Optimizing the Maintenance of Air Quality Stations

Pedro Figueiras

1. Introduction.

I am the owner of an electrical services company in Madrid, Spain. In a public contest, my company got awarded by the town hall with the maintenance contract for the "Air quality stations" (AQS) that are deployed in different areas of the city. According to the resolution, I have just two weeks per year to manage the job.

2. Business problem.

The area where the AQS are installed is big and my resources are limited. I will just have 4 service teams to address the job, so I need to optimize the travelling time and avoid wasting time during lunch time.

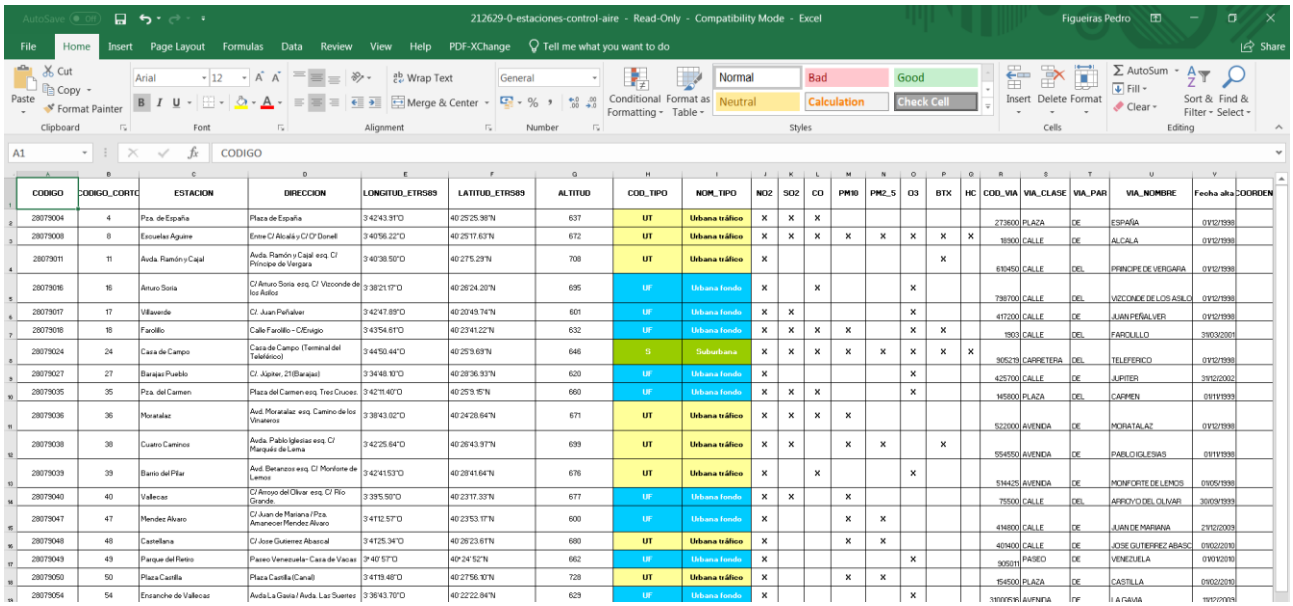
My idea is to assign the most convenient stations (location-wise) for each Service team. Also, to be productive reducing the restaurant search for having lunch, I've got the idea of finding the closest (but recommended) mid-price restaurant to each station. I'm sure my guys will be glad to know where to go.

3. Data sources.

I'm used two main sources for the data:

- Madrid's data webpage (https://datos.madrid.es): This is a new web service where the city is uploading multiple datasets with various information (traffic, air quality, weather, etc).

  In this case, I used the dataset with all the Air Quality Stations where my guys have to make maintenance. Here the link: https://datos.madrid.es/egob/catalogo/212629-0-estaciones-control-aire.xls; this is how it looked:



- Foursquare data using the API for developers. In this case, I managed a query using the new feature "Recommendations", where I was able to find the closest venue per type (restaurant) to a certain location (AQS location) with specific characteristics (medium price). This is the shape of the url:

  *https://api.foursquare.com/v2/search/**recommendations**?client_id={}&client_secret={}&ll={},{}&v={}&categoryId={}&radius={}&limit={}&prices{}&'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, cat, radius, LIMIT, prices)*

  The response that I got by Foursquare is in JSON format, something like this per query:


```
{'meta': {'code': 200, 'requestId': '5d7abf56d69ed0002c75ee9e'}, 'respons
e': {'group': {'results': [{'displayType': 'venue', 'venue': {'id': '4b65
d0a0f964a52079012be3', 'name': 'Casa Pedro', 'location': {'address': 'C.
Nuestra Señora de Valverde, 119', 'lat': 40.49754295286697, 'lng': -3.687
0403738295425, 'labeledLatLngs': [{'label': 'display', 'lat': 40.49754295
286697, 'lng': -3.6870403738295425}], 'distance': 404, 'postalCode': '280
34', 'cc': 'ES', 'city': 'Madrid', 'state': 'Madrid', 'country': 'España'
, 'formattedAddress': ['C. Nuestra Señora de Valverde, 119', '28034 Madri
d Madrid', 'España']}, 'categories': [{'id': '4bf58dd8d48988d150941735',
'name': 'Spanish Restaurant', 'pluralName': 'Spanish Restaurants', 'short
Name': 'Spanish', 'icon': {'prefix': 'https://ss3.4sqi.net/img/categories
_v2/food/spanish_', 'suffix': '.png'}, 'primary': True}], 'dislike': Fals
e, 'ok': False}, 'id': '5d7abf564e53600008a564c2', 'photo': {'id': '4f775
732e4b0e8caa5b7089d', 'createdAt': 1333221170, 'prefix': 'https://fastly.
4sqi.net/img/general/', 'suffix': '/f42QnQpkwXI5GJDwB-tBxZqgzsg3F0jZ-U7oA
Zxr5Y0.jpg', 'width': 720, 'height': 537, 'visibility': 'public'}, 'snipp
ets': {'count': 1, 'items': [{'detail': {'type': 'tip', 'object': {'id':
'4be9cb9c70c603bb691c9bb4', 'createdAt': 1273613212, 'text': 'Try the ste
ak tartar. Probably, the best of Madrid', 'type': 'user', 'canonicalUrl':
'https://foursquare.com/item/4be9cb9c70c603bb691c9bb4', 'logView': True,
'agreeCount': 0, 'disagreeCount': 0, 'todo': {'count': 0}, 'user': {'id':
'1198659', 'firstName': 'Jesus', 'lastName': 'Medina', 'gender': 'male',
'photo': {'prefix': 'https://fastly.4sqi.net/img/user/', 'suffix': '/BTYP
POH443MYBPDE.jpg'}}}}}]}}], 'totalResults': 26}, 'context': {'searchLocat
ionNearYou': True, 'searchLocationMapBounds': False, 'searchLocationType'
: 'NearYou', 'currentLocation': {'what': '', 'where': '', 'feature': {'cc
': 'ES', 'name': 'Valverde', 'displayName': 'Valverde, Madrid', 'woeType'
: 22, 'slug': 'valverde-spain', 'id': 'maponics:66861', 'longId': '66861'
, 'geometry': {'center': {'lat': 40.50024, 'lng': -3.678757}, 'bounds': {
'ne': {'lat': 40.520147, 'lng': -3.656786}, 'sw': {'lat': 40.482514, 'lng
': -3.700931}}}, 'encumbered': True}, 'parents': []}, 'boundsSummaryRadiu
s': 1000, 'relatedNeighborhoods': [], 'geoParams': {'ll': '40.5005477,-3.
6897307999999995', 'radius': '1000'}, 'geoBounds': {'circle': {'center':
{'lat': 40.5005477, 'lng': -3.6897307999999995}, 'radius': 1000}}}}}
```


4. Data acquisition and preparation.

The first thing I did was preparing the AQS dataset. These are the steps I followed:

- Firstly, I created a new data frame "df" using "Pandas" where I copied all info of the original Excel file.

- I copied the columns with the location (latitude and longitude) of all the stations into a new data frame ("df2"). I applied the K-means method on it with k=4 to optimally distribute (distance-wise) the 4 groups of stations between the 4 existing Service team.

- “df2” data frame was expanded with two new columns: “AQS” -> the name of the stations (copied from “df” dataframe) and “SER TEAM” -> the service team assignation as result of K-means method (K-means labels to be more specific). Here the result of the first 5 rows of the data frame:

	AQS	SER TEAM	LATITUD	LONGITUD
0	Pza. de España	0	40.423882	-3.712257
1	Escuelas Aguirre	0	40.421553	-3.682316
2	Pza. Elíptica	0	40.385034	-3.718768
3	Parque del Retiro	0	40.414444	-3.682500
4	Villaverde	0	40.347147	-3.713317

The next step was filtering and preparing the data that I got in JSON format as result of the API query to Foursquare:

- I used the “json_normalize” function from “pandas.io.json” to filter and create new data frame (named “dataframe”) with some information of the query results. Concretely, the name of the restaurant and the location (latitude, longitude):

	venue.name	venue.location.lat	venue.location.lng
0	Casa Pedro	40.497543	-3.68704

- Using a “for” loop, I created a new data frame “df3” where I appended all the new rows generated in “dataframe” with the information I mentioned in the previous point. Finally, I renamed the columns.

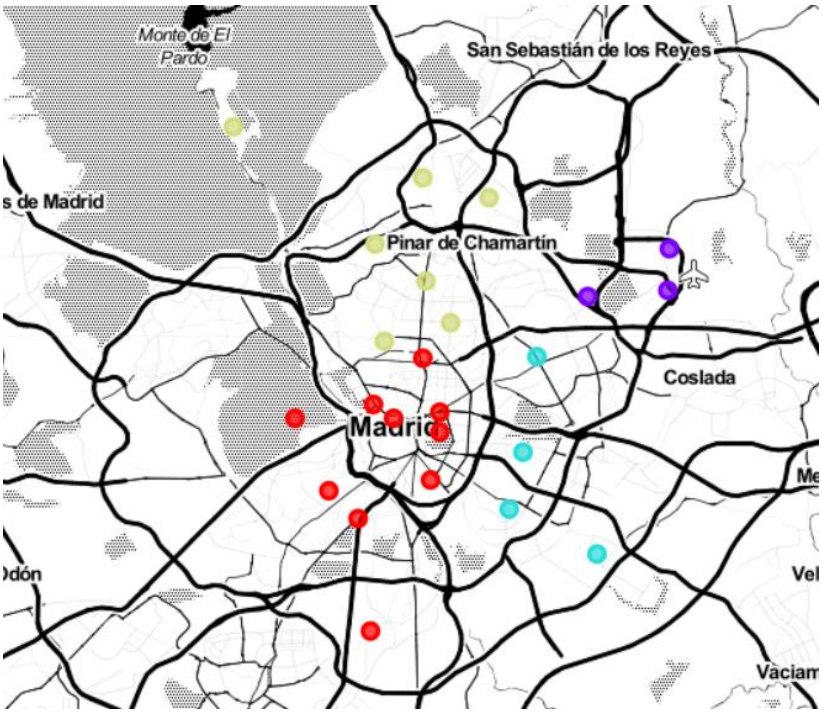
When “df3” was ready, I decided to merge “df2” and “df3” in a new data frame, called “df_map” where I compiled the AQS info and the related recommendation by Foursquare:

	AQS	SER TEAM	LATITUD	LONGITUD	RESTAURANTE	LAT_R	LON_R
0	Pza. de España	0	40.423882	-3.712257	Club Allard	40.423028	-3.713987
1	Escuelas Aguirre	0	40.421553	-3.682316	Kabuki Wellington	40.421891	-3.684217
2	Pza. Elíptica	0	40.385034	-3.718768	La Peña Soriana	40.382060	-3.711332
3	Parque del Retiro	0	40.414444	-3.682500	Levél Veggie Bistro	40.415730	-3.677641
4	Villaverde	0	40.347147	-3.713317	Restaurante la Mancheguita 1978	40.342981	-3.711628
5	Farolillo	0	40.394782	-3.731836	Döner Kebap Nemrut	40.388257	-3.730849

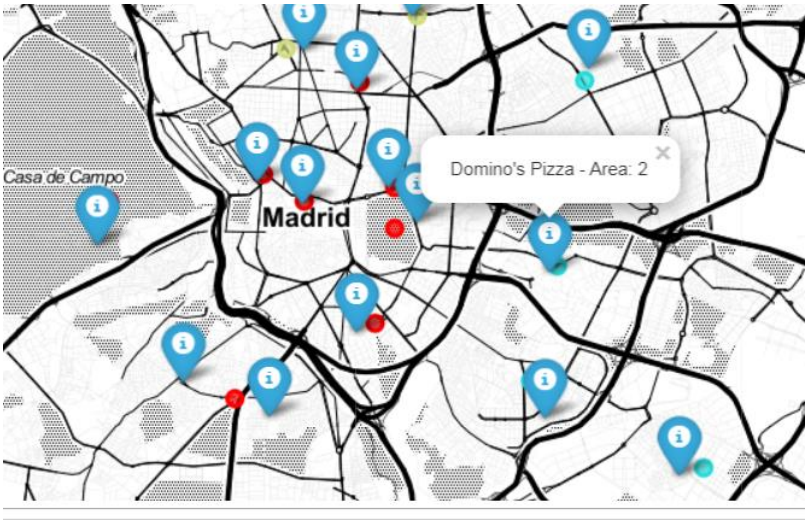
5. Data analysis.

Main comments about data analysis are related to the K-means method utilized for clustering the AQS in 4 areas. The purpose of such clustering was to assign each area to a specific Service team to optimize the travelling time. For that reason, it was clear that k = 4.

As it can be seen in the map (generated used Folium), the clustering made by the algorithm is very convenient to distribute the work. Each circle marker represents one AQS; the different colors represent the 4 areas determined through K-means:



Adding the recommended restaurants to the map, we can notice that Foursquare's API was very accurate with the search criteria we applied:



6. Conclusion

Clustering the AQS location gave me a clear advantage to assign the work to each Service team more efficiently, minimizing the travelling time. Also, just using the AQS location and Foursquare API, I was able to find a recommended restaurant close to each AQS for the guys.

I’m sure this will make their life easier and their days of work more effective.