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/py thon3.6/site-packages (0.10.1) Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/ python3.6/site-packages (from folium) (2.21.0) Requirement already satisfied: jinja2>=2.9 in /opt/conda/envs/Python36/l ib/python3.6/site-packages (from folium) (2.10) Requirement already satisfied: branca>=0.3.0 in /opt/conda/envs/Python3 6/lib/python3.6/site-packages (from folium) (0.4.0) Requirement already satisfied: numpy in /opt/conda/envs/Python36/lib/pyt hon3.6/site-packages (from folium) (1.15.4) Requirement already satisfied: idna<2.9,>=2.5 in /opt/conda/envs/Python3 6/lib/python3.6/site-packages (from requests->folium) (2.8) Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Pyt hon36/lib/python3.6/site-packages (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/Pytho n36/lib/python3.6/site-packages (from jinja2>=2.9->folium) (1.1.0) Requirement already satisfied: six in /opt/conda/envs/Python36/lib/pytho n3.6/site-packages (from branca>=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/cfabd5fb515b4b45a2714d48 fbcd2dd5_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', 'COLLEGENAM E'], dtype='object') Out[77]: Χ Y OBJECTID **ADDRESS** CITY COLLEGENAME 2101 Hillsborough NC State University - Main **0** 2.097579e+06 741713.299902 Raleigh Street 3800 Hillsborough **1** 2.092281e+06 744629.599844 2 Raleigh Meredith College Street 1007 Capability 2 2.095116e+06 736981.100049 Raleigh NCSU Centennial Campus Road **3** 2.107643e+06 735692.700066 118 South Street Raleigh **Shaw University** 1315 Oakwood **4** 2.112693e+06 741014.500118 Raleigh St. Augustine College Avenue df.shape In [37]: 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** Χ Υ 0 NC State University - Main Campus 2.097579e+06 741713.299902 1 Meredith College 2.092281e+06 744629.599844 NCSU Centennial Campus 2.095116e+06 736981.100049 Shaw University 2.107643e+06 735692.700066 3 St. Augustine College 2.112693e+06 741014.500118 5 Southeastern Baptist Seminary 2.144425e+06 812169.599934 Wake Technical College Southern Wake Campus 2.087623e+06 7 William Peace University 2.107440e+06 742210.100089 Wake Technical College Northern Wake Campus 2.134938e+06 770863.000133 Wake Technical College Western Wake Campus 2.060739e+06 9 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 Wake Technical College Public Safety Education... 2.104581e+06 722558.999956 12 13 Vernon Malone College and Career Academy 2.105868e+06 729089.000019 Campbell University ext. 2.055586e+06 14 15 Campbell University Law School 2.105852e+06 739116.799846 Wake Technical College Beltline Education Center 2.114660e+06 755005.982634 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/124c2187da8c41c59bde04fa 67eb2872_0.csv?session=1285114555.1585645226.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', 'FACILITYTYP Е', 'PERMITID', 'X', 'Y', 'GEOCODESTATUS'], dtype='object') Out[97]: CITY STATE POSTALCODE **OBJECTID HSISID** NAME ADDRESS1 ADDRESS2 9629 Tacos Don 509616 4092018236 **BITTER** NaN ANGIER NC 27501 Marcos **MELON DR** St Mary 625 Magdalene 509841 4092050026 **APEX** NC 27502 1 NaN School Magdala PL Cafeteria 758 **BAAN THAI** 509842 4092015002 2 **WILLIAMS APEX** NC 27502 NaN **RESTAURANT** 1403 W 509843 4092014554 KFC #J120103 **APEX** NC 27502 3 NaN Williams ST 760 W Brueggers's 509844 4092014694 **APEX** NC 27502 NaN Bagels Williams ST In [98]: df.shape Out[98]: (3481, 15) In [99]: | df4= df[['NAME', 'X', 'Y']] df4.head() Out[99]: NAME Χ Υ Tacos Don Marcos -78.747986 35.539550 1 St Mary Magdalene School Cafeteria -78.874487 35.724227 BAAN THAI RESTAURANT -78.864641 35.738078 3 KFC #J120103 -78.877270 35.750153 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_LA TITUDE, 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 = 13folium.Marker([CHD_LATITUDE, CHD_LONGITUDE]).add_to(raleigh_map) Out[55]: <folium.map.Marker at 0x7f64be258278> In [57]: raleigh_map Out[57]: + Leaflet | Data by © OpenStreetMap, under ODbL. 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 = 'VAYCGD5DPRVQT5URMRBXLYILFFXVG0AVYNAMPXYGABJF1MV foursquare_client_secret = 'I3TRZNORMFIVXTT0IJEFCBQC0SYS2ED4UQDMEV1UYWZL radius = 80467 # 50 mi = 80.4672Km no_of_venues = 150 version = '20200331' # Current date In [60]: def get_category_type(row): 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".forma t(venues_fetched, RADIUS/1000)) venues = result['response']['groups'][0]['items'] venues = json_normalize(venues) # Filter the columns filtered_columns = ['venue.name', 'venue.categories', 'venue.locatio n.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):</pre> break offset = offset + 100foursquare_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'], foursqua re_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]: + 0 Leaflet | Data by © OpenStreetMap, under ODbL. 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".forma t(venues_fetched, RADIUS/1000)) venues = result['response']['groups'][0]['items'] venues = json_normalize(venues) # Filter the columns filtered_columns = ['venue.name', 'venue.categories', 'venue.locatio n.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):</pre> break offset = offset + 100foursquare_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/c5/3cf9cdc39a6f 2552922f79915f36b45a95b71fd343cfc51170a5b6ddb6e8/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/Python3 6/lib/python3.6/site-packages (from geopandas) (0.24.1) Collecting pyproj>=2.2.0 (from geopandas) Downloading https://files.pythonhosted.org/packages/ce/37/705ee471f711 30d4ceee41bbcb06f3b52175cb89273cbb5755ed5e6374e0/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/c96d3461fda9 9ed8e82ff0b219ac2c8384694b4e640a611a1a8390ecd415/Shapely-1.7.0-cp36-cp36 m-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/20/4e63bc5c6e62 df889297b382c3ccd4a7a488b00946aaaf81a118158c6f09/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/l ib/python3.6/site-packages (from pandas>=0.23.0->geopandas) (2018.9) Requirement already satisfied: numpy>=1.12.0 in /opt/conda/envs/Python3 6/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/env s/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/85d8da5c9a45 e072301beb37ad7f833cd344e04c817d97e0cc75681d248f/munch-2.5.0-py2.py3-non Requirement already satisfied: six>=1.7 in /opt/conda/envs/Python36/lib/ python3.6/site-packages (from fiona->geopandas) (1.12.0) Collecting click-plugins>=1.0 (from fiona->geopandas) Downloading https://files.pythonhosted.org/packages/e9/da/824b92d9942f 4e472702488857914bdd50f73021efea15b4cad9aca8ecef/click_plugins-1.1.1-py 2.py3-none-any.whl Collecting cligj>=0.5 (from fiona->geopandas) Downloading https://files.pythonhosted.org/packages/e4/be/30a58b4b0733 850280d01f8bd132591b4668ed5c7046761098d665ac2174/cligj-0.5.0-py3-none-an y.whl Requirement already satisfied: click<8,>=4.0 in /opt/conda/envs/Python3 6/lib/python3.6/site-packages (from fiona->geopandas) (7.0) Requirement already satisfied: attrs>=17 in /opt/conda/envs/Python36/li b/python3.6/site-packages (from fiona->geopandas) (18.2.0) Installing collected packages: pyproj, shapely, munch, click-plugins, cl igj, fiona, geopandas Successfully installed click-plugins-1.1.1 cligj-0.5.0 fiona-1.8.13.post 1 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/pyt hon3.6/site-packages (1.18.1) Requirement already satisfied: geographiclib<2,>=1.49 in /opt/conda/env s/Python36/lib/python3.6/site-packages (from geopy) (1.49) In [155]: file_name='https://opendata.arcgis.com/datasets/cfabd5fb515b4b45a2714d48 fbcd2dd5_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', 'COLLEGENAM E'], dtype='object') Out[157]: Χ Y OBJECTID **ADDRESS CITY COLLEGENAME** 2101 Hillsborough NC State University - Main **0** 2.097579e+06 741713.299902 Raleigh Street Campus 3800 Hillsborough Raleigh **1** 2.092281e+06 744629.599844 Meredith College Street 1007 Capability Raleigh **2** 2.095116e+06 736981.100049 NCSU Centennial Campus Road **3** 2.107643e+06 735692.700066 118 South Street Raleigh **Shaw University** 1315 Oakwood **4** 2.112693e+06 741014.500118 Raleigh St. Augustine College Avenue 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: Using Nominatim with the default "geopy/1.18. 1" `user_agent` is strongly discouraged, as it violates Nominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ and may possibl y cause 403 and 429 HTTP errors. Please specify a custom `user_agent` wi th `Nominatim(user_agent="my-application")` or by overriding the default `user_agent`: `geopy.geocoders.options.default_user_agent = "my-applicat ion"`. 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 = 'VAYCGD5DPRVQT5URMRBXLYILFFXVGOAVYNAMPXYGABJF1MV foursquare_client_secret = 'I3TRZNORMFIVXTT0IJEFCBQC0SYS2ED4UQDMEV1UYWZL XYHS' radius = 80467 # 50 mi = 80.4672Km no of venues = 10version = '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 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 = 0total venues = 0foursquare_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".forma t(venues_fetched, RADIUS/1000)) venues = result['response']['groups'][0]['items'] venues = json_normalize(venues) # Filter the columns filtered_columns = ['venue.name', 'venue.categories', 'venue.locatio n.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):</pre> break else: offset = offset + 100foursquare_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 1315 Oakwood Avenue 15 E. Peace Street 2101 Hillsborough Street 219 S. Wingate Street 2200 South Wilmington Street 225 Hillsborough St 2901 Holston Ln 3200 Bush St 321 Chapanoke Rd 3434 Kildaire Farm Rd 3800 Hillsborough Street 519 N Industrial Dr 6600 Louisburg Rd 808 Aviation Pkwy #1100 9101 Fayetteville Road dtype: int64 In [277]: df2 = df[['ADDRESS', 'X', 'Y']] df2.head() Out[277]: **ADDRESS** Χ **0** 2101 Hillsborough Street 2.097579e+06 741713.299902 1 3800 Hillsborough Street 2.092281e+06 744629.599844 1007 Capability Road 2.095116e+06 736981.100049 118 South Street 2.107643e+06 735692.700066 1315 Oakwood Avenue 2.112693e+06 741014.500118 In [278]: | df2.columns = ['id', 'Latitude', 'Longitude'] df2.index = range(len(df2))df2.head() Out[278]: Latitude Longitude **0** 2101 Hillsborough Street 2.097579e+06 741713.299902 **1** 3800 Hillsborough Street 2.092281e+06 744629.599844 1007 Capability Road 2.095116e+06 736981.100049 118 South Street 2.107643e+06 735692.700066 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.la t', '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_typ e, axis=1) # clean columns nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.col nearby_venues.head() Out[281]: lat name categories Ing North Carolina Museum of Natural Sciences Science Museum 35.782217 -78.639375 Asian Restaurant 35.777227 -78.636812 Bida Manda Laotian Restaurant and Bar Artspace Visual Arts Center Art Gallery 35.775576 -78.636335 History Museum 35.781516 -78.638747 3 North Carolina Museum of History Beasley's Chicken + Honey Southern / Soul Food Restaurant 35.776968 -78.638175 In [282]: nearby_venues.tail() Out[282]: categories name lat Ing 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 Marbles Kids Museum Museum 35.778624 -78.636033 Bakery 35.777266 -78.640189 lucettegrace In [295]: import numpy as np import pandas as pd

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

[latitude, longitude],

fill_color = 'red', fill_opacity = 0.6

import requests

folium.CircleMarker(

).add_to(venues_map)

venues.categories):

radius=5,
color='red',
popup='Raleigh',
fill = True,

import matplotlib.colors as colors

from pandas.io.json import json_normalize

add all venues as blue circle markers

venues_map = folium.Map(location=[35.786319, -78.663314], zoom_start=15)

for lat, lng, label in zip(nearby_venues.lat, nearby_venues.lng, nearby_

Finding a Place for a Restaurant in Raleigh

people in the area, but so that you do not lose profits or customers to other surrounding

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

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.

• 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

entertainment venues?

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