recommendations basis their knowledge of the neighborhoods.

For the scope of this project we will refer to an online Toronto Neighborhood guide http://www.torontoneighbourhoods.net/neighbourhoods/scarborough and see what is mentioned for each of the top 5 recommendations:

Rouge: This neighbourhood features a good mix of affordable housing. It has it's own shopping, schools, parks and community centre, is well served by public transit, and is close to commuter highways.

Malvern: This neighbourhood has retained its rural roots by preserving mature trees, ravine woodlots and parklands.Malvern's affordable real estate has traditionally attracted many new Canadians to this neighbourhood.

Port Union – Centennial : A neighbourhood bound on the south by the railway and to the west by Colonel Danforth Park - a well wooded ravine valley that ushers the Highland Creek on the last leg of its journey into Lake Ontario

Highland Creek: This neighbourhood has a small town feel that emanates from its main street shopping district situated along Old Kingston Road

Guildwood: Guildwood Village is one of Toronto's most beautiful and inclusive neighbourhoods. Guildwood Village has an active community association that produces its own newsletter as well as sponsoring various neighbourhood events.

West Hill: West Hill is a culturally diverse, family oriented neighbourhood located in the south-east part of Toronto. West Hill's natural beauty is derived from Morningside Park and Colonel Danforth Park. West Hill has many fine attributes including a community centre, a public library, an abundance of parkland, a vibrant shopping district and affordable homes.

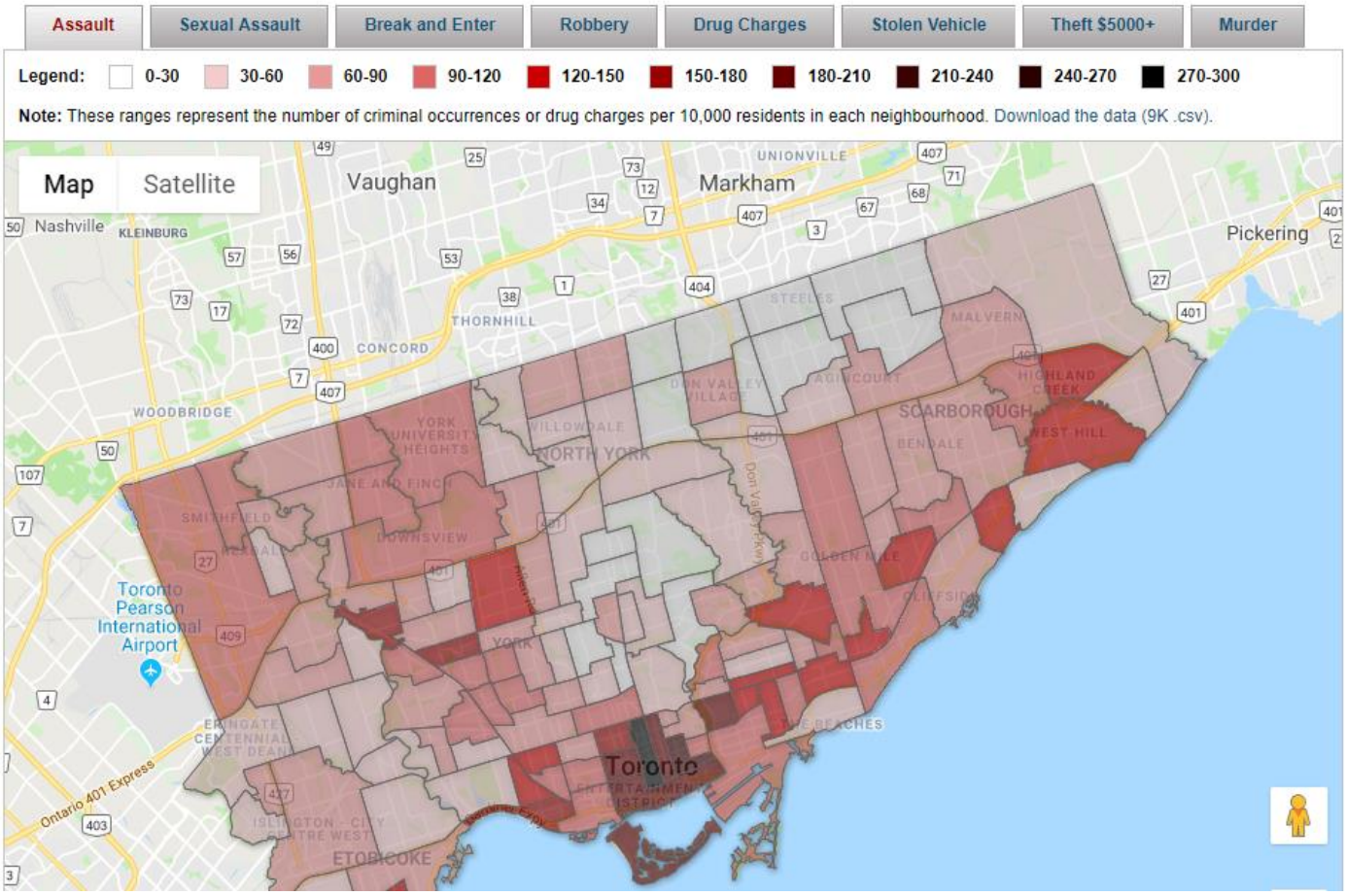
Woburn / Cedarbrae: Woburn is a quiet, family oriented neighbourhood comprised of winding, tree-lined streets that contain a good selection of moderately priced homes

5. Discussion:

Having a tool to suggest a neighborhood for an expat about to relocate to a new country or city without prior knowledge of the area would reduce significantly the time and effort needed for research by limiting the options to a number of recommended areas. The system presented showed a good performance in choosing neighborhoods that score highest on user rankings however it lacks other qualitative characteristics.

For example, let’s look another qualitative metric for the area which is the crime rate (source: https://www.cbc.ca/toronto/features/crimemap/)

The crime rate in some of the recommended areas is quite high and for sure wouldn’t be top options for anyone - let alone a relocating family as was the user in our test.



Furthermore, the system would benefit substantially from a further refinement by:

- Creating a user profile with preferences and check-ins from his current city/neighborhood and then correlate the features with neighborhoods with similar characteristics in the new city as a new input to the ranking matrix.
- Combining it with a Collaborative system filter to compare user profile with ratings and check-ins of other similar users that live in Toronto
- Filter the venues basis rankings before allocating them under a category. There are different types of restaurants, bars, schools which in a lot of cases are correlated to the quality of the venues in the Neighborhood and the type of Neighborhood itself.

6. Conclusion

Relocation is a big challenge for everyone. In this project we tried to visualize Toronto Neighborhoods as clusters created basis common features of a number of lifestyle categories and create a recommendation system for suggesting the top 5 Neighborhoods a new visitor/expat could select basis the importance each has to him. Such a system with necessary refinement and development as mentioned in the discussion section could be scaled to include all the major cities globally where a platform with records of user profiles and preferences could provide personalized recommendations for each user.

The more data were gathered the more the possibility to cluster the neighborhoods across major cities into areas that concentrate the interest of residents with specific lifestyle preferences, as i.e families, bachelors, foreign students, people into fitness etc. Such a platform could be of tremendous use to city designers, perspective regional business owners and marketers who could customize their products and services basis the “lifestyle” group of each Neighborhood and build their business model around smaller cluster centers rather than clutter the current urban centers.