Capstone Project The Battle of Neighborhoods

REPORT

JANUARY 22, 2020

Authored by: Abilio Ribeiro Duarte
CDO| Data & AI | Data Scientist
ajrduarte@gmail.com

MLA Inc.

Table of Contents

Executive Summary

MLA inc., is targeting to expand their business to UK, they want to launch 5 new Gym / Fitness Centers services in 2020 in the City of London. The rollout of those Gym's should follow for other main cities of UK from 2021 onwards.

Multiple approaches assessed and 5 districts out of total of 23 have been evaluated with focus on the neighborhood's where "Gym / Fitness Center" would features as most common venue and this way be able to recommend to senior management which neighborhood would best fit for the setup of the first 5 Gym's.

Recommendation is to setup the first five Gym / Fitness Center in the following neighborhood's:
EC2M 2PF, and EC2M 7LA at Tower Hamlets area.
EC3R 6BR, and EC3R 8AH at City of London area.
EC1M 4AA, at Islington, Camden, City of London area.

The Data Science evaluation highlighted that the main benefits come from setting up those Gym / Fitness Centers in areas where this venue has the highest trend and therefore those locations present higher probability of business success to disrupt and enhance with innovative Gym concept from MLA inc.

1. Introduction

The MLA Inc. founded in 2011, with headquarters in Chicago has experienced year on year a substantial growth on their business of supplying Gym and Gym appliances.

The innovative concept of Gym, associated with Digitization, AI, and his strong brand "GoGo Fitness", allow MLA Inc. to become the market leader in USA.

The MLA vision and strategy are to expand to markets outside USA. In 2019, the executive board from MLA Inc., chaired by Mr. J.L. Champs (CEO), has decided to expand their Gym concept to UK. The business plan is to start in 2020, with 5 new gym in the City of London, and grow from there to other main cities in the subsequent years.

The MLA business expansion to UK has enormous importance in shareholders strategy, therefore Mr. J. L. Champs has set this program at the highest priority for 2020.
Mr. Champs has assigned 3 of his champions for this program coded "The Battle of Neighborhoods".

The program main stakeholders and sponsors are Mrs. S. K. Voort (CMO), Mr. L. A. Raidillon (CDO), and Mrs. C. H. Nagasaki (COO).

Mr. Raidillon has agreed with his peers to bring on board his senior data scientist (Mr. A. R. Duarte) to help with business problem resolution.

1.1 Business Problem

Due to the high importance of "The Battle of Neighborhoods" program, it is from upmost importance to select the right locations for the first 5 Gyms in the City of London.

The stakeholders came up with following requirements (business problem):

#	Business Problem / Requirements
BR_01	Which neighborhood in the City of London has 'Gym' in top 10 venues?
BR_02	Which neighborhood in the City of London has "Gym" in the top 5 most common venues?
BR_03	How is the Gym market segmented in the City of London?
BR_04	Which are the best recommended districts in the City of London to setup the 5 gyms?

1.2 Business Understanding

The Data Science team has been tasked to produce a report to highlight the Gym market segment distribution across the boroughs and neighborhoods of the City of London.

The report should feature the top 5 districts where Gym is within the top 5 most common venues.

The report should explore and cluster the neighborhoods in the City of London regarding Gym venue, and present findings and conclusions.

2. Methodology

This section describes the research methods and data sources used for analysis as well as the Data requirements, data collection, data preparation, and modeling.

2.1 Analytical Approach

The business problem requests customer segmentation per location.
The data science team decided to explore the data with a cluster algorithm to see whether a natural statistical separation exists.

The clustering technique separates the data set into significant groups or buckets. Cluster algorithms attempt to divide the data into distinct groups by minimizing the distance between data points within a cluster and maximizing the distance between clusters.

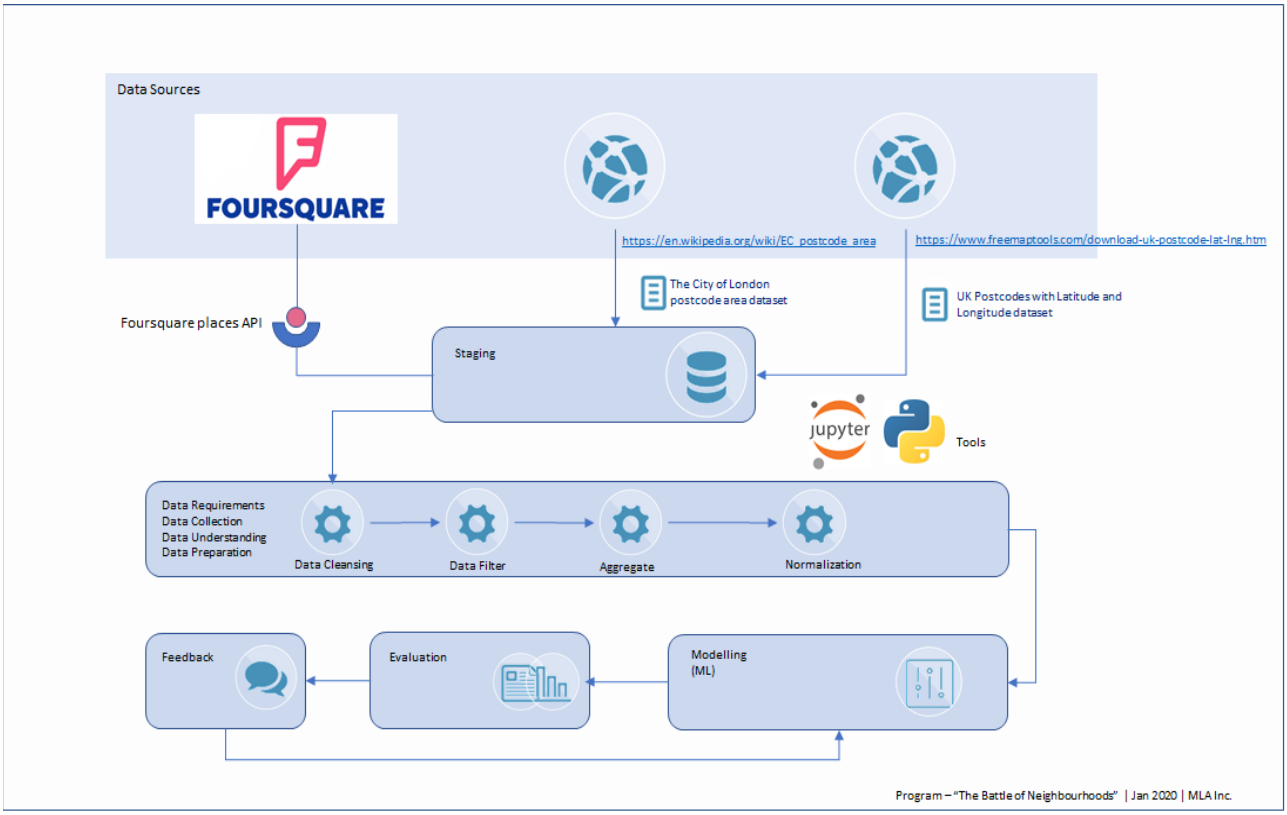
2.2 Data Sources

The team decided to leverage the Foursquare location data to solve the problem in combination with two other datasets.

The table below describes the data sources used in this analysis.

Data sources	Description
Foursquare	Using the Foursquare API (www.foursquare.com), one can search for specific type of venues or stores around a given location. The Places API offers real-time access to Foursquare' s global database of rich venue data and user content to power one's location-based experiences
UK Postcodes with Latitude and Longitude	This dataset contains all the UK postcodes with respective latitude and longitude, and can be found in the website **https://www.freemaptools.com/download-uk-postcode-lat-lng.htm**
The City of London postcode area	This dataset can be found in the Wikipedia **https://en.wikipedia.org/wiki/EC_postcode_area**

The diagram below depicts the high-level architecture used for this data science analysis.



The data collected via Foursquare places API will have the following format:

https://api.foursquare.com/v2/venues/_id/client_id=*****&client_secret=*******&v=version

The sample of the data collected in the UK postcode dataset and the City of London postcode dataset is shown below.

id	postcode	latitude	longitude
1	AB10 1XG	57.14417	-2.11485
2	AB10 6RN	57.13788	-2.12149
3	AB10 7JB	57.12427	-2.12719
4	AB11 5QN	57.1427	-2.0933
5	AB11 6UL	57.13755	-2.11223

Postcode district	Post town	Coverage	Local authority area
EC1A	LONDON	St Bartholomew's Hospital	City of London, Islington

EC1M	LONDON	Clerkenwell, Farringdon	Islington, Camden, City of London
EC1N	LONDON	Hatton Garden	Camden, City of London
EC1P	LONDON		non-geographic
EC1R	LONDON	Finsbury, Finsbury Estate (west)	Islington, Camden
EC1V	LONDON	Finsbury (east), Moorfields Eye Hospital	Islington, Hackney

2.3 Data Preparation

We have staged the "UK Postcodes with Latitude and Longitude" and "The City of London postcode area" datasets in a bucket in the IBM Cloud. After we have loaded those datasets in two pandas dataframe and did the data cleansing.

 # getting the datsets from the staging bucket.

!wget -q -O 'UKpostcodes_lat_long.csv' https://cloud-object-storage-kt-cos-standard-vw9.s3.us-south.cloud-object-storage.appdomain.cloud/ukpostcodes.csv
print('Data downloaded from url_1!')

 # load "UK postcodes" dataset to a pandas dataframe
uk_postcodes_df = pd.read_csv('UKpostcodes_lat_long.csv')

Let's inspect the 5 first items
uk_postcodes_df.head(5)

	id	postcode	latitude	longitude
0	1	AB10 1XG	57.144165	-2.114848
1	2	AB10 6RN	57.137880	-2.121487
2	3	AB10 7JB	57.124274	-2.127190
3	4	AB11 5QN	57.142701	-2.093295
4	5	AB11 6UL	57.137547	-2.112233

We have changed ast_note_interactivity kernel option in our Jupyter notebook in order to output any variable or statement on its own line, so we can see the value of multiple statements at once.

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

 # inpecting the dataframe for head and tail
uk_postcodes_df.head(5)
uk_postcodes_df.tail(5)

	id	postcode	latitude	longitude
0	1	AB10 1XG	57.144165	-2.114848
1	2	AB10 6RN	57.137880	-2.121487
2	3	AB10 7JB	57.124274	-2.127190
3	4	AB11 5QN	57.142701	-2.093295
4	5	AB11 6UL	57.137547	-2.112233

	id	postcode	latitude	longitude
1758221	2649010	YO8 9SN	53.775330	-1.129099
1758222	2649011	YO8 9SP	53.767117	-1.089415
1758223	2649012	YO8 9TR	53.771989	-1.120702
1758224	2649013	ZE2 9FJ	60.128408	-1.208825
1758225	2649014	ZE2 9FN	60.128408	-1.208825

Let's load also the "city of London district postcodes" to a pandas dataframe
load "City of London district postcodes" dataset to a pandas dataframe
!wget -q -O 'city-of-london-district-postcodes.csv' https://cloud-object-storage-kt-cos-standard-vw9.s3.us-south.cloud-object-storage.appdomain.cloud/city-of-london-district-postcode-areas.csv
print('Data downloaded from url_2!')

city_of_london_postcodes_df = pd.read_csv('city-of-london-district-postcodes.csv')

# inspect the 5 first items
city_of_london_postcodes_df.head()
city_of_london_postcodes_df.tail()
Data downloaded from url_2!

	Postcode district	Post town	Coverage	Local authority area
0	EC1A	LONDON	St Bartholomew's Hospital	City of London, Islington
1	EC1M	LONDON	Clerkenwell, Farringdon	Islington, Camden, City of London
2	EC1N	LONDON	Hatton Garden	Camden, City of London
3	EC1P	LONDON	NaN	non-geographic
4	EC1R	LONDON	Finsbury, Finsbury Estate (west)	Islington, Camden

	Postcode district	Post town	Coverage	Local authority area
23	EC4P	LONDON	NaN	non-geographic
24	EC4R	LONDON	Cannon Street	City of London
25	EC4V	LONDON	Blackfriars	City of London
26	EC4Y	LONDON	Temple	City of London, Westminster
27	EC50	LONDON	NaN	non-geographic

The dataframe uk_postcodes_df did contain 5 columns **dataframe_row_id| id (from original dataset) | postcode | latitude | longitude**. We don't need the id (from original dataset), so we have dropped this column.
we will drop the column 'id'

uk_postcodes_df.drop(['id'], axis=1, inplace=True)

uk_postcodes_df.tail()

	postcode	latitude	longitude
1758221	YO8 9SN	53.775330	-1.129099
1758222	YO8 9SP	53.767117	-1.089415
1758223	YO8 9TR	53.771989	-1.120702

1758224	ZE2 9FJ	60.128408	-1.208825
1758225	ZE2 9FN	60.128408	-1.208825

In this Uk_postcodes dataframe, we have observed that postcode is composed from District code + Neighborhood code. We have added a new column in this dataset which contains only the District code, since we will need this data when we plan to intersect this data frame with one from City of London district codes.

split_postcode_df = uk_postcodes_df["postcode"].str.split(" ", n = 1, expand = True)

 # making separate district column from split_postcode_df data frame
uk_postcodes_df['district'] = split_postcode_df[0]

 # making separate neighborhood postcode column from split_postcode_df data frame
uk_postcodes_df['neighborhood'] = split_postcode_df[1]

uk_postcodes_df.head()
uk_postcodes_df.tail()

	postcode	latitude	longitude	district	neighborhood
0	AB10 1XG	57.144165	-2.114848	AB10	1XG
1	AB10 6RN	57.137880	-2.121487	AB10	6RN
2	AB10 7JB	57.124274	-2.127190	AB10	7JB
3	AB11 5QN	57.142701	-2.093295	AB11	5QN
4	AB11 6UL	57.137547	-2.112233	AB11	6UL

	postcode	latitude	longitude	district	neighborhood
1758221	YO8 9SN	53.775330	-1.129099	YO8	9SN
1758222	YO8 9SP	53.767117	-1.089415	YO8	9SP
1758223	YO8 9TR	53.771989	-1.120702	YO8	9TR
1758224	ZE2 9FJ	60.128408	-1.208825	ZE2	9FJ
1758225	ZE2 9FN	60.128408	-1.208825	ZE2	9FN

After we have gotten the dataframe uk_postcodes_df2, with UK districts as well as latitude and longitude, we have started the data cleansing in the dataframe city_of_london_postcodes_df .

city_of_london_postcodes_df.head(1)

	Postcode district	Post town	Coverage	Local authority area
0	EC1A	LONDON	St Bartholomew's Hospital	City of London, Islington

From this dataframe, we were mainly interested in the columns **Postcode district | Post town | Local authority area**. So we have dropped the Coverage column.

to drop 'Coverage' column

city_of_london_postcodes_df.drop(['Coverage'], axis=1, inplace=True)

city_of_london_postcodes_df.head()
city_of_london_postcodes_df.tail()

	Postcode district	Post town	Local authority area
0	EC1A	LONDON	City of London, Islington
1	EC1M	LONDON	Islington, Camden, City of London
2	EC1N	LONDON	Camden, City of London
3	EC1P	LONDON	non-geographic
4	EC1R	LONDON	Islington, Camden

	Postcode district	Post town	Local authority area
23	EC4P	LONDON	non-geographic
24	EC4R	LONDON	City of London
25	EC4V	LONDON	City of London
26	EC4Y	LONDON	City of London, Westminster
27	EC50	LONDON	non-geographic

rename the column 'Postcode district' to 'district'
city_of_london_postcodes_df.rename(columns={'Postcode district': 'district'}, inplace=True)
city_of_london_postcodes_df.head()

	district	Post town	Local authority area
0	EC1A	LONDON	City of London, Islington

1	EC1M	LONDON	Islington, Camden, City of London
2	EC1N	LONDON	Camden, City of London
3	EC1P	LONDON	non-geographic
4	EC1R	LONDON	Islington, Camden

The dataframe city_of_london_postcodes_df has entries like the one in row 3, where the district is just an administrative code and do not have the corresponding latitude and longitude, therefore we need to remove all those entries from the dataframe.

We have removed from dataframe all rows where Local authority area = 'Not assigned'

index_names = city_of_london_postcodes_df[city_of_london_postcodes_df['Local authority area'] == 'non-geographic'].index

city_of_london_postcodes_df.drop(index_names, axis=0, inplace=True)

city_of_london_postcodes_df.head()

	district	Post town	Local authority area
0	EC1A	LONDON	City of London, Islington
1	EC1M	LONDON	Islington, Camden, City of London
2	EC1N	LONDON	Camden, City of London
4	EC1R	LONDON	Islington, Camden
5	EC1V	LONDON	Islington, Hackney

city_of_london_postcodes_df.shape
(23, 3)
city_of_london_postcodes_df.head(23)

	district	Post town	Local authority area
0	EC1A	LONDON	City of London, Islington
1	EC1M	LONDON	Islington, Camden, City of London
2	EC1N	LONDON	Camden, City of London
4	EC1R	LONDON	Islington, Camden
5	EC1V	LONDON	Islington, Hackney
6	EC1Y	LONDON	Islington, City of London

7	EC2A	LONDON	Islington, Hackney, City of London
8	EC2M	LONDON	Tower Hamlets
9	EC2N	LONDON	City of London
11	EC2R	LONDON	City of London
12	EC2V	LONDON	City of London
13	EC2Y	LONDON	City of London
14	EC3A	LONDON	City of London
15	EC3M	LONDON	City of London
16	EC3N	LONDON	Tower Hamlets, City of London
18	EC3R	LONDON	City of London
19	EC3V	LONDON	City of London
20	EC4A	LONDON	City of London, Westminster
21	EC4M	LONDON	City of London
22	EC4N	LONDON	City of London
24	EC4R	LONDON	City of London
25	EC4V	LONDON	City of London
26	EC4Y	LONDON	City of London, Westminster

We have merged both city_of_london_postcodes_df() and uk_postcodes_df. we used the option 'inner' to get a dataframe with only data relevant to City of London.

merge dataframes

city_of_london_df = pd.merge(uk_postcodes_df, city_of_london_postcodes_df, how='inner', on=['district'])

city_of_london_df.head()
city_of_london_df.tail()

	postcode	latitude	longitude	district	neighborhood	Post town	Local authority area
0	EC4Y 0AY	51.512616	-0.106828	EC4Y	0AY	LONDON	City of London, Westminster

1	EC4Y 0BH	51.512625	-0.108328	EC4Y	0BH	LONDON	City of London, Westminster
2	EC4Y 0BS	51.511985	-0.107144	EC4Y	0BS	LONDON	City of London, Westminster
3	EC4Y 0DA	51.511833	-0.108217	EC4Y	0DA	LONDON	City of London, Westminster
4	EC4Y 0DB	51.511833	-0.108217	EC4Y	0DB	LONDON	City of London, Westminster

	postcode	latitude	longitude	district	neighborhood	Post town	Local authority area
3371	EC1A 4JW	51.521952	-0.098229	EC1A	4JW	LONDON	City of London, Islington
3372	EC1A 7BD	51.518864	-0.098559	EC1A	7BD	LONDON	City of London, Islington
3373	EC1A 7BL	51.517671	-0.098753	EC1A	7BL	LONDON	City of London, Islington
3374	EC1A 7BF	51.518407	-0.098636	EC1A	7BF	LONDON	City of London, Islington
3375	EC1A 7BG	51.518578	-0.099220	EC1A	7BG	LONDON	City of London, Islington

city_of_london_df.shape

(3376, 7)

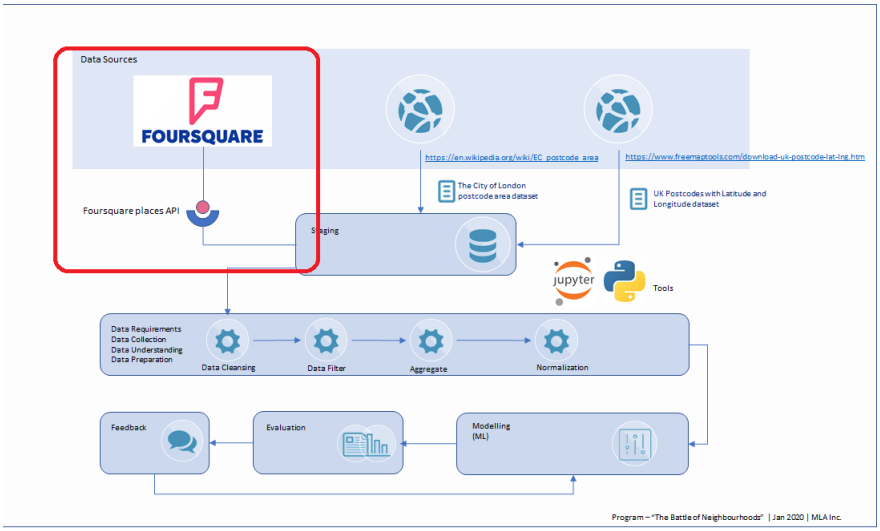
The number of districts and neighborhoods we have gotten on this dataframe was:

```
print('The dataframe has {} districts and {} neighborhoods.'.format(
        len(city_of_london_df['district'].unique()),
        city_of_london_df.shape[0]
    )
)
```

The dataframe has 23 districts and 3376 neighborhoods.

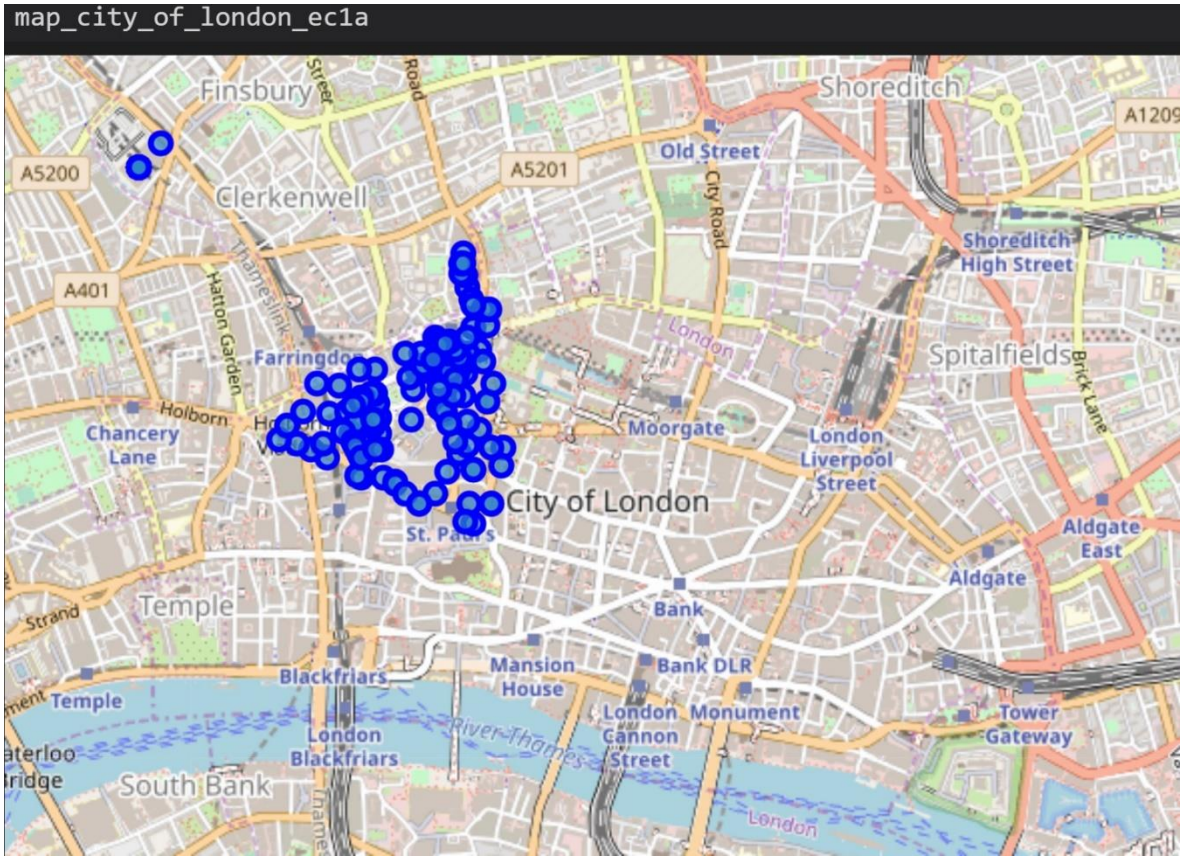
2.4 Modelling

We have downloaded the dataset related with UK postcodes with latitudes and longitudes as well as the dataset postcodes from the districts in the City of London. We have created two pandas dataframes with both datasets and after we have merged the two dataframes in order to have a single dataframe which would contain only neighborhoods in the City of London, with respective latitudes and longitudes.
The dataframe contained 23 districts and more than 3300 neighborhoods. So, we decided to create smaller dataframes. A dataframe per district, so we could analyze and use the Foursquare API with our development account.
We have used geopy library to get the latitude and longitude values of which district in the City of London.
We have used Folium library for visualization. With Folium we were able to visualize the neighborhood marks in a map per district in the City of London, and click on each circle mark to reveal the name of the neighborhood and its respective District.
We have used **One hot encoding** where categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. In our project we have used the ML K-means method of vector quantization for cluster analysis and data mining of neighborhoods in the City of London.
After we got the dataframe with one hot encoding we have calculated taking the mean of the frequency of occurrence of each category.
We have used Foursquare Location API to retrieve the 5 top most common venues for 5 out 23 districts in the City of London.



The queries with Foursquare API have used the location (lat, long) returned by the geopy library and the dataframe containing the neighborhoods, latitude, and longitude per district.
Foursquare has returned the results in JSON format, where we have printed each neighborhood along with the top 5 most common venues.
For each neighborhood we have highlighted the neighborhoods where "Gym / Fitness" was figuring within the top 5 most common venues.
After we have processed the JSON response from Foursquare and saved the results in dataframe.
We have created a grouped dataframe per district and we have run k-means to cluster each neighborhood from the district under evaluation into 5 clusters.
We have created a dataframe that includes the cluster as well as the top 10 venues for each neighborhood.
We have examined each cluster and determined the discriminating venue categories that distinguish each cluster.
Based on the defining categories, we have then assigned a name to each cluster.

We have Explored 23 districts and analyzed 5 in detail, and clustered the neighborhoods in the City of London.



2.5 Model Evaluation

We have used K=5 for clustering. The goal of K-means algorithm is to find groups in the data, with the number of groups represented by the variable K, and the data points are clustered based on feature similarity. On our project we aim to cluster all the neighborhoods where the Gym / Fitness center venue would the be the feature similarity.

3. Results

We have analyzed 5 Districts out of total of 23 districts for the City of London. Leveraging Foursquare Location API, we were able to find neighborhoods in 5 districts where the "Gym / Fitness Center" feature either the 1st or 2nd most common venue.

The table below lists the 5 recommended neighborhoods to setup the first 5 MLA inc., Gyms in the City of London.

Neighborhoods	Local Area	Recommended	Findings
EC3R 6BR	City of London	YES	Gym / Fitness Center features as 1st most common venue on this neighborhood
EC3R 8AH	City of London	YES	Gym / Fitness Center features as 1st most common venue on this neighborhood
EC2M 2PF	Tower Hamlets	YES	Gym / Fitness Center features as 1st most common venue on this neighborhood
EC2M 7LA	Tower Hamlets	YES	Gym / Fitness Center features as 1st most common venue on this neighborhood
EC1M 4AA	Islington, Camden, City of London	YES	Gym / Fitness Center features as 2nd most common venue on this neighborhood
EC4A	Westminster	NO	This District doesn't has any neighborhood where "Gym / Fitness Center" is within the 4 most common venue.

For this project we have used Foursquare Development account. The development account has limitations in the number of premium API queries, which one can do via Foursquare API. The API call for "venues" is a premium API call. Therefore, we have analyzed only 5 out of 23 districts, because City of London has many neighborhoods (3376 neighborhoods).

4. Discussion

On our analyses we have used Foursquare to find neighborhoods where the venue Gym / Fitness Center where featuring on the top 5. We have analyzed 5 out 23 districts from the City of London. We were able to recommend 5 neighborhoods, where 4 of them the venue Gym/Fitness Center features as the 1st Most common Venue.

For one of districts, we have clustered the neighborhoods in 5 different clusters.

During the model evaluation and feedback from stakeholders, it was decided to look for neighborhoods, where Gym/Fitness Center was featuring in the top 2 most common venues.

In a real scenario, the work from the data science team would be complete only after complete analyze of the 23 districts in the City of London.

5. Conclusions

The purpose of this project was to identify and recommend 5 best neighborhoods in the City of London, where MLA inc., would setup their first 5 Gyms in the MLA expansion program to UK. The criteria for the selection of those neighborhoods was based on the ranking of the most common venue returned by Foursquare API. where Gym / Fitness Center should rank on the top 5 most common. During the analyze we have reduced the criteria of Gym/Fitness Center to the top 2 most common venues.

From the 5 districts under analyze, there were a total of 7 neighborhood's where Gym / Fitness was the 1st most common venue.

EC3R 6BR 1st City of London EC3R 8AF EC3R 8AH

EC2M 2BS 1st Tower Hamlets EC2M 2PB EC2M 2PF EC2M 7LA EC2M 7UR

The final decision over which neighborhood will be select for the first 5 Gym's, will be taken by the stakeholders.

6. References

1. https://github.com/Ajrduarte/Coursera_Capstone/blob/master/Capstone_The_Battle_of_Neighborhoods.ipynb
2. https://github.com/Ajrduarte/Coursera_Capstone/blob/master/Capstone_The_Battle_of_Neighborhoods%20(1).html
3. IBM Data Science LAB 3: DP0701EN-3-3-2-Neighborhoods-New-York-py-v1.0.ipynb
4. IBM Data Science LAB 1: DP0701EN-3-3-1-Clustering-k-means-py-v1.0.ipynb
5. IBM Data Science LAB 2: DP0701EN-2-2-1-Foursquare-API-py-v1.0.ipynb
6. IBM Data Science: Course Data Science Methodology
7. UK Postcodes with Latitude and Longitude
8. City of London: EC postcode area
9. Foursquare (https://foursquare.com/)

7. Acknowledgments

This project and work were possible thanks to the excellent course and teachers in:

Course on Coursera called Applied Data Science Capstone
The entire specialization course set for IBM Data Science Professional.
The effort and quality put in the content and materials from this course by PhD. Alex Aklson, and PhD. Polong Lin.
Foursquare to provide such rich location data for free in the Developer account.

8. Appendices

Visualizing the resulting clusters in EC1M District

```
# create EC1M map
map_clusters_ec1m = folium.Map(location=[latitude_ec1m, longitude_ec1m], zoom_start=16)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(ec1m_merged_grouped['Latitude'], ec1m_merged_grouped['Longitude'], ec1m_me
rged_grouped['Neighborhood'], ec1m_merged_grouped['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters_ec1m)
```

map_clusters_ec1m