Capstone Project

June 24, 2019

1 Introduction

Hello and welcome to my IBM Data Science Capstone Project on Coursera. This notebook holds the source code for the project and has detailed headers explaining each step along the way. The last cell of the notebook contains the code necessary to generate the final outcomes (though be sure to run all preceding code cells beforehand). Thank you for reviewing my project and I hope you find it interesting!

```
In [1]: #All relevent imports
        import numpy as np
        import pandas as pd
        pd.set_option('display.max_columns',None)
        pd.set_option('display.max_rows',None)
        import json
        from geopy.geocoders import Nominatim
        import geocoder
        import requests
        from pandas.io.json import json_normalize
        import matplotlib.cm as cm
        import matplotlib.colors as colors
        import matplotlib.pyplot as plt
        from sklearn.cluster import KMeans
        import folium

        from bs4 import BeautifulSoup
```

2 Goal

to determine the best place a buisness owner can open up his/her buisness in chicago.

- we will utalize chicago census data and the foursquare api to generate a heatmap of recommended locations
- clustering will be used to translate the unstructured foursquare data into meaningful insights about the buisness enviroment of chicago neighborhoods

```
In [2]: #Some Resources Used
        #base API
```

```
#https://api.foursquare.com/v2/
#https://api.foursquare.com/v2/venues/
#https://api.foursquare.com/v2/users/
#https://api.foursquare.com/v2/tips/
#https://github.com/blackmad/neighborhoods/blob/master/chicago.geojson
#https://github.com/OpenDataDE/State-zip-code-GeoJSON/blob/master/il_illinois_zip_code
```

3 Getting Soup Output from website

Here we will be pulling the names and zipcodes of chicago neighborhoods from a website for later use

```
In [2]: df_chicago = pd.DataFrame(columns=['Zipcode','Neighborhood'])
        df_chicago
```

```
Out[2]: Empty DataFrame
        Columns: [Zipcode, Neighborhood]
        Index: []
```

```
In [3]: addr = 'https://data.mongabay.com/igapo/zip_codes/metropolitan-areas/metro-zip/Chicago
        source = requests.get(addr).text
        soup = BeautifulSoup(source,'lxml')
```

```
In [4]: table = soup.find('table',class_='boldtable')
```

```
In [5]: for i in table.find_all('tr'):
            content = i.td.text.split()
            df_chicago = df_chicago.append(dict(zip(df_chicago.columns,content)),ignore_index=
        df_chicago.head()
```

```
Out[5]:   Zipcode Neighborhood
        0   60001        Alden
        1   60002      Antioch
        2   60002          Old
        3   60002          Old
        4   60002    Wadsworth
```

```
In [7]: #df_chicago.to_csv(r'D:\Desktop\outcomes\chicago.csv')
```

```
In [6]: df_chicago_only = df_chicago[df_chicago["Neighborhood"] == "Chicago"]
```

```
In [7]: codes = df_chicago.groupby(df_chicago["Neighborhood"]).groups
```

```
In [8]: #Create Empty Pandas DF
        df_grouped = pd.DataFrame(columns=['Neighborhood','Zipcode'])
        df_grouped
```

```
Out[8]: Empty DataFrame
        Columns: [Neighborhood, Zipcode]
        Index: []
```

4 Combining Data By Neighborhood

For easy lookback here we combine the neightborhood names by each zipcode. This will aid in individual research outside of the datasets can we can later use to formulate a cost function.

```
In [9]: for nb in codes.keys():
            #print("NB",nb)
            zc = []
            for i in codes[nb]:
                zc.append(df_chicago.iloc[i][0])
                zc = list(set(zc))
                zcf = ', '.join(zc)
```

```
In [12]: for nb in codes.keys():
             content = [nb]
             zc = []
             for i in codes[nb]:
                 zc.append(df_chicago.iloc[i][0])
                 zc = list(set(zc))
                 zcf = ', '.join(zc)
             content.append(zcf)
             df_grouped = df_grouped.append(dict(zip(df_grouped.columns,content)),ignore_index=
         df_grouped.head()
```

```
Out[12]:   Neighborhood       Zipcode
         0           AT         60572
         1       Abbott         60064
         2      Addison         60101
         3        Alden  60001, 60033
         4    Algonquin  60156, 60102
```

5 Illinois geodata

In this section of code geojson data for the state of illinois is sorted to be used later in folium mapping. The file is rather large so it is sorted into only relevent sections and the remainder is dropped.

```
In [13]: #loading GeoJSON file
         with open('illinois.json','r') as jsonFile:
             data = json.load(jsonFile)

         geo = data
```

```
In [14]: #geo['features'][0]['properties']["ZCTA5CE10"]
         geo['features'][0]
```

```
Out[14]: {'type': 'Feature',
          'properties': {'STATEFP10': '17',
  'ZCTA5CE10': '62359',
  'GEOID10': '1762359',
  'CLASSFP10': 'B5',
  'MTFCC10': 'G6350',
  'FUNCSTAT10': 'S',
  'ALAND10': 10360074,
  'AWATER10': 7921,
  'INTPTLAT10': '+40.0338795',
  'INTPTLON10': '-091.2014548',
  'PARTFLG10': 'N'},
 'geometry': {'type': 'Polygon',
  'coordinates': [[[-91.182899, 40.026881],
    [-91.182577, 40.026761],
    [-91.182428, 40.026711],
    [-91.182125, 40.026608],
    [-91.181677, 40.02648],
    [-91.181419, 40.026419],
    [-91.18093, 40.026323],
    [-91.180498, 40.026255],
    [-91.180081, 40.026205],
    [-91.179637, 40.026169],
    [-91.179213, 40.026161],
    [-91.1788, 40.026162],
    [-91.178347, 40.026177],
    [-91.177816, 40.026225],
    [-91.177419, 40.026285],
    [-91.177051, 40.026348],
    [-91.176668, 40.026429],
    [-91.176692, 40.02387],
    [-91.176692, 40.023811],
    [-91.176698, 40.021465],
    [-91.176684, 40.021291],
    [-91.176632, 40.021046],
    [-91.176601, 40.020977],
    [-91.17656, 40.020918],
    [-91.176809, 40.020901],
    [-91.181786, 40.020983],
    [-91.183933, 40.021028],
    [-91.184539, 40.021028],
    [-91.185146, 40.021048],
    [-91.186161, 40.021048],
    [-91.187006, 40.021065],
    [-91.188558, 40.021081],
    [-91.189561, 40.021092],
    [-91.190867, 40.021127],
    [-91.192906, 40.021143],
    [-91.195194, 40.02118],
    [-91.195837, 40.021186],
[-91.195938, 40.021205],
[-91.19602, 40.021227],
[-91.196098, 40.021198],
[-91.196205, 40.021182],
[-91.19714, 40.021194],
[-91.198638, 40.021212],
[-91.199963, 40.021235],
[-91.20207, 40.021229],
[-91.20325, 40.021228],
[-91.203672, 40.021058],
[-91.204097, 40.020892],
[-91.204751, 40.020667],
[-91.205621, 40.020415],
[-91.209716, 40.019207],
[-91.211208, 40.018776],
[-91.212311, 40.018448],
[-91.213525, 40.018083],
[-91.215115, 40.01761],
[-91.215436, 40.017515],
[-91.216864, 40.017106],
[-91.218559, 40.016613],
[-91.218528, 40.0165],
[-91.219713, 40.015903],
[-91.223712, 40.014288],
[-91.223724, 40.014438],
[-91.223712, 40.014737],
[-91.223719, 40.014854],
[-91.22374, 40.014976],
[-91.223767, 40.015041],
[-91.223806, 40.015087],
[-91.223739, 40.015107],
[-91.223732, 40.015201],
[-91.223731, 40.015787],
[-91.22375, 40.016405],
[-91.223782, 40.016548],
[-91.223803, 40.016711],
[-91.223805, 40.016836],
[-91.223774, 40.016985],
[-91.223782, 40.017226],
[-91.223776, 40.017334],
[-91.223748, 40.017432],
[-91.223704, 40.017521],
[-91.223683, 40.017618],
[-91.223685, 40.017735],
[-91.22371, 40.018118],
[-91.223745, 40.01823],
[-91.223765, 40.018575],
[-91.22379, 40.019433],
 [-91.223837, 40.020565],
 [-91.223887, 40.021165],
 [-91.223889, 40.021508],
 [-91.223923, 40.021562],
 [-91.223934, 40.021838],
 [-91.223435, 40.033494],
 [-91.223308, 40.036756],
 [-91.223186, 40.047344],
 [-91.223201, 40.047611],
 [-91.223269, 40.047759],
 [-91.223207, 40.047765],
 [-91.223054, 40.04779],
 [-91.222968, 40.047873],
 [-91.222955, 40.049292],
 [-91.221448, 40.049295],
 [-91.204922, 40.049285],
 [-91.202708, 40.049284],
 [-91.195619, 40.049275],
 [-91.195618, 40.048968],
 [-91.19563, 40.048033],
 [-91.195671, 40.045868],
 [-91.195692, 40.045024],
 [-91.195707, 40.043059],
 [-91.195754, 40.039882],
 [-91.195772, 40.039441],
 [-91.195771, 40.039127],
 [-91.195772, 40.038993],
 [-91.195803, 40.035795],
 [-91.195442, 40.035798],
 [-91.193694, 40.035771],
 [-91.192651, 40.035747],
 [-91.186579, 40.035637],
 [-91.184871, 40.035597],
 [-91.183401, 40.035576],
 [-91.18266, 40.035561],
 [-91.180703, 40.035531],
 [-91.17831, 40.035483],
 [-91.176636, 40.035458],
 [-91.176645, 40.033093],
 [-91.17664, 40.031975],
 [-91.176666, 40.029181],
 [-91.176675, 40.028789],
 [-91.176672, 40.028705],
 [-91.177306, 40.02852],
 [-91.182777, 40.026919],
 [-91.182899, 40.026881]],
[[-91.182899, 40.026881],
 [-91.183078, 40.026948],
                [-91.183895, 40.027199],
                [-91.184208, 40.027285],
                [-91.184535, 40.027364],
                [-91.184908, 40.027436],
                [-91.185281, 40.027492],
                [-91.185605, 40.027535],
                [-91.185982, 40.027566],
                [-91.186473, 40.02759],
                [-91.18686, 40.027595],
                [-91.187246, 40.02758],
                [-91.187975, 40.027517],
                [-91.188361, 40.027459],
                [-91.188724, 40.027393],
                [-91.189076, 40.027318],
                [-91.189458, 40.027223],
                [-91.189847, 40.027106],
                [-91.190206, 40.026979],
                [-91.190531, 40.026856],
                [-91.190878, 40.026716],
                [-91.1912, 40.026561],
                [-91.191838, 40.026201],
                [-91.19221, 40.02596],
                [-91.192493, 40.025755],
                [-91.193464, 40.02498],
                [-91.193824, 40.024716],
                [-91.194436, 40.024343],
                [-91.194698, 40.024203],
                [-91.195045, 40.024042],
                [-91.195471, 40.023859],
                [-91.195954, 40.023686],
                [-91.195981, 40.023445],
                [-91.196013, 40.023319],
                [-91.196045, 40.023137],
                [-91.196055, 40.023002],
                [-91.192161, 40.024143],
                [-91.188828, 40.025144],
                [-91.182899, 40.026881]]]}}
```

```
In [15]: #Create Empty Pandas DF
         df_geoZips = pd.DataFrame(columns=['Zipcode','Latitude','Longitude'])
         df_geoZips
```

```
Out[15]: Empty DataFrame
         Columns: [Zipcode, Latitude, Longitude]
         Index: []
```

```
In [16]: validZips = []
         #zz = set(df_chicago.iloc[:,0].values)
```

```
zz = set(df_chicago_only.iloc[:,0].values)
for i in range(len(geo['features'])):
    zi = geo['features'][i]['properties']["ZCTA5CE10"]
    lat = geo['features'][i]['properties']['INTPTLAT10']
    long = geo['features'][i]['properties']['INTPTLON10']

    if(zi in zz):
        validZips.append(geo['features'][i])
        df_geoZips = df_geoZips.append(dict(zip(df_geoZips.columns,[zi,lat,long])),ig
df_geoZips.head()
```

```
Out[16]:   Zipcode     Latitude     Longitude
        0    60656  +41.9742801  -087.8271313
        1    60638  +41.7814424  -087.7705341
        2    60652  +41.7479398  -087.7148066
        3    60629  +41.7758678  -087.7114956
        4    60641  +41.9466055  -087.7467867
```

```
In [17]: #validZips = set(df_chicago.iloc[:,0].values)
         #geoData = []
         #for i in range(len(geo['features'])):
         #    z = geo['features'][i]['properties']["ZCTA5CE10"]
         #    if(z in zz):
         #
         #df_geoZips.to_csv(r'D:\Desktop\outcomes\chicago_geozips.csv')
```

```
In [18]: geo['features'][0]
```

```
Out[18]: {'type': 'Feature',
          'properties': {'STATEFP10': '17',
           'ZCTA5CE10': '62359',
           'GEOID10': '1762359',
           'CLASSFP10': 'B5',
           'MTFCC10': 'G6350',
           'FUNCSTAT10': 'S',
           'ALAND10': 10360074,
           'AWATER10': 7921,
           'INTPTLAT10': '+40.0338795',
           'INTPTLON10': '-091.2014548',
           'PARTFLG10': 'N'},
          'geometry': {'type': 'Polygon',
           'coordinates': [[[-91.182899, 40.026881],
             [-91.182577, 40.026761],
             [-91.182428, 40.026711],
             [-91.182125, 40.026608],
             [-91.181677, 40.02648],
             [-91.181419, 40.026419],
             [-91.18093, 40.026323],
             [-91.180498, 40.026255],
[-91.180081, 40.026205],
[-91.179637, 40.026169],
[-91.179213, 40.026161],
[-91.1788, 40.026162],
[-91.178347, 40.026177],
[-91.177816, 40.026225],
[-91.177419, 40.026285],
[-91.177051, 40.026348],
[-91.176668, 40.026429],
[-91.176692, 40.02387],
[-91.176692, 40.023811],
[-91.176698, 40.021465],
[-91.176684, 40.021291],
[-91.176632, 40.021046],
[-91.176601, 40.020977],
[-91.17656, 40.020918],
[-91.176809, 40.020901],
[-91.181786, 40.020983],
[-91.183933, 40.021028],
[-91.184539, 40.021028],
[-91.185146, 40.021048],
[-91.186161, 40.021048],
[-91.187006, 40.021065],
[-91.188558, 40.021081],
[-91.189561, 40.021092],
[-91.190867, 40.021127],
[-91.192906, 40.021143],
[-91.195194, 40.02118],
[-91.195837, 40.021186],
[-91.195938, 40.021205],
[-91.19602, 40.021227],
[-91.196098, 40.021198],
[-91.196205, 40.021182],
[-91.19714, 40.021194],
[-91.198638, 40.021212],
[-91.199963, 40.021235],
[-91.20207, 40.021229],
[-91.20325, 40.021228],
[-91.203672, 40.021058],
[-91.204097, 40.020892],
[-91.204751, 40.020667],
[-91.205621, 40.020415],
[-91.209716, 40.019207],
[-91.211208, 40.018776],
[-91.212311, 40.018448],
[-91.213525, 40.018083],
[-91.215115, 40.01761],
[-91.215436, 40.017515],
[-91.216864, 40.017106],
[-91.218559, 40.016613],
[-91.218528, 40.0165],
[-91.219713, 40.015903],
[-91.223712, 40.014288],
[-91.223724, 40.014438],
[-91.223712, 40.014737],
[-91.223719, 40.014854],
[-91.22374, 40.014976],
[-91.223767, 40.015041],
[-91.223806, 40.015087],
[-91.223739, 40.015107],
[-91.223732, 40.015201],
[-91.223731, 40.015787],
[-91.22375, 40.016405],
[-91.223782, 40.016548],
[-91.223803, 40.016711],
[-91.223805, 40.016836],
[-91.223774, 40.016985],
[-91.223782, 40.017226],
[-91.223776, 40.017334],
[-91.223748, 40.017432],
[-91.223704, 40.017521],
[-91.223683, 40.017618],
[-91.223685, 40.017735],
[-91.22371, 40.018118],
[-91.223745, 40.01823],
[-91.223765, 40.018575],
[-91.22379, 40.019433],
[-91.223837, 40.020565],
[-91.223887, 40.021165],
[-91.223889, 40.021508],
[-91.223923, 40.021562],
[-91.223934, 40.021838],
[-91.223435, 40.033494],
[-91.223308, 40.036756],
[-91.223186, 40.047344],
[-91.223201, 40.047611],
[-91.223269, 40.047759],
[-91.223207, 40.047765],
[-91.223054, 40.04779],
[-91.222968, 40.047873],
[-91.222955, 40.049292],
[-91.221448, 40.049295],
[-91.204922, 40.049285],
[-91.202708, 40.049284],
[-91.195619, 40.049275],
[-91.195618, 40.048968],
 [-91.19563, 40.048033],
 [-91.195671, 40.045868],
 [-91.195692, 40.045024],
 [-91.195707, 40.043059],
 [-91.195754, 40.039882],
 [-91.195772, 40.039441],
 [-91.195771, 40.039127],
 [-91.195772, 40.038993],
 [-91.195803, 40.035795],
 [-91.195442, 40.035798],
 [-91.193694, 40.035771],
 [-91.192651, 40.035747],
 [-91.186579, 40.035637],
 [-91.184871, 40.035597],
 [-91.183401, 40.035576],
 [-91.18266, 40.035561],
 [-91.180703, 40.035531],
 [-91.17831, 40.035483],
 [-91.176636, 40.035458],
 [-91.176645, 40.033093],
 [-91.17664, 40.031975],
 [-91.176666, 40.029181],
 [-91.176675, 40.028789],
 [-91.176672, 40.028705],
 [-91.177306, 40.02852],
 [-91.182777, 40.026919],
 [-91.182899, 40.026881]],
[[-91.182899, 40.026881],
 [-91.183078, 40.026948],
 [-91.183895, 40.027199],
 [-91.184208, 40.027285],
 [-91.184535, 40.027364],
 [-91.184908, 40.027436],
 [-91.185281, 40.027492],
 [-91.185605, 40.027535],
 [-91.185982, 40.027566],
 [-91.186473, 40.02759],
 [-91.18686, 40.027595],
 [-91.187246, 40.02758],
 [-91.187975, 40.027517],
 [-91.188361, 40.027459],
 [-91.188724, 40.027393],
 [-91.189076, 40.027318],
 [-91.189458, 40.027223],
 [-91.189847, 40.027106],
 [-91.190206, 40.026979],
 [-91.190531, 40.026856],
 [-91.190878, 40.026716],
[-91.1912, 40.026561],
[-91.191838, 40.026201],
[-91.19221, 40.02596],
[-91.192493, 40.025755],
[-91.193464, 40.02498],
[-91.193824, 40.024716],
[-91.194436, 40.024343],
[-91.194698, 40.024203],
[-91.195045, 40.024042],
[-91.195471, 40.023859],
[-91.195954, 40.023686],
[-91.195981, 40.023445],
[-91.196013, 40.023319],
[-91.196045, 40.023137],
[-91.196055, 40.023002],
[-91.192161, 40.024143],
[-91.188828, 40.025144],
[-91.182899, 40.026881]]]}}
```

6 Folium Beta Visual

This is a pre-calculations visual of our selected zipcodes. Each marker is placed in the center of our target zipcodes.

```
In [77]: map_chicago = folium.Map(location=[41.88, -87.62], zoom_start=10)

         for i in df_geoZips.values:
             t1 = float(i[1])
             t2 = float(i[2])
             #folium.Marker([t1,t2]).add_to(map_chicago)
             folium.CircleMarker([t1,t2],radius=5,color='blue',fill=True,fill_color='#3186cc',



         map_chicago

Out[77]: <folium.folium.Map at 0x21c821a2208>
```

7 Folium Geo visual

This is a pre-calculations visual of our illinois geojson data trimmed to the relevant zipcodes.

```
In [37]: map_chicago = folium.Map(location=[41.82, -87.62], zoom_start=9.5)

         print(len(validZips))
         for i in range(len(validZips)):
             folium.GeoJson(validZips[i],overlay=True,style_function= lambda x :{'fillColor':'
        map_chicago

65
```

```
Out[37]: <folium.folium.Map at 0x21c81cf4f28>
```

```
In [38]: #map_chicago.save("chicago.html")
```

8 chicago buisness data

Here we are bringing in data from the US Census to judge how many of each size of buisness are in our target areas. We can sort the data by NAICS code to mirror our foursquare data and gain further insight into where a good place for our buisness might be.

```
In [39]: #SOURCE
         #zbp16totals
         #https://www.census.gov/data/datasets/2016/econ/cbp/2016-cbp.html
```

Field Data
Name Type Description * ZIP C ZIP Code * NAICS C Industry Code - 6-digit NAICS code * EST N Total Number of Establishments * N1_4 N Number of Establishments: 1-4 Employee Size Class * N5_9 N Number of Establishments: 5-9 Employee Size Class * N10_19 N Number of Establishments: 10-19 Employee Size Class * N20_49 N Number of Establishments: 20-49 Employee Size Class * N50_99 N Number of Establishments: 50-99 Employee Size Class * N100_249 N Number of Establishments: 100-249 Employee Size Class * N250_499 N Number of Establishments: 250-499 Employee Size Class * N500_999 N Number of Establishments: 500-999 Employee Size Class * N1000 N Number of Establishments: 1,000 or More Employee Size Class

```
In [40]: df_ccd = pd.read_csv("zbp16detail.csv")
```

```
In [41]: print(df_ccd.head())
         #df_geoZips
         print(df_ccd.shape)
         distinct_zips = set(df_geoZips.iloc[:,0].values)
```

```
   zip   naics  est  n1_4  n5_9  n10_19  n20_49  n50_99  n100_249  n250_499  \
0  501  ------    2     1     0       0       1       0         0         0
1  501  81----    2     1     0       0       1       0         0         0
2  501  813///    2     1     0       0       1       0         0         0
3  501  8131//    2     1     0       0       1       0         0         0
4  501  81311/    2     1     0       0       1       0         0         0

   n500_999  n1000
0         0      0
1         0      0
2         0      0
3         0      0
```

```
4          0       0
(8418283, 12)
```

```
In [42]: df_ccd = df_ccd[df_ccd["zip"].isin(distinct_zips)]


         df_ccd.shape

Out[42]: (42701, 12)
```

```
In [43]: codes = df_ccd.groupby(df_ccd["zip"]).groups
         print(len(codes.keys()))

65
```

9 Foursquare Data

The bulk of our insight is gained from foursquare data. Here we will utalize the API to gain insight into the buisnesses in each zipcode and cluster different zipcodes accordingly. Those clusters can then be compaired to our target buisness to find which cluster best fits our target buisness and further compared to our Census dataset

```
In [44]: #trending venues endpoint
         #means venues with the most people checked in
         #we can use this data for each zipcode along with the chicago buisness data
         #to find the zipcodes with the least amount of establishments but most
         #trending
```

```
In [45]: #Foursquare credentials
         client_id = 'your_ud'
         client_secret = 'your_secret'
         version = '20190526'
```

```
In [46]: radius = 100000
         LIMIT = 50
```

url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius=
CLIENT_ID, CLIENT_SECRET, VERSION, lat, lng, radius, LIMIT)

```
In [47]: def getTrending(lat,long):
             url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={
                 client_id,
                 client_secret,
                 version,
                 lat,
                 long,
                 radius,
                 LIMIT)
          # make the GET request
             #results = requests.get(url).json()["response"]['groups'][0]['items']
             results = requests.get(url).json()
             return results
```

```
In [48]: #60649        +41.7634204        -087.5658787
         fTest = getTrending(+41.7634204,-087.5658787)
```

```
In [49]: k = fTest['response']['groups'][0]['items']
         k[0]['venue']
```

```
Out[49]: {'id': '42eeb780f964a520b4261fe3',
          'name': 'Museum of Science and Industry',
          'location': {'address': '5700 S Lake Shore Dr',
           'crossStreet': 'at 57th Dr',
           'lat': 41.791617208319984,
           'lng': -87.58306656501914,
           'labeledLatLngs': [{'label': 'display',
             'lat': 41.791617208319984,
             'lng': -87.58306656501914}],
           'distance': 3447,
           'postalCode': '60637',
           'cc': 'US',
           'city': 'Chicago',
           'state': 'IL',
           'country': 'United States',
           'formattedAddress': ['5700 S Lake Shore Dr (at 57th Dr)',
            'Chicago, IL 60637',
            'United States']},
          'categories': [{'id': '4bf58dd8d48988d191941735',
            'name': 'Science Museum',
            'pluralName': 'Science Museums',
            'shortName': 'Science Museum',
            'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/museu
             'suffix': '.png'},
            'primary': True}],
          'photos': {'count': 0, 'groups': []},
          'venuePage': {'id': '85626555'}}
```

```
In [50]: i = [1234]
         for j in fTest['response']['groups'][0]['items']:
             v = dict(j)['venue']
             content = [i[0],v['name'],v['location']['lat'],v['location']['lng'],]
             print(content)
             break;
```

```
[1234, 'Museum of Science and Industry', 41.791617208319984, -87.58306656501914]
```

10 Testing out Locations

Here we gain human insight into our data by seeing what categorys tend to show up for each zipcode. This insight was also useful because it exposed that some buisnesses were being duplicated by the API (Airports, resturants, ect) and this was corrected.

```
In [51]: #Create Empty Pandas DF
         df_trends = pd.DataFrame(columns=['Zipcode','Name','Latitude','Longitude','Category']
         target_category = '5454144b498ec1f095bff2f2'
         #https://developer.foursquare.com/docs/resources/categories
         df_trends
```

```
Out[51]: Empty DataFrame
         Columns: [Zipcode, Name, Latitude, Longitude, Category]
         Index: []
```

```
In [52]: #df_geoZips

         for i in df_geoZips.values:
             trending_venues = getTrending(i[1],i[2])['response']['groups'][0]['items']
             for j in trending_venues:
                 v = dict(j)['venue']
                 content = [i[0],v['name'],v['location']['lat'],v['location']['lng'],v['catego
                 df_trends = df_trends.append(dict(zip(df_trends.columns,content)),ignore_inde
```

```
In [53]: df_trends.drop_duplicates(["Zipcode","Name"],inplace = True)
         len(df_trends)
         df_trends.head()
```

```
Out[53]:   Zipcode                                Name   Latitude  Longitude  \
         0   60656                  The Capital Grille  41.974923 -87.862916
         1   60656  Frank Lloyd Wright Home and Studio  41.894157 -87.799517
         2   60656                          Smoque BBQ  41.950168 -87.727684
         3   60656                        Trader Joe's  41.890123 -87.804593
         4   60656                          Portillo's  41.907365 -87.912586

                     Category
         0  American Restaurant
         1        Historic Site
         2            BBQ Joint
         3        Grocery Store
         4        Hot Dog Joint
```

```
In [54]: # one hot encoding
         df_trends_onehot = pd.get_dummies(df_trends[['Category']], prefix="", prefix_sep="")

         # add neighborhood column back to dataframe
         df_trends_onehot['Zipcode'] = df_trends['Zipcode']
```

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

df_trends_onehot.head()
```

Out[54]:

```
  Zipcode  African Restaurant  American Restaurant  Amphitheater  \
0   60656                   0                    1             0   
1   60656                   0                    0             0   
2   60656                   0                    0             0   
3   60656                   0                    0             0   
4   60656                   0                    0             0   

   Antique Shop  Art Gallery  Art Museum  Asian Restaurant  BBQ Joint  Bakery  \
0             0            0           0                 0          0       0   
1             0            0           0                 0          0       0   
2             0            0           0                 0          1       0   
3             0            0           0                 0          0       0   
4             0            0           0                 0          0       0   

   Bar  Baseball Stadium  Beach  Beer Bar  Beer Store  Boat or Ferry  \
0    0                 0      0         0           0              0   
1    0                 0      0         0           0              0   
2    0                 0      0         0           0              0   
3    0                 0      0         0           0              0   
4    0                 0      0         0           0              0   

   Bookstore  Breakfast Spot  Brewery  Butcher  Café  Chinese Restaurant  \
0          0               0        0        0     0                   0   
1          0               0        0        0     0                   0   
2          0               0        0        0     0                   0   
3          0               0        0        0     0                   0   
4          0               0        0        0     0                   0   

   Chocolate Shop  Climbing Gym  Clothing Store  Cocktail Bar  Coffee Shop  \
0               0             0               0             0            0   
1               0             0               0             0            0   
2               0             0               0             0            0   
3               0             0               0             0            0   
4               0             0               0             0            0   

   Comedy Club  Concert Hall  Cosmetics Shop  Cupcake Shop  Cycle Studio  \
0            0             0               0             0             0   
1            0             0               0             0             0   
2            0             0               0             0             0   
3            0             0               0             0             0   
4            0             0               0             0             0   
```

```
   Deli / Bodega  Dessert Shop  Diner  Donut Shop  Electronics Store  \
0              0             0      0           0                  0
1              0             0      0           0                  0
2              0             0      0           0                  0
3              0             0      0           0                  0
4              0             0      0           0                  0

   Farmers Market  Field  Flower Shop  French Restaurant  Frozen Yogurt Shop  \
0               0      0            0                  0                   0
1               0      0            0                  0                   0
2               0      0            0                  0                   0
3               0      0            0                  0                   0
4               0      0            0                  0                   0

   Furniture / Home Store  Garden  Garden Center  Gourmet Shop  Grocery Store  \
0                       0       0              0             0              0
1                       0       0              0             0              0
2                       0       0              0             0              0
3                       0       0              0             0              1
4                       0       0              0             0              0

   Gym  Gym / Fitness Center  Historic Site  History Museum  Hot Dog Joint  \
0    0                     0              0               0              0
1    0                     0              1               0              0
2    0                     0              0               0              0
3    0                     0              0               0              0
4    0                     0              0               0              1

   Hotel  Ice Cream Shop  Indie Movie Theater  Italian Restaurant  \
0      0               0                    0                   0
1      0               0                    0                   0
2      0               0                    0                   0
3      0               0                    0                   0
4      0               0                    0                   0

   Japanese Restaurant  Korean Restaurant  Lingerie Store  Liquor Store  \
0                    0                  0               0             0
1                    0                  0               0             0
2                    0                  0               0             0
3                    0                  0               0             0
4                    0                  0               0             0

   Mediterranean Restaurant  Mexican Restaurant  \
0                         0                   0
1                         0                   0
2                         0                   0
3                         0                   0
```

```
4                         0                   0

   Molecular Gastronomy Restaurant  Museum  Music School  Music Venue  \
0                                0       0             0            0
1                                0       0             0            0
2                                0       0             0            0
3                                0       0             0            0
4                                0       0             0            0

   Nature Preserve  New American Restaurant  Optical Shop  \
0                0                        0             0
1                0                        0             0
2                0                        0             0
3                0                        0             0
4                0                        0             0

   Other Great Outdoors  Outdoor Sculpture  Park  Pie Shop  Pizza Place  \
0                     0                  0     0         0            0
1                     0                  0     0         0            0
2                     0                  0     0         0            0
3                     0                  0     0         0            0
4                     0                  0     0         0            0

   Rock Club  Salad Place  Salon / Barbershop  Sandwich Place  Science Museum  \
0          0            0                   0               0               0
1          0            0                   0               0               0
2          0            0                   0               0               0
3          0            0                   0               0               0
4          0            0                   0               0               0

   Seafood Restaurant  Spa  Stadium  Sushi Restaurant  Tapas Restaurant  \
0                   0    0        0                 0                 0
1                   0    0        0                 0                 0
2                   0    0        0                 0                 0
3                   0    0        0                 0                 0
4                   0    0        0                 0                 0

   Theater  Trail  Vegetarian / Vegan Restaurant  Waterfront  Yoga Studio  \
0        0      0                              0           0            0
1        0      0                              0           0            0
2        0      0                              0           0            0
3        0      0                              0           0            0
4        0      0                              0           0            0

   Zoo  Zoo Exhibit
0    0            0
1    0            0
2    0            0
```

```
3    0            0
4    0            0
```

```
In [55]: df_trends_grouped = df_trends_onehot.groupby('Zipcode').mean().reset_index()
         df_trends_grouped.head()
```

```
Out[55]:   Zipcode  African Restaurant  American Restaurant  Amphitheater  \
        0    60411            0.021277                  0.0      0.021277
        1    60415            0.020833                  0.0      0.000000
        2    60601            0.000000                  0.0      0.020000
        3    60602            0.000000                  0.0      0.020408
        4    60603            0.000000                  0.0      0.020408

           Antique Shop  Art Gallery  Art Museum  Asian Restaurant  BBQ Joint  \
        0           0.0     0.021277    0.021277          0.021277   0.021277
        1           0.0     0.020833    0.020833          0.020833   0.020833
        2           0.0     0.000000    0.020000          0.000000   0.020000
        3           0.0     0.000000    0.020408          0.000000   0.020408
        4           0.0     0.000000    0.020408          0.000000   0.020408

             Bakery       Bar  Baseball Stadium  Beach  Beer Bar  Beer Store  \
        0  0.021277  0.042553               0.0    0.0       0.0         0.0
        1  0.020833  0.041667               0.0    0.0       0.0         0.0
        2  0.000000  0.020000               0.0    0.0       0.0         0.0
        3  0.000000  0.020408               0.0    0.0       0.0         0.0
        4  0.000000  0.020408               0.0    0.0       0.0         0.0

           Boat or Ferry  Bookstore  Breakfast Spot   Brewery  Butcher  Café  \
        0       0.021277        0.0        0.021277  0.042553      0.0   0.0
        1       0.000000        0.0        0.020833  0.062500      0.0   0.0
        2       0.040000        0.0        0.000000  0.000000      0.0   0.0
        3       0.040816        0.0        0.000000  0.000000      0.0   0.0
        4       0.040816        0.0        0.000000  0.000000      0.0   0.0

           Chinese Restaurant  Chocolate Shop  Climbing Gym  Clothing Store  \
        0            0.021277        0.021277           0.0             0.0
        1            0.020833        0.000000           0.0             0.0
        2            0.000000        0.020000           0.0             0.0
        3            0.000000        0.020408           0.0             0.0
        4            0.000000        0.020408           0.0             0.0

           Cocktail Bar  Coffee Shop  Comedy Club  Concert Hall  Cosmetics Shop  \
        0           0.0     0.021277          0.0      0.021277        0.000000
        1           0.0     0.020833          0.0      0.020833        0.000000
        2           0.0     0.040000          0.0      0.020000        0.020000
        3           0.0     0.061224          0.0      0.020408        0.020408
        4           0.0     0.061224          0.0      0.020408        0.020408
```

```
   Cupcake Shop  Cycle Studio  Deli / Bodega  Dessert Shop  Diner  Donut Shop  \
0          0.00           0.0       0.021277           0.0    0.0    0.000000
1          0.00           0.0       0.020833           0.0    0.0    0.000000
2          0.02           0.0       0.020000           0.0    0.0    0.020000
3          0.00           0.0       0.020408           0.0    0.0    0.020408
4          0.00           0.0       0.020408           0.0    0.0    0.020408

   Electronics Store  Farmers Market  Field  Flower Shop  French Restaurant  \
0           0.000000             0.0    0.0     0.000000           0.000000
1           0.000000             0.0    0.0     0.020833           0.020833
2           0.020000             0.0    0.0     0.000000           0.020000
3           0.020408             0.0    0.0     0.000000           0.020408
4           0.020408             0.0    0.0     0.000000           0.020408

   Frozen Yogurt Shop  Furniture / Home Store    Garden  Garden Center  \
0                 0.0                     0.0  0.000000            0.0
1                 0.0                     0.0  0.020833            0.0
2                 0.0                     0.0  0.000000            0.0
3                 0.0                     0.0  0.000000            0.0
4                 0.0                     0.0  0.000000            0.0

   Gourmet Shop  Grocery Store       Gym  Gym / Fitness Center  Historic Site  \
0      0.000000       0.021277  0.021277              0.000000       0.000000
1      0.000000       0.041667  0.020833              0.000000       0.020833
2      0.020000       0.020000  0.020000              0.020000       0.000000
3      0.020408       0.020408  0.020408              0.020408       0.000000
4      0.000000       0.020408  0.020408              0.020408       0.000000

   History Museum  Hot Dog Joint     Hotel  Ice Cream Shop  \
0        0.042553       0.021277  0.042553        0.042553
1        0.041667       0.020833  0.020833        0.062500
2        0.000000       0.000000  0.100000        0.000000
3        0.000000       0.000000  0.102041        0.000000
4        0.020408       0.000000  0.102041        0.000000

   Indie Movie Theater  Italian Restaurant  Japanese Restaurant  \
0                  0.0            0.000000                  0.0
1                  0.0            0.000000                  0.0
2                  0.0            0.020000                  0.0
3                  0.0            0.020408                  0.0
4                  0.0            0.020408                  0.0

   Korean Restaurant  Lingerie Store  Liquor Store  Mediterranean Restaurant  \
0                0.0        0.021277      0.021277                  0.021277
1                0.0        0.020833      0.020833                  0.020833
2                0.0        0.020000      0.020000                  0.040000
3                0.0        0.020408      0.020408                  0.040816
4                0.0        0.020408      0.020408                  0.020408
```

```
   Mexican Restaurant  Molecular Gastronomy Restaurant    Museum  \
0                 0.0                              0.0  0.000000   
1                 0.0                              0.0  0.000000   
2                 0.0                              0.0  0.020000   
3                 0.0                              0.0  0.020408   
4                 0.0                              0.0  0.020408   

   Music School  Music Venue  Nature Preserve  New American Restaurant  \
0           0.0          0.0         0.021277                 0.000000   
1           0.0          0.0         0.020833                 0.000000   
2           0.0          0.0         0.000000                 0.020000   
3           0.0          0.0         0.000000                 0.020408   
4           0.0          0.0         0.000000                 0.020408   

   Optical Shop  Other Great Outdoors  Outdoor Sculpture      Park  Pie Shop  \
0      0.000000              0.021277           0.021277  0.127660       0.0   
1      0.000000              0.000000           0.000000  0.083333       0.0   
2      0.020000              0.000000           0.020000  0.080000       0.0   
3      0.020408              0.000000           0.020408  0.081633       0.0   
4      0.020408              0.000000           0.020408  0.102041       0.0   

   Pizza Place  Rock Club  Salad Place  Salon / Barbershop  Sandwich Place  \
0     0.021277   0.021277     0.000000                0.00        0.000000   
1     0.041667   0.020833     0.000000                0.00        0.020833   
2     0.000000   0.000000     0.020000                0.02        0.000000   
3     0.000000   0.000000     0.020408                0.00        0.020408   
4     0.000000   0.000000     0.020408                0.00        0.000000   

   Science Museum  Seafood Restaurant  Spa   Stadium  Sushi Restaurant  \
0        0.021277            0.000000  0.0  0.000000               0.0   
1        0.020833            0.000000  0.0  0.020833               0.0   
2        0.000000            0.040000  0.0  0.000000               0.0   
3        0.000000            0.040816  0.0  0.000000               0.0   
4        0.000000            0.040816  0.0  0.000000               0.0   

   Tapas Restaurant   Theater     Trail  Vegetarian / Vegan Restaurant  \
0               0.0  0.021277  0.021277                            0.0   
1               0.0  0.020833  0.000000                            0.0   
2               0.0  0.060000  0.020000                            0.0   
3               0.0  0.061224  0.020408                            0.0   
4               0.0  0.081633  0.020408                            0.0   

   Waterfront  Yoga Studio  Zoo  Zoo Exhibit  
0    0.042553     0.021277  0.0          0.0  
1    0.020833     0.041667  0.0          0.0  
2    0.040000     0.040000  0.0          0.0  
3    0.040816     0.020408  0.0          0.0  
```

```
4    0.040816     0.020408  0.0          0.0
```

11 Venue Categories

To make calculations easier later on and create a nicer input interface the venue categories are called down from the API and sorted according to category teirs.

```
In [56]: #https://api.foursquare.com/v2/venues/categories
         #Create Empty Pandas DF
         df_category = pd.DataFrame(columns=['Category','Subcategory','Sub-Subcategory'])
         df_category
```

```
Out[56]: Empty DataFrame
         Columns: [Category, Subcategory, Sub-Subcategory]
         Index: []
```

```
In [57]: url = 'https://api.foursquare.com/v2/venues/categories?&client_id={}&client_secret={}&
                     client_id,
                     client_secret,version )

         categories = requests.get(url).json()['response']['categories']
```

```
In [58]: categories[0]['categories'][20]['categories']
```

```
Out[58]: [{'id': '4bf58dd8d48988d18f941735',
           'name': 'Art Museum',
           'pluralName': 'Art Museums',
           'shortName': 'Art Museum',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/museu
            'suffix': '.png'},
           'categories': []},
          {'id': '559acbe0498e472f1a53fa23',
           'name': 'Erotic Museum',
           'pluralName': 'Erotic Museums',
           'shortName': 'Erotic Museum',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/nightlife/stripclub_',
            'suffix': '.png'},
           'categories': []},
          {'id': '4bf58dd8d48988d190941735',
           'name': 'History Museum',
           'pluralName': 'History Museums',
           'shortName': 'History Museum',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/museu
            'suffix': '.png'},
           'categories': []},
          {'id': '4bf58dd8d48988d192941735',
           'name': 'Planetarium',
           'pluralName': 'Planetariums',
           'shortName': 'Planetarium',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/museu
            'suffix': '.png'},
           'categories': []},
          {'id': '4bf58dd8d48988d191941735',
           'name': 'Science Museum',
           'pluralName': 'Science Museums',
           'shortName': 'Science Museum',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/museu
            'suffix': '.png'},
           'categories': []}]
```

```
In [59]: for k in categories:
             for i in k['categories']:
                 if(len(i['categories']) > 0):
                     for j in i['categories']:
                         df_category = df_category.append(dict(zip(df_category.columns,[k['name
                     else:
                         df_category = df_category.append(dict(zip(df_category.columns,[k['name



         df_category.head()
```

```
Out[59]:                Category    Subcategory      Sub-Subcategory
         0  Arts & Entertainment  Movie Theater     Drive-in Theater
         1  Arts & Entertainment  Movie Theater  Indie Movie Theater
         2  Arts & Entertainment  Movie Theater            Multiplex
         3  Arts & Entertainment  Movie Theater        Movie Theater
         4  Arts & Entertainment         Museum           Art Museum
```

```
In [60]: index = df_category[df_category['Subcategory'] == 'Stadium']
         print(index)
         df_trends[df_trends['Category'].isin(index['Sub-Subcategory'].values)]
```

```
                Category Subcategory     Sub-Subcategory
22  Arts & Entertainment     Stadium    Baseball Stadium
23  Arts & Entertainment     Stadium  Basketball Stadium
24  Arts & Entertainment     Stadium      Cricket Ground
25  Arts & Entertainment     Stadium    Football Stadium
26  Arts & Entertainment     Stadium        Hockey Arena
27  Arts & Entertainment     Stadium       Rugby Stadium
28  Arts & Entertainment     Stadium      Soccer Stadium
29  Arts & Entertainment     Stadium      Tennis Stadium
30  Arts & Entertainment     Stadium       Track Stadium
31  Arts & Entertainment     Stadium             Stadium
```

```
Out[60]:      Zipcode           Name   Latitude  Longitude          Category
         67     60638  United Center  41.880759 -87.673974           Stadium
```

```
129    60652  United Center  41.880759 -87.673974           Stadium
174    60629  United Center  41.880759 -87.673974           Stadium
278    60625  Wrigley Field  41.948160 -87.655562  Baseball Stadium
373    60626  Wrigley Field  41.948160 -87.655562  Baseball Stadium
591    60630  Wrigley Field  41.948160 -87.655562  Baseball Stadium
627    60651  United Center  41.880759 -87.673974           Stadium
678    60645  Wrigley Field  41.948160 -87.655562  Baseball Stadium
788    60803  United Center  41.880759 -87.673974           Stadium
836    60712  Wrigley Field  41.948160 -87.655562  Baseball Stadium
854    60623  United Center  41.880759 -87.673974           Stadium
911    60608  United Center  41.880759 -87.673974           Stadium
950    60612  United Center  41.880759 -87.673974           Stadium
1076   60659  Wrigley Field  41.948160 -87.655562  Baseball Stadium
1132   60415  United Center  41.880759 -87.673974           Stadium
1209   60624  United Center  41.880759 -87.673974           Stadium
1332   60607  United Center  41.880759 -87.673974           Stadium
1367   60657  Wrigley Field  41.948160 -87.655562  Baseball Stadium
1405   60613  Wrigley Field  41.948160 -87.655562  Baseball Stadium
1584   60805  United Center  41.880759 -87.673974           Stadium
1720   60640  Wrigley Field  41.948160 -87.655562  Baseball Stadium
1811   60632  United Center  41.880759 -87.673974           Stadium
1895   60643  United Center  41.880759 -87.673974           Stadium
1940   60620  United Center  41.880759 -87.673974           Stadium
1988   60636  United Center  41.880759 -87.673974           Stadium
2033   60609  United Center  41.880759 -87.673974           Stadium
2339   60655  United Center  41.880759 -87.673974           Stadium
2420   60644  United Center  41.880759 -87.673974           Stadium
2484   60618  Wrigley Field  41.948160 -87.655562  Baseball Stadium
2670   60660  Wrigley Field  41.948160 -87.655562  Baseball Stadium
2713   60804  United Center  41.880759 -87.673974           Stadium
2793   60707  United Center  41.880759 -87.673974           Stadium
2944   60621  United Center  41.880759 -87.673974           Stadium
3089   60646  Wrigley Field  41.948160 -87.655562  Baseball Stadium
3147   60639  United Center  41.880759 -87.673974           Stadium
3178   60622  United Center  41.880759 -87.673974           Stadium
```

12 Collection of Dataframes

Below is a detail of all of our collected dataframe thusfar and their held data. In total 7 dataframe were examined to give us great insight into the chicago buisness climate. With this data we can now proceed into final calculations.

- df_category = [CATEGORY,SUBCATEGORY,SUB-Subcategory]

- df_trends_grouped = [Onehot encoded near buisnesses by category]

- df_trends = [closest buisnesses and their categories]

- df_ccd = [chicago census data for buisnesses]
- df_geoZips = [zip, lat ,long]
- df_grouped = [all zipcodes for each neighborhood]
- df_chicago = [original scrapped data]

```
In [61]: def mainCatPrintout():
             types = df_category.Subcategory.unique()
             print("Please Select a type:")
             for i in range(0,len(types),3):
                 print("%-30s %-30s %s" %(str(i)+":"+types[i],str(i+1)+":"+types[i+1],str(i+2)


         def getmainCatSelection(index):
             index = int(index)
             if(index >= 0 and index < 52):
                 sc = getGeoCats(df_category.Subcategory.unique()[index])
                 #print(sc)
                 return sc
             else:
                 return "Selection Not Found. Please Try Again"


         def getGeoCats(category_name):
             index = df_category[df_category['Subcategory'] == category_name]
             mc = index.values[0,0]
             sc = index['Sub-Subcategory'].values
             #print(sc)
             return sc

         #mapping of input below to a NAICS code
         #https://www.naics.com/business-lists/counts-by-naics-code/?#countsByNAICS
         naics_codes = {0:71,1:61,2:71,3:71,4:81,5:71,6:71,7:71,8:61,9:61,10:72,
                        11:72,12:72,13:72,14:72,15:72,16:72,17:72,18:72,19:72,20:72,
                        21:72,22:72,23:72,24:72,25:72,26:72,27:72,28:72,29:72,30:72,
                        31:72,32:44,33:11,34:11,35:92,36:71,37:71,38:92,39:62,40:55,
                        41:61,42:71,43:62,44:44,45:72,46:42,47:81,48:48,49:53,50:48}

         def getNaicsData(index):
             return df_ccd[df_ccd["naics"].str[0:2] == str(naics_codes[int(selection)])]

         def getFoursquareData():
             limit = 10
             indicators = ['st', 'nd', 'rd']
             # create columns according to number of top venues
             columns = ['Zipcode']
             for ind in np.arange(limit):
                 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
             df_commons = pd.DataFrame(columns=columns)
             df_commons['Zipcode'] = df_trends_grouped['Zipcode']
             for ind in np.arange(df_trends_grouped.shape[0]):
                 df_commons.iloc[ind, 1:] = getMostCommon(df_trends_grouped.iloc[ind, :], limi

             return df_commons

         def getMostCommon(row, limit):
             row_categories = row.iloc[1:]
             row_categories_sorted = row_categories.sort_values(ascending=False)

             return row_categories_sorted.index.values[0:limit]

In [62]: getFoursquareData().head()

Out[62]:   Zipcode 1st Most Common Venue 2nd Most Common Venue 3rd Most Common Venue  \
         0   60411                  Park        History Museum            Waterfront
         1   60415                  Park               Brewery        Ice Cream Shop
         2   60601                 Hotel                  Park               Theater
         3   60602                 Hotel                  Park               Theater
         4   60603                  Park                 Hotel               Theater

           4th Most Common Venue 5th Most Common Venue     6th Most Common Venue  \
         0                 Hotel        Ice Cream Shop                       Bar
         1         Grocery Store                   Bar               Pizza Place
         2           Coffee Shop           Yoga Studio                Waterfront
         3           Coffee Shop    Seafood Restaurant  Mediterranean Restaurant
         4           Coffee Shop    Seafood Restaurant             Boat or Ferry

               7th Most Common Venue     8th Most Common Venue  \
         0                   Brewery  Mediterranean Restaurant
         1            History Museum               Yoga Studio
         2  Mediterranean Restaurant        Seafood Restaurant
         3             Boat or Ferry                Waterfront
         4                Waterfront  Mediterranean Restaurant

              9th Most Common Venue 10th Most Common Venue
         0          Nature Preserve           Concert Hall
         1              Coffee Shop                 Garden
         2            Boat or Ferry   Gym / Fitness Center
         3           Cosmetics Shop           Concert Hall
         4  New American Restaurant                 Museum
```

13 Clustering on Foursqure Data

Here we utalize the encoded data from the foursquare API to cluster zipcodes according to buisness climates. This will form a large part of our predictions

```
In [63]: #Num clusters
         k = 5
         #dataSet = getFoursquareData().drop('Zipcode',1)

         kmc = KMeans(random_state=0)
         kmc.fit(df_trends_grouped.drop('Zipcode',1))
```

```
Out[63]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
             random_state=0, tol=0.0001, verbose=0)
```

```
In [64]: kmc.labels_
```

```
Out[64]: array([5, 5, 4, 4, 4, 4, 4, 4, 4, 1, 1, 4, 4, 3, 6, 0, 1, 1, 1, 2, 1, 1,
                1, 3, 3, 3, 6, 6, 5, 5, 2, 2, 5, 1, 7, 1, 1, 5, 7, 6, 7, 0, 5, 3,
                6, 2, 7, 1, 3, 5, 1, 4, 5, 2, 0, 6, 6, 4, 7, 7, 2, 5, 3, 5, 5])
```

```
In [65]: df_geoZips.sort_values(by=["Zipcode"],inplace=True)
         df_geoZips.insert(3,"Cluster",kmc.labels_,True)
```

```
In [66]: df_geoZips.head()
```

```
Out[66]:    Zipcode     Latitude     Longitude  Cluster
         44   60411  +41.5087744  -087.5903141        5
         22   60415  +41.7029482  -087.7788303        5
         20   60601  +41.8853104  -087.6221295        4
         52   60602  +41.8830726  -087.6291494        4
         33   60603  +41.8801879  -087.6255095        4
```

```
In [67]: validZips[0]['properties']['ZCTA5CE10']

         for i in validZips[0:1]:
             print(i['properties']['ZCTA5CE10'])
```

```
60656
```

```
In [68]: df_geoZips.head()
```

```
Out[68]:    Zipcode     Latitude     Longitude  Cluster
         44   60411  +41.5087744  -087.5903141        5
         22   60415  +41.7029482  -087.7788303        5
         20   60601  +41.8853104  -087.6221295        4
         52   60602  +41.8830726  -087.6291494        4
         33   60603  +41.8801879  -087.6255095        4
```

14 Cluster Map

This map represents the clustered data. All that remains is a cost function analysis cloropleth map to be overlayed atop it to create final recommendations.

```
In [69]: map_chicago = folium.Map(location=[41.88, -87.62], zoom_start=10)
         numClusters = df_geoZips["Cluster"].max()
         x = np.arange(numClusters)
         ys = [i + x + (i*x)**2 for i in range(numClusters)]
         colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
         rainbow = [colors.rgb2hex(i) for i in colors_array]


         for i in df_geoZips.values:
             t1 = float(i[1])
             t2 = float(i[2])
             folium.CircleMarker([t1,t2],radius=5,color=rainbow[int(i[3]-1)],fill=True,fill_co


         for i in range(len(validZips)):
             clust = df_geoZips[df_geoZips["Zipcode"]==validZips[i]['properties']['ZCTA5CE10']]
             folium.GeoJson(validZips[i],style_function= lambda x: {'fillColor':'grey','color'


         map_chicago

Out[69]: <folium.folium.Map at 0x21c821a2ef0>
```

15 Recommendation Logic

Here lays the recommendation cost function for our analysis it attempts to score zipcodes based on the business opportunity by balancing the right amount of existing business presence (signaling a market/want) and threat of competition (too many small businesses or a few large businesses)

```
In [70]: fqd = getFoursquareData()
         def zipcodeScore(zipCode,ccdSmallBuisnessNum,selection):
             buisMod = 0
             clusterMod = 0
             if(ccdSmallBuisnessNum != 0 and (ccdSmallBuisnessNum > 8 or ccdSmallBuisnessNum <
                 buisMod = (abs(ccdSmallBuisnessNum-6)-2)*-1
             weight = -5
             for i in fqd[fqd["Zipcode"]==str(zipCode)].values[0]:
                 if(i not in getmainCatSelection(selection)):
                     clusterMod-=(5-abs(weight))
                     weight+=1
             return clusterMod+buisMod
         def recommendationEngine(selection):
             naics = getNaicsData(selection)
             scores = []
             for i in df_geoZips.values:
                 buisnesses = naics[naics["zip"]==int(i[0])]
                 bNum = buisnesses["n1_4"].sum() + buisnesses["n5_9"].sum()
                 scores.append(zipcodeScore(i[0],bNum,selection))
             df_geoZips.insert(4,"Score",scores,True)
             bestCluster = df_geoZips.iloc[df_geoZips[['Score']].idxmax()].values[0][3]
             for i in range(len(df_geoZips)):
                 if(df_geoZips.iloc[i,3]==bestCluster):
                     df_geoZips.iloc[i,4]+=30
             return df_geoZips
```

```
In [71]: df_geoZips.head()
```

```
Out[71]:    Zipcode     Latitude     Longitude  Cluster
         44   60411  +41.5087744  -087.5903141        5
         22   60415  +41.7029482  -087.7788303        5
         20   60601  +41.8853104  -087.6221295        4
         52   60602  +41.8830726  -087.6291494        4
         33   60603  +41.8801879  -087.6255095        4
```

```
In [76]: mainCatPrintout()
         selection = input()
         if((selection != None) and int(selection) >= 0 and int(selection) < 51):
             getmainCatSelection(selection)
             recommendationEngine(selection)
             map_chicago = folium.Map(location=[41.88, -87.62], zoom_start=10)

             # Add the color for the chloropleth:
             map_chicago.choropleth(
              geo_data=dict({"Type":"FeatureCollection","features":list(validZips)}),
              name='choropleth',
              data=df_geoZips,
              columns=['Zipcode', 'Score'],
              key_on='properties.ZCTA5CE10',
              fill_color='BuGn',
              fill_opacity=0.9,
              line_opacity=0.5,
              legend_name="Recommedation Cost Estimate"
             )
             folium.LayerControl().add_to(map_chicago)

             for i in df_geoZips.values:
                 t1 = float(i[1])
                 t2 = float(i[2])
                 folium.CircleMarker([t1,t2],radius=5,color=rainbow[int(i[3]-1)],fill=True,fil
             display(map_chicago)
             df_geoZips.drop(["Score"], axis=1,inplace=True)
         else:
             print("Please enter a valid Selection")
```

```
Please Select a type:
0:Movie Theater                1:Museum                       2:Music Venue
3:Performing Arts Venue        4:Public Art                   5:Stadium
6:Theme Park                   7:Zoo                          8:College Academic Building
9:College Stadium              10:African Restaurant          11:American Restaurant
12:Asian Restaurant            13:Caribbean Restaurant        14:Dessert Shop
15:Eastern European Restaurant 16:French Restaurant           17:German Restaurant
18:Greek Restaurant            19:Hawaiian Restaurant         20:Indian Restaurant
21:Italian Restaurant          22:Jewish Restaurant           23:Latin American Restaurant
24:Mediterranean Restaurant    25:Mexican Restaurant          26:Middle Eastern Restaurant
27:Russian Restaurant          28:Spanish Restaurant          29:Turkish Restaurant
30:Ukrainian Restaurant        31:Bar                         32:Athletics & Sports
33:Beach                       34:Ski Area                    35:States & Municipalities
36:Convention Center           37:Event Space                 38:Government Building
39:Medical Center              40:Office                      41:School
42:Spiritual Center            43:Child Care Service          44:Clothing Store
45:Food & Drink Shop           46:Furniture / Home Store      47:Airport
48:Bus Station                 49:Hotel                       50:Train Station
12
```

```
<folium.folium.Map at 0x21c828ddcf8>
```

```
In [ ]:
```