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

Applied Data Science Capstone by IBM/Coursera

Report on public schools in Chicago

Maciej Rybczyński

Introduction: Business Problem

The aim of this project is to find a public schools in Chicago, US placed in a suitable location equipped witha proper commercial establishments. In particular this report will be targeted to people moving from other cities/states/countries to Chicago and interested in sending their children to the right school in Chicago.

Data description

Based on definition of the problem, the following factors that will influence our decission are:

- ✓ finding the geographical location of the schools in Chicago,
- ✓ finding the most common venues surrounding a particular school.

We will be using the geographical coordinates of Chicago and geographical location of the schools to plot school location, and finally cluster our schools and present our findings.

Following data sources will be needed to extract/generate the required information:

- ✓ Part 1: Using a real world data set from City of Chicago containing information on Chicago public schools in 2011-2012 school year, updated in 2018. A dataset consisting of location of the school, its type and other optional parameters describing a school.
- ✓ Part2: Foursquare API Data
- ✓ Part 3: Creating a new consolidated dataset of the schools, the most common venues and the respective Community Areas along with co-ordinates.: This data will be fetched using Four Square API to explore the venues around schools and to apply machine learning algorithm to cluster the schools and present the findings by plotting it on maps using Folium.

Part 1: Using a real world data set from City of Chicago containing information on Chicago public schools in 2011-2012 school year, updated in 2018.

Chicago Public Schools - Progress Report Cards (2011-2012)

This is a very detailed dataset containing many useful information about each public school in Chicago Some properties of dataset include:

- ✓ Name of School
- ✓ Type of School (Elementary, Middle, or High School)
- ✓ Street Address
- ✓ ZIP Code
- ✓ Phone Number
- ✓ Website URL
- ✓ Safety Score
- ✓ Family Involvement Score
- ✓ Environment Score
- ✓ Leaders Score
- ✓ Teachers Score
- ✓ Latitude
- ✓ Longitude
- ✓ Community Area Name

Data set URL:

https://data.cityofchicago.org/Education/Chicago-Public-Schools-Progress-Report-Cards-2011-/9xs2-f89t

Part 2: Foursquare API Data

We will need data about different venues surrounding schools. In order to gain that information we will use Foursquare locational information. *Foursquare* is a location data provider with information about all manner of venues and events within an area of interest. Such information includes venue names, locations, menus and even photos. As such, the foursquare location platform will be used as the sole data source since all the stated required information can be obtained through the API.

After finding the list of schools, we then connect to the Foursquare API to gather information about venues around every considered school. For each school surrounding, we have chosen the radius to be 100 meters.

The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the school location. The information obtained per venue as follows:

- ✓ Name of School,

- ✓ School Latitude,
- ✓ School Longitude,
- ✓ Name of Venue,
- ✓ Venue Latitude,
- ✓ Venue Longitude,
- ✓ Venue Category.

Based on all the above described information we have collected a sufficient data to build our model. We cluster the schools together based on similar venue categories. We then present our observations and findings. Using this data, our stakeholders can take the necessary decision.

Methodology

We will be creating our model with the help of Python so we start off by importing all the required packages.

Importing libraries:

```
import pandas as pd

!pip install geopy
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

!pip install folium

#Importing folium to visualise Maps and plot based on Lat and Lng
import folium

#Requests to request web pages by making get requests to FourSquare REST Client
import requests

#To normalise data returned by FourSquare API
from pandas.io.json import json_normalize

#Importing KMeans from SciKit library to Classify neighborhoods into clusters
from sklearn.cluster import KMeans
```

Package breakdown:

- ✓ pandas: To collect and manipulate data in JSON and HTMl and then data analysis,
- ✓ Nominatim: To convert an address into latitude and longitude values,
- ✓ folium: Generating maps Chicago,
- ✓ requests: Handle http requests,
- ✓ json_normalize: To normalise data returned by FourSquare API,
- ✓ sklearn: To import Kmeans which is the machine learning model that we are using.

The approach taken here is to explore each surrounding of the considered school, plot the map to show the schools being considered and then build our model by clustering all of the similar school surroundings together and finally plot the new map with the clustered schools. We draw insights and then compare and discuss our findings.

Data Collection

Data downloaded from the website of City of Chicago were stored in an csv file:

Reading from dataset:

```
chicago_schools_df = pd.read_csv('Chicago_Public_Schools_2011-2012_updated.csv', index_col=None, error_bad_lines=False)
chicago_schools_df.head()
```

	School ID	Name of School	Elementary, Middle, or High School	Street Address	City	State	ZIP Code	Phone Number	Link	Network Manager	...	RCDTS Code
0	609966	Charles G Hammond Elementary School	ES	2819 W 21st Pl	Chicago	IL	60623	(773) 535-4580	http://schoolreports.cps.edu/SchoolProgressRep...	Pilsen-Little Village Elementary Network	...	150000000000000
1	610539	Marvin Camras Elementary School	ES	3000 N Mango Ave	Chicago	IL	60634	(773) 534-2960	http://schoolreports.cps.edu/SchoolProgressRep...	Fullerton Elementary Network	...	150000000000000
2	609852	Eliza Chappell Elementary School	ES	2135 W Foster Ave	Chicago	IL	60625	(773) 534-2390	http://schoolreports.cps.edu/SchoolProgressRep...	Ravenswood-Ridge Elementary Network	...	150000000000000
3	609835	Daniel R Cameron Elementary School	ES	1234 N Monticello Ave	Chicago	IL	60651	(773) 534-4290	http://schoolreports.cps.edu/SchoolProgressRep...	Garfield-Humboldt Elementary Network	...	150000000000000
4	610521	Sir Miles Davis Magnet Elementary Academy	ES	6730 S Paulina St	Chicago	IL	60636	(773) 535-9120	http://schoolreports.cps.edu/SchoolProgressRep...	Englewood-Gresham Elementary Network	...	150000000000000

5 rows × 79 columns

This is a very detailed dataset and we extracted the information which is the most useful to us:

	Name of School	Street Address	City	ZIP Code	Elementary, Middle, or High School	Safety Score	Latitude	Longitude	Community Area Name
0	Charles G Hammond Elementary School	2819 W 21st Pl	Chicago	60623	ES	40.0	41.852691	-87.696278	SOUTH LAWNDALE
1	Marvin Camras Elementary School	3000 N Mango Ave	Chicago	60634	ES	54.0	41.934966	-87.770165	BELMONT CRAGIN
2	Eliza Chappell Elementary School	2135 W Foster Ave	Chicago	60625	ES	70.0	41.975867	-87.683254	LINCOLN SQUARE
3	Daniel R Cameron Elementary School	1234 N Monticello Ave	Chicago	60651	ES	42.0	41.903785	-87.717963	HUMBOLDT PARK
4	Sir Miles Davis Magnet Elementary Academy	6730 S Paulina St	Chicago	60636	ES	35.0	41.771222	-87.666567	WEST ENGLEWOOD

Data exploration

We start data exploration by finding how many schools exists in each Community Area of Chicago.

How many schools in each Chicago Community Area:

```
chicago_schools_sel['Community Area Name'].value_counts()
```

```
AUSTIN             23
SOUTH LAWNDALE     22
WEST TOWN          20
ENGLEWOOD          17
NEAR WEST SIDE     16
                   ..
BURNSIDE            1
MONTCLARE           1
LOOP                1
OAKLAND             1
OHARE               1
Name: Community Area Name, Length: 77, dtype: int64
```

There are three types of public schools in Chicago: Elementary (ES), Middle (MS) and High (HS) schools:

How many public schools of a particular type are in Chicago:

```
seriesObjE = chicago_schools_sel.apply(lambda x: True if x['Elementary, Middle, or High School'] == 'ES' else False , axis=1)
# Count number of True in series
numOfRowsE = len(seriesObjE[seriesObjE == True].index)
print('Number of elementary schools : ', numOfRowsE)

seriesObjM = chicago_schools_sel.apply(lambda x: True if x['Elementary, Middle, or High School'] == 'MS' else False , axis=1)
numOfRowsM = len(seriesObjM[seriesObjM == True].index)
print('Number of middle schools : ', numOfRowsM)

seriesObjH = chicago_schools_sel.apply(lambda x: True if x['Elementary, Middle, or High School'] == 'HS' else False , axis=1)
numOfRowsH = len(seriesObjH[seriesObjH == True].index)
print('Number of high schools : ', numOfRowsH)
```

```
Number of elementary schools :  462
Number of middle schools :  11
Number of high schools :  93
```

Pivoting table to show a type of a particular school (ES - elementary school, MS - middle school, HS - high school)

```
chicago_schools_cat = pd.pivot_table(chicago_schools_sel,
                                values=['City'],
                                index=['Name of School'],
                                columns=['Elementary, Middle, or High School'],
                                aggfunc=len,
                                fill_value=0,
                                margins=True)
chicago_schools_cat
```

	City			
Elementary, Middle, or High School	**ES**	**HS**	**MS**	**All**
Name of School				
A.N. Pritzker School	1	0	0	1
Abraham Lincoln Elementary School	1	0	0	1
Adam Clayton Powell Paideia Community Academy Elementary School	1	0	0	1
Adlai E Stevenson Elementary School	1	0	0	1
Agustin Lara Elementary Academy	1	0	0	1
...	...	...	...	...
Wilma Rudolph Elementary Learning Center	1	0	0	1
Wolfgang A Mozart Elementary School	1	0	0	1
Woodlawn Community Elementary School	1	0	0	1
World Language Academy High School	0	1	0	1
All	462	93	11	566

567 rows × 4 columns

Next we use *Nomatim geolocator* to find geographical coordinates of Chicago, what will be needed to plot map of Chicago.

Using geopy Nominatim geolocator to fing geographical coordinates of Chicago:

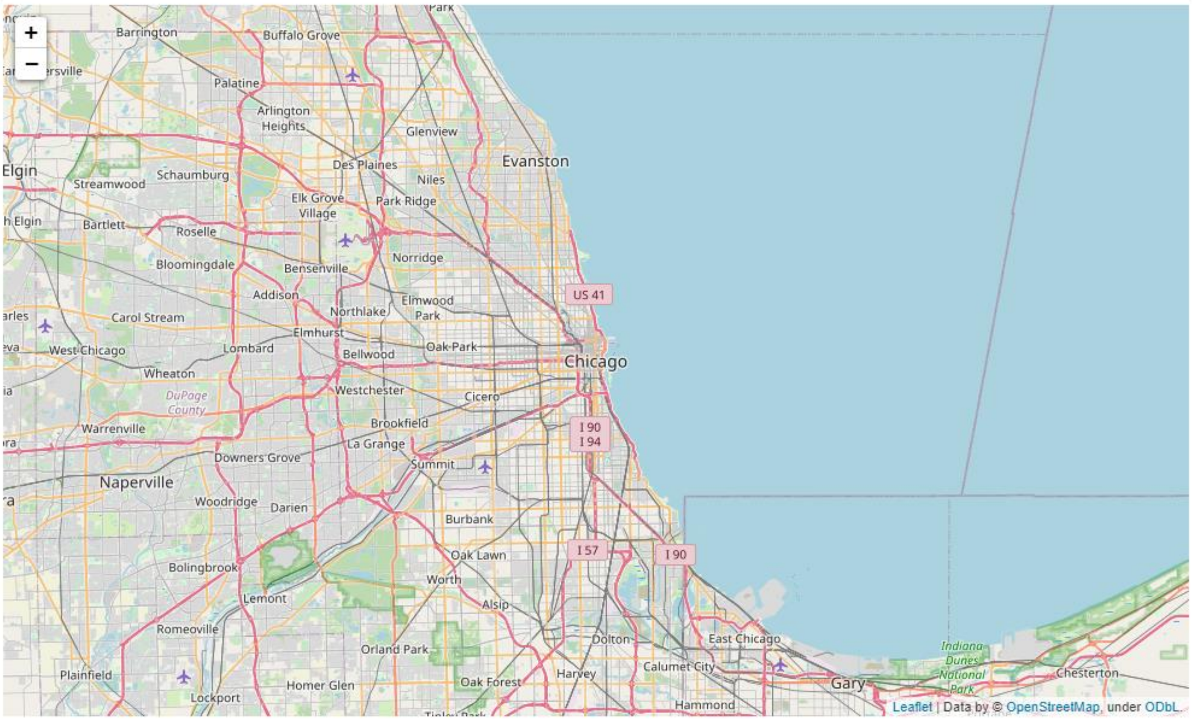
```
address = "Chicago, IL"

geolocator = Nominatim(user_agent="chicago_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinates of Chicago are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinates of Chicago are 41.8755616, -87.6244212.
```

Plotting map of Chicago using Folium:

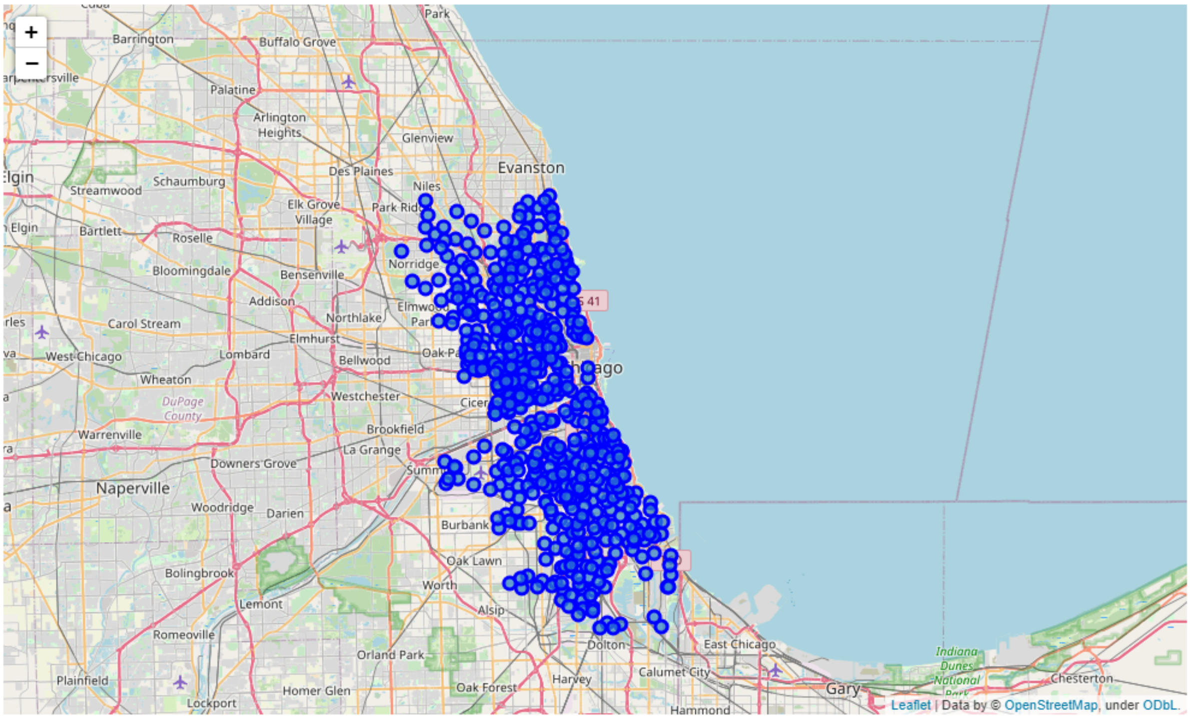
```
map_chicago = folium.Map(location=[latitude, longitude], zoom_start=10)
map_chicago
```



Next, we can overly the schools locations on the map of Chicago:

```python
for lat, lng, ncomm, nschool in zip(
        chicago_schools_sel['Latitude'],
        chicago_schools_sel['Longitude'],
        chicago_schools_sel['Community Area Name'],
        chicago_schools_sel['Name of School']):
    label = '{}, {}'.format(ncomm, nschool)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_chicago)

map_chicago
```



Due to substantial number of public schools in Chicago, focusing on high schools only.

Due to substantial number of public schools in Chicago, focusing on high schools only:

```
chicago_high_schools = chicago_schools_sel[chicago_schools_sel['Elementary, Middle, or High School'] == 'HS']
chicago_high_schools
```

	Name of School	Street Address	City	ZIP Code	Elementary, Middle, or High School	Safety Score	Latitude	Longitude	Community Area Name
8	Walter Payton College Preparatory High School	1034 N Wells St	Chicago	60610	HS	98.0	41.901552	-87.634537	NEAR NORTH SIDE
15	Manley Career Academy High School	2935 W Polk St	Chicago	60612	HS	41.0	41.870912	-87.699887	EAST GARFIELD PARK
17	Northside College Preparatory High School	5501 N Kedzie Ave	Chicago	60625	HS	99.0	41.981352	-87.708672	NORTH PARK
28	Michele Clark Academic Prep Magnet High School	5101 W Harrison St	Chicago	60644	HS	NaN	41.872857	-87.753355	AUSTIN
30	Uplift Community High School	900 W Wilson Ave	Chicago	60640	HS	50.0	41.965574	-87.652522	UPTOWN
...	...	...	...	...	...	...	...	...	...
554	Chicago High School for Agricultural Sciences	3857 W 111th St	Chicago	60655	HS	87.0	41.691194	-87.717739	MOUNT GREENWOOD
559	Stephen T Mather High School	5835 N Lincoln Ave	Chicago	60659	HS	58.0	41.987595	-87.702449	WEST RIDGE
560	High School of Leadership at South Shore	7627 S Constance Ave	Chicago	60649	HS	NaN	41.756194	-87.579607	SOUTH SHORE
561	TEAM Englewood Community Academy High School	6201 S Stewart Ave	Chicago	60621	HS	45.0	41.781493	-87.634942	ENGLEWOOD
564	Infinity Math Science and Technology High School	3120 S Kostner Ave	Chicago	60623	HS	58.0	41.836020	-87.734195	SOUTH LAWNDALE

93 rows × 9 columns

Part 3: Creating a new consolidated dataset of the schools, and the most common venues and the respective Community Areas along with co-ordinates.:

This data will be fetched using Four Square API to explore the venues and to apply machine learning algorithm to cluster the schools and present the findings by plotting it on maps using Folium.

We start with setting up Foursquare credentials.

Setting Up Foursquare Credentials (to be removed from github version of notebook):

```
#Four Square Credentials

CLIENT_ID = ''
CLIENT_SECRET = ''
VERSION = '20210516'
LIMIT = 10

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID:
CLIENT_SECRET:
```

Next, we prepare a function to fetch venues around a given location and use it.

A function to fetch venues around a given location:

```
def getNearbyVenues(names, latitudes, longitudes, radius=100):

    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={}'.form
            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']['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 of School',
                  'School Latitude',
                  'School Longitude',
                  'Venue',
                  'Venue Category']

    return(nearby_venues)
```

Fetching venues around public high schools:

```
chicago_school_venues = getNearbyVenues(names=chicago_high_schools['Name of School'],
                                   latitudes=chicago_high_schools['Latitude'],
                                   longitudes=chicago_high_schools['Longitude']
                                  )
```

```
Walter Payton College Preparatory High School
Manley Career Academy High School
Northside College Preparatory High School
Michele Clark Academic Prep Magnet High School
Uplift Community High School
Morgan Park High School
Bronzeville Scholastic Academy High School
William J Bogan High School
Emil G Hirsch Metropolitan High School
Austin Polytechnical Academy High School
World Language Academy High School
Multicultural Academy of Scholarship
Mason High School
Marie Sklodowska Curie Metropolitan High School
George Washington High School
Robert Lindblom Math & Science Academy High School
Benito Juarez Community Academy High School
Hyde Park Academy High School
John Marshall Metropolitan High School
Friedrich W von Steuben Metropolitan Science High School
Southside Occupational Academy High School
Chicago Military Academy High School
Eric Solorio Academy High School
Neal F Simeon Career Academy High School
John Hancock College Preparatory High School
Roald Amundsen High School
Edwin G Foreman High School
Paul Laurence Dunbar Career Academy High School
Charles P Steinmetz Academic Centre High School
Gurdon S Hubbard High School
Albert G Lane Technical High School
Carl Schurz High School
Dyett High School
Phoenix Military Academy High School
Chicago Vocational Career Academy High School
```

The collected data we put into a new data frame and then we group them with respect to school location.

Data frame containing venues aroung each public high school in Chicago:

```
print(chicago_school_venues.shape)
chicago_school_venues.head()
```

```
(69, 5)
```

	Name of School	School Latitude	School Longitude	Venue	Venue Category
0	Northside College Preparatory High School	41.981352	-87.708672	Lake Shore Symphony Rehersal	Music Venue
1	Michele Clark Academic Prep Magnet High School	41.872857	-87.753355	YWCA of Metropolitan Chicago	Gym / Fitness Center
2	Uplift Community High School	41.965574	-87.652522	Citizen Skate Cafe	Café
3	Uplift Community High School	41.965574	-87.652522	CVS pharmacy	Pharmacy
4	William J Bogan High School	41.749348	-87.721097	Dollar Tree	Discount Store

Groupping of venues with respect to school location:

```
chicago_school_venues.groupby('Name of School').count().drop(['School Latitude','School Longitude','Venue Category'], axis = 1)
```

	Venue
Name of School	
Alcott High School for the Humanities	2
Benito Juarez Community Academy High School	1
Carl Schurz High School	4
Chicago High School for Agricultural Sciences	2
Chicago Military Academy High School	8
DeVry University Advantage Academy High School	2
Friedrich W von Steuben Metropolitan Science High School	2
Gage Park High School	1
George H Corliss High School	1
Gwendolyn Brooks College Preparatory Academy High School	1
Hyman G Rickover Naval Academy High School	1
Jacqueline B Vaughn Occupational High School	3
Lake View High School	8
Lincoln Park High School	5
Marie Sklodowska Curie Metropolitan High School	1
Michele Clark Academic Prep Magnet High School	1
New Millennium High School of Health at Bowen	1
Nicholas Senn High School	1
Northside College Preparatory High School	1
Orr Academy High School	1
Roald Amundsen High School	2
Roberto Clemente Community Academy High School	3
Spry Community Links High School	1
Stephen T Mather High School	1
Thomas Kelly High School	1
Uplift Community High School	2
Wells Community Academy High School	4
William J Bogan High School	3
William Jones College Preparatory High School	5

```
print('There are {} uniques categories.'.format(len(chicago_school_venues['Venue Category'].unique())))
```

```
There are 47 uniques categories.
```

One Hot Encoding

Since we are trying to find out what are the different kinds of venue categories present in each high school surrounding and then calculate the top 5 common venues to base our similarity on, we use the One Hot Encoding to work with our categorical datatype of the venue categories. This helps to convert the categorical data into numeric data.

One Hot Encoding to analyze each high school surrounding:

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

# add neighborhood column back to dataframe
chicago_school_onehot['Name of School'] = chicago_school_venues['Name of School']

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

chicago_school_onehot.head()
```

	Name of School	American Restaurant	Art Gallery	Asian Restaurant	BBQ Joint	Basketball Court	Breakfast Spot	Burger Joint	Bus Station	Café	...	Pool	Record Shop	Restaurant	River	Salon / Barbershop	Sandwic Plac
0	Northside College Preparatory High School	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	Michele Clark Academic Prep Magnet High School	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	Uplift Community High School	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	
3	Uplift Community High School	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	William J Bogan High School	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

5 rows × 48 columns

```
chicago_school_onehot.shape
```

(69, 48)

Top 5 most common venues around high school:

```
num_top_venues = 5

for school in chicago_school_grouped['Name of School']:
    print("----"+school+"----")
    temp = chicago_school_grouped[chicago_school_grouped['Name of School'] == school].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Alcott High School for the Humanities----
                venue  freq
0             Dog Run   0.5
1                Park   0.5
2  American Restaurant   0.0
3           Hookah Bar   0.0
4    Korean Restaurant   0.0


----Benito Juarez Community Academy High School----
                venue  freq
0   Mexican Restaurant   1.0
1  American Restaurant   0.0
2          Pizza Place   0.0
3    Korean Restaurant   0.0
4               Market   0.0


----Carl Schurz High School----
```

Building a new dataframe and display the top 5 venues around each high school:

```
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']

# create columns according to number of top venues
columns = ['Name of School']
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
chicago_school_venues_sorted = pd.DataFrame(columns=columns)
chicago_school_venues_sorted['Name of School'] = chicago_school_grouped['Name of School']

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

chicago_school_venues_sorted.head()
```

	Name of School	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Alcott High School for the Humanities	Dog Run	Park	Women's Store	Historic Site	Furniture / Home Store
1	Benito Juarez Community Academy High School	Mexican Restaurant	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
2	Carl Schurz High School	Thai Restaurant	Asian Restaurant	Martial Arts School	Convenience Store	Women's Store
3	Chicago High School for Agricultural Sciences	BBQ Joint	Dive Bar	Women's Store	Coffee Shop	Furniture / Home Store
4	Chicago Military Academy High School	History Museum	Pizza Place	Historic Site	Wings Joint	Cosmetics Shop

Model Building - KMeans

We will be using KMeans Clustering Machine learning algorithm to cluster similar school's surroundings together. We will be going with the number of clusters as 5.

Clustering Chicago high schools:

```
# set number of clusters
kclusters = 5

chicago_school_grouped_clustering = chicago_school_grouped.drop('Name of School', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
array([1, 0, 0, 0, 0, 0, 0, 0, 2, 1])
```

```
# add clustering labels
chicago_school_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
chicago_merged = chicago_schools_sel
chicago_merged = chicago_merged.join(chicago_school_venues_sorted.set_index('Name of School'), on='Name of School')
chicago_merged
```

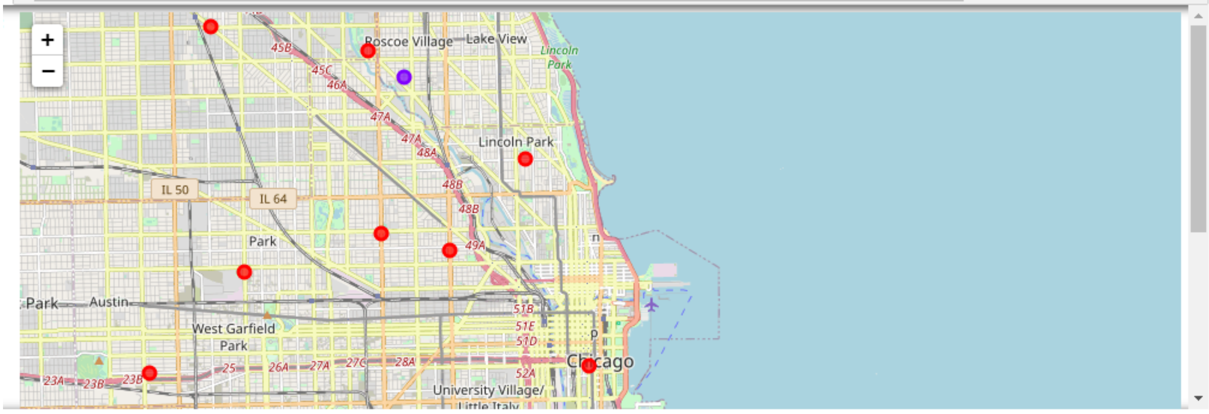
Name of School	Street Address	City	ZIP Code	Elementary, Middle, or High School	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th M Comm Ver
Charles G														

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=12)

# set color scheme for the clusters
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['Latitude'], chicago_merged['Longitude'], chicago_merged['Name of School'], chic
    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
```



Clusters Analysis

Examining the resulting Clusters:

Cluster 1:

```
chicago_merged.loc[chicago_merged['Cluster Labels'] == 0, chicago_merged.columns[[1] + list(range(5, chicago_merged.shape[1]))]]
```

	Street Address	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
17	5501 N Kedzie Ave	99.0	41.981352	-87.708672	NORTH PARK	0	Music Venue	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
28	5101 W Harrison St	NaN	41.872857	-87.753355	AUSTIN	0	Gym / Fitness Center	Coffee Shop	Furniture / Home Store	Fried Chicken Joint	Football Stadium
30	900 W Wilson Ave	50.0	41.965574	-87.652522	UPTOWN	0	Pharmacy	Café	Coffee Shop	Furniture / Home Store	Fried Chicken Joint
38	3939 W 79th St	20.0	41.749348	-87.721097	ASHBURN	0	Furniture / Home Store	Fast Food Restaurant	Discount Store	Women's Store	Coffee Shop
77	2150 S Laflin St	46.0	41.852673	-87.663769	LOWER WEST SIDE	0	Mexican Restaurant	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
95	5039 N Kimball Ave	70.0	41.973193	-87.713350	NORTH PARK	0	River	Bus Station	Women's Store	Coffee Shop	Furniture / Home Store
107	3519 S Giles Ave	32.0	41.830538	-87.619178	DOUGLAS	0	History Museum	Pizza Place	Historic Site	Wings Joint	Cosmetics Shop
116	5110 N Damen Ave	51.0	41.975079	-87.679521	LINCOLN SQUARE	0	Basketball Court	Pool	Women's Store	Coffee Shop	Furniture / Home Store
139	3601 N Milwaukee Ave	48.0	41.946408	-87.735625	IRVING PARK	0	Thai Restaurant	Asian Restaurant	Martial Arts School	Convenience Store	Women's Store
190	4015 N Ashland Ave	64.0	41.954784	-87.668916	LAKE VIEW	0	Chinese Restaurant	Coffee Shop	Thai Restaurant	Fried Chicken Joint	Breakfast Spot
238	3300 N Campbell	NaN	41.941426	-87.690799	NORTH CENTER	0	Furniture / Home Store	Salon / Barbershop	Women's Store	Coffee Shop	Fried Chicken Joint
264	5630 S Rockwell St	14.0	41.791014	-87.688991	GAGE PARK	0	Clothing Store	Coffee Shop	Furniture / Home Store	Fried Chicken Joint	Football Stadium
298	2001 N Orchard St	65.0	41.918304	-87.645974	LINCOLN PARK	0	Women's Store	Art Gallery	BBQ Joint	Burger Joint	Mediterranean Restaurant
311	730 N Pulaski Rd	NaN	41.894448	-87.726203	HUMBOLDT PARK	0	Fast Food Restaurant	Women's Store	Coffee Shop	Furniture / Home Store	Fried Chicken Joint

Cluster 2:

```
chicago_merged.loc[chicago_merged['Cluster Labels'] == 1, chicago_merged.columns[[1] + list(range(5, chicago_merged.shape[1]))]]
```

	Street Address	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
244	4136 S California Ave	36.0	41.818711	-87.694675	BRIGHTON PARK	1	Park	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
463	2957 N Hoyne Ave	70.0	41.935761	-87.680524	NORTH CENTER	1	Dog Run	Park	Women's Store	Historic Site	Furniture / Home Store
473	250 E 111th St	64.0	41.692790	-87.616381	ROSELAND	1	Park	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint

```
Cluster 3:
```

```
chicago_merged.loc[chicago_merged['Cluster Labels'] == 2, chicago_merged.columns[[1] + list(range(5, chicago_merged.shape[1]))]]
```

	Street Address	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
198	821 E 103rd St	33.0	41.707391	-87.603078	PULLMAN	2	Football Stadium	Women's Store	Coffee Shop	Furniture / Home Store	Fried Chicken Joint

Cluster 4:

```
chicago_merged.loc[chicago_merged['Cluster Labels'] == 3, chicago_merged.columns[[1] + list(range(5, chicago_merged.shape[1]))]]
```

	Street Address	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
68	4959 S Archer Ave	43.0	41.803046	-87.722007	ARCHER HEIGHTS	3	Hotel	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
176	5900 N Glenwood Ave	64.0	41.989051	-87.665262	EDGEWATER	3	Hotel	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint
274	5900 N Glenwood Ave	48.0	41.989051	-87.665262	EDGEWATER	3	Hotel	Women's Store	Historic Site	Furniture / Home Store	Fried Chicken Joint

Cluster 5:

```
chicago_merged.loc[chicago_merged['Cluster Labels'] == 4, chicago_merged.columns[[1] + list(range(5, chicago_merged.shape[1]))]]
```

	Street Address	Safety Score	Latitude	Longitude	Community Area Name	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
521	2710 E 89th St	17.0	41.733761	-87.557753	SOUTH CHICAGO	4	American Restaurant	Coffee Shop	Furniture / Home Store	Fried Chicken Joint	Football Stadium

Results and Discussion

The object of the business problem was to help Chicago migrants to identify suitable public school to their children, located in area surrounded with the appropriate venues. This has been achieved by first making use of Chicago Public Schools data to identify a proper place with considerable number of venues. Due to substantial number of public schools in Chicago focus was made on the public high schools only. Next, groupping of the high schools into clusters was done to assist the migrants by providing them with relevant data about venues and safety of a given school surrounding.

Conclusion

We have explored the Chicago Public Schools data to understand different types of public schools in all Community Areas of Chicago and later categorized them into different types. This helped us group the schools. We further shortlist the high schools based on the common venues, to choose clusters of schools which best suits the business problem.