Food-delivery in Manhattan

Guglielmin Federico

July 19, 2020

1. Business Problem

A food delivery company has decided to open a new office in Manhattan. They hired a data scientist in order to find the most suitable location of this new office, in which the shipment costs are minimized and the demand is high.
This project will explore where the highest concentration of restaurants is located. This minimizes the shipment costs (since we would have an higher number of venues close to the new office) and improves the productivity, since the single shipment requires less time and more can be done in the same time interval.
We will also look at the highest average score of each cluster of restaurants, since each costumer is likely to order from an higher rated restaurant (in terms of value for money). Higher ratings also favours customer loyalty and those venues are less likely to go out of business. This also means that if the food delivery company establishes business relationships with those restaurants, those relationships are likely to be

durable and the problem of finding new partners to sustain the company is minimized.

2. Data

Data containing information on all the neighborhoods in New York are found at https://cocl.us/new_york_dataset. From those data, we selected only the ones of Manhattan. Using the Neighborhood name, the Latitude and Longitude values we performed API calls (spacifying User ID, User Secret and Version) to FourSquare. Those iterative calls give back several json containing information on all the venues in Manhattan. Since we are interested in Restaurants, we selected only venues classified as 'Restaurants' in the 'Venue Category' key.
From all those restaurants we selected only a few, through clustering techniques. For the three most densly populated sub-cluster of a previously defined bigger cluster, we made premium API calls to FourSquare, obtaining a json file for each restaurant. From this file we retrieved all the scores.

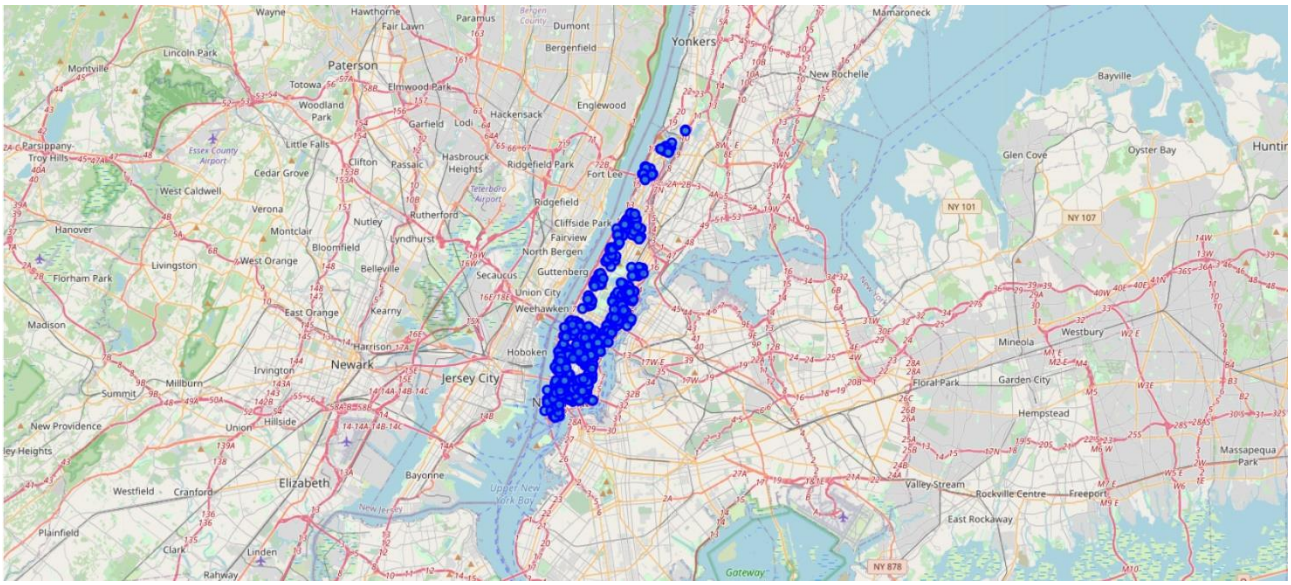
3. Methodology

From the data of New York neighborhoods, we only selected the ones with 'Manhattan' label. With this smaller database, we performed a cycle of API calls to FourSquare in order to obtain latitude, longitude, name, id and category of all the venues in Manhattan.
Since we are interested in venues that can be linked to a food-delivery business, we selected all the venues having the key word 'Restaurant' in their venue category.
We used Folium in order to visually represent our results. The represented points were grouped with the use of a machine learning technique: DBSCAN (Density-based spatial clustering of applications with noise) Clustering. This technique (Unsupervised Machine Learning) works well with points characterized by their geographical

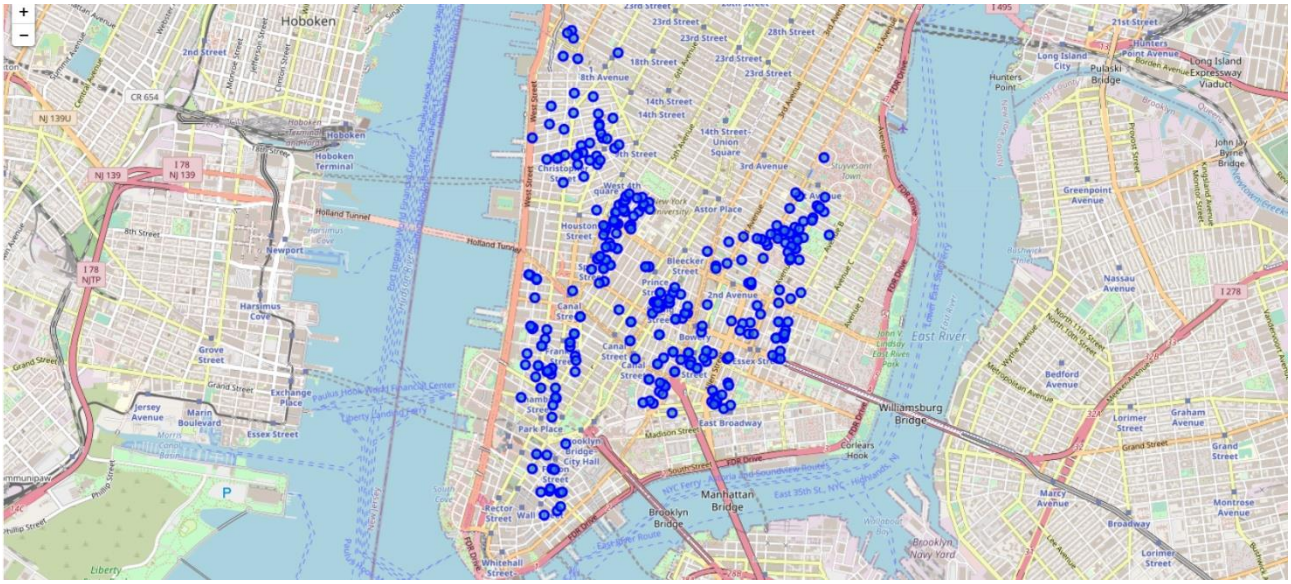
coordinates and forms Clusters considering two parameters: density and number of points. The number and dimensions of clusters are controlled by two parameters in the Scikit-Learn method DBSCAN: ‘eps’ and ‘min_samples’. Two points are considered neighbors if the distance between the two points is below the threshold ‘eps’, while ‘min_samples’ represents the minimum number of neighbors a given point should have in order to be classified as a core point.
This clustering technique was applied two times. The first time in order to select the area of Manhattan with the highest nr of restaurants, the second time to restrict the area covered by the food-delivery office.
API calls were sent in order to get the rating of each restaurant in three sub-clusters. The cluster with highest average rating was then selected.

4. Results Section

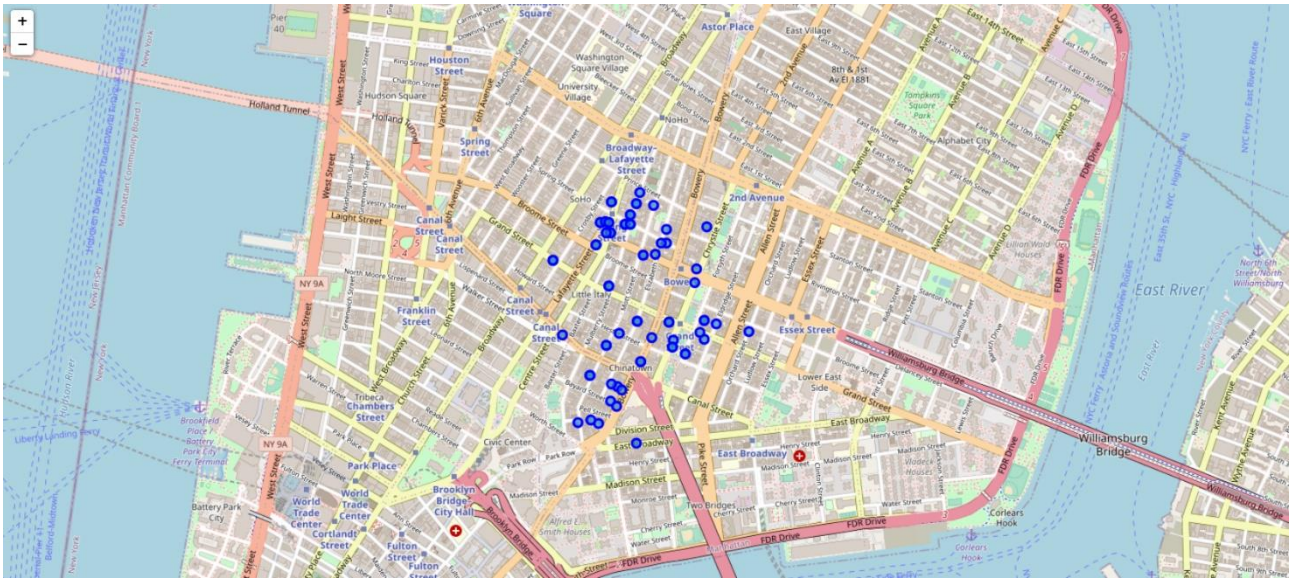
We first represented all the restaurants of Manhattan through the Python Folium library:



In this map, a total of 891 restaurants are represented.
In order to restrict the area of our delivery office, DBSCAN Clustering was performed on the total number of restaurants in Manhattan. The cluster with higher density and number of restaurants was selected.



In this cluster we have a total of 316 restaurants, which are too many for our office to cover. So we proceded to form sub-clusters of this cluster using the same technique used before (DBSCAN Clustering). We then selected 3 clusters out of the 9 formed, considering the number of restaurants in each one and their density. Even if one cluster had the high number of restaurants, this was not selected among those three since its point density was very low and it was covering a large area.
The obtained clusters are the following:



With a total of 70 restaurants is the biggest sub-cluster in terms of number of points. It covers the neighborhoods of China Town (24

restaurants), Little Italy (30 restaurants), Soho (15 restaurants) and Noho (only 1 restaurant).

With a total of 45 restaurants. It covers the neighborhoods of East Village (37 restaurants) and Noho (8 restaurants).

With a total of 53 restaurants. It covers the neighborhoods of Greenwich Village (42 restaurants) and Soho (11 restaurants).

In order to select one out of these cluster, we introduced a second criterion: the rating of each restaurants.
We obtained those values through premium API calls to FourSquare (by specifying the venue id of the restaurants) and performed the average of those values for each cluster.

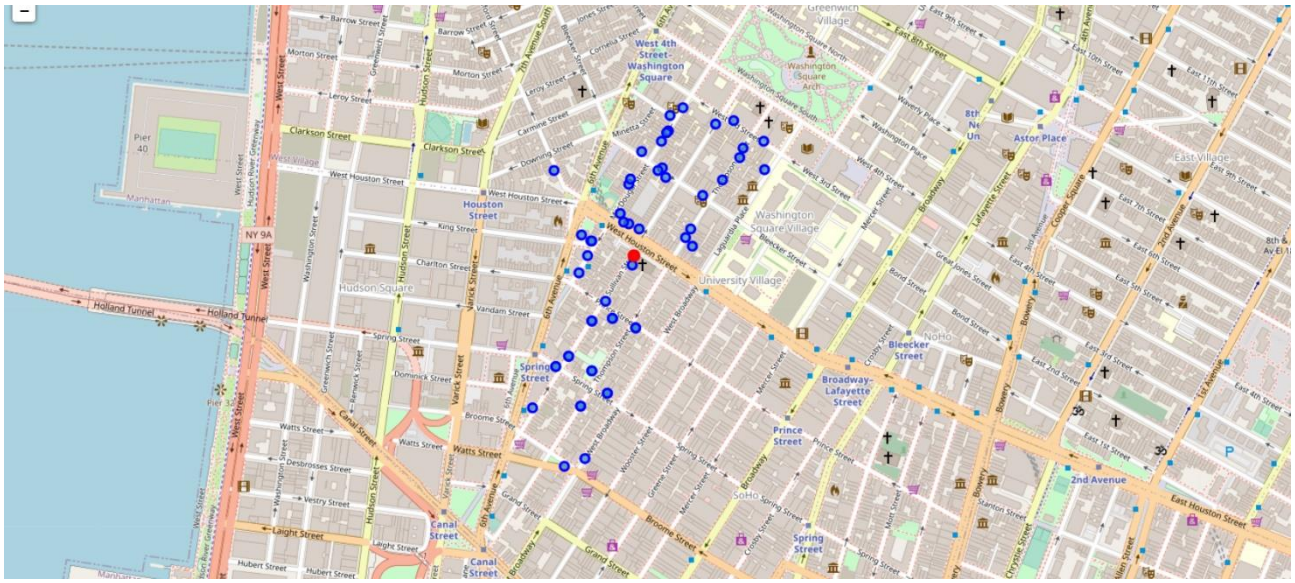
The retrieved average ratings were:

- 8.60857142857143 for the first cluster.
- 8.637777777777776 for the second cluster.
- 8.752830188679246 for the third cluster.

The third cluster was chosen since its average rating is the highest. It is also worth noticing that the street connecting all restaurants a parallel to each other, thus granting better viability from a restaurant to the other.

Finally, the office location was computed by averaging the latitude and longitude coordinates of all points in the selected cluster. This defines our food-delivery office as the centroid of the cluster.

Office location: 40°43'38.7"N 74°00'05.6"W. Red point in figure.



In Google Maps:

5. Discussion Section

We were able to determine a precise location of the food-delivery office, but in the clustering process some aggregation of restaurants were sparse. This problem (which in our project affected only one cluster) can be solved by optimizing the ‘eps’ and ‘min_samples’ parameters in the DBSCAN method. We also decided to discard every other cluster in the first step based only on the number of restaurants; but it is highly probable that in some other clusters with lower total number of restaurants there are clusters with higher densities and number of restaurants respect to the selected one. This brings us to conclude that the suggested choice is a very good location, but it is possible that it’s not the optimal one in Manhattan.

6. Conclusion

Thisproject is a simple proof of concept. Many other factors can be taken into account in order to determine the best location of the food-delivery office, as number of reviews, number of daily customers, road traffic etc.
The number of information was limited by the daily quota of premium API calls to FourSquare and only restaurants information were retrieved. However, many other venues produce food that can be delivered (for ex. Ice-cream shops, bagel shops etc.).