Select the Ideal Community to Launch a Local Sharing Service App

By: Yao Xie
February 2020

1. Introduction

1.1 Background

Chicago-based startup firm - Enso Street is developing a local sharing platform that focuses on family tools and equipment. The firm is planning to launch the beta version of the platform in March 2020. It is of paramount importance for the firm to select the most suitable community in Chicago to roll out its service. The success in this effort would help the startup to find product-market fit and gain meaningful operating metrics before their next funding event. The founding team engaged us to carry out the research and analytics to identify the three target communities in Chicago.

1.2 Problem

Given the fact that the founding team has already completed the customer segmentation and determined the ideal marketing personas are financially sound millennials that recently started their families. Data that might contribute to understanding the suitability of each community might include the education level, unemployment rate, per capita income and types of venues that implies what type of community it is. This analysis aims to examine both the demographic and geographic factors of each community in order to make the right recommendation.

1.3 Target Audience

Obviously, the founding team of Enso Street is the stakeholder. They will select the right community to debut their local sharing services based on this analysis. Meanwhile, Enso Street's future investors might be interested in understanding the team's methodology and process in selecting target locations to launch their services. A thorough and analytical approach would enhance the team's creditability in capital raising events.

2. Data Acquisition and Cleaning

2.1 Data Sources

In order to perform our analysis, we need to obtain the data listed below:

- **Complete List of Communities in Chicago.** The project scope is confined to the city of Chicago. Therefore, we obtained the entire community list from the Chicago Data Portal (https://data.cityofchicago.org/Health-Human-Services/Uptown-Census-Data/vdfh-mxit)

- **Socioeconomic Data.** The firm's marketing personas are home improvement/maintenance DIYers and financially sound millennials that recently started their families. In this light, the socioeconomic indicators for each Chicago community are of critical importance for our analysis. The data we acquired is from the Chicago Data Portal (https://data.cityofchicago.org/Health-Human-Services/Uptown-Census-Data/vdfh-mxit)

- **Geographic data:**
    a. **Coordinates of Communities in Chicago.** The data is required to help us plot maps and get venue data for each individual community. We employed the geocoder package offered by ArcGIS (https://developers.arcgis.com/features/geocoding/) to obtain the coordinates required to complete our analysis.
    b. **Venue Data.** The firm will compete, to some extent, with stores (i.e., Home Depot, ACE Hardware) that provide tools and equipment rental services. Relevant geographic data provided by Foursquare can help us to identify the communities that have a relatively low density of such stores. We can also gain other insights about the communities based on the venue numbers and types. For instance, a community with a high density of parks and coffee shops typically would have a relatively large number of young families. We obtained the data through Foursquare's developer API (https://foursquare.com/developers/apps).

2.2 Data Cleaning

After loading socioeconomic data into a data-frame, we first deleted rows that have no 'Community Area Number' information. This allowed us to include each individual community data and exclude any aggregate data, which offered no value in our analysis. We then decided to drop the 'Community Area Number' column in our data-frame so that we kept the columns that are useful for our analysis. We also noticed that unnecessary space after 'Per Capita Income', the name of the last column. We deleted the space by using the data-frame rename function.

3. Methodology

We employed a portfolio of methods in order to complete our comprehensive analysis. We are going to discuss them one by one below.

3.1 Dataframe Filtering

There are 77 communities in Chicago for us to consider. Our first selection criteria are socioeconomic factors. For each socioeconomic indicator, we only selected communities with value better than the group average value. For example, a community with a lower hardship index is more suitable for our purposes. Therefore, we filtered the communities that have a hardship value below the group average. Meanwhile, a community with a higher per capita income index is more suitable for our purposes. In this light, we filtered the communities that have per capita income above the group average. The process allowed us to narrow down the list of communities that are suitable and subject to our further analysis.

```
#let's select the communities with above average socioeconomic indicators

pct_poverty = df_cleaned['PERCENT HOUSEHOLDS BELOW POVERTY']
pct_hsdiploma = df_cleaned['PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA']
pct_crowded = df_cleaned['PERCENT OF HOUSING CROWDED']
pct_unemployed = df_cleaned['PERCENT AGED 16+ UNEMPLOYED']
hardship = df_cleaned['HARDSHIP INDEX']
pct_agegroup = df_cleaned['PERCENT AGED UNDER 18 OR OVER 64']
pc_income = df_cleaned['PER CAPITA INCOME']

df_filtered = df_cleaned.loc[(pct_poverty    < pct_poverty.mean())
                            &(pct_hsdiploma  < pct_hsdiploma.mean())
                            &(pct_crowded    < pct_crowded.mean())
                            &(pct_unemployed < pct_unemployed.mean())
                            &(hardship       < hardship.mean())
                            &(pct_agegroup   < pct_agegroup.mean())
                            &(pc_income      > pc_income.mean())]

df_filtered.head(15)
```

3.2 Geocoding

We employed the geocoding API provided by ArcGIS, the world's leading mapping and location analytics platform, to retrieve latitude and longitude data for each community. We then stored the coordinates data into a data-frame named df_geodata.

```
# define a function to get coordinates

def get_latlng(community):
    # initialize your variable to None
    lat_lng_coords = None
    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Chicago, Illinois'.format(community))
        lat_lng_coords = g.latlng
    return lat_lng_coords
```

```
# create temporary dataframe to populate the coordinates into Latitude and Longitude

import geocoder
coords = [get_latlng(community) for community in df_filtered['COMMUNITY AREA NAME'].tolist()]
df_coords = pd.DataFrame(coords, columns=['COMMUNITY LATITUDE', 'COMMUNITY LONGITUDE'])
df_coords.head(15)
```

```
# merge the coordinates into the df_filtered dataframe and name the new dataframe df_geodata

df_coords['COMMUNITY AREA NAME'] = df_filtered['COMMUNITY AREA NAME'].tolist()

df_geodata = df_coords.reindex(columns= ['COMMUNITY AREA NAME', 'COMMUNITY LATITUDE', 'COMMUNITY LONGITUDE'])

df_geodata.head(15)
```

3.3 One Hot Encoding

This process enabled us to convert categorical variables into a form that could be utilized by machine learning algorithms to provide optimal predictions. All unique items under Venue Category were one-hot encoded before K-means clustering algorithm was employed.

```
#now that we have the nearby venues information, let's employ one hot encoding to prepare the data for further analysis

chicago_onehot = pd.get_dummies(chicago_venues[['Venue Category']], prefix='', prefix_sep='')

chicago_onehot.head()
```

```python
#add community column back to dataframe and move it to the first column
chicago_onehot['COMMUNITY AREA NAME'] = chicago_venues['COMMUNITY AREA NAME']
fixed_columns = [chicago_onehot.columns[-1]] + list(chicago_onehot.columns[:-1])
chicago_onehot = chicago_onehot[fixed_columns]

chicago_onehot.head()
```

```python
#next, we are going to group rows by community and by taking the means of the frequency of occurence of each category
chicago_grouped = chicago_onehot.groupby('COMMUNITY AREA NAME').mean().reset_index()

chicago_grouped.head()
```

3.4 10 Most Common Venues

We selected the 10 most common venues ONLY for our analysis. We then created a new data-frame named comm_venues_sorted to store all the values, before we conducted our K-means clustering analysis.

```python
#now let's put the output into a pandas dataframe
#first, we are going to define a function to sort the venues in descending order

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]
```

```python
#Let's create a new dataframe and display the top 10 venues for each neighborhood.

num_top_venues = 10

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

```python
# create columns according to number of top venues
import numpy as np

columns = ['COMMUNITY AREA NAME']

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))
```

```python
# create a new dataframe
comm_venues_sorted = pd.DataFrame(columns=columns)
comm_venues_sorted['COMMUNITY AREA NAME'] = chicago_grouped['COMMUNITY AREA NAME']

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

print(comm_venues_sorted.head())
```

3.5 K-Means Clustering

We employed the unsupervised learning algorithm K-Means clustering twice in our analysis. We first trained our socioeconomic data and clustered them into 3 buckets.

```python
# use k-means to cluster the neighborhood into 3 clusters based on all socioeconomic data
kclusters = 3
df_filtered_clustering = df_filtered.drop('COMMUNITY AREA NAME', 1)
```

```python
#run k-means clustering
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(df_filtered_clustering)
```

```python
#add clustering labels to comm_venues_sorted dataframe

df_filtered.insert(0, 'Demo Cluster Labels', kmeans.labels_)

df_filtered.head()
```

We then trained our venue data using the same algorithm. Based on the number of communities, we employed a cluster number of 3. After the clustering is complete, we added the cluster label information to the data-frame for further analysis.

```
# use k-means to cluster the neighborhood into 3 clusters
kclusters = 3
chicago_grouped_clustering = chicago_grouped.drop('COMMUNITY AREA NAME', 1)
```

```
#run k-means clustering
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(chicago_grouped_clustering)
```

```
#add clustering labels to comm_venues_sorted dataframe

comm_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
#comm_venues_sorted = comm_venues_sorted.drop(columns=['Cluster Labels'])
```

4. Results

The communities are categorized into three different clusters. We then visualized our clustered communities on a Leaflet map using Folium with different colors.

```
#create map
chicago_coort = (41.8336, -87.8720)
latitude = chicago_coort[0]
longitude = chicago_coort[1]

map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)
```

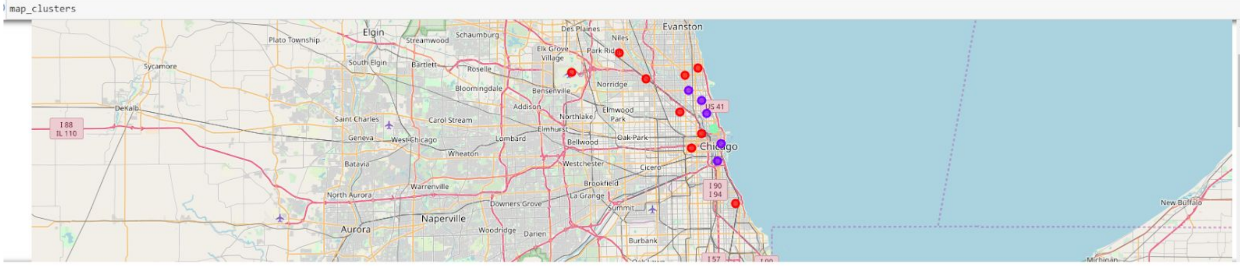
```
# set color scheme for the clusters

import matplotlib.cm as cm
import matplotlib.colors as colors

x = np.arange(kclusters)
ys = [i + x + (i * x) ** 2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]
```

```
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(chicago_merged['COMMUNITY LATITUDE'], chicago_merged['COMMUNITY LONGITUDE'],
                                  chicago_merged['COMMUNITY AREA NAME'], chicago_merged['Cluster Labels'].astype('int64')):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster - 1],
        fill=True,
        fill_color=rainbow[cluster - 1],
        fill_opacity=0.7).add_to(map_clusters)
```

```
map_clusters
```



5. Discussion

By examing each individual cluster, we found that Cluster No. 3 is the most family-friendly community based on the types of 10 most common venues.

13 out of 15 communities are clustered in #1, indicating a high level of homogeneity in these Chicago communities. This cluster has bars, coffee shops, and restaurants topping the most common venues. It might imply that these communities have a large population of young professionals.

In [166]:
```python
#examine each cluster
#Cluster 1

chicago_merged.loc[chicago_merged['Cluster Labels'] == 0, chicago_merged.columns[[0] + list(range(4, chicago_merged.shape[1]))]]
```

Out[166]:

	COMMUNITY AREA NAME	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	Lincoln Square	Bar	Hot Dog Joint	Liquor Store	Bus Station	Convenience Store	Pizza Place	Korean Restaurant	Grocery Store	Football Stadium	Food Truck
1	North Center	Bar	Coffee Shop	Bank	Dive Bar	Pub	Pharmacy	Mobile Phone Shop	Boutique	American Restaurant	Latin American Restaurant
2	Lake View	Café	Japanese Restaurant	Coffee Shop	Bakery	Bagel Shop	Sandwich Place	Performing Arts Venue	Gym / Fitness Center	Pizza Place	Pharmacy
3	Lincoln Park	Pizza Place	Sandwich Place	Coffee Shop	Gym / Fitness Center	Bar	Taco Place	Breakfast Spot	Mexican Restaurant	Vietnamese Restaurant	Art Gallery
4	Near North Side	Gym / Fitness Center	Gym	Restaurant	Spa	American Restaurant	Bar	Breakfast Spot	Sandwich Place	Pub	Pool
6	Jefferson Park	Bar	Pharmacy	Video Store	Coffee Shop	Ice Cream Shop	Park	Chinese Restaurant	Sushi Restaurant	Supermarket	Restaurant
7	Logan Square	Bar	Cocktail Bar	Mexican Restaurant	Pizza Place	Restaurant	Ice Cream Shop	Discount Store	Donut Shop	Coffee Shop	Fast Food Restaurant
8	West Town	Sandwich Place	Pizza Place	Bar	Sushi Restaurant	Yoga Studio	Theater	Deli / Bodega	Pub	Burger Joint	Coffee Shop
10	Loop	Harbor / Marina	Park	Boat or Ferry	Ice Cream Shop	Concert Hall	Sushi Restaurant	Museum	Coffee Shop	Garden	Sandwich Place
11	Near South Side	Chinese Restaurant	Pizza Place	Dessert Shop	Bubble Tea Shop	Bakery	Rental Car Location	Nightclub	Grocery Store	Caribbean Restaurant	Candy Store
12	Hyde Park	Coffee Shop	Bookstore	Sandwich Place	Train Station	Pharmacy	Shipping Store	Bubble Tea Shop	Café	Rental Car Location	Gym
13	O'Hare	Airport Service	Coffee Shop	Snack Place	Airport Lounge	American Restaurant	Accessories Store	Bar	Tea Room	Tex-Mex Restaurant	Dessert Shop
14	Edgewater	Mexican Restaurant	Bus Station	Indian Restaurant	Sushi Restaurant	Asian Restaurant	Antique Shop	Bakery	Deli / Bodega	Yoga Studio	Convenience Store

Cluster No. 2 has only one community. From the venue type information, this community Near West Side might have a large population of students.

In [167]:
```python
#Cluster 2

chicago_merged.loc[chicago_merged['Cluster Labels'] == 1, chicago_merged.columns[[0] + list(range(4, chicago_merged.shape[1]))]]
```

Out[167]:

	COMMUNITY AREA NAME	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
9	Near West Side	Coffee Shop	Mexican Restaurant	Sandwich Place	College Gym	Baseball Field	Fast Food Restaurant	Train Station	Flower Shop	Fish & Chips Shop	Filipino Restaurant

Finally, there is one community, namely Edison Park, in Cluster #3. The three most venues are theaters, neighborhoods, and parks, all of which are highly popular among families.

In [168]:
```python
#Cluster 3

chicago_merged.loc[chicago_merged['Cluster Labels'] == 2, chicago_merged.columns[[0] + list(range(4, chicago_merged.shape[1]))]]
```

Out[168]:

	COMMUNITY AREA NAME	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
5	Edison Park	Theater	Neighborhood	Park	Yoga Studio	Flower Shop	Fish & Chips Shop	Filipino Restaurant	Fast Food Restaurant	Farmers Market	Exhibit

6. Conclusion

After analyzing the socioeconomic indicators and venue types of each community in Chicago, we strongly recommend that Enso Street select Edison Park as the community to launch their services.

COMMUNITY AREA NAME	PERCENT HOUSEHOLDS BELOW POVERTY	PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA	PERCENT OF HOUSING CROWDED	PERCENT AGED 16+ UNEMPLOYED	HARDSHIP INDEX	PERCENT AGED UNDER 18 OR OVER 64	PER CAPITA INCOME
Edison Park	3.3	7.4	1.1	6.5	8.0	35.3	40959

As Enso Street gradually acquires customers after it launches, we can help to finetune our methodologies and analysis based on actual customer data. This will help us to more accurately select communities for future operation expansion.