Battle of Zip codes – Charlotte Edition

Introduction/Business Problem

The goal of this capstone project is to develop and deliver recommendation for the best location (zip code) to open a new restaurant in the city of Charlotte NC.

Our client is planning to open a marquee fine dining establishment that serves Indian food in the city of Charlotte. Although there are a few Indian restaurants, our client sees a need for a date-worthy establishment that caters not just to people of Indian origin, but to a wide variety of patrons. Our client, though has an idea of a good location for the restaurant, she would like a recommendation based on data. She'd like to know the best zip code to open the restaurant. She'd also like to understand the parameters used for coming up with the recommendation and how each factor affected the recommendation. She wants the model to be flexible enough to get a new recommendation without starting from scratch, if she were to change her mind about the concept of the restaurant.

About Charlotte, NC:

According to Wikipedia, Charlotte, NC is the most populous city in the state of North Carolina and 17th most populous city in the US. It is also the third most populous city in the US with approximately eight hundred thousand residents. Bank of America and east coast operations of WellsFargo are based out of Charlotte among several other companies. It is also home to a diverse population.

Approach

Solving this problem starts with identifying and collecting data that can help us. We need to collect information along the following four dimensions.

- **Crime Rate/Safety:** Safety is of prime concern to the patrons. A restaurant won't be successful if it is not in a safe location. For the kind of restaurant that our client is considering, we need to treat this factor very important
- **Demographics:** Understanding demographics helps us understand if the location is the right location. A high-end restaurant will not do well in a low-income neighborhood, for example. Among all the data that is being considered, we'd expect this to be of highest importance after safety.
- **Competition:** Number of competing restaurants vs complimentary restaurants are in a location (when combined with demographic data) will give us valuable information on future success of the project. I'd consider this to be the third in the list in the order of importance.
- **Economic activity in the area:** Commerce and economic vibrancy in an area would indicate its viability for new business. To measure economic vibrancy, we will look at all the shopping centers/areas in a particular location. Compare one location relative to the other to determine what is are the preferable locations. Though I would not consider

this to be the most important, it nevertheless is a valuable data point in triangulating best location for the restaurant.

Once we collect the data, we will need to profile each zip code along the respective dimensions. For each dimension, we will group all the zip codes that are alike. K-means clustering algorithm will come in handy, in this case.

For the sake of simplicity, we will try to group them into three clusters. We will analyze each cluster and assign a somewhat subjective weight/rating on their desirability as a location for the restaurant.

In some cases, for example – Crime/Safety, we might conclude that a high-crime zip code is categorically undesirable. Where as in other cases, as in the case of economic activity in the area, even if a cluster/zip code rates low, we may not entirely rule it out depending on how the zip code measures on other dimensions.

Once we cluster zip codes along the four dimensions, we'll evaluate each zip code on all four dimensions and pick the zip code that comes out at the top.

Specific details of how the methodology is used will be elaborated in Methodology section of this report. We will focus on specific data that we collected and used in the next section.

Data

The data that was collected/used for this project is from three sources.

- US Zipcodes database from uszipcodes package in Python
- Data from Charlotte NC Open Data Portal at data.Charlottenc.gov
- Foursquare

- ### Crime Rate/Safety
  The City of Charlotte does not have all the crime data openly available to public. However, they do have three APIs/files for the three categories below
  - CMPD Officer-Involved Shootings – Individuals at http://data.charlottenc.gov/datasets/cmpd-officer-involved-shootings-individuals-1/geoservice
  - CMPD Officer Involved Shootings – Officers at http://data.charlottenc.gov/datasets/cmpd-officer-involved-shootings-officers-1/geoservice

  - CMPD Officer-Involved Shootings – Incidents at http://data.charlottenc.gov/datasets/cmpd-officer-involved-shootings-incidents-1/geoservice

  As you can see this data only accounts for crime/safety incidents where a fire-arm was discharged. However, we can safely assume that this data is a good proxy for Safety/Crime in general.

  Since I would consider all the incidents above are of same magnitude, I will only interested in latitude and longitude at which the incident occurred and total number of incidents. This will help us in identifying the zip code in which the incident occurred and how many incidents occurred. Since the data set is pretty small, I'll be using all of the data regardless of when the crime occurred. Please follow the links if you are interested in all the data provided by the API.

  Note that I will be using # incidents / zip code population as input to standard scaler while clustering the data (to make comparison apples to apples)

- Demographics
  I have used built-in US Zipcodes module in Python to gather demographic information for each zip code. I have taken the following into account while clustering the data
  - Zipcode
  - Total population
  - Population density
  - # of house holds
  - Median household income

- Economic activity in the area
  City of Charlotte provides an API for Existing Shopping Centers at
  http://data.charlottenc.gov/datasets/existing-shopping-centers/geoservice
  This API provides information about all shopping centers in and around Charlotte with shape objects for each of the shopping centers. Name, shopping center type (convenience, community, regional, super regional etc.), area of the shopping center and anchor stores along with other details.

  Among this data, I will be using the shape (or the boundaries) of the shopping centers to extract latitude and longitude of the shopping center to get the zip code as to where it is located.

  Shopping center type and area are also of significance. They are proxies for economic vibrancy in that area and this are crucial for our model.

- Competition

  I'll be using FourSquare to gather information about Restaurants in each zip code. Number of restaurants in relation to population density give us a good idea on if a zip code is saturated vs there is an opportunity waiting in plain sight. Category of each of these restaurants will help us understand if the zip is ripe for a specific type of restaurant (or replete with it). I will be using the following data for each zip code:
  - Category
  - Longitude, latitude

  I will again be using longitude and latitude to get the zip code for each restaurant. I would be interested in looking at Number of restaurants in each category by zip code (in relation to population density). This information will be used to cluster the competition to identify opportunities/threats.

Methodology

Given that success of a restaurant largely driven by location, it is imperative for us to collect as much data as possible (and available) and identifying key data among that to use for our purpose. Given the data that was available (see above), I feel confident that we will be able to make a sound recommendation.

At a high level, our methodology strives to cluster zip codes into three categories

- Yes – meaning it is a great location for a restaurant with respect to the feature/dimension that we are examining
- Maybe – the zip code can be considered in absence of better alternatives or if the zip code is rated 'Yes' in other dimensions
- No – this zip code is not suitable for opening a restaurant

For one dimension (demographics), we have an extra category i.e. Strong Yes, to differentiate its appeal from other zip codes

Strong Yes, Yes, Maybe and No categories are assigned by analyzing the cluster a particular zip code falls into. I used K-means clustering for clustering the zip codes.

As you know, K-means clustering while helpful in grouping like items together, we still have to look at the clusters to figure out what separate a particular cluster from the rest. It is these characteristics that I will use to tag the zip code more or less favorable to open a new restaurant.

Note that all four dimensions are not weighted equally. Demographics is weighted the heaviest. Followed by competition, economic activity (shopping centers) and crime rate.

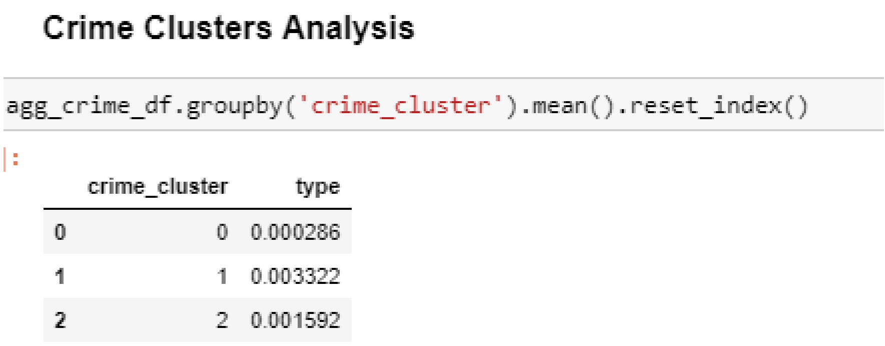
Once clustering is complete for all dimensions, we will look at all zip codes along all four dimensions together. The zip code that has best classification among all will be considered as the ideal location for opening a restaurant. In case of a tie, we will pick a location that measure better in bigger dimensions in relation to the other.

Discussion and Results

I have combined Discussion and Results section as they go hand in hand for this particular example. In this section, I will be elaborating on methodology, and approach along with results for each dimension.

Crime Rate/Safety

As mentioned in Data section, all available crime data was considered to calculate number of incidents by zip code. Since zip code was not directly available, latitude and longitude obtained from the API are used to get zip codes from us zip codes package in python. Total number of crimes are divided by population of the zip code so that they are compared fairly against one another.

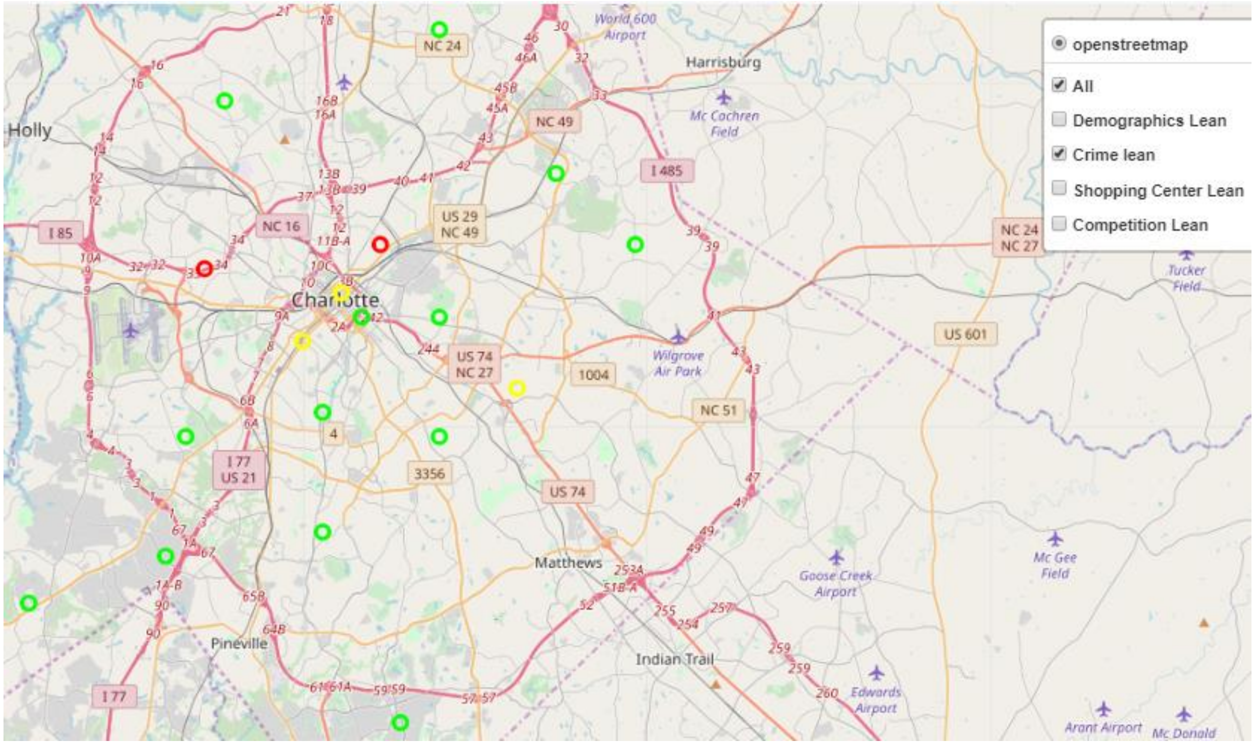


Crime Clusters Analysis

```
agg_crime_df.groupby('crime_cluster').mean().reset_index()
```

	crime_cluster	type
0	0	0.000286
1	1	0.003322
2	2	0.001592

The representation of crime cluster by zip code looks as follows:



● openstreetmap

☑ All

□ Demographics Lean

☑ Crime lean

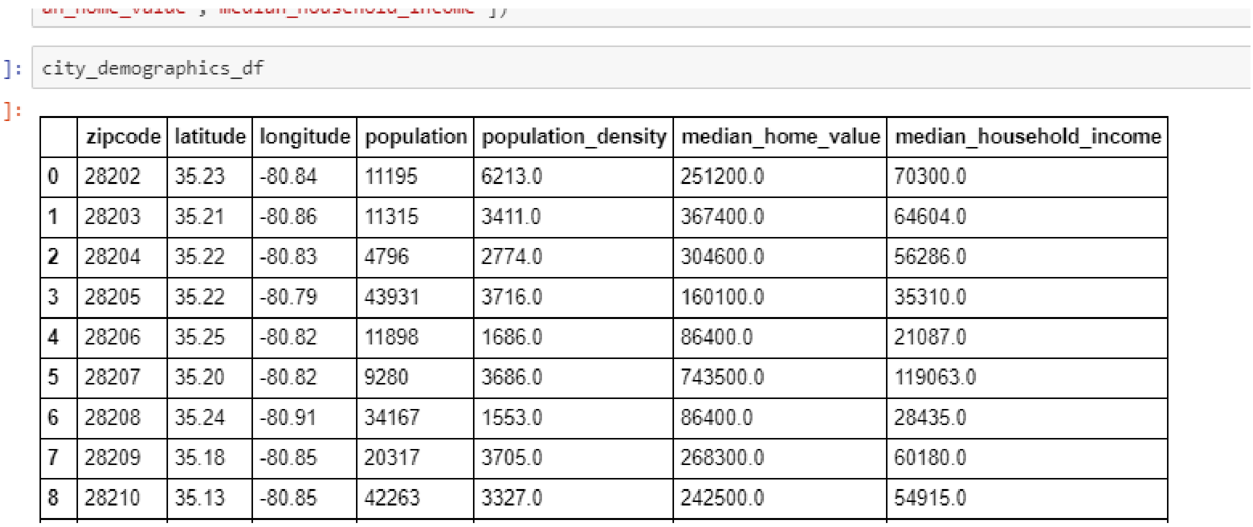
□ Shopping Center Lean

□ Competition Lean

We can see that zip codes close to uptown fell into ‘Maybe’ (yellow) category. A couple of zip codes (one North East of uptown and one North West of uptown) are tagged as unsafe. Cluster #0 seem to be the most favorable in this aspect. As mentioned elsewhere in the document I have used size of the circle to indicate it’s weight and color to represent lean. These circles are placed on the map at latitude and longitude of the zip code they represent.

Demographics

From uszipcodes package from Python, I have collected the following data.



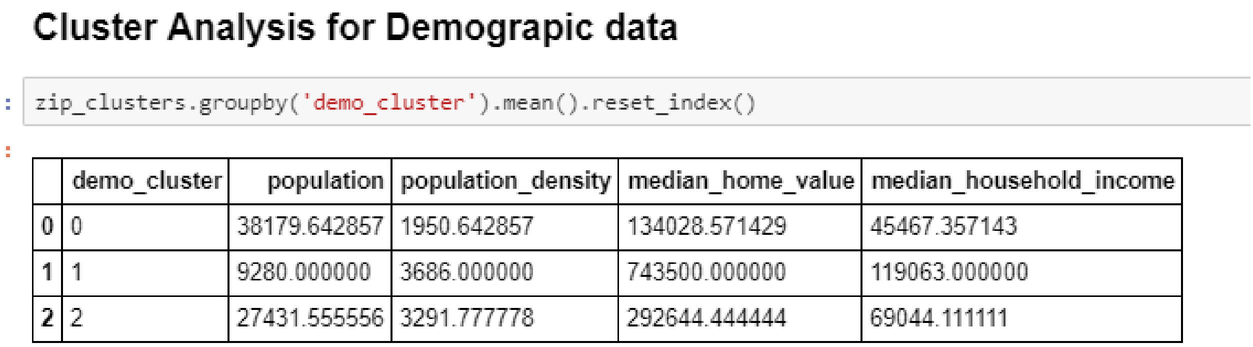
```
an_home_value', 'median_household_income'])
]: city_demographics_df
]:
```

	zipcode	latitude	longitude	population	population_density	median_home_value	median_household_income
0	28202	35.23	-80.84	11195	6213.0	251200.0	70300.0
1	28203	35.21	-80.86	11315	3411.0	367400.0	64604.0
2	28204	35.22	-80.83	4796	2774.0	304600.0	56286.0
3	28205	35.22	-80.79	43931	3716.0	160100.0	35310.0
4	28206	35.25	-80.82	11898	1686.0	86400.0	21087.0
5	28207	35.20	-80.82	9280	3686.0	743500.0	119063.0
6	28208	35.24	-80.91	34167	1553.0	86400.0	28435.0
7	28209	35.18	-80.85	20317	3705.0	268300.0	60180.0
8	28210	35.13	-80.85	42263	3327.0	242500.0	54915.0

Only Zip codes within the city are considered (not the suburbs) and any zip code with less than 100 households are removed from consideration.

Since our client wants to open a high-end restaurant, we want to pick affluent zip codes for the location. And all the data we got from the zip codes database will tell us exactly that.

Profile for all three clusters for demographics is as shown below.



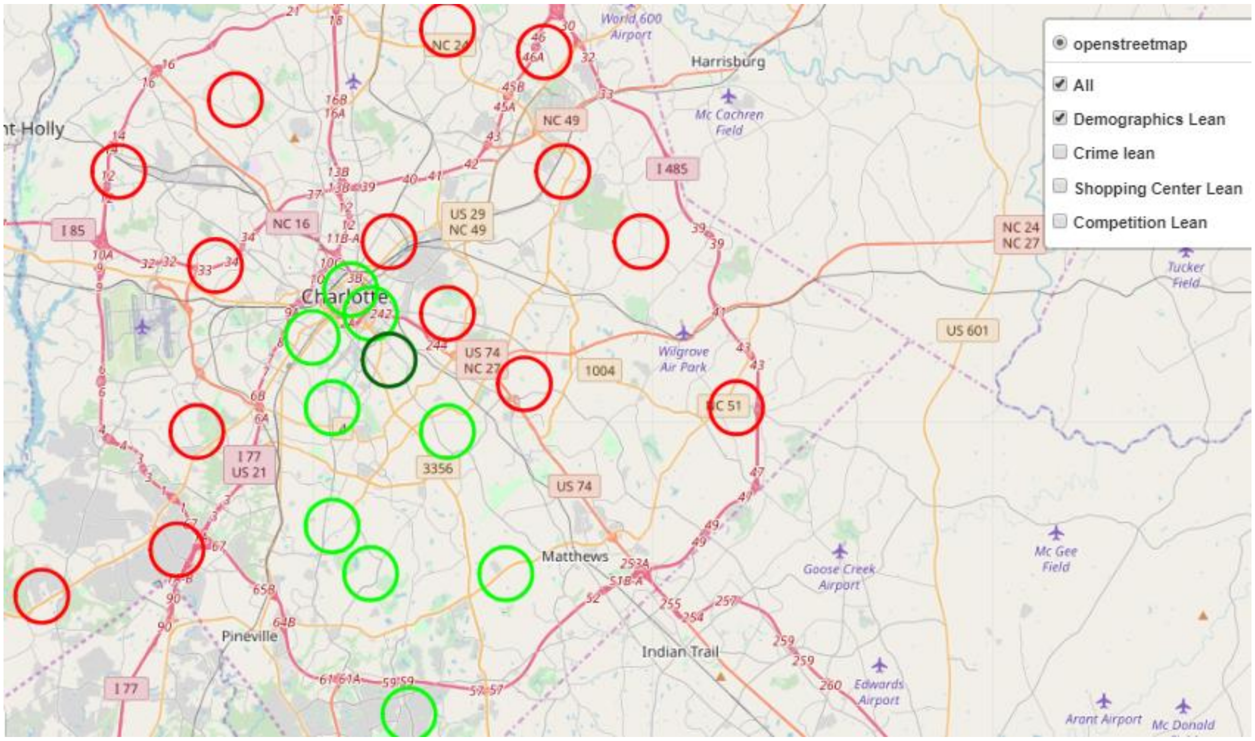
Cluster Analysis for Demograpic data

```
: zip_clusters.groupby('demo_cluster').mean().reset_index()
:
```

	demo_cluster	population	population_density	median_home_value	median_household_income
0	0	38179.642857	1950.642857	134028.571429	45467.357143
1	1	9280.000000	3686.000000	743500.000000	119063.000000
2	2	27431.555556	3291.777778	292644.444444	69044.111111

As we can see Cluster # 1 seem most favorable, followed by Cluster #2. Cluster #0 though has high population, it is not affluent and nor it is densely populated.

When these clusters are represented on the map (based on longitude and latitude of the zip code), it looks as below.



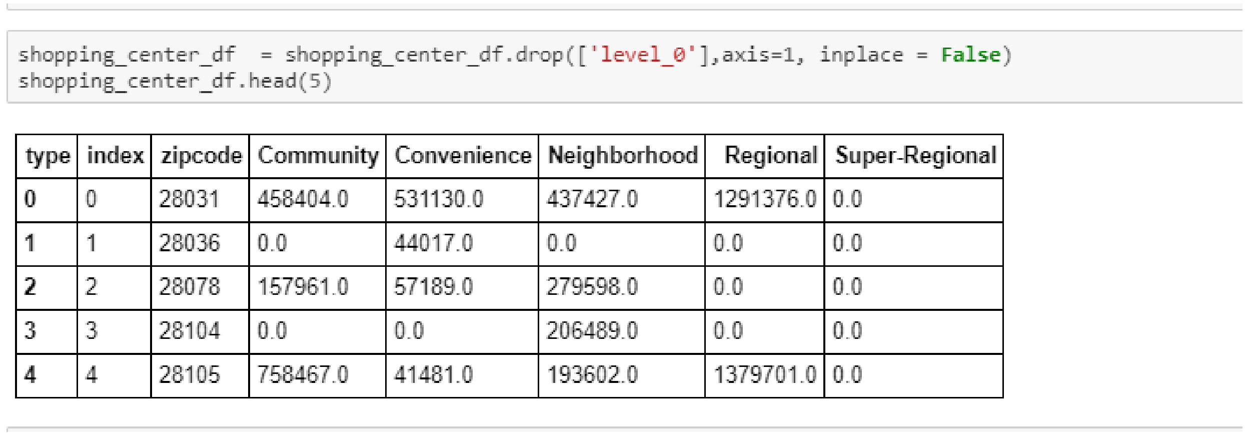
As you will see, I have used size of the circle to indicated weight of that particular dimension in helping.

Most clusters at the center and south of Charlotte seem to be good candidates. The one in dark green (Strong Yes) is zip code 28207, is favored among all. The representation you see here is Strong Yes, Yes and No. It appears that more than half the zip codes are eliminated by this criteria.

Economic Activity in the area (a.k.a. shopping centers)

Type of shopping center and total size of shopping center are good indicators of economic vibrancy. Not only that, they also tell us which zip codes have highest foot traffic (an important aspect for a restaurant location).

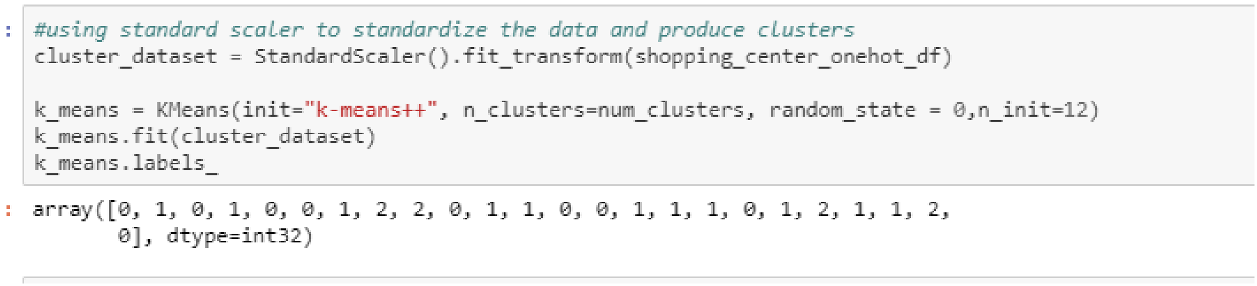
A sample of the shopping center data (after wrangling) looks as follows. The numbers under each column represent total area for that type shopping center. Note that only zip codes within Charlotte are considered. Since zip code is not readily available in the API, I used the shopping center shape dimensions from the API to extract latitude and longitude and then get zip code from there.



```
shopping_center_df  = shopping_center_df.drop(['level_0'],axis=1, inplace = False)
shopping_center_df.head(5)
```

type	index	zipcode	Community	Convenience	Neighborhood	Regional	Super-Regional
0	0	28031	458404.0	531130.0	437427.0	1291376.0	0.0
1	1	28036	0.0	44017.0	0.0	0.0	0.0
2	2	28078	157961.0	57189.0	279598.0	0.0	0.0
3	3	28104	0.0	0.0	206489.0	0.0	0.0
4	4	28105	758467.0	41481.0	193602.0	1379701.0	0.0

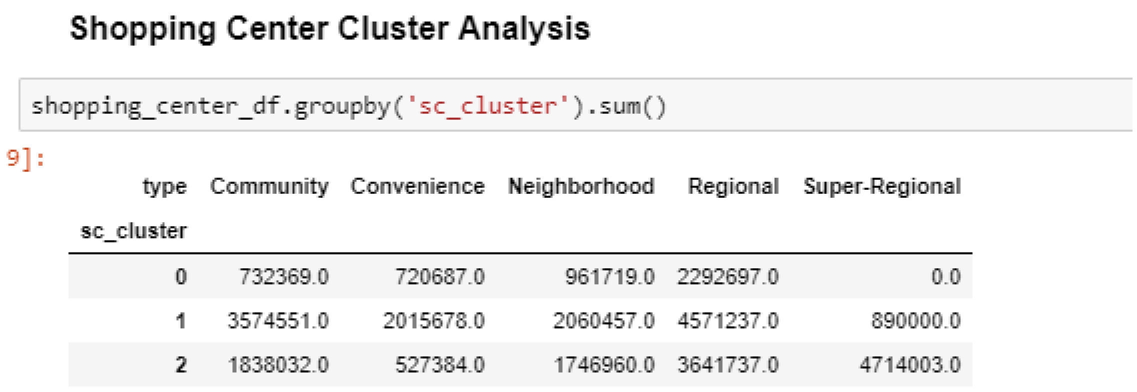
After K-means clustering the data the three clusters look as below.



```
#using standard scaler to standardize the data and produce clusters
cluster_dataset = StandardScaler().fit_transform(shopping_center_onehot_df)

k_means = KMeans(init="k-means++", n_clusters=num_clusters, random_state = 0,n_init=12)
k_means.fit(cluster_dataset)
k_means.labels_
```

```
array([0, 1, 0, 1, 0, 0, 1, 2, 2, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 2, 1, 1, 2,
       0], dtype=int32)
```



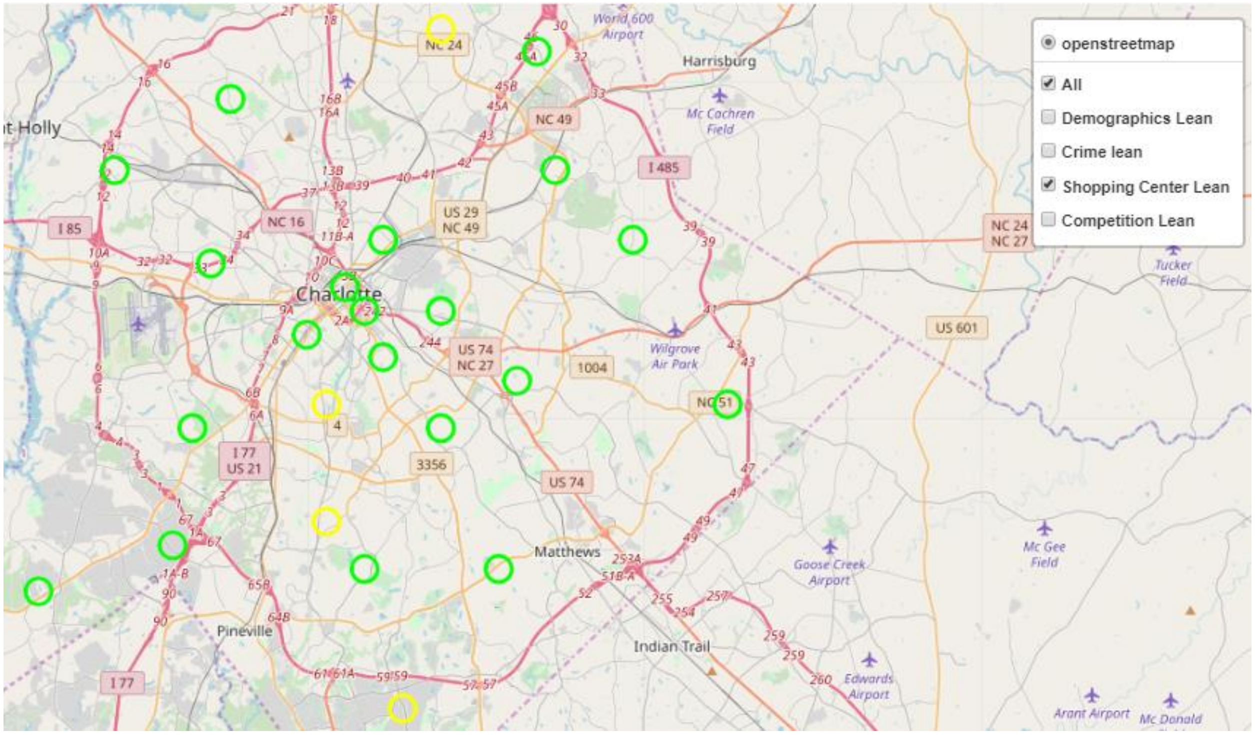
Shopping Center Cluster Analysis

```
shopping_center_df.groupby('sc_cluster').sum()
```

9]:

type	Community	Convenience	Neighborhood	Regional	Super-Regional
sc_cluster					
0	732369.0	720687.0	961719.0	2292697.0	0.0
1	3574551.0	2015678.0	2060457.0	4571237.0	890000.0
2	1838032.0	527384.0	1746960.0	3641737.0	4714003.0

Pictorial representation of the clusters above looks as below. As we can see, most zip codes where data is available are rated favorably except some in south of Charlotte.



● openstreetmap
☑ All
□ Demographics Lean
□ Crime lean
☑ Shopping Center Lean
□ Competition Lean

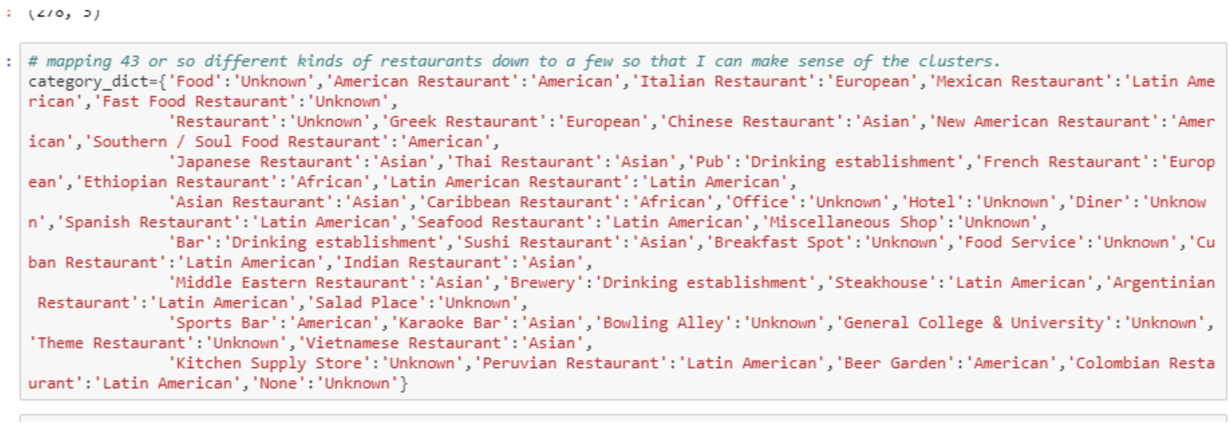
Competition

We would want to look at other restaurants in the zip code to assess competition. A zip code crowded with restaurants (after normalized for population) is less appealing than one where there is little to no competition (with all else being equal).

I have leveraged FourSqaure API to get list of restaurants and their categories. I used longitude and latitude that we already collected for each zipcode for this purpose.

One could take total number of restaurants (normalized for population) in a zip code to make a recommendation. However, looking at categories of restaurants would potentially give us additional insight into a restaurant being complimentary vs. competing.

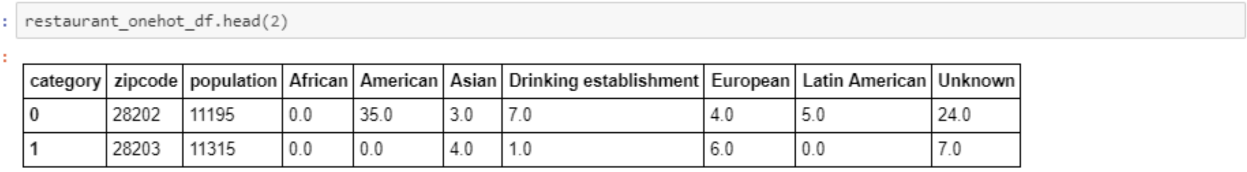
But there were 43 categories of restaurants listed in Charlotte. I had to condense the list down so that I can reasonably analyze the clusters. I used the dictionary below to condense 43 categories into 7 categories.



```
(278, 5)
```

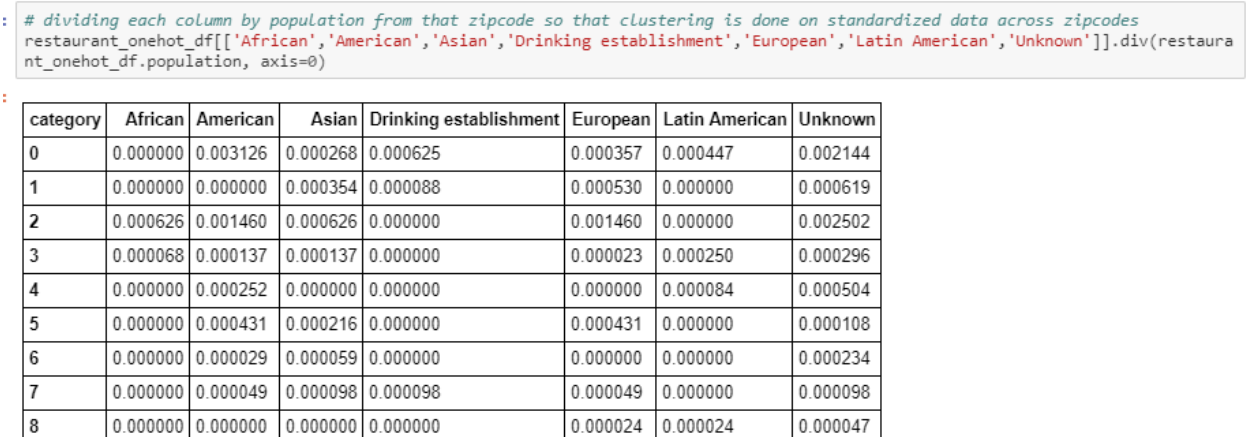
```python
# mapping 43 or so different kinds of restaurants down to a few so that I can make sense of the clusters.
category_dict={'Food':'Unknown','American Restaurant':'American','Italian Restaurant':'European','Mexican Restaurant':'Latin American','Fast Food Restaurant':'Unknown',
               'Restaurant':'Unknown','Greek Restaurant':'European','Chinese Restaurant':'Asian','New American Restaurant':'American','Southern / Soul Food Restaurant':'American',
               'Japanese Restaurant':'Asian','Thai Restaurant':'Asian','Pub':'Drinking establishment','French Restaurant':'European','Ethiopian Restaurant':'African','Latin American Restaurant':'Latin American',
               'Asian Restaurant':'Asian','Caribbean Restaurant':'African','Office':'Unknown','Hotel':'Unknown','Diner':'Unknown','Spanish Restaurant':'Latin American','Seafood Restaurant':'Latin American','Miscellaneous Shop':'Unknown',
               'Bar':'Drinking establishment','Sushi Restaurant':'Asian','Breakfast Spot':'Unknown','Food Service':'Unknown','Cuban Restaurant':'Latin American','Indian Restaurant':'Asian',
               'Middle Eastern Restaurant':'Asian','Brewery':'Drinking establishment','Steakhouse':'Latin American','Argentinian Restaurant':'Latin American','Salad Place':'Unknown',
               'Sports Bar':'American','Karaoke Bar':'Asian','Bowling Alley':'Unknown','General College & University':'Unknown','Theme Restaurant':'Unknown','Vietnamese Restaurant':'Asian',
               'Kitchen Supply Store':'Unknown','Peruvian Restaurant':'Latin American','Beer Garden':'American','Colombian Restaurant':'Latin American','None':'Unknown'}
```

A sample of restaurant data that we wrangled is given below. Note that I have divided the number of restaurants in each zip code by population to normalize the data before feeding it to Standard Scaler.



```python
restaurant_onehot_df.head(2)
```

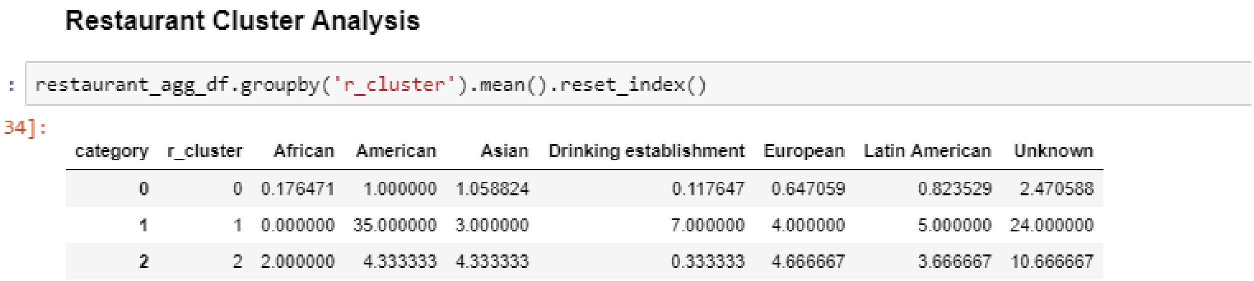
category	zipcode	population	African	American	Asian	Drinking establishment	European	Latin American	Unknown
0	28202	11195	0.0	35.0	3.0	7.0	4.0	5.0	24.0
1	28203	11315	0.0	0.0	4.0	1.0	6.0	0.0	7.0



```python
# dividing each column by population from that zipcode so that clustering is done on standardized data across zipcodes
restaurant_onehot_df[['African','American','Asian','Drinking establishment','European','Latin American','Unknown']].div(restaurant_onehot_df.population, axis=0)
```

category	African	American	Asian	Drinking establishment	European	Latin American	Unknown
0	0.000000	0.003126	0.000268	0.000625	0.000357	0.000447	0.002144
1	0.000000	0.000000	0.000354	0.000088	0.000530	0.000000	0.000619
2	0.000626	0.001460	0.000626	0.000000	0.001460	0.000000	0.002502
3	0.000068	0.000137	0.000137	0.000000	0.000023	0.000250	0.000296
4	0.000000	0.000252	0.000000	0.000000	0.000000	0.000084	0.000504
5	0.000000	0.000431	0.000216	0.000000	0.000431	0.000000	0.000108
6	0.000000	0.000029	0.000059	0.000000	0.000000	0.000000	0.000234
7	0.000000	0.000049	0.000098	0.000098	0.000049	0.000000	0.000098
8	0.000000	0.000000	0.000000	0.000000	0.000024	0.000024	0.000047

The result of restaurant clustering is below.



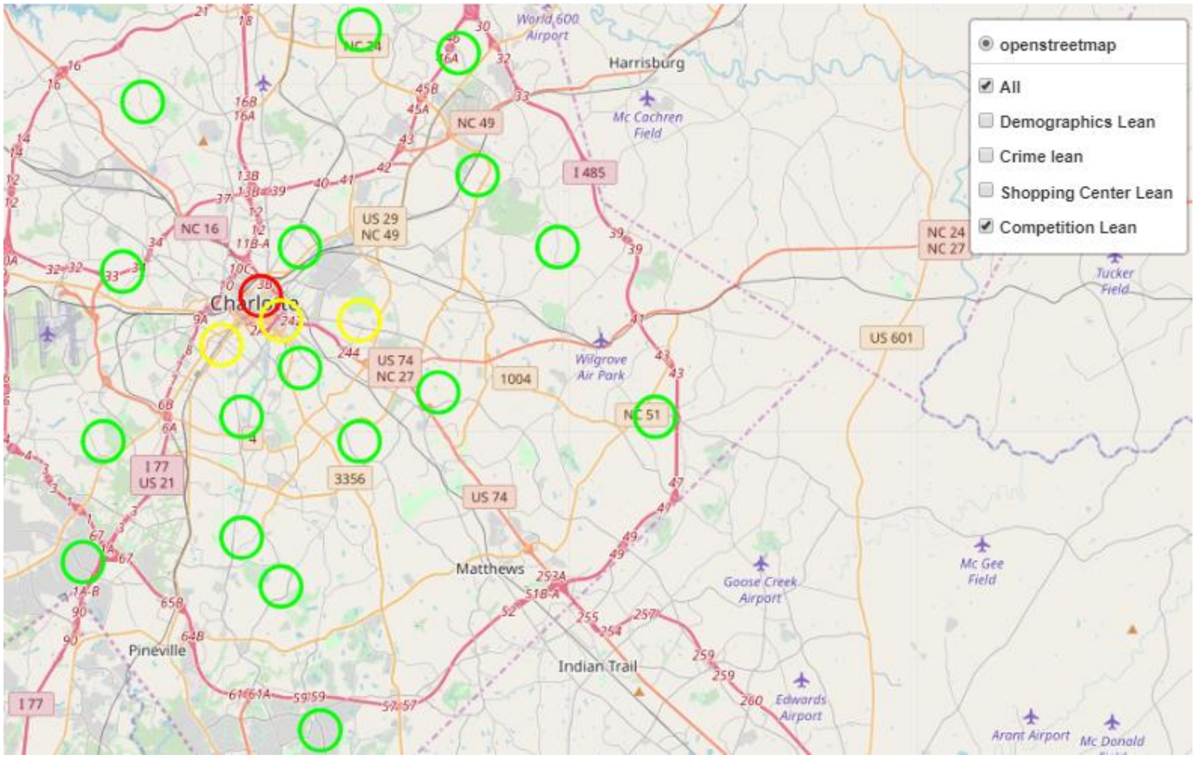
Restaurant Cluster Analysis

```python
restaurant_agg_df.groupby('r_cluster').mean().reset_index()
```

34]:

category	r_cluster	African	American	Asian	Drinking establishment	European	Latin American	Unknown
0	0	0.176471	1.000000	1.058824	0.117647	0.647059	0.823529	2.470588
1	1	0.000000	35.000000	3.000000	7.000000	4.000000	5.000000	24.000000
2	2	2.000000	4.333333	4.333333	0.333333	4.666667	3.666667	10.666667

As we can see from the data Cluster #0 seem to be most favorable (with low number of restaurants, particularly in Asain category, given Indian falls into that category). Cluster #2 is least favorable with high number of total number of restaurants (in relation to population). Cluster #1 seems to next in the list, with fewer number of restaurants (though have higher # of Asian restaurants).



● openstreetmap

☑ All

□ Demographics Lean

□ Crime lean

□ Shopping Center Lean

☑ Competition Lean

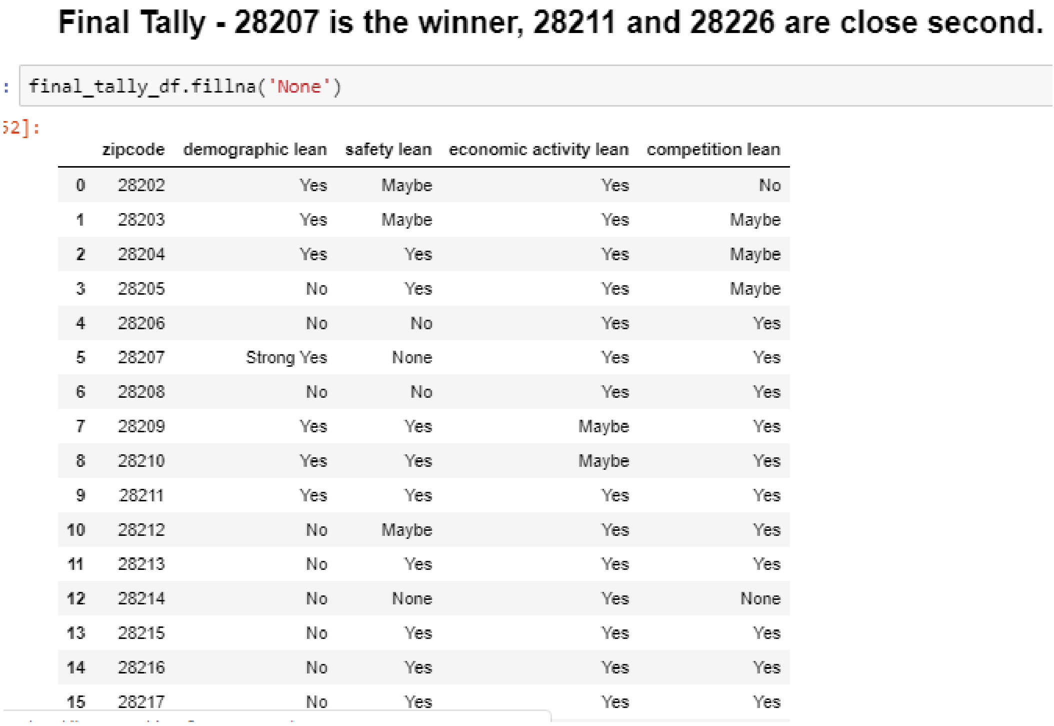
Predictably, zip codes closer to center of the city seem to be less favorable.

Conclusion / Final results

Until know we looked at individual dimensions and discussed what are the favored and least favored zip codes for the restaurant of our client. We need to put this all together to figure out what is the best location when we consider all the dimensions.

If we are able to find a zip code that is rated positively on all dimensions that would be our recommendation. But if we do not, then we may need to go back and look at the criteria we used for clustering. We may also need to reconsider the weights we placed on each dimension. In rare cases, we may have to tell our client that there is no good place to open a restaurant in the city of Charlotte. But let's hope it does not get to that (at least the data so far doesn't indicate so)

Here's the final tally of rating of each zip code along all four dimensions.



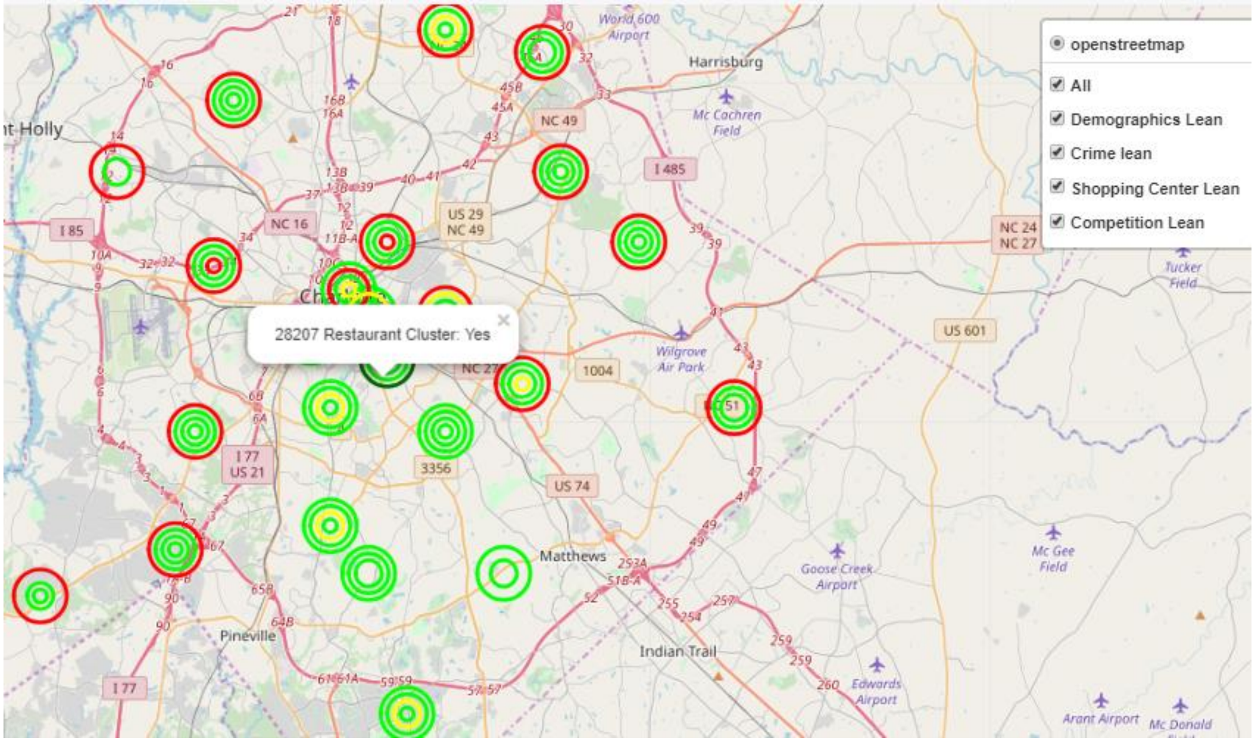
Final Tally - 28207 is the winner, 28211 and 28226 are close second.

```
: final_tally_df.fillna('None')
```

52]:

	zipcode	demographic lean	safety lean	economic activity lean	competition lean
0	28202	Yes	Maybe	Yes	No
1	28203	Yes	Maybe	Yes	Maybe
2	28204	Yes	Yes	Yes	Maybe
3	28205	No	Yes	Yes	Maybe
4	28206	No	No	Yes	Yes
5	28207	Strong Yes	None	Yes	Yes
6	28208	No	No	Yes	Yes
7	28209	Yes	Yes	Maybe	Yes
8	28210	Yes	Yes	Maybe	Yes
9	28211	Yes	Yes	Yes	Yes
10	28212	No	Maybe	Yes	Yes
11	28213	No	Yes	Yes	Yes
12	28214	No	None	Yes	None
13	28215	No	Yes	Yes	Yes
14	28216	No	Yes	Yes	Yes
15	28217	No	Yes	Yes	Yes

The same table when represented on the map, is shown below.



28207 Restaurant Cluster: Yes

● openstreetmap

☑ All

☑ Demographics Lean

☑ Crime lean

☑ Shopping Center Lean

☑ Competition Lean

Caveats

This model was built for this project with time and resource constraints. Not to mention limited knowledge of modeling, python and limited data available. We could enhance this model further.

- We could use PAC (probably approximately correct) learning to condense number of variables we used in clustering. I know such thing existed, but did not have time to learn about it, yet.
- Population and Population density may not always tell the story. For example, uptown of the city is mostly office space that represents tremendous opportunity for a restaurant (as rightly reflected by the # of restaurants in the area). But our model tell us to avoid uptown. While this still could be correct, more research need to be done.
- Weightage for each dimension are subjective. So is disposition we assigned for each cluster
- It is entirely possible and probably true that this is a truly novice version and there are far better methods and models to solve this problem.