Finding a Place for a Restaurant in Raleigh

Introduction/Business Problem: In the city of Raleigh, if one wants to find an area to create a restaurant venue, where would you recommend opening one? There are a lot of restaurants in Raleigh, especially since it is the home of one of the state's largest universities, NC State. In addition, there are 15 other colleges and universities within local area. The background of the problem is, where might someone open a profitable restaurant venue that would maximize profits, as you want to not only take advantage of the fact that there are a significant amount of young people in the area, but so that you do not lose profits or customers to other surrounding entertainment venues?

The basic idea is to find a suitable location for an restaurant that is within Raleigh, that is suitable for not only business, but takes advantage of the proximity to colleges and universities in the area.

Data Collection: The project will collect data from the following sources:

• Foursquare API: use Foursquare to collect data from Raleigh and surrounding areas within 50 miles • Raleigh Open Data: contains API for both Colleges and Universities and Restaurants

Raleigh: Raleigh geographical area covers 144.8 square miles. The objective is to find probable areas that may be advantageous to opening a restaurant.

To begin, we need to identify the tools that are needed to pull data from the file.

In [6]:

```
# import the libraries we need
import pandas as pd
import geopy
from geopy.geocoders import Nominatim
import numpy as np
!pip install folium
import folium
```

```
Requirement already satisfied: folium in /opt/conda/envs/Python36/lib/python3.6/site-packages (0.10.
1)
Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (fro
m folium) (2.21.0)
Requirement already satisfied: jinja2>=2.9 in /opt/conda/envs/Python36/lib/python3.6/site-packages (
from folium) (2.10)
Requirement already satisfied: branca>=0.3.0 in /opt/conda/envs/Python36/lib/python3.6/site-packages
(from folium) (0.4.0)
Requirement already satisfied: numpy in /opt/conda/envs/Python36/lib/python3.6/site-packages (from f
olium) (1.15.4)
Requirement already satisfied: idna<2.9,>=2.5 in /opt/conda/envs/Python36/lib/python3.6/site-package
s (from requests->folium) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python36/lib/python3.6/site-pac
kages (from requests->folium) (2019.11.28)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /opt/conda/envs/Python36/lib/python3.6/site-
packages (from requests->folium) (3.0.4)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /opt/conda/envs/Python36/lib/python3.6/site-
packages (from requests->folium) (1.24.1)
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/envs/Python36/lib/python3.6/site-packa
ges (from jinja2>=2.9->folium) (1.1.0)
Requirement already satisfied: six in /opt/conda/envs/Python36/lib/python3.6/site-packages (from bra
nca>=0.3.0->folium) (1.12.0)
```

In [76]:

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
from tabulate import tabulate

file_name='https://opendata.arcgis.com/datasets/cfabd5fb515b4b45a2714d48fbcd2dd5_0.csv?outSR=%7B%22latestWkid%22%
3A2264%2C%22wkid%22%3A102719%7D&session=1196661311.1585645749.csv'
df=pd.read_csv(file_name)
```

In [77]:

```
print('shape: ',df.shape)
print('columns: ', df.columns)
df.head()
```

```
shape:  (18, 6)
columns:  Index(['X', 'Y', 'OBJECTID', 'ADDRESS', 'CITY', 'COLLEGENAME'], dtype='object')
```

Out[77]:

	X	Y	OBJECTID	ADDRESS	CITY	COLLEGENAME
0	2.097579e+06	741713.299902	1	2101 Hillsborough Street	Raleigh	NC State University - Main Campus
1	2.092281e+06	744629.599844	2	3800 Hillsborough Street	Raleigh	Meredith College
2	2.095116e+06	736981.100049	3	1007 Capability Road	Raleigh	NCSU Centennial Campus
3	2.107643e+06	735692.700066	4	118 South Street	Raleigh	Shaw University
4	2.112693e+06	741014.500118	5	1315 Oakwood Avenue	Raleigh	St. Augustine College

In [37]:

```
df.shape
```

Out[37]:

```
(3481, 15)
```

Methodology

For the below, wanted to provide location data for all the colleges and universities located in Raleigh.

In [80]:

```
df3 = df[['COLLEGENAME', 'X', 'Y']]
df3.head(20)
```

Out[80]:

	COLLEGENAME	X	Y
0	NC State University - Main Campus	2.097579e+06	741713.299902
1	Meredith College	2.092281e+06	744629.599844
2	NCSU Centennial Campus	2.095116e+06	736981.100049
3	Shaw University	2.107643e+06	735692.700066
4	St. Augustine College	2.112693e+06	741014.500118
5	Southeastern Baptist Seminary	2.144425e+06	812169.599934
6	Wake Technical College Southern Wake Campus	2.087623e+06	692108.900095
7	William Peace University	2.107440e+06	742210.100089
8	Wake Technical College Northern Wake Campus	2.134938e+06	770863.000133
9	Wake Technical College Western Wake Campus	2.060739e+06	711927.599959
10	Wake Technical College Perry Health Sciences C...	2.122848e+06	740698.100161
11	Wake Technical College RTP Campus	2.049020e+06	766080.999854
12	Wake Technical College Public Safety Education...	2.104581e+06	722558.999956
13	Vernon Malone College and Career Academy	2.105868e+06	729089.000019
14	Campbell University ext.	2.055586e+06	758889.400064
15	Campbell University Law School	2.105852e+06	739116.799846
16	Wake Technical College Beltline Education Center	2.114660e+06	755005.982634
17	Wake Technical College Eastern Wake Education ...	2.205384e+06	757267.557179

Next, we need to pull the data for restaurants in the Raleigh Area.

In [96]:

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
from tabulate import tabulate

file_name='https://opendata.arcgis.com/datasets/124c2187da8c41c59bde04fa67eb2872_0.csv?session=1285114555.1585645
226.csv'
df=pd.read_csv(file_name)
```

In [97]:

```
print('shape: ',df.shape)
print('columns: ', df.columns)
df.head()
```

```
shape:  (3481, 15)
columns:  Index(['OBJECTID', 'HSISID', 'NAME', 'ADDRESS1', 'ADDRESS2', 'CITY', 'STATE',
       'POSTALCODE', 'PHONENUMBER', 'RESTAURANTOPENDATE', 'FACILITYTYPE',
       'PERMITID', 'X', 'Y', 'GEOCODESTATUS'],
      dtype='object')
```

Out[97]:

	OBJECTID	HSISID	NAME	ADDRESS1	ADDRESS2	CITY	STATE	POSTALCODE	PHONENUMBER	RESTAURANTO
0	509616	4092018236	Tacos Don Marcos	9629 BITTER MELON DR	NaN	ANGIER	NC	27501	(919) 272-7062	2019-11-19T00:(
1	509841	4092050026	St Mary Magdalene School Cafeteria	625 Magdala PL	NaN	APEX	NC	27502	(919) 657-4800 ext. 298	2004-08-13T00:(
2	509842	4092015002	BAAN THAI RESTAURANT	758 WILLIAMS ST	NaN	APEX	NC	27502	(919) 629-6399	2007-11-09T00:(
3	509843	4092014554	KFC #J120103	1403 W Williams ST	NaN	APEX	NC	27502	(919) 303-7997	2006-04-25T00:(
4	509844	4092014694	Brueggers's Bagels	760 W Williams ST	NaN	APEX	NC	27502	(919) 367-7720	2006-09-29T00:(

In [98]:

```
df.shape
```

Out[98]:

```
(3481, 15)
```

In [99]:

```
df4= df[['NAME', 'X', 'Y']]
df4.head()
```

Out[99]:

	NAME	X	Y
0	Tacos Don Marcos	-78.747986	35.539550
1	St Mary Magdalene School Cafeteria	-78.874487	35.724227
2	BAAN THAI RESTAURANT	-78.864641	35.738078
3	KFC #J120103	-78.877270	35.750153
4	Brueggers's Bagels	-78.862761	35.738032

We also need to identify and map the city and gain the coordinates.

In [39]:

```
address = 'Raleigh City'
geolocator = Nominatim(user_agent='to_explorer')
location = geolocator.geocode(address)
print(location.latitude, location.longitude)
```

```
35.7803977 -78.6390989
```

In [41]:

```
CHD_LATITUDE = '35.7803977'
CHD_LONGITUDE = '-78.6390989'
print('The geograpical coordinates of Raleigh are {}, {}.'.format(CHD_LATITUDE, CHD_LONGITUDE))
```

```
The geograpical coordinates of Raleigh are 35.7803977, -78.6390989.
```

In [55]:

```
raleigh_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start = 13)
folium.Marker([CHD_LATITUDE, CHD_LONGITUDE]).add_to(raleigh_map)
```

Out[55]:

```
<folium.map.Marker at 0x7f64be258278>
```

In [57]:

```
raleigh_map
```

Out[57]:

After identifying Raleigh on the map, we then need to identify the venues using Foursquare as well as map the venues.

In [59]:

```
foursquare_client_ID = 'VAYCGD5DPRVQT5URMRBXLYILFFXVGOAVYNAMPXYGABJF1MV0'
foursquare_client_secret = 'I3TRZNORMFIVXTTOIJEFCBQC0SYS2ED4UQDMEV1UYWZLXYHS'
radius = 80467 # 50 mi = 80.4672Km
no_of_venues = 150
version = '20200331' # Current date
```

In [60]:

```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [61]:

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from pandas.io.json import json_normalize
import requests

pd.set_option('display.max_rows', None)

offset = 0
total_venues = 0
foursquare_venues = pd.DataFrame(columns = ['name', 'categories', 'lat', 'lng'])

while (True):
    url = ('https://api.foursquare.com/v2/venues/explore?client_id={}'
           '&client_secret={}&v={}&ll={},{}&radius={}&limit={}&offset={}').format(foursquare_client_ID,
                                                                              foursquare_client_secret,
                                                                              version,
                                                                              CHD_LATITUDE,
                                                                              CHD_LONGITUDE,
                                                                              radius,
                                                                              no_of_venues,
                                                                              offset)
    result = requests.get(url).json()
    venues_fetched = len(result['response']['groups'][0]['items'])
    total_venues = total_venues + venues_fetched
    print("Total {} venues fetched within a total radius of {} Km".format(venues_fetched, RADIUS/1000))

    venues = result['response']['groups'][0]['items']
    venues = json_normalize(venues)

    # Filter the columns
    filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
    venues = venues.loc[:, filtered_columns]

    # Filter the category for each row
    venues['venue.categories'] = venues.apply(get_category_type, axis = 1)

    # Clean all column names
    venues.columns = [col.split(".")[-1] for col in venues.columns]
    foursquare_venues = pd.concat([foursquare_venues, venues], axis = 0, sort = False)

    if (venues_fetched < 100):
        break
    else:
        offset = offset + 100

foursquare_venues = foursquare_venues.reset_index(drop = True)
print("\nTotal {} venues fetched".format(total_venues))
```

```
Total 100 venues fetched within a total radius of 80.467 Km
Total 100 venues fetched within a total radius of 80.467 Km
Total 46 venues fetched within a total radius of 80.467 Km

Total 246 venues fetched
```

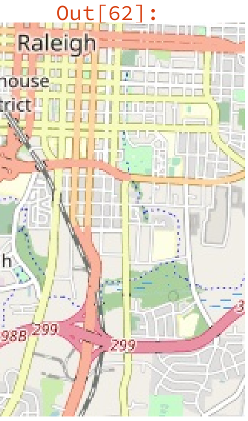
In [62]:

```
raleigh_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start = 13)

for name, latitude, longitude in zip(foursquare_venues['name'], foursquare_venues['lat'], foursquare_venues['lng'
]):
    label = '{}'.format(name)
    label = folium.Popup(label, parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 5,
        popup = label,
        color = 'green',
        fill = True,
        fill_color = '#3186cc',
        fill_opacity = 0.7,
        parse_html = False).add_to(raleigh_map)


raleigh_map
```

Out[62]:



In [146]:

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from pandas.io.json import json_normalize
import requests

pd.set_option('display.max_rows', None)

offset = 0
total_venues = 0
foursquare_venues = pd.DataFrame(columns = ['name', 'categories', 'lat', 'lng'])

while (True):
    url = ('https://api.foursquare.com/v2/venues/explore?client_id={}'
           '&client_secret={}&v={}&ll={},{}&radius={}&limit={}&offset={}').format(foursquare_client_ID,
                                                                        foursquare_client_secret,
                                                                        version,
                                                                        CHD_LATITUDE,
                                                                        CHD_LONGITUDE,
                                                                        radius,
                                                                        no_of_venues,
                                                                        offset)
    result = requests.get(url).json()
    venues_fetched = len(result['response']['groups'][0]['items'])
    total_venues = total_venues + venues_fetched
    print("Total {} venues fetched within a total radius of {} Km".format(venues_fetched, RADIUS/1000))

    venues = result['response']['groups'][0]['items']
    venues = json_normalize(venues)

    # Filter the columns
    filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
    venues = venues.loc[:, filtered_columns]

    # Filter the category for each row
    venues['venue.categories'] = venues.apply(get_category_type, axis = 1)

    # Clean all column names
    venues.columns = [col.split(".")[-1] for col in venues.columns]
    foursquare_venues = pd.concat([foursquare_venues, venues], axis = 0, sort = False)

    if (venues_fetched < 100):
        break
    else:
        offset = offset + 100

foursquare_venues = foursquare_venues.reset_index(drop = True)
print("\nTotal {} venues fetched".format(total_venues))
```

```
Total 100 venues fetched within a total radius of 80.467 Km
Total 100 venues fetched within a total radius of 80.467 Km
Total 46 venues fetched within a total radius of 80.467 Km

Total 246 venues fetched
```

We need to identify the proximity of the universities to the restaurants. However, there are 17 colleges and universities, and 3481 restaurants in the Raleigh area. There are 246 venues within a 50 mile radius of Raleigh. So we now need to reduce the amount of data that we will use, as it goes beyond the scope of what the Capstone intends. The data isn't as compatible as I believed when I began. I now will adjust the scope of the project to the scope of my experience. I will use a specific location, NC State University, and will attempt to determine the venues surrounding it.

In [154]:

```
!pip install geopandas
!pip install geopy
```

```
Collecting geopandas
  Downloading https://files.pythonhosted.org/packages/83/c5/3cf9cdc39a6f2552922f79915f36b45a95b71fd3
43cfc51170a5b6ddb6e8/geopandas-0.7.0-py2.py3-none-any.whl (928kB)
     |████████████████████████████████| 931kB 14.8MB/s eta 0:00:01
Requirement already satisfied: pandas>=0.23.0 in /opt/conda/envs/Python36/lib/python3.6/site-package
s (from geopandas) (0.24.1)
Collecting pyproj>=2.2.0 (from geopandas)
  Downloading https://files.pythonhosted.org/packages/ce/37/705ee471f71130d4ceee41bbcb06f3b52175cb89
273cbb5755ed5e6374e0/pyproj-2.6.0-cp36-cp36m-manylinux2010_x86_64.whl (10.4MB)
     |████████████████████████████████| 10.4MB 47.5MB/s eta 0:00:01
Collecting shapely (from geopandas)
  Downloading https://files.pythonhosted.org/packages/20/fa/c96d3461fda99ed8e82ff0b219ac2c8384694b4e
640a611a1a8390ecd415/Shapely-1.7.0-cp36-cp36m-manylinux1_x86_64.whl (1.8MB)
     |████████████████████████████████| 1.8MB 45.0MB/s eta 0:00:01
Collecting fiona (from geopandas)
  Downloading https://files.pythonhosted.org/packages/ec/20/4e63bc5c6e62df889297b382c3ccd4a7a488b009
46aaaf81a118158c6f09/Fiona-1.8.13.post1-cp36-cp36m-manylinux1_x86_64.whl (14.7MB)
     |████████████████████████████████| 14.7MB 25.9MB/s eta 0:00:01
Requirement already satisfied: pytz>=2011k in /opt/conda/envs/Python36/lib/python3.6/site-packages (
from pandas>=0.23.0->geopandas) (2018.9)
Requirement already satisfied: numpy>=1.12.0 in /opt/conda/envs/Python36/lib/python3.6/site-packages
(from pandas>=0.23.0->geopandas) (1.15.4)
Requirement already satisfied: python-dateutil>=2.5.0 in /opt/conda/envs/Python36/lib/python3.6/site
-packages (from pandas>=0.23.0->geopandas) (2.7.5)
Collecting munch (from fiona->geopandas)
  Downloading https://files.pythonhosted.org/packages/cc/ab/85d8da5c9a45e072301beb37ad7f833cd344e04c
817d97e0cc75681d248f/munch-2.5.0-py2.py3-none-any.whl
Requirement already satisfied: six>=1.7 in /opt/conda/envs/Python36/lib/python3.6/site-packages (fro
m fiona->geopandas) (1.12.0)
Collecting click-plugins>=1.0 (from fiona->geopandas)
  Downloading https://files.pythonhosted.org/packages/e9/da/824b92d9942f4e472702488857914bdd50f73021
efea15b4cad9aca8ecef/click_plugins-1.1.1-py2.py3-none-any.whl
Collecting cligj>=0.5 (from fiona->geopandas)
  Downloading https://files.pythonhosted.org/packages/e4/be/30a58b4b0733850280d01f8bd132591b4668ed5c
7046761098d665ac2174/cligj-0.5.0-py3-none-any.whl
Requirement already satisfied: click<8,>=4.0 in /opt/conda/envs/Python36/lib/python3.6/site-packages
(from fiona->geopandas) (7.0)
Requirement already satisfied: attrs>=17 in /opt/conda/envs/Python36/lib/python3.6/site-packages (fr
om fiona->geopandas) (18.2.0)
Installing collected packages: pyproj, shapely, munch, click-plugins, cligj, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.5.0 fiona-1.8.13.post1 geopandas-0.7.0 munch-2.5.
0 pyproj-2.6.0 shapely-1.7.0
Requirement already satisfied: geopy in /opt/conda/envs/Python36/lib/python3.6/site-packages (1.18.1
)
Requirement already satisfied: geographiclib<2,>=1.49 in /opt/conda/envs/Python36/lib/python3.6/site
-packages (from geopy) (1.49)
```

In [155]:

```
file_name='https://opendata.arcgis.com/datasets/cfabd5fb515b4b45a2714d48fbcd2dd5_0.csv?outSR=%7B%22latestWkid%22%
3A2264%2C%22wkid%22%3A102719%7D&session=1196661311.1585645749.csv'
df=pd.read_csv(file_name)
```

In [157]:

```
print('shape: ',df.shape)
print('columns: ', df.columns)
df.head()
```

```
shape:  (18, 6)
columns:  Index(['X', 'Y', 'OBJECTID', 'ADDRESS', 'CITY', 'COLLEGENAME'], dtype='object')
```

Out[157]:

	X	Y	OBJECTID	ADDRESS	CITY	COLLEGENAME
0	2.097579e+06	741713.299902	1	2101 Hillsborough Street	Raleigh	NC State University - Main Campus
1	2.092281e+06	744629.599844	2	3800 Hillsborough Street	Raleigh	Meredith College
2	2.095116e+06	736981.100049	3	1007 Capability Road	Raleigh	NCSU Centennial Campus
3	2.107643e+06	735692.700066	4	118 South Street	Raleigh	Shaw University
4	2.112693e+06	741014.500118	5	1315 Oakwood Avenue	Raleigh	St. Augustine College

In [160]:

```
from geopy.geocoders import Nominatim
geolocator = Nominatim()
location = geolocator.geocode("2101 Hillsborough Street NC")
print(location.address)
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/ipykernel/__main__.py:2: DeprecationWarning: Us
ing Nominatim with the default "geopy/1.18.1" `user_agent` is strongly discouraged, as it violates N
ominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ and may possibly cause 403 a
nd 429 HTTP errors. Please specify a custom `user_agent` with `Nominatim(user_agent="my-application"
)` or by overriding the default `user_agent`: `geopy.geocoders.options.default_user_agent = "my-appl
ication"`. In geopy 2.0 this will become an exception.
  from ipykernel import kernelapp as app

2101, Hillsborough Street, Oberlin, Wake County, North Carolina, 27607, United States of America
```

In [161]:

```
print((location.latitude, location.longitude))
```

```
(35.786319, -78.663314)
```

In [164]:

```
foursquare_client_ID = 'VAYCGD5DPRVQT5URMRBXLYILFFXVGOAVYNAMPXYGABJF1MV0'
foursquare_client_secret = 'I3TRZNORMFIVXTTOIJEFCBQC0SYS2ED4UQDMEV1UYWZLXYHS'
radius = 80467 # 50 mi = 80.4672Km
no_of_venues = 10
version = '20200331' # Current date
```

In [165]:

```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [274]:

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from pandas.io.json import json_normalize
import requests

pd.set_option('display.max_rows', None)

offset = 0
total_venues = 0
foursquare_venues = pd.DataFrame(columns = ['name', 'categories', 'lat', 'lng'])

while (True):
    url = ('https://api.foursquare.com/v2/venues/explore?client_id={}'
           '&client_secret={}&v={}&ll={},{}&radius={}&limit={}&offset={}').format(foursquare_client_ID,
                                                                        foursquare_client_secret,
                                                                        version,
                                                                        CHD_LATITUDE,
                                                                        CHD_LONGITUDE,
                                                                        radius,
                                                                        no_of_venues,
                                                                        offset)
    result = requests.get(url).json()
    venues_fetched = len(result['response']['groups'][0]['items'])
    total_venues = total_venues + venues_fetched
    print("Total {} venues fetched within a total radius of {} Km".format(venues_fetched, RADIUS/1000))

    venues = result['response']['groups'][0]['items']
    venues = json_normalize(venues)

    # Filter the columns
    filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
    venues = venues.loc[:, filtered_columns]

    # Filter the category for each row
    venues['venue.categories'] = venues.apply(get_category_type, axis = 1)

    # Clean all column names
    venues.columns = [col.split(".")[-1] for col in venues.columns]
    foursquare_venues = pd.concat([foursquare_venues, venues], axis = 0, sort = False)

    if (venues_fetched < 100):
        break
    else:
        offset = offset + 100

foursquare_venues = foursquare_venues.reset_index(drop = True)
print("\nTotal {} venues fetched".format(total_venues))
```

```
Total 10 venues fetched within a total radius of 80.467 Km

Total 10 venues fetched
```

In [275]:

```
df.shape
```

Out[275]:

```
(18, 6)
```

In [276]:

```
df.groupby('ADDRESS').size()
```

Out[276]:

```
ADDRESS
1007 Capability Road               1
10908 Chapel Hill Rd               1
118 South Street                   1
1315 Oakwood Avenue                1
15 E. Peace Street                 1
2101 Hillsborough Street           1
219 S. Wingate Street              1
2200 South Wilmington Street       1
225 Hillsborough St                1
2901 Holston Ln                    1
3200 Bush St                       1
321 Chapanoke Rd                   1
3434 Kildaire Farm Rd              1
3800 Hillsborough Street           1
519 N Industrial Dr                1
6600 Louisburg Rd                  1
808 Aviation Pkwy #1100            1
9101 Fayetteville Road             1
dtype: int64
```

In [277]:

```
df2 = df[['ADDRESS', 'X', 'Y']]
df2.head()
```

Out[277]:

	ADDRESS	X	Y
0	2101 Hillsborough Street	2.097579e+06	741713.299902
1	3800 Hillsborough Street	2.092281e+06	744629.599844
2	1007 Capability Road	2.095116e+06	736981.100049
3	118 South Street	2.107643e+06	735692.700066
4	1315 Oakwood Avenue	2.112693e+06	741014.500118

In [278]:

```
df2.columns = ['id', 'Latitude', 'Longitude']
df2.index = range(len(df2))
df2.head()
```

Out[278]:

	id	Latitude	Longitude
0	2101 Hillsborough Street	2.097579e+06	741713.299902
1	3800 Hillsborough Street	2.092281e+06	744629.599844
2	1007 Capability Road	2.095116e+06	736981.100049
3	118 South Street	2.107643e+06	735692.700066
4	1315 Oakwood Avenue	2.112693e+06	741014.500118

In [279]:

```
df3 = df2.loc[:99, :]
df3.shape
```

Out[279]:

```
(18, 3)
```

In [280]:

```
def getNearbyVenues(id, latitudes, longitudes, radius = 80467):

    venues_list = []
    for index, lat, lng in zip(id, latitudes, longitudes):
        print(index)

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude,
    longitude,
    radius,
    LIMIT)
results = requests.get(url).json()
nearby_venues = pd.DataFrame()

venues = results['response']['groups'][0]['items']
nearby_venues = json_normalize(venues)
```

In [281]:

```
# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]
# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)
# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]
nearby_venues.head()
```

Out[281]:

	name	categories	lat	lng
0	North Carolina Museum of Natural Sciences	Science Museum	35.782217	-78.639375
1	Bida Manda Laotian Restaurant and Bar	Asian Restaurant	35.777227	-78.636812
2	Artspace Visual Arts Center	Art Gallery	35.775576	-78.636335
3	North Carolina Museum of History	History Museum	35.781516	-78.638747
4	Beasley's Chicken + Honey	Southern / Soul Food Restaurant	35.776968	-78.638175

In [282]:

```
nearby_venues.tail()
```

Out[282]:

	name	categories	lat	lng
5	Burial Beer Co.	Brewery	35.775241	-78.632247
6	Nature Research Center	Science Museum	35.782486	-78.640745
7	Logan Trading Co.	Garden Center	35.789426	-78.640663
8	Marbles Kids Museum	Museum	35.778624	-78.636033
9	lucettegrace	Bakery	35.777266	-78.640189

In [295]:

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from pandas.io.json import json_normalize
import requests
venues_map = folium.Map(location=[35.786319, -78.663314], zoom_start=15)

folium.CircleMarker(
    [latitude, longitude],
    radius=5,
    color='red',
    popup='Raleigh',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)
# add all venues as blue circle markers
for lat, lng, label in zip(nearby_venues.lat, nearby_venues.lng, nearby_venues.categories):
    folium.CircleMarker(
        [lat, lng],
        radius=10,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)
```

[296]:

```
ues_map
```

[296]:

Results

Based on the data collected, a probable area to set up a restaurant near NC State in Raleigh would be within the range of the warehouse district.

Conclusion: Discussion: Trial, Error, and Changing Scope:

I now understand that while I have the training to apply certain tools, it takes experience to apply them effectively. The original scope was to find the best location for a restaurant using all data from surrounding colleges and universities. However, after much confusion and frustration, I realized that I do not have the experience to work with complex tables that do not match as well as hoped. From researching data and results, it would have been better to select data that was compatible with each other, i.e. both having longitude and latitude data on it. From the data, we can see a number of surrounding sources, but not all of them are restaurants.