Finding a location for a restaurant in New York

Karen Simoes

February 2019

Table of Contents

1. Introduction

Restaurants are extremely popular with people of all ages. Busy and changing lifestyles have seen an increase in the number of restaurants. It's a lucrative business. But starting a business is not an easy task. It requires good research, proper planning and management. Success or failure depends on a lot of factors and the 2 of the important ones are picking the right location and knowing your customers.

1.1. Business Problem

Imagine that you want to open a restaurant in your city. You start researching all of your prospective competitors. You find out where they are located, what types of food they serve, at what prices and what their customers have to say about them. You also find out what customers usually prefer.

In order to be profitable, your restaurant:

- should be easily reachable and accessible to not only your customers but also your personnel and suppliers
- should be located in a part of the city that is frequently visited by people matching your customer profile
- should have a menu and prices that attract customers

There is a lot to consider. So how can a potential restaurant owner use this information to choose a place for their restaurant that will be the most profitable?

The goal of this project was to find the best location or neighbourhood for an Italian restaurant in Manhattan, New York.

1.2. Target Audience

The target audience for this project could be an entrepreneur or investor who wishes to set-up a restaurant in New York and needs to find a location that would be profitable based on data from other similar restaurants. The analysis in this project could also help existing restaurant owners expand their business in other locations.

2. Data

2.1. Data sources

The data for this project was collected from 3 main sources: Wikipedia, geopy python package and Foursquare. The list of New York neighbourhoods was first obtained from Wikipedia, geographic co-ordinates for the neighbourhoods were obtained using geopy and the details of restaurants located in each neighbourhood were fetched using the Foursquare API.

2.2. Data Collection

The names of the neighbourhoods were obtained from the wikipedia page, Neighborhoods in New York City [1]. The data was available in an HTML table with the New York community board names, area, population and neighbourhoods in each board.

Community Board (CB)	Area km^2	Pop. Census 2010	Pop./ km^2	Neighborhoods
Bronx CB 1	7.17	91,497	12,761	Melrose, Mott Haven, Port Morris
Bronx CB 2	5.54	52,246	9,792	Hunts Point, Longwood
Bronx CB 3	4.07	79,762	19,598	Claremont, Concourse Village, Crotona Park, Morrisania
Bronx CB 4	5.28	146,441	27,735	Concourse, High Bridge
Bronx CB 5	3.55	128,200	36,145	Fordham, Morris Heights, Mount Hope, University Heights
Bronx CB 6	4.01	83,268	20,765	Bathgate, Belmont, East Tremont, West Farms
Bronx CB 7	4.84	139,286	28,778	Bedford Park, Norwood, University Heights
Bronx CB 8	8.83	101,731	11,521	Fieldston, Kingsbridge, Kingsbridge Heights, Marble Hill, Riverdale, Spuyten Duyvil, Van Cortlandt Village
Bronx CB 9	12.41	172,298	13,884	Bronx River, Bruckner, Castle Hill, Clason Point, Harding Park, Parkchester, Soundview, Unionport
Bronx CB 10	16.76	120,392	7,183	City Island, Co-op City, Locust Point, Pelham Bay, Silver Beach, Throgs Neck, Westchester Square
Bronx CB 11	9.32	113,232[2]	12,149	Allerton, Bronxdale, Indian Village, Laconia, Morris Park, Pelham Gardens, Pelham Parkway, Van Nest[3][4]

Figure 1: List of neighbourhoods in New York (Wikipedia)

The table data from Figure 1 was scraped using the BeautifulSoup [2] python library, cleaned and stored in a pandas dataframe.

Only Manhattan neighbourhoods were chosen. The geographical coordinates for each of them were fetched using the geopy [3] python package. The address input required by geopy was constructed using the neighbourhood and borough names.

```
Battery Park City, Manhattan, New York 40.7110166 -74.0169369
Financial District, Manhattan, New York 40.7076124 -74.009378
TriBeCa, Manhattan, New York 40.7153802 -74.0093063
Chinatown, Manhattan, New York 40.7164913 -73.9962504
Greenwich Village, Manhattan, New York 40.7319802 -73.9965658
Little Italy, Manhattan, New York 40.7192728 -73.9982152
Lower East Side, Manhattan, New York 40.7159357 -73.9868057
NoHo, Manhattan, New York 40.7258746 -73.9939566
```

Figure 2: Neighbourhood names, latitude and longitude

To get the list of restaurants and their details, the Foursquare API [4] was used. The explore, venues and search API endpoints were used. The following figure shows the JSON response of the explore endpoint for a single neighbourhood.


```
{'meta': {'code': 200, 'requestId': '5c6587e2351e3d4e704c2e54'},
 'response': {'suggestedFilters': {'header': 'Tap to show:',
   'filters': [{'name': 'Open now', 'key': 'openNow'},
    {'name': '$-$$$$', 'key': 'price'}]},
  'headerLocation': 'Battery Park City',
  'headerFullLocation': 'Battery Park City, New York',
  'headerLocationGranularity': 'neighborhood',
  'query': 'restaurants',
  'totalResults': 71,
  'suggestedBounds': {'ne': {'lat': 40.7155166045, 'lng': -74.0110113729468},
   'sw': {'lat': 40.7065165955, 'lng': -74.02286242705321}},
  'groups': [{'type': 'Recommended Places',
    'name': 'recommended',
    'items': [{'reasons': {'count': 0,
       'items': [{'summary': 'This spot is popular',
         'type': 'general',
         'reasonName': 'globalInteractionReason'}]},
      'venue': {'id': '5362a2ae498e3b18c22334be',
       'name': 'Hudson Eats',
       'location': {'address': '225 Liberty St',
        'crossStreet': 'South End Avenue',
        'lat': 40.712802562960505,
        'lng': -74.01610221841605,
        'labeledLatLngs': [{'label': 'display',
          'lat': 40.712802562960505,
          'lng': -74.01610221841605}],
        'distance': 210,
        'postalCode': '10281',
        'cc': 'US',
        'city': 'New York',
        'state': 'NY',
        'country': 'United States',
        'formattedAddress': ['225 Liberty St (South End Avenue)',
         'New York, NY 10281',
         'United States']},
       'categories': [{'id': '4bf58dd8d48988d120951735',
         'name': 'Food Court',
         'pluralName': 'Food Courts',
         'shortName': 'Food Court',
         'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/food_foodcourt_',
          'suffix': '.png'},
         'primary': True}],
```


Figure 3: API response for restaurants within a 500 metre radius of Battery Park City, Manhattan

The explore endpoint was used to fetch restaurants around a 500 metre radius of each Manhattan neighbourhood. Only the Italian restaurants were chosen.
The details for each Italian restaurant were fetched using the venues endpoint of the API. The venue details response can be seen in Figure 4.


```
{'meta': {'code': 200, 'requestId': '5c6828039fb6b740fecfc76e'},
 'response': {'venue': {'id': '556cb8c6498e751e51fbcf41',
   'name': 'Parm',
   'contact': {'phone': '2127764927', 'formattedPhone': '(212) 776-4927'},
   'location': {'address': '250 Vesey St',
    'crossStreet': 'West St',
    'lat': 40.71451391801664,
    'lng': -74.01626377138824,
    'labeledLatLngs': [{'label': 'display',
      'lat': 40.71451391801664,
      'lng': -74.01626377138824}],
    'postalCode': '10080',
    'cc': 'US',
    'city': 'New York',
    'state': 'NY',
    'country': 'United States',
    'formattedAddress': ['250 Vesey St (West St)',
     'New York, NY 10080',
     'United States']},
   'canonicalUrl': 'https://foursquare.com/v/parm/556cb8c6498e751e51fbcf41',
   'categories': [{'id': '4bf58dd8d48988d110941735',
     'name': 'Italian Restaurant',
     'pluralName': 'Italian Restaurants',
     'shortName': 'Italian',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
      'suffix': '.png'},
     'primary': True},
    {'id': '4bf58dd8d48988d1c5941735',
     'name': 'Sandwich Place',
     'pluralName': 'Sandwich Places',
     'shortName': 'Sandwiches',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/deli_',
      'suffix': '.png'}}],
   'verified': True,
   'stats': {'tipCount': 67},
   'url': 'http://www.parmnyc.com',
   'price': {'tier': 2, 'message': 'Moderate', 'currency': '$'},
   'likes': {'count': 239,
    'groups': [{'type': 'others', 'count': 239, 'items': []}],
    'summary': '239 Likes'},
   'dislike': False,
   'ok': False,
   'rating': 8.5,
```


Figure 4: API response for venue details

The closest venues around each restaurant were also fetched to understand how they affect restaurant ratings and reviews. The search endpoint of the API was used to find

the venues around a 250 metre radius of each restaurant. The raw response can be seen in Figure 5.


```
{'meta': {'code': 200, 'requestId': '5c63d3d01ed2192b5a2c1d53'},
 'response': {'venues': [{'name': 'Gild Hall',
    'hasPerk': False,
    'location': {'city': 'New York',
     'formattedAddress': ['15 Gold St (at Platt St)',
      'New York, NY 10038',
      'United States'],
     'lat': 40.7078871,
     'labeledLatLngs': [{'lat': 40.7078871,
       'label': 'display',
       'lng': -74.0071031}],
     'cc': 'US',
     'lng': -74.0071031,
     'address': '15 Gold St',
     'postalCode': '10038',
     'state': 'NY',
     'distance': 8,
     'country': 'United States',
     'crossStreet': 'at Platt St'},
    'id': '4a6e3e75f964a5204ed41fe3',
    'categories': [{'name': 'Hotel',
      'pluralName': 'Hotels',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/travel/hotel_',
       'suffix': '.png'},
      'primary': True,
      'shortName': 'Hotel',
      'id': '4bf58dd8d48988d1fa931735'}],
    'referralId': 'v-1550046160'},
   {'name': 'Felice 15 Gold Street',
    'delivery': {'provider': {'name': 'seamless',
      'icon': {'name': '/delivery_provider_seamless_20180129.png',
       'prefix': 'https://fastly.4sqi.net/img/general/cap/',
       'sizes': [40, 50]}},
     'id': '299433',
     'url': 'https://www.seamless.com/menu/felice-15-gold-st-new-york/299433?affiliate=1131&utm_source=foursquare-affili
ate-network&utm_medium=affiliate&utm_campaign=1131&utm_content=299433'},
    'hasPerk': False,
    'location': {'postalCode': '10038',
     'city': 'New York',
     'formattedAddress': ['15 Gold St', 'New York, NY 10038', 'United States'],
     'lat': 40.70783228127862,
     'labeledLatLngs': [{'lat': 40.70783228127862,
       'label': 'display',
       'lng': -74.00704045580463}],
     'cc': 'US',
```


Figure 5: API response for venues within a 250 metre radius of a restaurant

3. Methodology

3.1 Exploratory Data Analysis

As can be seen in Figure 1, the data obtained from the wikipedia page listed multiple neighbourhoods for each community board in New York City. For example, Bronx CB 1 has the neighbourhoods, Melrose, Mott Haven, Port Morris. Each of these were split into their own row and duplicate neighbourhoods were removed. Only the text content was extracted from the html table cells on the web page, excluding the reference link text in some of the cells. The Borough name for each neighbourhood was extracted from the Community Board column. Borough names were added to the dataframe as a new column.

	Borough	Community Board	Neighbourhood
0	Bronx	Bronx CB 1	Melrose
1	Bronx	Bronx CB 1	Mott Haven
2	Bronx	Bronx CB 1	Port Morris
3	Bronx	Bronx CB 2	Hunts Point
4	Bronx	Bronx CB 2	Longwood

Figure 6: NY Neighbourhoods dataframe

The geographical co-ordinates were fetched for each neighbourhood using geopy and added to the dataframe from Figure 6. Any rows without co-ordinates were dropped.

	Borough	Community Board	Neighbourhood	Latitude	Longitude
0	Manhattan	Manhattan CB 1	Battery Park City	40.711017	-74.016937
1	Manhattan	Manhattan CB 1	Financial District	40.707612	-74.009378
2	Manhattan	Manhattan CB 1	TriBeCa	40.715380	-74.009306
3	Manhattan	Manhattan CB 2	Chinatown	40.716491	-73.996250
4	Manhattan	Manhattan CB 2	Greenwich Village	40.731980	-73.996566
5	Manhattan	Manhattan CB 2	Little Italy	40.719273	-73.998215

Figure 7: Location data for Manhattan neighbourhoods

The Foursquare API's response from Figure 3 contained the names, location details, categories, address, photos available, delivery details, etc. Only the venue id, name, geographical co-ordinates, address and categories were extracted from the response and stored in a dataframe.

	Address	Categories	DistanceToNeighbourhood	Latitude	Longitude	Neighbourhood	Venue
0	225 Liberty St (at Hudson Eats),New York, NY 1...	American Restaurant	216	40.712835	-74.016013	Battery Park City	Dig Inn
1	200 Vesey St,New York, NY 10281,United States	Sushi Restaurant	205	40.712742	-74.016065	Battery Park City	Blue Ribbon Sushi Bar
2	250 Vesey St (West St),New York, NY 10080,Unit...	Italian Restaurant	393	40.714514	-74.016264	Battery Park City	Parm
3	101 Liberty St (at Greenwich St),New York, NY ...	Italian Restaurant	376	40.710554	-74.012519	Battery Park City	Osteria della Pace
4	259 Vesey St (North End Avenue),New York, NY 1...	Mexican Restaurant	458	40.714940	-74.015300	Battery Park City	El Vez

Figure 8: Manhattan restaurants

Duplicate and overlapping venues located in adjacent neighbourhoods were removed. This was done by retaining the rows having the least distance to a neighbourhood. On grouping the remaining restaurants by categories, it was observed that most of the restaurants were Italian.

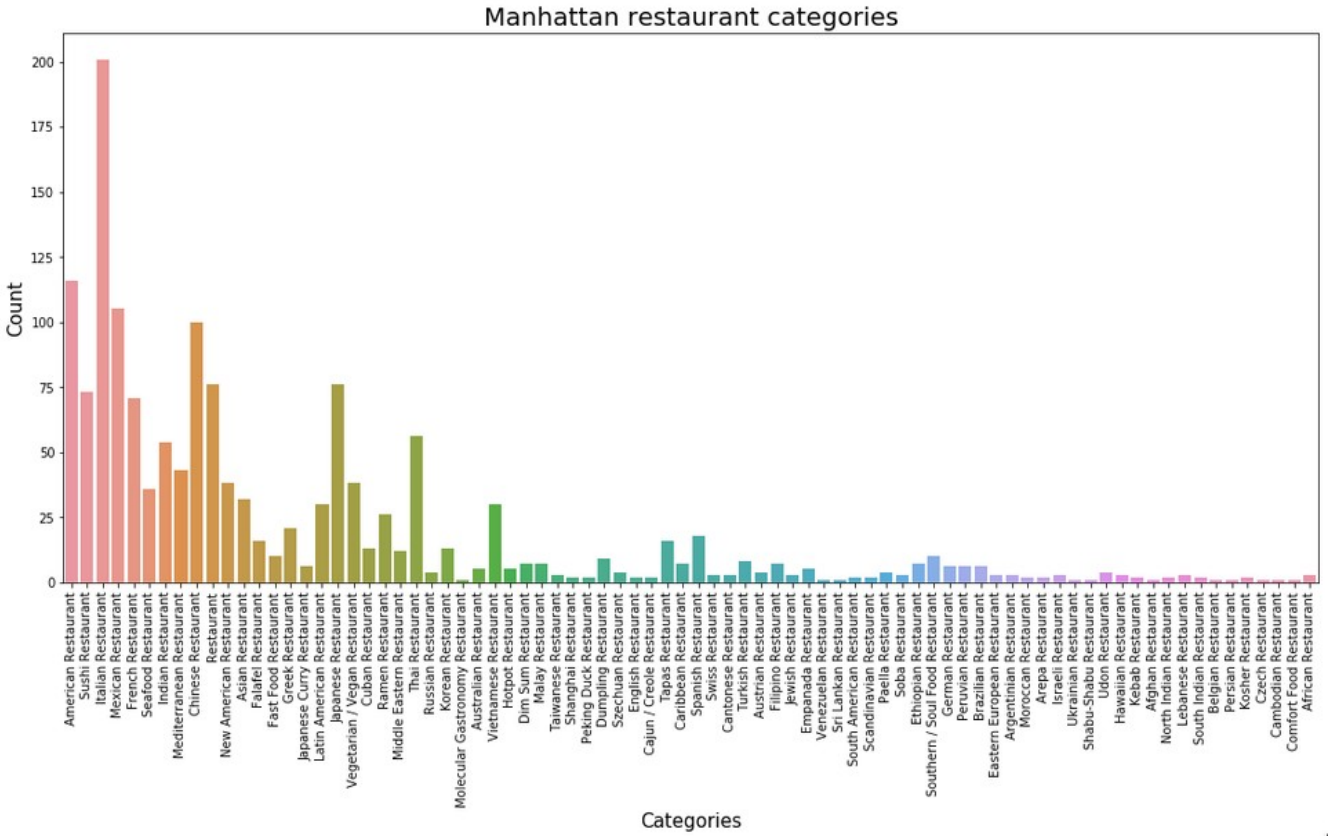

Figure 9: Manhattan restaurant categories

201 of the restaurants were Italian, making them most popular restaurants in Manhattan. These were therefore chosen for further analysis.

The response for Italian restaurant details, from Figure 4, included the following properties: venue id, venue name, price tier, price message, ratings, rating signals, likes, dislike, OK, review summary, timings, specials and attributes. Attributes had values indicating whether the restaurant accepted credit cards, had wheelchair access, music, parking, served lunch, dinner, breakfast, etc.

The attributes with the features of a restaurant, review summary, timings and specials properties from the response were available as dictionary values. These were parsed to string format. The attributes were expanded into new columns for each list item. The timings list had the opening and closing times for each day of the week. Not all the restaurants had this information, so the timings column was dropped. The dislike and OK columns had only one value, False, for all the rows. It was also dropped. The specials column had values for only one restaurant, so it was also dropped. Rows without ratings were dropped. Null values were filled with zeroes and column cells with values were

replaced with binary values. Only attributes with frequencies more than 10 were retained.

Likes	On Lists	Price Message	Price Tier	Rating	Rating Signals	Tips	Verified	Beer	Breakfast	Brunch	Cocktails	Credit Cards	Delivery	Dessert	Dinner	Full Bar
239	172	Moderate	2	8.5	335.0	67	1	1	0	1	1	1	1	1	1	1
41	18	Moderate	2	7.9	51.0	3	0	0	0	0	0	0	0	0	1	0
305	251	Expensive	3	8.3	503.0	140	1	1	0	1	1	1	1	0	1	1
33	13	Very Expensive	4	8.2	52.0	17	0	0	0	0	1	1	0	0	1	1
140	131	Very Expensive	4	8.2	217.0	50	1	0	1	1	1	1	1	0	1	1

Figure 10: Italian restaurant details

The Italian restaurants were grouped using the lunch and dinner columns to analyse how many of them served either one or both. 8 of the restaurants served neither lunch nor dinner, so they were dropped.

		Venue_id
Lunch	**Dinner**	
0	**0**	8
	1	59
1	**0**	5
	1	122

Figure 11: Italian restaurants grouped by lunch and dinner attributes

The search endpoint response, from Figure 5, included the distance of each venue from the restaurant. The categories and distance of the search results from each restaurant were extracted from the JSON raw response. Categories with empty string values were excluded. The rows with distance greater than 250 metres were dropped and only the 10 closest venues were retained.

	Neighbour - 0	Neighbour - 1	Neighbour - 2	Neighbour - 3	Neighbour - 4	Neighbour - 5	Neighbour - 6	Neighbour - 7	Neighbour - 8	Neighbour - 9
0	Boat or Ferry	Office	Burrito Place	Bus Station	Office	Coffee Shop	Doctor's Office	Neighborhood	Office	Bus Line
0	Pizza Place	Gift Shop	Restaurant	Wine Bar	Italian Restaurant	Financial or Legal Service	Office	Coffee Shop	Tech Startup	Chocolate Shop
0	Italian Restaurant	Bakery	Wine Bar	Candy Store	Gourmet Shop	Dessert Shop	Beer Store	Coffee Shop	Ice Cream Shop	Mexican Restaurant
0	Italian Restaurant	Hardware Store	Physical Therapist	Residential Building (Apartment / Condo)	Lounge	School	Parking	Parking	American Restaurant	Gym
0	Italian Restaurant	Residential Building (Apartment / Condo)	Lounge	Snack Place	Mattress Store	Hardware Store	Parking	Italian Restaurant	Laundry Service	Building

Figure 12: 10 closest venues around Italian restaurants

3.2. Modeling

In order to choose a location for a new restaurant based on restaurant data, the existing restaurants needed to be grouped based on their attributes. The groups or clusters would then be analysed and used to make recommendations. The clustering technique was chosen for this task. Clustering is a form of unsupervised machine learning which segments data, without labels or classes, into clusters with similar features. To determine which data points are similar, the distance between each point is measured. The clusters are formed so that the distance between points within each cluster is minimum and distance between points in different clusters is maximum. K-Means, hierarchical clustering, DBSCAN are some of the algorithms available for clustering.

Unlike K-Means, the hierarchical clustering algorithm does not require the number of clusters to be set. Hierarchical clustering creates a hierarchy of clusters by merging neighbouring clusters. The resulting clusters can be visualized by creating a dendrogram. The dendrogram can then be used to find the clusters in the dataset. There are 2 types of hierarchical clustering algorithms:

1. Agglomerative: Follows a bottom-up approach, starting with each data point as a cluster and merging similar clusters as it move up the hierarchy
2. Divisive: Follows a top-down approach, starting with the entire dataset as a single cluster and splitting them based on their similarities as it moves down the hierarchy

The agglomerative clustering algorithm was used for this analysis as the number of clusters in the dataset could be determined without being explicitly set. The scipy python package was used to create a distance matrix which was used to create a dendrogram. The distance matrix computes distances between data points. A dendrogram is a tree-like representation of the clusters produced by hierarchical clustering. The dedrogram indicated 2 clusters in the dataset, but 4 was chosen as the cluster size by splitting the tree at the 0.25 on the y-axis (4^{th} value from the bottom).

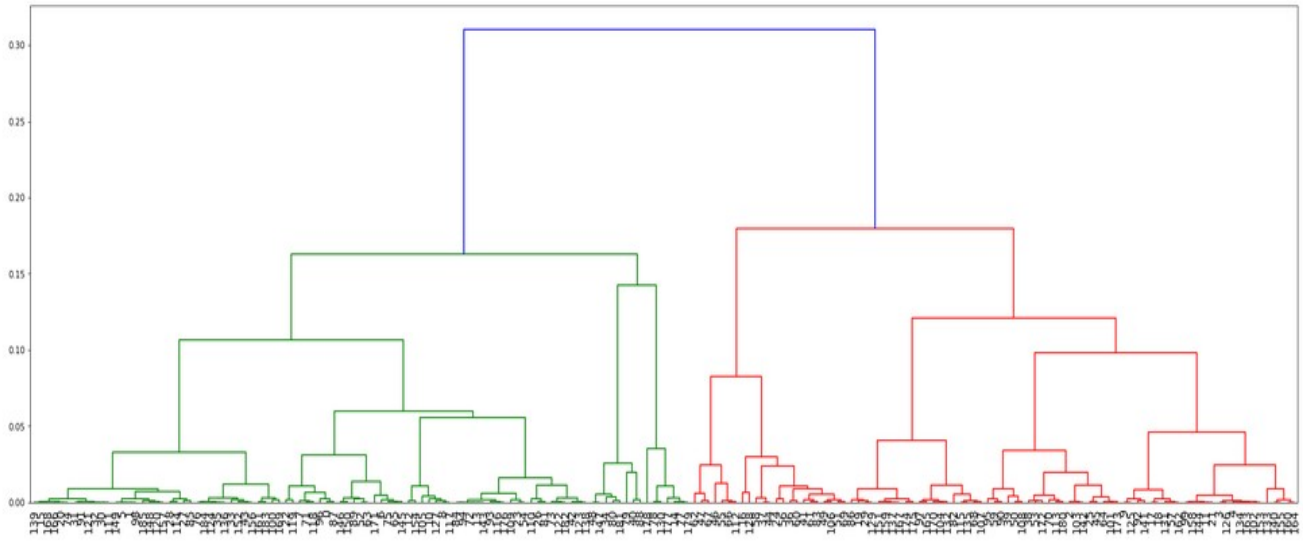

Figure 13: Dendrogram indicating 2 clusters

Only the following features were chosen for clustering: likes, on lists, price tier, rating, rating signals, and tips. The others were dropped and added after clustering. The dataframe columns contained values ranging from 0 to more than 1000. Since the clustering algorithms use distance between data points for clustering, the values in the dataframe were standardized to avoid larger values from influencing the distance measures. The values in each column were scaled within a range of 0 to 1 using the MinMaxScaler class from scikit-learn.

The AgglomerativeClustering class was used from scikit-learn. The number of clusters was set to 4, cosine distance was used as the metric to compute distances between data points and the linkage criterion was set to complete to use maximum distances between

all points within a cluster. All 4 clusters were analysed and were plotted on a map using the folium [5] python library.

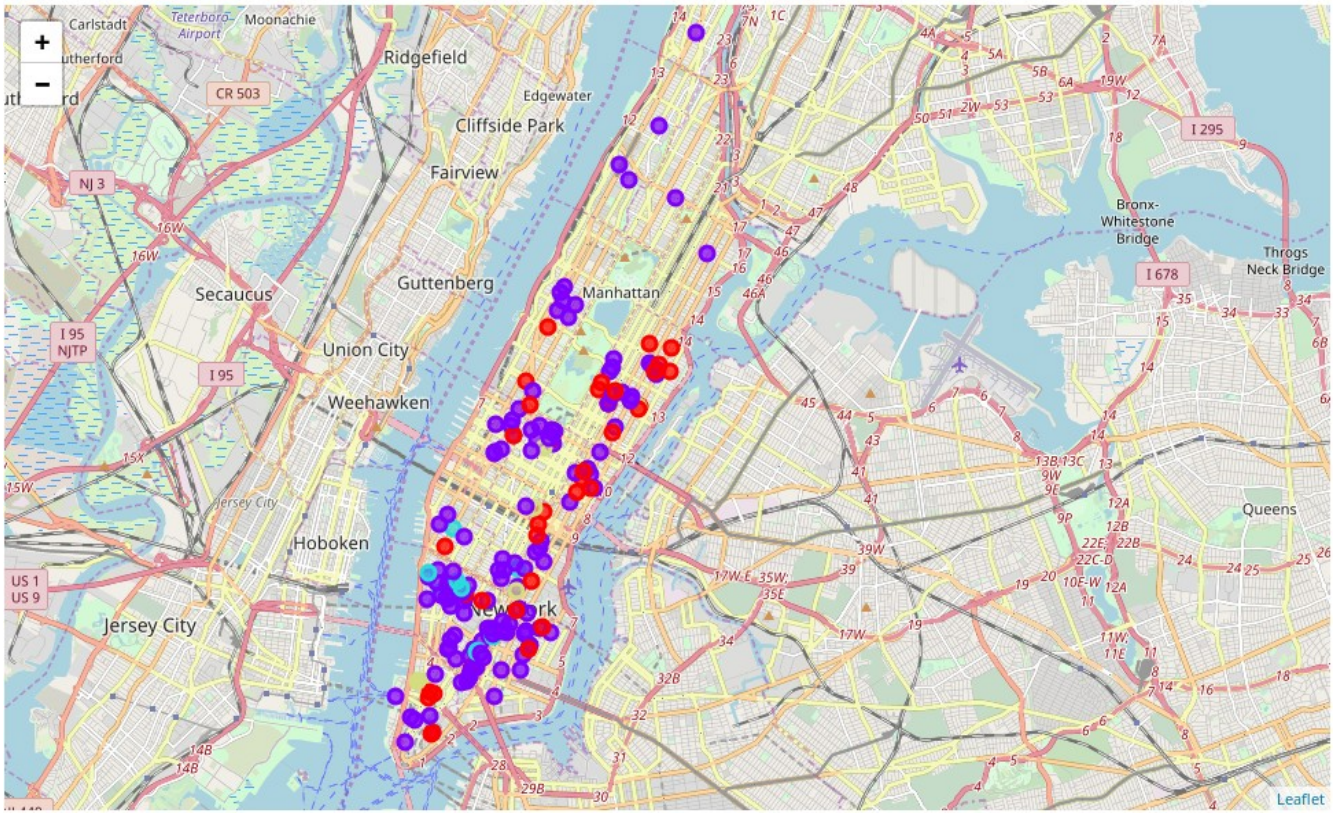

Figure 14: Clusters visualized on a map

4. Results

The 1st cluster had 38 restaurants. All of them served dinner and had no music or take-out. The median rating of this cluster was 7.7 with a median rating signal of 47.5.

The 2nd cluster had 131 restaurants. The median rating of this cluster was 8.3 with a median rating signal of 179.

The 3rd cluster had 9 restaurants. All of them served dinner, cocktails and wine. They also accepted credit cards. The median rating of this cluster was 9.2 with a median rating signal of 1420.

The 4th cluster has 8 restaurants. All of them served lunch and none of them served cocktails nor had music. All were in the same price tier 1. The median rating of this cluster was 8.4 with a median rating signal of 189.5.

5. Discussion

On plotting the Italian restaurants on a map, it was observed that most of them were located in the lower half of Manhattan.

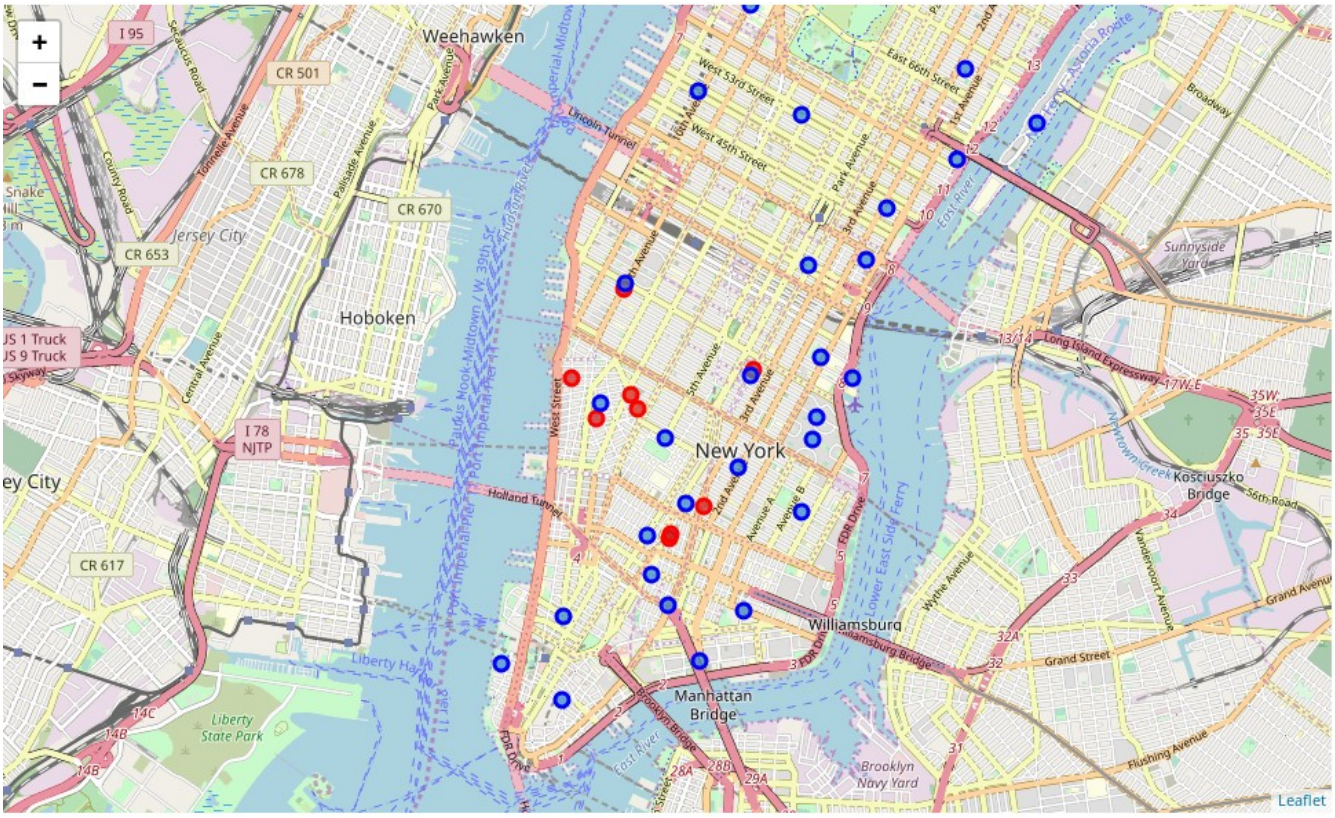

Figure 15: Italian restaurants from cluster 2

Out of the 4 clusters, the 3rd cluster (cluster 2) had the highest median rating, rating signals, tips and likes. 8 of the 9 restaurants in this cluster had other Italian restaurants as their nearest neighbours, which did not seem to affect their ratings. There were 4 restaurants located in West Village, 2 in SoHo, and 1 each in Chelsea, Gramercy Park and NoHo.

Since West Village already had more than one Italian restaurant, SoHo, Chelsea, Gramercy Park and NoHo neighbourhoods were short-listed to setup a new restaurant.

Some of the restaurants in the 3rd cluster were located in semi-residential areas while others were located in commercial areas. Since the restaurants in the short-listed neighborhoods are highly rated and reviewed, setting up another restaurant in similar surroundings could be profitable.

6. Conclusion

In this project, the existing Italian restaurants in Manhattan neighbourhoods were analysed to find the best location to set-up a new one. SoHo, Chelsea, Gramercy Park and NoHo neighbourhoods were chosen as possible locations.

This analysis was based on user ratings and reviews. It could be improved with data like population density in each of the neighbourhoods, type of customers and their frequency of visits and menu details.

7. Appendix

Links to data sources, libraries, packages and API used in this project:

[1] https://en.wikipedia.org/wiki/Neighborhoods_in_New_York_City
[2] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[3] https://geopy.readthedocs.io/en/stable/
[4] https://developer.foursquare.com/docs/api/endpoints
[5] http://python-visualization.github.io/folium/