Capstone Project - The Battle of the Neighborhoods (Week 2)

Applied Data Science Capstone by IBM/Coursera

Introduction

Business problem

In this project we will try to find an optimal location that a family of three can move into, based on their preferences and requirements. Specifically, this report will be targeted to stakeholders interested in moving to Minneapolis, Minnesota.

Minneapolis is known as one of the most comfortable cities in the US. It is praised for its parks and theaters; it also reportedly has rather low crime rates. Most importantly, the Minneapolis-St.Paul metropolitan area is home to some 24 Fortune 1000 companies, and it is one of the country's top economies. Some people even like to compare this city to New York.

In our case, a family of three is considering moving to Minneapolis since one of the parents was offered an interesting job opportunity there. They have never been to the state of Minnesota (and Minneapolis in particular), so they decided to consult a specialist on which neighborhood they should pick.

Objective

The objective is to identify a neighborhood in Minneapolis that is most suitable for the given family. This analysis would be relevant and useful for other individuals planning to move to this city.

Problem Statement

The family needs to find a neighborhood that will meet the following criteria:

1) be reasonably close to the earning parent's workplace in Downtown East 2) have a some grocery shops/farmers markets etc 3) have a selection of places to eat out 4) have a green zone in the vicinity 5) have low crime rates 6) offer some form of entertainment and sports facilities (movies, gyms, etc.)

Data analysis will help to find the most suitable neighborhoods that the family will consider for relocation.

Data

We will begin our analysis by identifying the neighborhoods in Minneapolis using information from OpenData (http://opendata.minneapolismn.gov/datasets/minneapolis-neighborhoods/data). We will use geodata to visualize neighborhoods on the map to see which neighborhoods are closer to Downtown East.

Below is the dataframe with the geometry data that will be used to map out neighborhood borders.

As we can see, neighborhoods are not in all caps (as they are in our crime data), so we need to adjust them for consistency

In [15]:
```
df_neighb['BDNAME'] = df_neighb['BDNAME'].str.upper()
df_neighb.drop(columns=['BDNUM','TEXT_NBR','NCR_LINK','IMAGE'],axis=1,inplace=True)
```

In [16]:
```
df_neighb.head()
```

Out[16]:

	FID	BDNAME	Shape_STAr	Shape_STLe	SHAPE_Length	SHAPE_Area	geometry
0	1	PHILLIPS WEST	1.066925e+07	14403.885934	0.045801	0.000113	MULTIPOLYGON (((-93.26258 44.96091, -93.26258 ...
1	2	DOWNTOWN WEST	2.075613e+07	19220.602541	0.063671	0.000220	MULTIPOLYGON (((-93.26011 44.98300, -93.26010 ...
2	3	DOWNTOWN EAST	1.025499e+07	13436.601356	0.045179	0.000109	MULTIPOLYGON (((-93.24499 44.97893, -93.24499 ...
3	4	VENTURA VILLAGE	1.263526e+07	16988.532717	0.059590	0.000134	MULTIPOLYGON (((-93.24958 44.96630, -93.24951 ...
4	5	SUMNER - GLENWOOD	5.741860e+06	11065.343364	0.035535	0.000061	MULTIPOLYGON (((-93.28830 44.98904, -93.28830 ...

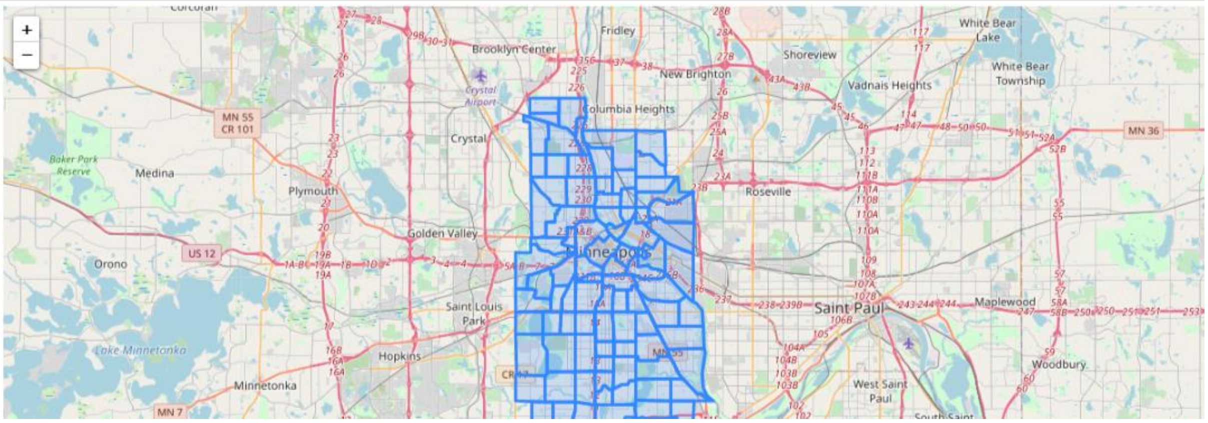
Below is the map we created, with borders of all Minneapolis neighborhoods.

```
map_minneapolis = folium.Map(location=[latitude, longitude], zoom_start=10)

#show neighborhoods on the map

folium.GeoJson(data = df_neighb, name='Neighborhoods', tooltip=folium.features.GeoJsonTooltip(fields=['BDNAME'], localize=True)).add_to(map_minneapolis)
map_minneapolis
```

Out[19]:



We also need to extract latitude and longitude values for each neighborhood (to further work with Foursquare API)

The geojson file contained only the geometry, so we needed to first extract centroids

In [23]:
```
centroids = pd.DataFrame(df_neighb['geometry'].centroid)
header = ['Centroid']
centroids.columns = header
centroids.head()
```

Out[23]:

	Centroid
0	POINT (-93.26734 44.95386)
1	POINT (-93.27012 44.97784)
2	POINT (-93.25411 44.97606)
3	POINT (-93.25790 44.96278)
4	POINT (-93.29160 44.98403)

Now we can add the column with centroids to our neighborhood dataframe

In [24]:
```
neighb_w_centroids = df_neighb.join(centroids, sort = False)
neighb_w_centroids.head()
```

Out[24]:

	FID	BDNAME	Shape_STAr	Shape_STLe	SHAPE_Length	SHAPE_Area	geometry	Centroid
0	1	PHILLIPS WEST	1.066925e+07	14403.885934	0.045801	0.000113	MULTIPOLYGON (((-93.26258 44.96091, -93.26258 ...	POINT (-93.26734 44.95386)
1	2	DOWNTOWN WEST	2.075613e+07	19220.602541	0.063671	0.000220	MULTIPOLYGON (((-93.26011 44.98300, -93.26010 ...	POINT (-93.27012 44.97784)
2	3	DOWNTOWN EAST	1.025499e+07	13436.601356	0.045179	0.000109	MULTIPOLYGON (((-93.24499 44.97893, -93.24499 ...	POINT (-93.25411 44.97606)
3	4	VENTURA VILLAGE	1.263526e+07	16988.532717	0.059590	0.000134	MULTIPOLYGON (((-93.24958 44.96630, -93.24951 ...	POINT (-93.25790 44.96278)
4	5	SUMNER - GLENWOOD	5.741860e+06	11065.343364	0.035535	0.000061	MULTIPOLYGON (((-93.28830 44.98904, -93.28830 ...	POINT (-93.29160 44.98403)

and then from the cetroid points extract the latitude and longitude values:

```
In [48]: neighb_latlng = neighb_w_centroids.join(lat_lng, sort = False)
         neighb_latlng.head()
```

Out[48]:

	FID	BDNAME	Shape_STAr	Shape_STLe	SHAPE_Length	SHAPE_Area	geometry	Centroid	Latitude	Longitude
0	1	PHILLIPS WEST	1.066925e+07	14403.885934	0.045801	0.000113	MULTIPOLYGON (((-93.26258 44.96091, -93.26258 ...	POINT (-93.26734 44.95386)	NaN	NaN
1	2	DOWNTOWN WEST	2.075613e+07	19220.602541	0.063671	0.000220	MULTIPOLYGON (((-93.26011 44.98300, -93.26010 ...	POINT (-93.27012 44.97784)	NaN	NaN
2	3	DOWNTOWN EAST	1.025499e+07	13436.601356	0.045179	0.000109	MULTIPOLYGON (((-93.24499 44.97893, -93.24499 ...	POINT (-93.25411 44.97606)	NaN	NaN
3	4	VENTURA VILLAGE	1.263526e+07	16988.532717	0.059590	0.000134	MULTIPOLYGON (((-93.24958 44.96630, -93.24951 ...	POINT (-93.25790 44.96278)	NaN	NaN
4	5	SUMNER - GLENWOOD	5.741860e+06	11065.343364	0.035535	0.000061	MULTIPOLYGON (((-93.28830 44.98904, -93.28830 ...	POINT (-93.29160 44.98403)	NaN	NaN

Now we can populate the empty latitude and longitude columns with data from the centroid column

```
In [72]: for i, row in enumerate(neighb_latlng.values):
             neighb_latlng['Latitude'][i] = neighb_latlng['Centroid'][i].y
             neighb_latlng['Longitude'][i] = neighb_latlng['Centroid'][i].x
             #looping through centroids to extract latitude and longitude
```

Our final dataframe that was used to work with the Foursquare API:

```
In [73]: neighb_latlng.head() #check the results
```

Out[73]:

	FID	BDNAME	Shape_STAr	Shape_STLe	SHAPE_Length	SHAPE_Area	geometry	Centroid	Latitude	Longitude
0	1	PHILLIPS WEST	1.066925e+07	14403.885934	0.045801	0.000113	MULTIPOLYGON (((-93.26258 44.96091, -93.26258 ...	POINT (-93.26734 44.95386)	44.9539	-93.2673
1	2	DOWNTOWN WEST	2.075613e+07	19220.602541	0.063671	0.000220	MULTIPOLYGON (((-93.26011 44.98300, -93.26010 ...	POINT (-93.27012 44.97784)	44.9778	-93.2701
2	3	DOWNTOWN EAST	1.025499e+07	13436.601356	0.045179	0.000109	MULTIPOLYGON (((-93.24499 44.97893, -93.24499 ...	POINT (-93.25411 44.97606)	44.9761	-93.2541
3	4	VENTURA VILLAGE	1.263526e+07	16988.532717	0.059590	0.000134	MULTIPOLYGON (((-93.24958 44.96630, -93.24951 ...	POINT (-93.25790 44.96278)	44.9628	-93.2579
4	5	SUMNER - GLENWOOD	5.741860e+06	11065.343364	0.035535	0.000061	MULTIPOLYGON (((-93.28830 44.98904, -93.28830 ...	POINT (-93.29160 44.98403)	44.984	-93.2916

Let's create a new dataframe with only the neighborhood names and corresponding lat/lng values, which will be used for working with Foursquare

```
In [74]: neighborhood_ll = neighb_latlng[['BDNAME','Latitude','Longitude']]
         neighborhood_ll.rename(columns={'BDNAME':'Neighborhood'}, inplace=True)
         neighborhood_ll.head()
```

Out[74]:

	Neighborhood	Latitude	Longitude
0	PHILLIPS WEST	44.9539	-93.2673
1	DOWNTOWN WEST	44.9778	-93.2701
2	DOWNTOWN EAST	44.9761	-93.2541

X Cancel ✓ Capture

We will then examine crime data in each neighborhood (also using data available at OpenData; the most recent information is for 2018: http://opendata.minneapolismn.gov/datasets/police-incidents-2018/data). We will identify the types of crimes that worry the family the most and then visualize the corresponding data on the map:

Let's make a new dataframe with only the relevant data (neighborhood name, type of crime, and lat/lng values)

```
In [7]: neighborhoods_crime = crime_2018[['Neighborhood','Description','Lat', 'Long']]
        neighborhoods_crime = neighborhoods_crime.rename(columns={'Description':'Type of Crime', 'Lat':'Latitude', 'Long': 'Longitude'})
        neighborhoods_crime.head()
```

Out[7]:

	Neighborhood	Type of Crime	Latitude	Longitude
0	REGINA	Burglary Of Business	44.919705	-93.269608
1	MARCY HOLMES	Motor Vehicle Theft	44.986351	-93.237514
2	MARCY HOLMES	Theft From Motr Vehc	44.985314	-93.233748
3	NORTH LOOP	Shoplifting	44.982576	-93.268417
4	NORTHEAST PARK	Other Theft	45.003639	-93.228834

Data prepared for the map:

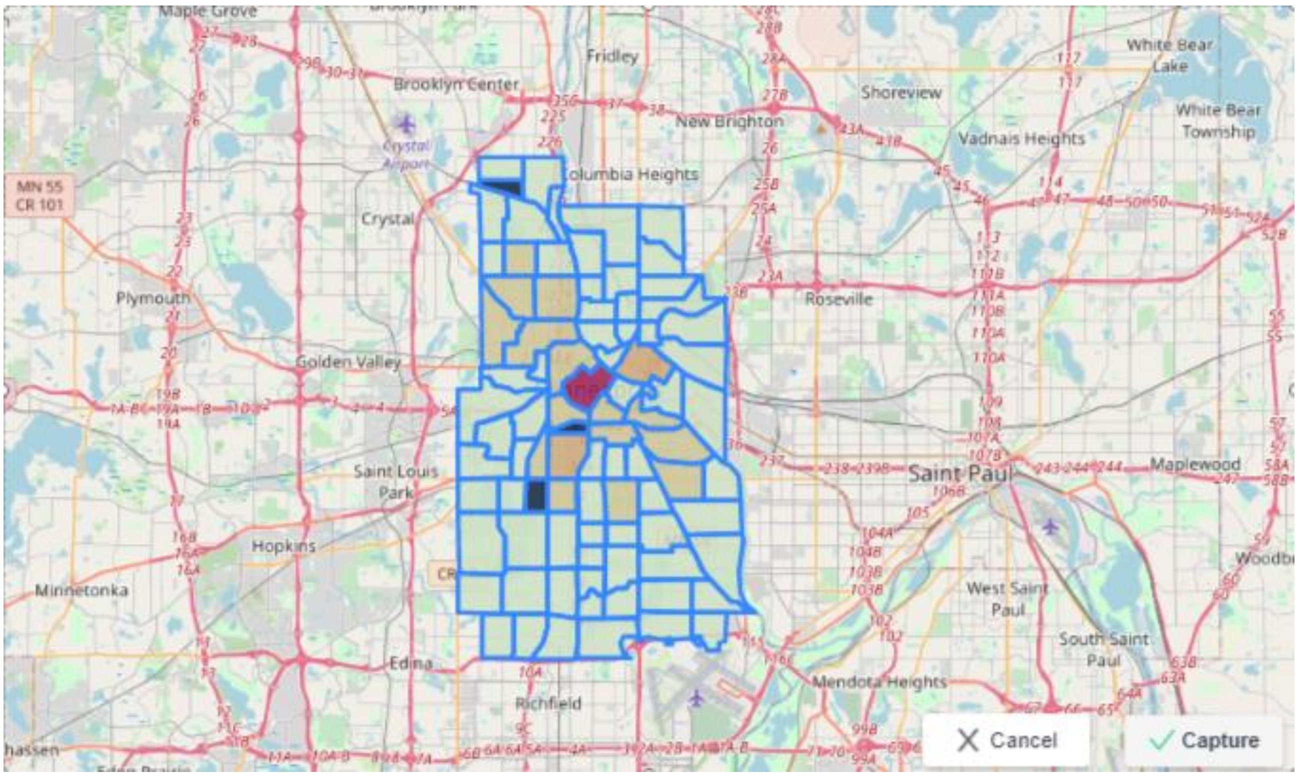
First we need t o create a new dataframe with generalized crime data per neighborhood

In [20]:
```
neigh_crime_counts = neighborhoods_crime['Neighborhood'].value_counts().to_frame()
crime_per_neighborhood = neigh_crime_counts.reset_index().rename(columns={'index': 'Neighborhood', 'Neighborhood':'Count'})
crime_per_neighborhood.head()
```

Out[20]:

	Neighborhood	Count
0	DOWNTOWN WEST	568
1	WHITTIER	202
2	MARCY HOLMES	201
3	JORDAN	173
4	LOWRY HILL EAST	168

The map itself:



As we can see, only several neighborhoods have significant crime levels, so, in making the final decision, this factor will only be used together with the list of popular venues.

We will use Matplotlib to visualize crime rates, so let’s prepare the dataframe:

Out[129]:

	Neighborhood	Arson	Aslt-great Bodily Hm	Aslt-sgnfcnt Bdly Hm	Asslt W/dngrs Weapon	Burglary Of Dwelling	Crim Sex Cond-rape	Motor Vehicle Theft	Murder (general)	Other Theft	Robbery Of Person	Theft From Motr Vehc	Theft From Person
1	MARCY HOLMES	0	0	0	0	0	0	1	0	0	0	0	0
2	MARCY HOLMES	0	0	0	0	0	0	0	0	0	0	1	0
4	NORTHEAST PARK	0	0	0	0	0	0	0	0	1	0	0	0
5	ST. ANTHONY EAST	0	0	0	0	0	0	0	0	0	0	1	0
6	ELLIOT PARK	0	0	0	0	0	0	1	0	0	0	0	0

Group rows by neighborhood and by taking the mean of category occurence frequency

In [130]:
```
crime_grouped =crime_onehot.groupby('Neighborhood').mean().reset_index()
crime_grouped.head()
```

Out[130]:

	Neighborhood	Arson	Aslt-great Bodily Hm	Aslt-sgnfcnt Bdly Hm	Asslt W/dngrs Weapon	Burglary Of Dwelling	Crim Sex Cond-rape	Motor Vehicle Theft	Murder (general)	Other Theft	Robbery Of Person	Theft From Motr Vehc	Theft From Person
0	ARMATAGE	0.0	0.000000	0.0	0.000000	0.190476	0.000000	0.000000	0.0	0.190476	0.000000	0.619048	0.000000
1	AUDUBON PARK	0.0	0.000000	0.0	0.000000	0.485714	0.028571	0.100000	0.0	0.242857	0.000000	0.128571	0.014286
2	BANCROFT	0.0	0.000000	0.0	0.000000	0.333333	0.000000	0.190476	0.0	0.238095	0.000000	0.238095	0.000000
3	BELTRAMI	0.0	0.000000	0.0	0.076923	0.076923	0.000000	0.000000	0.0	0.230769	0.000000	0.615385	0.000000
4	BOTTINEAU	0.0	0.043478	0.0	0.043478	0.173913	0.000000	0.217391	0.0	0.173913	0.043478	0.260870	0.043478

Set neighborhood as index

In [131]:
```
crime_grouped.set_index('Neighborhood', inplace=True)
```

X Cancel ✓ Capture

Methodology

In this project we will direct our efforts on detecting neighborhoods of Minnesota that have low crime rates, particularly those with low number of crimes that involve individuals (as opposed to businesses/establishments), as well as neighborhoods that offer infrastructure that is appealing to the stakeholder (i.e. the family). The most suitable neighborhoods will have some green zones (parks, trails, playgrounds), grocery shops or farmers' markets, as well as a sufficient number of cafes/restaurants.

In first step we will collect the required data:

- We gathered data on crime rates for each neighborhood, cleaning all unnecessary data from the dataset
- We extracted latitude and longitude values for each neighborhood (to use with foursquare API);

  Now

- Using foursquare API we will get location and type (category) of venues in each neighborhood of Minneapolis.

In [84]:
```
minneapolis_venues.head()
```

Out[84]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	PHILLIPS WEST	44.953863	-93.267343	American Swedish Institute	44.954641	-93.265787	Museum
1	PHILLIPS WEST	44.953863	-93.267343	FIKA	44.954555	-93.265717	Scandinavian Restaurant
2	PHILLIPS WEST	44.953863	-93.267343	Minneapolis Institute of Art	44.958554	-93.273284	Art Museum
3	PHILLIPS WEST	44.953863	-93.267343	Taqueria La Hacienda	44.948605	-93.271181	Mexican Restaurant
4	PHILLIPS WEST	44.953863	-93.267343	Children's Theatre Company	44.957949	-93.273466	Theater

Check how many unique venue categories we have

In [86]:
```
print('There are {} uniques categories.'.format(len(minneapolis_venues['Venue Category'].unique())))
```

```
There are 253 uniques categories.
```

Most common venues for every neighborhood:

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

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

neighborhoods_venues_sorted.head()
```

Out[91]:

	Neighborhood	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	ARMATAGE	Pizza Place	Trail	Restaurant	Arts & Crafts Store	Flower Shop	Mexican Restaurant	Pet Store	French Restaurant	Supermarket	Chinese Restaurant
1	AUDUBON PARK	Park	Bakery	Pharmacy	Brewery	Pizza Place	Coffee Shop	Thai Restaurant	Hotel Bar	Organic Grocery	Liquor Store
2	BANCROFT	Park	Food & Drink Shop	Coffee Shop	Pharmacy	BBQ Joint	Mexican Restaurant	Food Truck	South American Restaurant	Chinese Restaurant	Furniture / Home Store
3	BELTRAMI	Brewery	BBQ Joint	Dive Bar	Coffee Shop	Yoga Studio	Park	Cosmetics Shop	Donut Shop	Event Space	Café
4	BOTTINEAU	Dive Bar	Coffee Shop	Theme Restaurant	Ice Cream Shop	American Restaurant	Café	Pet Store	Bookstore	Speakeasy	New American Restaurant

Now we can also prepare the data for visualization with Matplotlib

Since we cannot display all 253 venue categories on a plot (and keep it visually appealing and easy t
neighborhoods

[269]:

```
cols_to_remove = set()

for i in mn_venues:
    if mn_venues[i].mean() < 0.01: #Let's leave only those venue categories that have at L
        cols_to_remove.add(i)

for i in cols_to_remove:
    mn_venues.drop(i, axis=1, inplace=True)
```

[270]:

```
mn_venues.head() #a preview
```

Out[270]:

	American Restaurant	Art Gallery	Asian Restaurant	Bakery	Bar	Brewery	Café	Chinese Restaurant	
Neighborhood									
ARMATAGE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.043478	0.
AUDUBON PARK	0.033333	0.000000	0.000000	0.100000	0.000000	0.066667	0.000000	0.000000	0.
BANCROFT	0.033333	0.033333	0.000000	0.033333	0.033333	0.000000	0.000000	0.033333	0.
BELTRAMI	0.033333	0.033333	0.033333	0.000000	0.000000	0.133333	0.033333	0.000000	0.
BOTTINEAU	0.066667	0.033333	0.000000	0.000000	0.033333	0.033333	0.066667	0.000000	0.

5 rows × 23 columns

The second step in our analysis will be finding the most popular venues in each neighborhood and clustering the neighborhoods based on this information (using KMeans clustering).

Group rows by neighborhood and by taking the mean of category occurence frequency

```
In [88]: minneapolis_grouped = minneapolis_onehot.groupby('Neighborhood').mean().reset_index()
minneapolis_grouped.head()
```

Out[88]:

	Neighborhood	ATM	Accessories Store	Acupuncturist	Adult Boutique	Advertising Agency	African Restaurant	American Restaurant	Antique Shop	Arcade	...	Vegetarian / Vegan Restaurant	Video Store	Vietnamese Restaurant	Warehouse Store
0	ARMATAGE	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.043478	0.000000	0.0
1	AUDUBON PARK	0.0	0.000000	0.0	0.0	0.0	0.0	0.033333	0.0	0.0	...	0.0	0.033333	0.033333	0.0
2	BANCROFT	0.0	0.033333	0.0	0.0	0.0	0.0	0.033333	0.0	0.0	...	0.0	0.033333	0.000000	0.0
3	BELTRAMI	0.0	0.000000	0.0	0.0	0.0	0.0	0.033333	0.0	0.0	...	0.0	0.000000	0.033333	0.0
4	BOTTINEAU	0.0	0.000000	0.0	0.0	0.0	0.0	0.066667	0.0	0.0	...	0.0	0.000000	0.033333	0.0

5 rows × 254 columns

Clustered neighborhoods:

```
minneapolis_merged.head()
```

Out[101]:

	Neighborhood	Latitude	Longitude	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	PHILLIPS WEST	44.9539	-93.2673	3	Mexican Restaurant	Vietnamese Restaurant	American Restaurant	Performing Arts Venue	Theater	Sandwich Place	Scandinavian Restaurant	Bar	Taco Place	College Arts Building
1	DOWNTOWN WEST	44.9778	-93.2701	4	Theater	Italian Restaurant	New American Restaurant	Music Venue	Baseball Stadium	Restaurant	Bar	Sushi Restaurant	Pedestrian Plaza	Donut Shop
2	DOWNTOWN EAST	44.9761	-93.2541	4	Bar	Hotel	Park	Japanese Restaurant	Yoga Studio	Scenic Lookout	Public Art	Non-Profit	Music Venue	Music School
3	VENTURA VILLAGE	44.9628	-93.2579	2	Coffee Shop	Park	Grocery Store	Pharmacy	Theater	Café	Flower Shop	Bridge	Shopping Mall	Breakfast Spot
4	SUMNER - GLENWOOD	44.984	-93.2916	4	Brewery	Thrift / Vintage Store	Farmers Market	Food Truck	Deli / Bodega	Business Service	Café	Fish Market	Bar	Clothing Store

Cluster example:

Cluster 0

```
n [110]: minneapolis_merged.loc[minneapolis_merged['Cluster Labels'] == 0, minneapolis_merged.columns[[0] + list(range(3, minneapolis_merged.shape[1]))]]
```

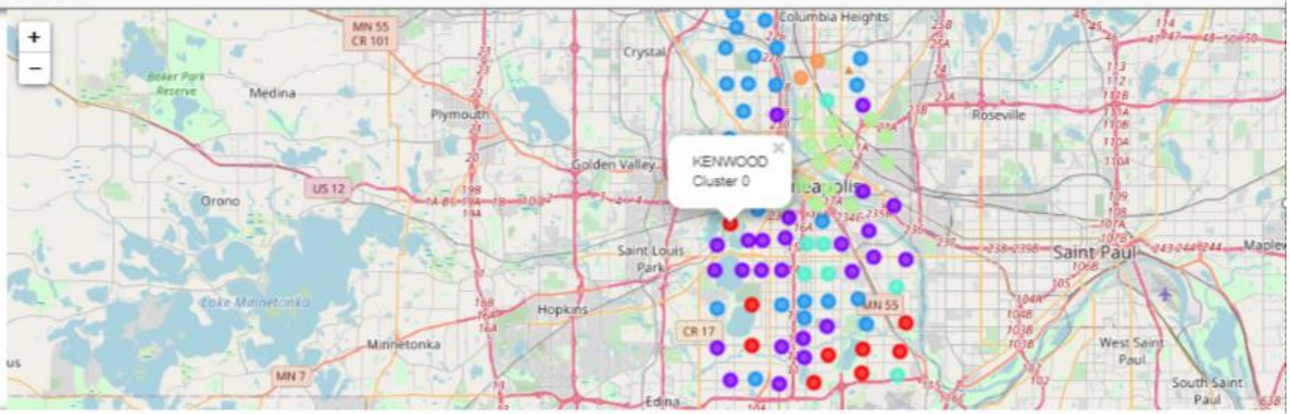
Out[110]:

	Neighborhood	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 Cc
12	HIAWATHA	0	Park	Trail	Scenic Lookout	Pizza Place	History Museum	Brewery	Bus
28	MINNEHAHA	0	Park	Trail	Historic Site	Liquor Store	Breakfast Spot	Shoe Store	Seafood Res
29	DIAMOND LAKE	0	Food Truck	Pet Store	Gas Station	Dog Run	Park	Grocery Store	Asian Res
33	HALE	0	Lake	Farmers Market	Restaurant	Sculpture Garden	Asian Restaurant	Furniture / Home Store	Thrift / Vintag
34	KEEWAYDIN	0	Park	Beach	Yoga Studio	Grocery Store	Restaurant	Playground	Mexican Res
36	LYNNHURST	0	Italian Restaurant	Park	Sporting Goods Shop	Taco Place	Bike Rental / Bike Share	French Restaurant	Bo
46	WENONAH	0	Mexican Restaurant	Yoga Studio	Dog Run	Baseball Field	Sandwich Place	Asian Restaurant	Fast Food Res
55	EAST HARRIET	0	Playground	Park	Garden	Snack Place	Carpet Store	Beach	Ce
73	KENWOOD	0	Lake	Beach	Park	Trail	Café	Bookstore	American Res

Map with clusters:

```
map_clusters
```

Out[102]:

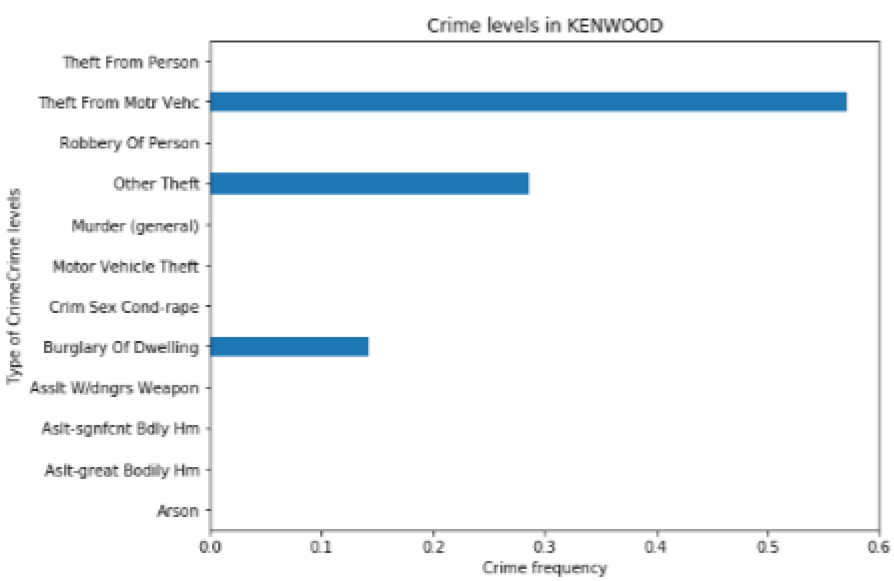


In third and final step we will focus on most promising neighborhoods in each cluster and choose the most suitable neighborhoods by visualizing crime and venue data for the most promising neighborhoods (visualization will help to decide on the final neighborhood recommendations).

Analyzing crime levels in a neighborhood chosen from a cluster:

Crime

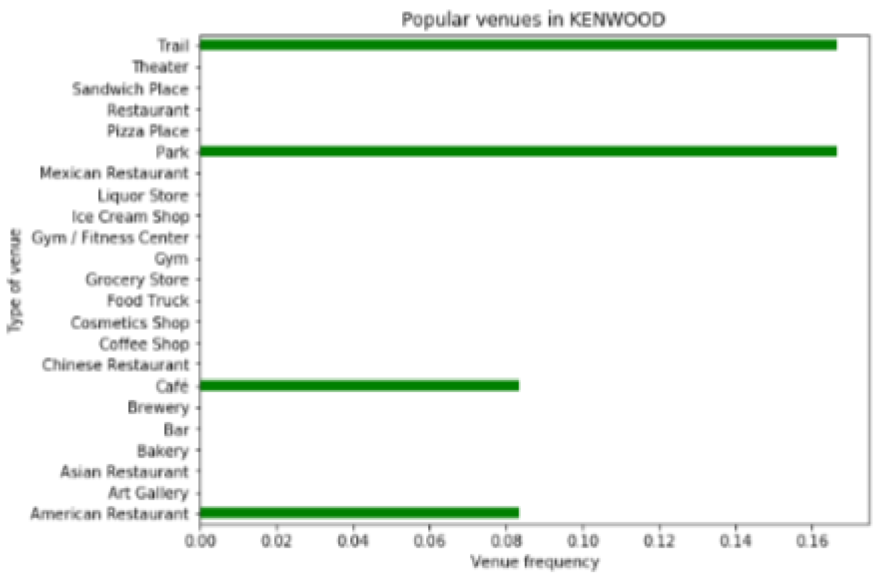
```
[250]: kenwood = crime_grouped.loc['KENWOOD']
       kenwood.plot(kind='barh', figsize = (8, 6))
       plt.title('Crime levels in KENWOOD')
       plt.ylabel('Type of Crime''Crime levels')
       plt.xlabel('Crime frequency')
       plt.show()
```



Analyzing popular venues in the same neighborhood:

Venues

[271]:
```
kenwood_venues = mn_venues.loc['KENWOOD']
kenwood_venues.plot(kind='barh', figsize = (8, 6), color='green')
plt.title('Popular venues in KENWOOD')
plt.ylabel('Type of venue')
plt.xlabel('Venue frequency')
plt.show()
```



Results and Discussion

Cluster analysis showed that two (2) out of six (6) formed clusters did not meet the requirements of the stakeholders, so neighborhoods from those clusters were not considered.

Out of other four (4) clusters, a total of five (5) neighborhoods were chosen as suitable options for the stakeholders.

The first cluster (cluster 0) includes relatively quiet neighborhoods with access to green zones. Two neighborhoods (**Kenwood and Keewaydin**) were identified as suitable options. These neighborhoods are generally more family-friendly and have low crime rates. However, they are both farther from Downtown East, which is the neighborhood where one of the parents in the family plans to work. So, the stakeholders will need to consider whether the commute time is acceptable.

The third cluster (cluster 2) also has neighborhoods that are close to parks and have a good selection of restaurants/shops. However, most neighborhoods in this cluster also have relatively high crime rates (because they are closer to downtown), so only one neighborhood was recommended - **Harrison**.

In the fifth cluster (cluster 4), two neighborhoods were identified as suitable (both have access to a park, in addition to various options to eat out); both have sufficiently low crime rates, so both were recommended - **St.Anthony West and Beltrami**.

Conclusion

Purpose of this project was to identify neighborhoods in Minneapolis that the stakeholder family could relocate to. Using data from Foursquare API we identified the most common venue categories in each neighborhood and were able to cluster Minneapolis neighborhoods using that information. Each cluster was analyzed to identify neighborhoods that suited the family's preferences (if such neighborhoods were present in the cluster). The chosen neighborhoods were compared using data on crime levels and the most popular venues.

The final decision on optimal neighborhood will be made by the stakeholders (i.e. the family) based on specific characteristics of the recommended neighborhoods.