# IBM Capstone Project
# The Battle of Neighborhoods

Ines Ben Ameur

May 15, 2019

## Contents

# 1 Introduction

This report details the specifications of my final capstone project for IBM Applied Data Science Capstone course in Coursera.
For this project, I'm going to use the Foursquare location data and machine learning tools in order to segment and cluster neighborhoods in Athens, Greece.
Being a fan of both Italian food and the grandiose Greek capital, I chose to setup a tool that helps an Italian restaurateur to open an Italian restaurant in Athens. Since Italian food is very spread worldwide, it is quite common to come across a Neapolitan or Sicilian restaurant in any neighborhood or city. Therefore finding the most suitable location to open such a restaurant is one of the most important decisions for this entrepreneur as a good location is one of the pillars of a profitable business.

# 2 Business Problem

The objective of this capstone project is to help an Italian entrepreneur to look for the most fitting location to open a new Italian restaurant in Athens, Greece. By using machine learning methods such as clustering and a location data platform like Foursquare, we will extract features and characteristics of neighborhoods in Greece, cluster them and finally determine the ideal place for opening the restaurant.
In order to find this ideal location, we will try to detect locations that are not already crowded with Italian restaurants. We are also particularly interested in areas with attractions in vicinity (movie theatres, Amphitheater, etc).

# 3 Explored Data

The first step of building the tool would be naturally to specify the data we're going to need and then collect it from the adequate sources.

## 3.1 Data specification

In order to be able to cluster the neighborhoods of Athens we need:

- List of neighborhoods in Athens: It is necessary to collect the neighborhoods of the city of Athens to apply the clustering.
- Latitude and Longitude of the neighborhoods: We need the geolocation coordinates of each neighborhood in order to extract the venues located in each one.
- Venues: Finally we need to extract the list of Italian restaurants along with different types of attraction venues in each neighborhood to complete our data collection. For the first step of data collection, we will load all sort of venues provided by Foursquare, then we will filter the places with the types we desire.

## 3.2 Data Sources

We used the following data sources to collect the features specified above:

- List of neighborhoods in Athens: This part is collected from this Wikipedia page [1].
- Latitude and Longitude of the neighborhoods: The coordinates are extracted via **geopy.geocoders** python package .
- Venues: The venues and their attributes are provided by the Foursquare API [2]

### 3.3 Data Samples

- List of neighborhoods in Athens and their coordinates:

| Neighborhood | Latitude | Longitude |
|---|---|---|
| Sepolia | 38.006449 | 23.717277 |
| Skouze Hill | 37.977363 | 23.728436 |
| Thiseio | 37.976766 | 23.720329 |
| Treis Gefyres | 38.014098 | 23.718425 |
| Vathi | 35.359413 | 23.595234 |

- Venues: The venues and their attributes are provided by the Foursquare API (before Filtering:

| Neigh | Neigh.Lat | Neigh.Long | Venue | V.Lat | V.Long | V.Cat |
|---|---|---|---|---|---|---|
| Aerides | 36.147765 | 22.989737 | Fossa | 36.148441 | 22.988234 | Café |
| Aerides | 36.147765 | 22.989737 | SOCHORA | 36.148275 | 22.988419 | Boutique |
| Aerides | 36.147765 | 22.989737 | Mercato | 36.148107 | 22.988716 | Bar |
| Aerides | 36.147765 | 22.989737 | Coffee Island | 38.020038 | 23.731612 | Coffee Shop |
| Aerides | 36.147765 | 22.989737 | choraki | 36.148251 | 22.988503 | Cocktail Bar |

- Venues: We filter the locations and keep only Italian restaurants and attractions:

| Neigh | Neigh.Lat | Neigh.Long | Venue | V.Lat | V.Long | V.Cat |
|---|---|---|---|---|---|---|
| Aerides | 36.147765 | 22.989737 | Belvedere | 36.148118 | 22.989718 | Italian Restaurant |
| Akadimia | 37.980285 | 23.734528 | Piadina L' Umbro | 37.979424 | 23.735630 | Italian Restaurant |
| Akadimia | 37.980285 | 23.734528 | La Pasteria | 37.978384 | 23.736132 | Italian Restaurant |
| Akadimia | 37.980285 | 23.734528 | Frankie | 37.980583 | 23.737988 | Italian Restaurant |
| Akadimia | 37.980285 | 23.734528 | Il Postino | 37.982145 | 23.736906 | Italian Restaurant |

## 4 Methodology

In this section, we're going to specify and discuss all the steps that led us to cluster neighborhoods in Athens and help us determine the perfect location for an Italian business setup.

### 4.1 Loading Neighborhoods

As mentioned in the data section, we obtained the list of neighborhoods in Athens from a Wikipedia page. In order to construct a Python object (Dataframe) to analyze and treat the data, we need to sparse the page and collect the names in the tables.
For that purpose, we used BeautifulSoup [3] and stored the list in a Dataframe.

| | Neighborhood |
|---|---|
| 0 | Aerides |
| 1 | Agios Eleftherios |
| 2 | Agios Panteleimonas |
| 3 | Akadimia Platonos |
| 4 | Akadimia |

Figure 1: First 5 rows of the neighborhoods

### 4.2 Neighborhoods' coordinates

Next, we need to look for the geolocation coordinates (Latitude, Longitude) of each neighborhood so we can send a query to Foursquare and get a list of the venues. The following Figure 2 represents the list of places we obtained with their coordinates:

| | Neighborhood | Latitude | Longitude |
|---|---|---|---|
| 0 | Aerides | 36.147765 | 22.989737 |
| 1 | Agios Eleftherios | 38.020044 | 23.731724 |
| 2 | Agios Panteleimonas | 37.607478 | 26.096458 |
| 3 | Akadimia Platonos | 37.989357 | 23.711217 |
| 4 | Akadimia | 37.980285 | 23.734528 |
| 5 | Ampelokipoi | 37.758008 | 20.871949 |
| 6 | Anafiotika | 37.972351 | 23.728043 |
| 7 | Ano Petralona | NaN | NaN |
| 8 | Asteroskopeio | 37.973125 | 23.719985 |
| 9 | Asyrmatos | NaN | NaN |

Figure 2: Coordinates of the neighborhoods

As we can see above, we need to clean the data since we couldn't extract the coordinates of some of the neighborhoods, therefore we must drop all the rows with empty coordinates.

### 4.3 Neighborhoods' Map

Figure 3 shows a map to visualize the data we cleaned:

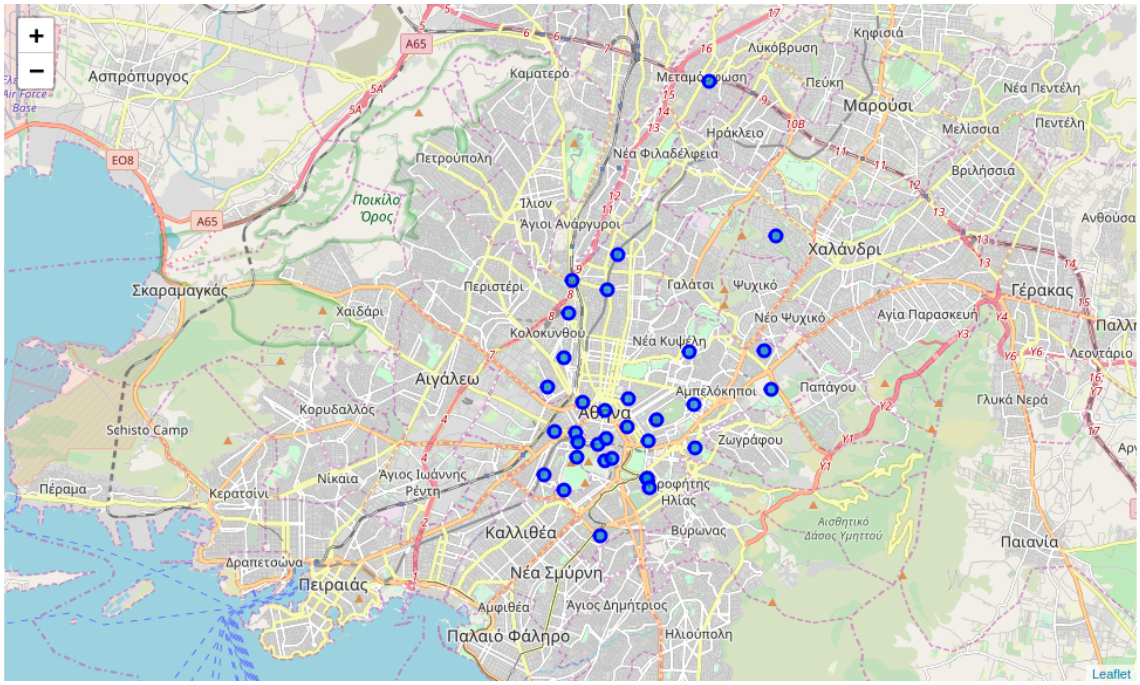


Figure 3: Map of the neighborhoods

## 4.4 Collect Venues

Now that we have our location candidates, let's use Foursquare API to collect all the venues in each neighborhood. After we get the venues from the API, we need to filter them and keep Italian restaurants and attractions. We need to send an **explore** query to the API for each neighborhood containing:

- Both Client ID and secret key.
- The version of the API that we desire.
- Latitude and longitude of the location.
- The radius within which we're going to explore the venues.
- The limit number of venues to explore.

We get a table looking like this:

| Neigh | Neigh.Lat | Neigh.Long | Venue | V.Lat | V.Long | V.Cat |
|---|---|---|---|---|---|---|
| Aerides | 36.147765 | 22.989737 | Fossa | 36.148441 | 22.988234 | Café |
| Aerides | 36.147765 | 22.989737 | SOCHORA | 36.148275 | 22.988419 | Boutique |
| Aerides | 36.147765 | 22.989737 | Mercato | 36.148107 | 22.988716 | Bar |
| Aerides | 36.147765 | 22.989737 | Coffee Island | 38.020038 | 23.731612 | Coffee Shop |
| Aerides | 36.147765 | 22.989737 | choraki | 36.148251 | 22.988503 | Cocktail Bar |

## 4.5 Filter Venues

As we mentioned in the previous sections, we're only going to need to count the number of italian restaurants per location. Therefore we should get rid of the unneeded venues and keep a list containing the following attractions:

- Art Gallery
- Concert Hall
- Movie Theater

- Music Venue
- Museum
- Pub
- Performing Arts Venue

Then we should form a new list out of this one, containing the number of attractions per location. We get a table like this one:

| | Neighborhood | Number of attractions |
|---|---|---|
| 0 | Akadimia | 1.0 |
| 1 | Akadimia Platonos | 1.0 |
| 2 | Anafiotika | 2.0 |
| 3 | Asteroskopeio | 3.0 |
| 4 | Ellinoroson | 1.0 |

Figure 4: Number of attractions per location

Using the same procedure, we create a new column containing the number of Italian restaurants:

| | Neighborhood | Number of Italian Restaurants |
|---|---|---|
| 0 | Aerides | 1.0 |
| 1 | Akadimia | 1.0 |
| 2 | Ellinoroson | 2.0 |
| 3 | Kallimarmaro | 6.0 |
| 4 | Kallimarmaro | 6.0 |

Figure 5: Number of Italian restaurants per location

### 4.6 Clustering

After merging the 2 tables we created above, we get our final dataset that will help us partition our neighborhoods into groups of individuals that have similar characteristics, based on the 2 features we have.
For that purpose we are going to use a very popular clustering algorithm, called **k-means** which is a form of unsupervised machine learning. K-means can group data only unsupervised based on the similarity of candidates to each other. That is, it divides the data into k non-overlapping subsets or clusters without any cluster internal structure or labels. Which means that objects within a cluster are very similar and objects across different clusters are very different or dissimilar.
Using a ready python library called **sklearn.cluster.KMeans** that has this algorithm implemented, we'e going to fit our data with a 3-clusters algorithm to partition the neighborhoods into 3 groups.

## 5 Results

After running the kmeans algorithm on our dataset, we need to prepare a new table containing a new column for the cluster labels, that is the number of cluster to which each neighborhood belongs:

| | Neighborhood | Latitude | Longitude | Cluster Labels | Number of Italian Restaurants | Number of attractions |
|---|---|---|---|---|---|---|
| 0 | Aerides | 36.147765 | 22.989737 | 1 | 1.0 | 0.0 |
| 1 | Agios Eleftherios | 38.020044 | 23.731724 | 1 | 0.0 | 0.0 |
| 2 | Agios Panteleimonas | 37.607478 | 26.096458 | 1 | 0.0 | 0.0 |
| 3 | Akadimia Platonos | 37.989357 | 23.711217 | 1 | 0.0 | 1.0 |
| 4 | Akadimia | 37.980285 | 23.734528 | 1 | 1.0 | 1.0 |

Figure 6: List of clusters and neighborhoods

Let's visualize the clustered neighborhoods on the map:

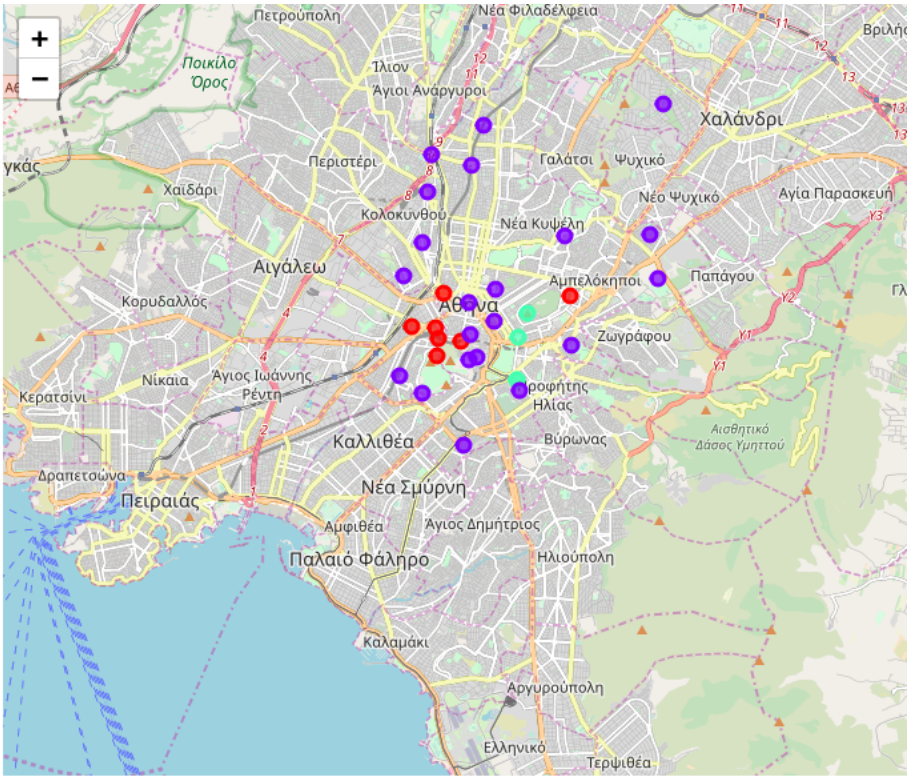


Figure 7: Clusters on the map

In order to analyse the clusters we have, let's visualize histograms for each cluster:

- **Cluster 1:**

| Neighborhood | Latitude | Longitude | Cluster Labels | Number of Italian Restaurants | Number of attractions |
|---|---|---|---|---|---|
| **Asteroskopeio** | 37.973125 | 23.719985 | 0 | 0.0 | 3.0 |
| **Kerameikos** | 37.978730 | 23.719506 | 0 | 0.0 | 3.0 |
| **Kountouriotika** | 37.985335 | 23.754196 | 0 | 0.0 | 4.0 |
| **Metaxourgeio** | 37.985853 | 23.721380 | 0 | 0.0 | 3.0 |
| **Monastiraki** | 37.976273 | 23.725929 | 0 | 0.0 | 3.0 |
| **Pedion tou Areos** | 40.622526 | 22.955472 | 0 | 0.0 | 4.0 |
| **Rouf** | 37.979114 | 23.713333 | 0 | 0.0 | 7.0 |
| **Thiseio** | 37.976766 | 23.720329 | 0 | 0.0 | 6.0 |

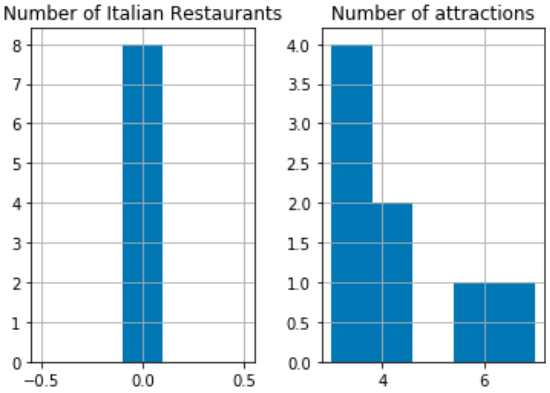


Figure 8: First Cluster

- **Cluster 2:**

| Neighborhood | Latitude | Longitude | Cluster Labels | Number of Italian Restaurants | Number of attractions |
|---|---|---|---|---|---|
| **Aerides** | 36.147765 | 22.989737 | 1 | 1.0 | 0.0 |
| **Agios Eleftherios** | 38.020044 | 23.731724 | 1 | 0.0 | 0.0 |
| **Agios Panteleimonas** | 37.607478 | 26.096458 | 1 | 0.0 | 0.0 |
| **Akadimia Platonos** | 37.989357 | 23.711217 | 1 | 0.0 | 1.0 |
| **Akadimia** | 37.980285 | 23.734528 | 1 | 1.0 | 1.0 |

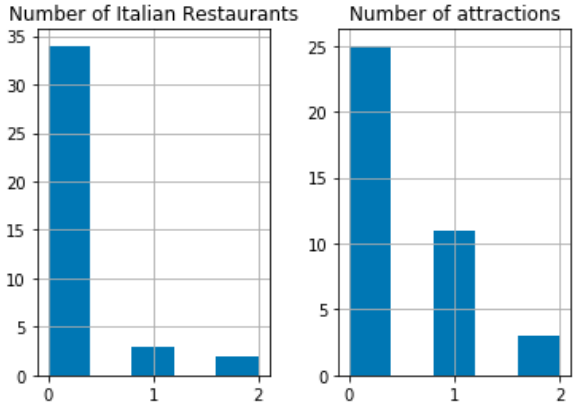


Figure 9: Second Cluster

- **Cluster 3:**

| Neighborhood | Latitude | Longitude | Cluster Labels | Number of Italian Restaurants | Number of attractions |
|---|---|---|---|---|---|
| **Kallimarmaro** | 37.968424 | 23.740403 | 2 | 6.0 | 2.0 |
| **Kallimarmaro** | 37.968424 | 23.740403 | 2 | 6.0 | 2.0 |
| **Kolonaki** | 37.976975 | 23.740814 | 2 | 4.0 | 2.0 |
| **Mount Lycabettus** | 37.981886 | 23.743152 | 2 | 7.0 | 4.0 |

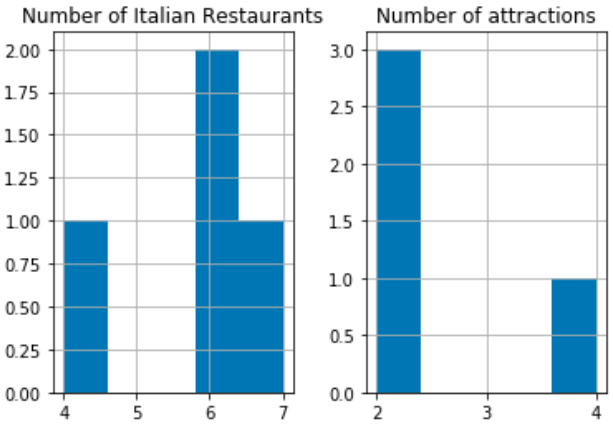


Figure 10: Third Cluster

# 6 Discuss Observations

Based on the graphs shown above, we can clearly notice that we obtained 3 types of neighborhoods:

- Cluster 1: This cluster is characterized by a low number of italian restaurants (0) and a high number of attractions
- Cluster 2: This group is not characterized by the presence of italian restaurants nor attractions.
- Cluster 3: Finally, we have a group of neighborhoods that has a high number of italian restaurants but not a very high number of attractions.
  Obviously, our Italian entrepreneur should consider the neighborhoods within **cluster 1** to setup his business.

# 7 Conclusion

In conclusion, in this project we tried to find a solution for a recurrent and quite interesting problem that many entrepreneurs can face: Finding the best location for a profitable investment. We took the example of an Italian restaurant in the city of Athens, Greece but note that we can follow the same procedure for any type of business in any city in the world!
There are two main important steps to follow in order to get good and convincing results:

- Collect and create a clean and solid data set: The features and the values present in the data set should make sense and have added value. Also we should make sure that the entries we posses are consistent and integrated.
- Choose the number of clusters wisely, a small number of clusters can hide some characteristics and information also a high number of clusters can lead to over fitting, dilute the information that the features bring and therefore we cannot have conclusive results.

## References


[1] Wikipedia, "Category:neighbourhoods in athens." [Online]. Available: https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Athens

[2] Foursquare, "Foursquare api." [Online]. Available: https://foursquare.com/

[3] BeautifulSoup, "Beautiful soup documentation." [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/